A User Comment Draft of the Joint Committee on the NTCIP

NTCIP 1202 Version 04

National Transportation Communications for ITS Protocol

Object Definitions for Actuated Signal Controllers (ASC) Interface

Draft v04.07b January 28, 2025

Published by

American Association of State Highway and Transportation Officials (AASHTO)
444 North Capitol Street, N.W., Suite 249
Washington, D.C. 20001

Institute of Transportation Engineers (ITE)
1627 Eye Street, N.W., Suite 550
Washington, D.C. 20006

National Electrical Manufacturers Association (NEMA)
1300 North 17th Street, Suite 900
Rosslyn, Virginia 22209-3801

REVISION HISTORY

Filename	Version	Date	Notes
NTCIP 1202v04.07b.docx	04.07b	1/28/25	User Comment Draft
NTCIP 1202v04.07a.docx	04.07a	01/24/25	Addressed comments from ASC WG 1/23 meeting. Replaced references to badValue(3) with updated errors. Removed disable remote command and manual backup from unitControlV4. Added unitManualBackup. Replaced patternStatus with coordPatternFaultStatus. Replaced badPlan(7) with badPattern(7)
NTCIP 1202v04.06b.docx	04.06b	01/20/25	Proposed User Comment Draft. Submitted to NTCIP JC.
NTCIP 1202v04.06a.docx	04.06a	01/20/25	Various fixes. Additional test cases.
NTCIP 1202v04.05c.docx	04.05c	12/31/24	Proposed User Comment Draft. Submitted to ASC WG.
NTCIP 1202v04.05b.docx	04.05b	12/25/24	Updated new Connected Vehicle Requirements following for Red Light Violation Warning. Updated System Engineering Traceability. Updated Section 4 and 5.
NTCIP 1202v04.05.docx	04.05	12/5/24	Updated Block Definitions. Updated requirements following walkthrough.
NTCIP 1202v04.04b.docx	04.04a	10/17/24	Draft submitted to USDOT for comment prior to walkthrough
NTCIP 1202v04.04a.docx	04.04a	10/16/24	Draft System Design Details
NTCIP 1202v04.03a.docx	04.03a	4/2/24	Final Requirements Document
NTCIP 1202v04.02fSRSWalkthroughDay2.docx	04.02f	3/15/24	Post walkthrough – Day 2
NTCIP 1202v04.02fSRSWalkthroughDay1.docx	04.02f	3/14/24	Post walkthrough – Day 1
NTCIP 1202v04.02f-pre.docx	04.02f	3/14/24	Pre-walkthrough
NTCIP 1202v04.02f.docx	04.02f	2/9/24	Chan – Submit to USDOT
NTCIP 1202v04.02e.docx	04.02e	2/6/24	Chan – Sent to co-chairs, Wyatt, Lahiri for QA
NTCIP 1202v04.02a.docx	04.02a	6/9/23	Chan – QA, New requirements from new user needs
NTCIP 1202v04.02.docx	04.02	6/1/23	Lahiri – Initial draft with Functional Requirements
NTCIP 1202v04.01a.docx	04.01a	5/1/23	Chan – Submitted to ASC WG and USDOT. Final ConOps
NTCIP 1202v04.01.docx	04.01	4/27/23	Lahiri – Addresses ConOps Walkthrough comments
NTCIP 1202v04.00a.docx	04.00a	3/15/23	Chan – Submitted to ASC WG and USDOT. Initial Draft ConOps
NTCIP 1202v04.00.docx	04.00	3/11/23	Lahiri – Initial Draft

NOTICES

Copyright Notice

PDF File License Agreement

Data Dictionary and MIB Distribution Permission

TRF Distribution Permission

A Testing Requirements Form ("TRF") may be a Testing Requirements Traceability Table and/or Test Procedures. To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of a TRF, AASHTO / ITE / NEMA extend the following permission:

a) you may make and/or distribute unlimited electronic or hard copies, including derivative works of the TRF, provided that each copy you make and/or distribute contains the citation "Based on NTCIP 0000 [insert the standard number] TRF. Used by permission. Original text © AASHTO / ITE / NEMA.";
b) you may not modify the logical flow of any test procedure, without clearly noting and marking any such modification; and
c) if the TRF excerpt is made from an unapproved draft, add to the citation "TRF excerpted from a draft standard containing preliminary information that is subject to change."

Content and Liability Disclaimer

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

AASHTO, ITE, and NEMA standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and seeks out the views of persons who have an interest in the topic covered by this publication. While AASHTO, ITE, and NEMA administer the process and establish rules to promote fairness in the development of consensus, they do not write the document and they do not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in their standards and guideline publications.

AASHTO, ITE, and NEMA disclaim liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. AASHTO, ITE, and NEMA disclaim and make no guaranty or warranty, express or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. AASHTO, ITE, and NEMA do not undertake to guarantee the performance of any individual manufacturer or seller's products or services by virtue of this standard or guide.

In publishing and making this document available, AASHTO, ITE, and NEMA are not undertaking to render professional or other services for or on behalf of any person or entity, nor are AASHTO, ITE, and NEMA undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

AASHTO, ITE, and NEMA have no power, nor do they undertake to police or enforce compliance with the contents of this document. AASHTO, ITE, and NEMA do not certify, test, or inspect products, designs, or installations for safety or health purposes. Any certification or other statement of compliance with any health or safety-related information in this document shall not be attributable to AASHTO, ITE, or NEMA and is solely the responsibility of the certifier or maker of the statement.

Acknowledgements

NTCIP 1202 v04 was prepared by the NTCIP Actuated Signal Controller Working Group (ASC WG), which is a subdivision of the Joint Committee on the NTCIP. The NTCIP Joint Committee is organized under a Memorandum of Understanding among the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). The NTCIP Joint Committee consists of six representatives from each of the standards organizations, and provides guidance for NTCIP development.

When NTCIP 1202 v04 was prepared, the following individuals were voting (indicated by an asterisk) or alternate voting members of the NTCIP ASC WG: **TO BE FINALIZED**

- California Department of Transportation, Mike Robinson*, Jay Schultz
- Caliper Corporation, Wuping Xin*
- City of Anaheim, John Thai* (Co-Chair)
- Econolite Control Products, Inc., Doug Tarico* (Co-Chair), Dustin DeVoe
- Florida Department of Transportation, Matthew DeWitt*, Jeffrey Morgan, Derek Vollmer
- Minnesota Department of Transportation, Derek Lehrke*, Ray Starr, Terry Haukom, Peter Skweres
- Nissan Motor Co., Roy Goudy*
- Oregon Department of Transportation, Christopher Primm*, Roger Boettcher
- Pillar Consulting, Ralph Boaz*
- Q-Free, Douglas Crawford*, Jorge Perez
- TransCore ITS, Robert Rausch*, Keith Patton
- Yunex Traffic, Jonathan Grant*, Dave Miller, Iouri Nemirovski

Observing members include:

- Consensus Systems Technologies, Kevin Balke, Patrick Chan, Manny Insignares, AJ Lahiri, Joerg 'Nu' Rosenbohm
- Florida DOT, Ronald Meyer
- Michigan DOT, Bashir Choudry
- Noblis, Justin Anderson, Mike Mercer
- Oriux, Ray Deer, Joe Socci
- Parsons Transportation Group, Jon Wyatt, Dmitri Khijniak
- SWARCO McCain Inc., Mark Simpson
- Texas A&M University (TTI), Hassan Charara, Srinivasa Sunkari
- Trevilon Corp., Kenneth Vaughn
- Utah DOT, Mike Blanchette

Additional stakeholders who provided input or monitored development include:

- CAMP, Jay Parikh
- Cubic, Shaun Alford
- City of Toronto, Rakesh Patel
- DKS & Associates, Alan Clelland
- Gresham Smith, James Anderson
- GreenFactor, Canny Quach
- Iteris, Frank Provenzano
- KLD, Satya Muthuswamy
- Leidos, Archak Mittal
- Maricopa County DOT, April Wire
- Miovision, Greg Barlow, Christian Kulus
- Mixon Hill, Christopher Poe, Jackson Holderman
- NoTraffic, Craig Hinners
- OmniAir, Randy Roebuck
- Oriux, Ray Deer, Joe Socci
- SANDAG, Peter Thompson
- Seattle DOT, Michael Mullen
- Skyline Technology Solution, Israel Lopez
- City of Toronto, Rakesh Patel
- UTA, Taylor Li
- WSP, Eric Gionet

In addition to the many volunteer efforts, recognition is also given to those organizations that supported the effort by providing funding:

- U.S. Department of Transportation

Foreword

NTCIP 1202 v04, an NTCIP standards publication, identifies and defines how a management station may wish to interface with a field device to control and monitor traffic signal controllers and associated detectors in an NTCIP-conformant fashion. NTCIP 1202 v04 may also identify and define how a traffic signal controller NTCIP 1202 v04 uses only metric units.

NTCIP 1202 v04 is titled Actuated Signal Controllers (ASC) Interface Protocol to express the multiple sections and annexes that are included in NTCIP 1202 v04. This NTCIP 1200-series standards publication has grown beyond the "object definitions" that were reflected in the title for its predecessors, NTCIP 1202 versions v01 (1996), v02 (2005), v03 (2018).

NTCIP 1202 v04 defines data elements for use with Actuated Signal Controller Units. The data is defined using the Simple Network Management Protocol (SNMP) object-type format as defined in RFC 1212 and the defined NTCIP format defined in NTCIP 8004. This data would typically be exchanged using one of the NTCIP 1103 recognized Application Layers (e.g., SNMP). Previous versions of NTCIP 1202 used SNMPv1. NTCIP 1202 v04 uses SNMPv3 and does not support SNMPv1.

NTCIP 1202 v04 follows an established systems engineering approach to support procurement processes. The PRL is designed to allow an agency to indicate what user needs are applicable to a procurement, and to select which requirements are to be implemented in a project specific implementation. Proper completion of the PRL by the agency results in a specification that is more likely to satisfy the agency’s project needs and that is conformant to NTCIP 1202 v04. The RTM defines the interface specifications for those requirements selected, and can be used to develop the test plans and test procedures.

The following keywords apply to this document: AASHTO, ITE, NEMA, NTCIP, ASC, data, data dictionary, object, MIB, PRL and RTM.

NTCIP 1202 v04 includes a number of normative and informative annexes.

NTCIP 1202 v04 is also an NTCIP Data Dictionary standard. Data Dictionary standards provide definitions of data concepts (messages, data frames, and data elements) for use within NTCIP systems; and are approved by AASHTO, ITE, and NEMA through a ballot process, after a recommendation by the NTCIP Joint Committee. For more information about NTCIP standards, or to acquire the related NTCIP 1202 v04 MIB, visit www.ntcip.org.

User Comment Instructions

The term "User Comment" includes any type of written inquiry, comment, question, or proposed revision, from an individual person or organization, about any NTCIP 1202 v04 content. A "Request for Interpretation" is also classified as a User Comment. User Comments are solicited at any time. In preparation of this NTCIP standards publication, input of users and other interested parties was sought and evaluated.

User Comments are generally referred to the committee responsible for developing and/or maintaining NTCIP 1202 v04. The committee chairperson, or their designee, may contact the submitter for clarification of the User Comment. When the committee chairperson or designee reports the committee’s consensus opinion related to the User Comment, that opinion is forwarded to the submitter. The committee chairperson may report that action on the User Comment may be deferred to a future committee meeting and/or a future revision of the standards publication. Previous User Comments and their disposition may be available for reference and information at www.ntcip.org.

A User Comment should be submitted to this address:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 900
Rosslyn, Virginia 22209-3801
e-mail: ntcip@nema.org

A User Comment should be submitted in the following form:

Standard Publication number and version:
Page:
Section, Paragraph, or Clause:
Comment:
Editorial or Substantive?:
Suggested Alternative Language:

Please include your name, organization, and address in your correspondence.

Approvals

To be completed prior to publication.

History

In 1992, the NEMA 3TS Transportation Management Systems and Associated Control Devices Section began the effort to develop NTCIP. Under the guidance of the Federal Highway Administration's NTCIP Steering Group, the NEMA effort was expanded to include the development of communications standards for all transportation field devices that could be used in an ITS network.

In September 1996, an agreement was executed among AASHTO, ITE, and NEMA to jointly develop, approve, and maintain the NTCIP standards. In late 1998, the Actuated Signal Controller Working Group was tasked with the effort to update the Actuated Traffic Signal Controller Object Definitions document. The first meeting of this working group was held in October 1999. From 1996 to 1999, this document was referenced as NEMA TS 3.5-1996. However, to provide an organized numbering scheme for the NTCIP documents, this document is now referenced as NTCIP 1202. As included in the following development history, NTCIP 1202 has experienced revisions over time:

NEMA TS 3.5-1996. 1996 – Approved by NEMA. 1996 – Accepted as a Recommended Standard by the Joint Committee on the NTCIP. 1997 – Approved by AASHTO and ITE.
v01.07a printed with NEMA cover.

NTCIP 1202 v01. v01.07b printed with joint cover. v01.07c printed to PDF in November 2002.
v01.07d printed to PDF for no-cost distribution January 2005.

NTCIP 1202 Amendment 1. November 1999 – Accepted as a User Comment Draft Amendment by the Joint Committee on the NTCIP. April 2000 – NTCIP Standards Bulletin B0049 sent NTCIP 1202 Amendment 1 v01.06b for user comment. NTCIP 1202 Amendment 1, a User Comment Draft, was incorporated into 1202v02, and was not advanced further.

NTCIP 1202 v02.10. June 2001 – Accepted as a User Comment Draft by the Joint Committee on the NTCIP. February 2002 – NTCIP Standards Bulletin B0068 referred v02.13 for user review and comment.

NTCIP 1202 v02.16. October 2002 – Accepted as a Recommended Standard by the Joint Committee on the NTCIP. April 2004 – NTCIP Standards Bulletin B0091 referred v02.18 for balloting. Approved by AASHTO in November 2004, approved by ITE in March 2005, and approved by NEMA in November 2004.

NTCIP 1202:2005 v02.19. November 2005 – Edited document for publication. By the terms of MOU on CTPA article 1.2, the ownership of version 02 was assigned to AASHTO, ITE, and NEMA because the preexisting work was revised by more than 50%.

NTCIP 1202 v03 was developed to reflect lessons learned, to update the document to the new documentation formats, and to add new features such as support for a connected vehicle interface. NTCIP 1202 v03 also follows an established systems engineering approach. Several new sections were added to relate user needs identified in a concept of operations, functional requirements, interface specifications and a requirements traceability matrix to the existing sections.

As NTCIP 1202 v03 was about to be published and distributed, a user provided proposed clarifications/corrections associated with experience in implementing the Flashing Yellow Arrow (FYA) functionality. The clarifications/corrections constitute the FYA errata, and is published as NTCIP 1202 v03A in May 2019.

The NTCIP 1202 v03B Amendment was published in October 2023 to address an urgent need to fix errors, clarify definitions, and provide additional guidance for traffic signal controllers to support the SAE J2735 SPaT Message in response to guidance detailed in Connected Transportation Interoperability (CTI) 4501, Connected Intersections (CI) Implementation Guide.

NTCIP 1202 v04 to be published in 2025.

Compatibility of Versions

To distinguish NTCIP 1202 v04 (as published) from previous drafts, NTCIP 1202 v04 also includes NTCIP 1202 v04.00a on each page header. All NTCIP Standards Publications have a major and minor version number for configuration management. The version number SYNTAX is "v00.00a," with the major version number before the period, and the minor version number and edition letter (if any) after the period.

The MIB associated with NTCIP 1202 v04 (as published) is tbd.

NTCIP 1202 v04 is designated, and should be cited as, NTCIP 1202 v04. Anyone using NTCIP 1202 v04 should seek information about the version number that is of interest to them in any given circumstance. The PRL, RTM and the MIB should all reference the version number of the standards publication that was the source of the excerpted material.

> Note: Users of the Test Procedure Generator (TPG) should enter Standard Number 1202, Major Version Number 04, Minor Version Number 0, and browse for the TPG-enabled version of NTCIP 1202 v04.

Compliant systems based on later, or higher, version numbers MAY NOT be compatible with compliant systems based on earlier, or lower, version numbers. Anyone using NTCIP 1202 v04 should also consult NTCIP 8004 v02 for specific guidelines on compatibility.

TABLE OF CONTENTS

Note: The following Contents listing includes seven heading levels (for annexes) to permit TPG evaluation.

LIST OF FIGURES

LIST OF TABLES

<This page is intentionally left blank.>

Section 1
General [Informative]

1.1 Scope

NTCIP 1202 v04 specifies the logical interface between an Actuated Signal Controller (ASC) and the host systems that control them; and the logical interface between an ASC and other roadside devices or processes at the intersection, such as a signal monitoring unit (SMU) in the transportation cabinet, a connected vehicle (CV) application process, or external control local application (ECLA). NTCIP 1202 v04 describes the supported ASC functionality in terms of user needs and requirements; however, the nature of the interfaces is determined in part by the operational nature of the devices being controlled, and therefore NTCIP 1202 v04 touches on such operational issues on occasion.

Prior to the development of NTCIP 1202, there were no standards defining how ASCs communicate with host systems or roadside devices/processes. As a result, each manufacturer has developed its own protocol to meet its own particular needs. This approach has resulted in systems that are not interchangeable or interoperable. If an agency wishes to use either a central management system or additional ASC from a different vendor, the agency encounters significant systems integration challenges, requiring additional resources to address. These additional resource requirements inhibit information sharing within and between various potential users of the data and prevent vendor independence. Without manufacturer independence, resource requirements further increase because of a lack of a competitive market.

These problems have not been limited to traffic signal controllers. Many other devices also need to exchange information. In surface transportation, examples include dynamic message signs, bus priority sensors, weather, and environmental monitoring, etc.

To address these problems, NTCIP is a family of open standards for communications between field devices and central management systems. NTCIP 1202 v04 is part of that larger family and is designed to define an interoperable and interchangeable interface between a transportation management system and an ASC, while still allowing for extensions beyond NTCIP 1202 v04 to allow for new functions as needed; and between an ASC and other roadside devices/processes. This approach is expected to support the deployment of ASC from one or more vendors in a consistent and resource-efficient way.

NTCIP 1202 v04 standardizes the communications interface by identifying the various operational needs of the users (Section 2) and subsequently identifying the necessary requirements (Section 3) that support each need. NTCIP standardized communications interface used to fulfill these requirements are identified by dialogs (**Error! Reference source not found.**) and related data concepts (0) that support each requirement. Traceability among the various sections is defined by the Protocol Requirements List (Section 3.3) and the Requirements Traceability Matrix (Annex A). Conformance requirements for NTCIP 1202 v04 are provided in Section 3.3.3. NTCIP 1202 v04 only addresses a subset of the requirements needed for procurement. It does not address requirements related to the performance of the traffic detectors (e.g., accuracy, the supported detection range, the time it takes to detect conditions, etc.), hardware components, or mounting details.

Previous versions of NTCIP 1202 addressed only ASCs that employ vehicle or pedestrian detectors to activate a particular phase – the scope did not include pre-timed, or fixed-time signal controllers that cycle through phases regardless of the number of vehicles or pedestrians present. ASCs included both fully actuated traffic signals, where all phases are actuated, and phases are skipped if no vehicles or pedestrians are detected; and semi-actuated traffic signals, where at least one phase is guaranteed to be served regardless of whether pedestrians or vehicles are detected. For the NTCIP 1202 purposes, controllers that allow different phases to be active (or skipped) at any point in time phase are known as phase-based controllers.

Beginning with NTCIP 1202 v03, the scope was expanded to standardize the communications interface between an ASC and a CV Application Process, which may be located inside an RSU. An RSU is any connected vehicle field device that is used to broadcast messages to, and receive messages from, nearby vehicles using Vehicle-to-Everything (V2X) communications, such as Cellular Vehicle-To-Everything (C-V2X).

An implementation of NTCIP 1202 requires lower-level services to structure, encode, and exchange the data concepts defined by NTCIP 1202. NTCIP 1202 assumes that the data concepts are exchanged by one of the protocols defined in NTCIP 2301.

1.2 References

1.2.1 Normative References

Normative references contain provisions that, through reference in this text, constitute provisions of NTCIP 1202 v04. Other references in NTCIP 1202 v04 might provide a complete understanding or provide additional information. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on NTCIP 1202 v04 are encouraged to investigate the possibility of applying the most recent editions of the standards listed.

Identifier	Title
ATC 5301 v02	Advanced Transportation Controller (ATC) Cabinet Standard Version 02, v02.02, AASHTO / ITE / NEMA, published March 18, 2019.
ISO 26048-1	Intelligent transport systems — Field device Simple Network Management Protocol (SNMP) data interface — Part 1: Global Objects
IETF RFC 1907	Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2), January 1996.
NEMA TS 1-1989 (R2020)	NEMA Standards Publication TS 1-1989 (R1994, R2000, R2005, R2020), Traffic Control Systems. (Not Recommended for New Designs)
NEMA TS 2-2021	NEMA Standards Publication TS 2-2021, Traffic Controller Assemblies with NTCIP Requirements Version 03.08, NEMA, published 2021.
SAE J2735	V2X Communications Message Set Dictionary, SAE International, published September 2024.
SAE J3161/1	Onboard System Requirements for LTE-V2X V2V Safety Communications, SAE International, published September 2024.

1.2.2 Other References

The following documents and standards may provide the reader with a better understanding of the entire protocol and the relations between all parts of the protocol. However, these documents do not contain direct provisions that are required by NTCIP 1202 v04. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on NTCIP 1202 v04 are encouraged to investigate the possibility of applying the most recent editions of the standard listed.

Identifier	Title
ATC 5201 API v06	Advanced Transportation Controller (ATC) Standard Version v06A, v06A.37, AASHTO / ITE / NEMA, published July 29, 2020.
ATC 5401 v02b	Application Programming Interface (API) Standard for the Advanced Transportation Controller (ATC) v02B, v02B.45, published February 16, 2023.
CTI 4001	Connected Transportation Interoperability 4001 - Roadside Unit (RSU) Standard, September 2022, v01.01.

Identifier	Title
CTI 4501	Connected Transportation Interoperability 4501 – Connected Intersections (CI) Implementation Guide, June 2022, v01.01.
Caltrans TEES 2020	Caltrans Transportation Electrical Equipment Specifications (TEES), 2020.
IEEE Std 100-2000	The Authoritative Dictionary of IEEE Standards Terms, IEEE, December 11, 2000.
Indiana Traffic Signal Hi Resolution Data Logger Enumerations	Indiana Traffic Signal Hi Resolution Data Logger Enumerations, November 2012. http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1002&context=jtrpdata
FHWA MUTCD 2023 Edition	Manual of Uniform Traffic Control Devices, FHWA, 11th Edition, December 2023.
Multimodal Intelligent Traffic Signal System	Multi-Modal Intelligent Traffic Signal System – Phase II: System Development, Deployment and Field Test, Final Report, Connected Vehicle Pooled Fund Study, September 2016. https://engineering.virginia.edu/sites/default/files/common/Centers/CTS/CVPFS/projects/53 MMITSS Phase 2 - Final Report – FINAL 09252016-compressed.pdf
NTCIP 1209	Object Definitions for Transportation Sensor Systems (TSS) Version 02, AASHTO / ITE / NEMA, published May 2014.
NTCIP 8004	Structure and Identification of Management Information (SMI) Version 03, AASHTO / ITE / NEMA, published xxx.
NTCIP 8007	Testing and Conformity Assessment documentation within NTCIP Standards Publications (TSS) Version 01, AASHTO / ITE / NEMA, published May 2008.
Preemption of Traffic Signals Near RR Grade Crossings, 2006	The Preemption of Traffic Signals Near RR Grade Crossings, an ITE Recommended Practice, Institute of Transportation Engineers, 2006.
Signal Timing Manual	Signal Timing Manual - Second Edition, National Academies of Sciences, Engineering, and Medicine, 2015.
U.S. Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT)	Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT), USDOT, Version 9.2.
V2I Hub Interface Control Document	Integrated Vehicle-to-Infrastructure Prototype (IVP), V2I Hub Interface Control Document (ICD) - Final Report March 2017, FHWA JPO.

1.2.3 Contact Information

1.2.3.1 Architecture Reference for Cooperative and Intelligent Transportation

The Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT) may be viewed at:

http://arc-it.net/

ARC-IT is also known as US National ITS Architecture and combines the US National ITS Architecture and the Connected Vehicle Reference Implementation Architecture (CVRIA).

1.2.3.2 ATC Standards

The Advanced Transportation Controller (ATC) Family of Standards may be view at:

https://www.ite.org/technical-resources/standards/

1.2.3.3 Caltrans Standards

Caltrans' Transportation Electrical Equipment Specifications (TEES) may be viewed at:

https://dot.ca.gov/programs/traffic-operations/tees

1.2.3.4 Connected Transportation Interoperability (CTI) Standards

CTI 4001 may be viewed at:

https://www.ite.org/technical-resources/standards/rsu-standardization/

CTI 4501 may be viewed at:

https://www.ite.org/technical-resources/standards/connected-intersections/

1.2.3.5 Internet Engineering Task Force (IETF) Documents

Obtain Request for Comment (RFC) electronic documents from several repositories on the World Wide Web, or by "anonymous" File Transfer Protocol (FTP) with several hosts. Browse or FTP to:

www.rfc-editor.org
www.rfc-editor.org/repositories.html
for FTP sites, read ftp://ftp.isi.edu/in-notes/rfc-retrieval.txt

1.2.3.6 NEMA Standards

Copies of NEMA standards may be obtained from:

National Electrical Manufacturers Association
1300 N.17th Street, Suite 900
Rosslyn, Virginia 22209-3801
www.nema.org

1.2.3.7 NTCIP Standards

Copies of NTCIP standards may be obtained from:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 N.17th Street, Suite 900
Rosslyn, Virginia 22209-3801
www.ntcip.org
e-mail: ntcip@nema.org

Draft amendments, which are under discussion by the relevant NTCIP Working Group, and amendments recommended by the NTCIP Joint Committee are available.

1.2.3.8 SAE International Standards

Copies of SAE International standards may be obtained from:

SAE International
400 Commonwealth Drive
Warrendale, PA 15096
www.sae.org

1.3 General Statements

<In the opinion of the responsible NTCIP working group, Section 1.3 does not apply in the context of NTCIP 1202 v04.>

1.4 Terms

For the purposes of NTCIP 1202 v04, the following terms, definitions, acronyms, and abbreviations apply. Meteorological terms not defined in this section are in accordance with their definitions in the Glossary of Meteorology. Electrical and electronic terms not defined here are used in accordance with their definitions in IEEE Std 100-2000. English words not defined here or in IEEE Std 100-2000 are used in accordance with their definitions in Webster's New Collegiate Dictionary.

Term	Definition
Actuated Signal Controller (ASC)	Any traffic signal controller, regardless if it is a phase-based controller or interval-based controller.
actuation	The operation of any type of detector.
advanced preemption time	The period of time between the minimum warning time needed for railroad operations and the maximum preemption time required for highway traffic signal operations.
automatic flash	Automatic programmed flash mode not caused by manual switch activation or fault condition or startup.
auxiliary function	A control that may activate auxiliary functions or outputs in an actuated controller unit.
backup mode	Control by local TBC or Interconnect based on absence of central or external command.
barrier	A barrier (compatibility line) is a reference point in the preferred sequence of a multi-ring CU at which all rings are interlocked. Note: Barriers assure there is no concurrent selection and timing of conflicting phases for traffic movement in different rings. All rings cross the barrier simultaneously for the selection and timing of phases on the other side.
Basic Safety Message (BSM)	The Basic Safety Message (BSM) is used in a variety of applications to exchange safety data regarding vehicle state. Source: SAE J2735_202309
call	A registration of a demand for right-of-way by traffic (vehicles or pedestrians) to a controller unit.
call, serviceable conflicting	A call which: a) Occurs on a conflicting phase not having the right-of-way at the time the call is placed. b) Occurs on a conflicting phase which is capable of responding to a call. c) When occurring on a conflicting phase operating in an occupancy mode, remains present until given its right-of-way.

Term	Definition
channel	Three circuits of a Monitor Device wired to monitor the green, yellow, and red outputs of the associated load switch position in the Terminal & Facilities. Channel 1 is assumed to monitor Load Switch 1, etc.
check	An output from a controller unit that indicates the existence of unanswered call(s).
clear track change interval	The yellow change interval following the clear track green interval and preceding the railroad hold intervals. A red clearance interval shall follow the clear track change interval if such an interval follows the normal yellow change interval. (Preemption of Traffic Signals Near Railroad Crossings - ITE, 2006)
computed lane	A lane that has a similar geometry and attributes of another lane.
concurrency group	A group of phases which describes possible timing combinations. Note: A phase within the group is required to be able to time concurrently with any other phase from another ring contained in the group. For example, in the typical dual-ring eight phase design, phases 1, 2, 5, and 6 form one concurrency group, and phases 3, 4, 7, and 8 form another concurrency group.
concurrent timing	A mode of controller unit operation whereby a traffic phase can be selected and timed simultaneously and independently with another traffic phase.
connected device	A mobile device, such as a vehicle or smartphone, equipped to broadcast, transmit or receive messages using V2X communications.
Connected Vehicle (CV) Application Process	A logical, functional process consisting of sub-processes that support the connected vehicle environment. From the context of an ASC, the relevant sub-processes include running intersection CV applications, broadcasting the SPAT and MAP messages to connected devices, and processing Basic Safety Messages (BSMs) and Personal Safety Messages (PSMs) received from connected devices by the CV Application Process. Physically, this may be a roadside unit (RSU).
Controller Assembly (CA)	A complete electrical device mounted in a cabinet for controlling the operation of a traffic control signal display(s).
Controller Unit (CU)	A controller unit is that portion of a controller assembly that is devoted to the selection and timing of signal displays.
Coordinated Universal Time (UTC)	UTC is the time standard commonly used across the world. The world’s timing centers have agreed to keep their time scales closely synchronized – or coordinated. This 24-hour time standard is kept using highly precise atomic clocks combined with the Earth’s rotation. UTC is similar to Greenwich Mean Time, but while UTC is a time standard, GMT refers to a time zone (similar to Eastern Standard Time). UTC never changes to account for daylight saving time.

Term	Definition
coordination	The control of controller units in a manner to provide a relationship between specific green indications at adjacent intersections in accordance with a time schedule to permit continuous operation of groups of vehicles along the street at a planned speed.
coordinator	A device or program/routine which provides coordination.
cycle	The total time to complete one sequence of signalization around an intersection. In an actuated controller unit, a complete cycle is dependent on the presence of calls on all phases. Note: In a pre-timed controller unit, it is a complete sequence of signal indications.
cycle length	The time period in seconds required for one complete cycle.
detector, pedestrian	Pedestrian detectors may be pushbuttons or passive detection devices. Passive detection devices register the presence of a pedestrian in a position indicative of a desire to cross, without requiring the pedestrian to push a button. Some passive detection devices are capable of tracking the progress of a pedestrian as the pedestrian crosses the roadway for the purpose of extending or shortening the duration of certain pedestrian timing intervals. Source: Manual of Uniform Traffic Control Devices, FHWA, December 2023
detector, system	Any type of vehicle detector used to obtain representative traffic flow information.
detector, vehicle	A detector that is responsive to operation by or the presence of a vehicle.
dial	The cycle timing reference or coordination input activating same. Dial is also frequently used to describe the cycle.
display map	A graphic display of the street system being controlled showing the status of the signal indications and the status of the traffic flow conditions.
dual entry	Dual entry is a mode of operation (in a multi-ring CU) in which one phase in each ring is required to be in service. Note: If a call does not exist in a ring when it crosses the barrier, a phase is selected in that ring to be activated by the CU in a predetermined manner.
dwell	The interval portion of a phase when present timing requirements have been completed.
dynamic timing pattern	A transient timing plan to be used for the next cycle only.
enabled lanes (list)	A sequence of lane identifiers for lanes that are identified to be enabled (active) and can be used by the appropriate travelers at the current time.
external control local application (ECLA)	An application that asserts a higher-level control over the traffic signal controller.

Term	Definition
fault monitor state	Internal CU diagnostics have determined that the CU device is not in a safe operational state. Note: An output may be asserted to indicate this condition.
first coordinated phase	The coordinated phase which occurs first within the concurrent group of phases containing the coordinated phase(s) when there are constant calls on all phases.
Flash	An operation where one section in each vehicle signal (yellow or red) is alternately on and off with a one second cycle time and a 50 percent duty cycle.
force off	A command to force the termination of the green indication in the actuated mode or Walk Hold in the non-actuated mode of the associated phase. Note: Termination is subject to the presence of a serviceable conflicting call. The Force Off function is not effective during the timing of the Initial, Walk, or Pedestrian Clearance. The Force Off is only effective as long as the condition is sustained. If a phase-specific Force Off is applied, the Force Off does not prevent the start of green for that phase.
Free	Operation without coordination control from any source.
gap reduction	A feature whereby the Unit Extension or allowed time spacing between successive vehicle actuations on the phase displaying the green in the extensible portion of the Green indication is reduced.
Group	Any portion of a traffic control network (system) that can be controlled by a common set of timing patterns.
Hold	A command that retains the existing Green indication.
Hold-on line	A signal to an intersection controller commanding it to remain under computer control.
interchangeability	A condition which exists when two or more items possess such functional and physical characteristics as to be equivalent in performance and durability and are capable of being exchanged one for the other without alteration of the items themselves, or adjoining items, except for adjustment, and without selection for fit and performance. Source: National Telecommunications and Information Administration, U.S. Department of Commerce
Interconnect	A means of remotely controlling some or all of the functions of a traffic signal.
interoperability	The ability of two or more systems or components to exchange information and use the information that has been exchanged Source: IEEE Std. 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology
intersection status	The knowledge of whether a controlled intersection is on-line and which mode it is currently operating in.

Term	Definition
indication	The part or parts of the signal cycle during which signal indication displays do not change.
Interval-based controller	A traffic signal controller implementing a sequence of defined, discrete steps (i.e., an interval), each interval driving their associated signal indications, in a repeating cycle according to the timing constraints programmed into the device. Note that some step sequences may be displayed or skipped in response to traffic conditions.
load switch driver group	The set of three outputs which are used to drive load switch inputs to provide a Green, Yellow, or Red output condition for vehicle signals or Walk, Ped Clear, or Don’t Walk output condition for pedestrian signals.
Malfunction Management Unit (MMU)	A device used to detect and respond to improper and conflicting signals and improper operating voltages in a traffic controller assembly.
Management Information Base (MIB)	A structured collection or database of related managed objects defined using Abstract Syntax Notation One (ASN.1). Source: NTCIP 8004 v02 and ISO/IEC 8824-1:2008 and ISO/IEC 8825-1:2008.
Manual Control Enable	An input to place calls on all actuated movements, stop CU timing in programmed variable intervals, and inhibit the operation of Interval Advance during all other intervals.
MAP message	The MapData message used to convey many types of geographic road information. At the current time, its primary use is to convey one or more intersection lane geometry maps within a single message. Source: SAE J2735_202309
maximum green	The maximum green time with a serviceable opposing actuation, which may start during the initial portion.
Movement	An action that is taken to traverse through an intersection, reflecting the user perspective and defined by the user type.
Multi-ring controller unit	A multi-ring CU contains two or more interlocked rings which are arranged to time in a preferred sequence and to allow concurrent timing of all rings, subject to barrier restraint.
Node point	A point defining the centerline of the pathway of a lane.
nonlocking memory	A mode of actuated-controller-unit operation which does not require the retention of a call for future utilization by the controller assembly.
Occupancy	A measurement of vehicle presence within a zone of detection, expressed in seconds of time a given point or area is occupied by a vehicle.
Off-line	A controller assembly not under the control of the normal control source.

Term	Definition
Offset	The time relationship, expressed in seconds, between the starting point of the first coordinated phase Green and a system reference point. (See definition of First Coordinated Phase)
omit, phase	A command that causes omission of a selected phase.
On-line	A controller assembly under the control of the normal control source.
Overlap	A Green display that allows traffic movement during the green indications of and clearance indications between two or more phases.
Passage time	The time allowed for a vehicle to travel at a selected speed from the detector to the stop line.
Pattern	A unique set of coordination parameters (cycle value, split values, offset value, and either signal plan or phase sequence). Note: A phase-based timing pattern consists of a cycle length, offset, set of minimum green and maximum green values, force off (determined by splits in some cases), and phase sequence. It also includes specification of phase parameters for minimum or maximum vehicle recall, pedestrian recall, or phase omit. An interval-based timing pattern consists of a cycle length, offset, set of minimum and programmed interval duration values, and a signal plan sequence.
Pedestrian clearance interval	The first clearance interval for the pedestrian signal following the pedestrian WALK indication.
Pedestrian recycle	A method of placing a recurring demand for pedestrian service on the movement when that movement is not in its Walk interval.
permissive	A time period, during which the CU is allowed to leave the coordinated phase(s) under coordination control to go to other phases.
Personal Safety Message (PSM)	The Personal Safety Message (PSM) is used to broadcast safety data regarding the kinematic state of various types of Vulnerable Road Users (VRU), such as pedestrians, cyclists or road workers. Source: SAE J2735_202309
phase	A timing unit associated with the control of one or more movements. Phases are often assigned to vehicular and pedestrian movements. Source: Signal Timing Manual Note: unless otherwise noted, a phase may be assigned to a vehicular, transit vehicle, or bicycle movement.
phase sequence	A predetermined order in which the phases of a cycle occur.

Term	Definition
phase, active	The indicated phase is currently timing. A phase is always active if it is Green or Yellow (Walk or Pedestrian Clear for Pedestrian Phases). It is also active if it is timing Red Clearance. It may be considered active during Red Dwell.
phase, conflicting	Conflicting phases are two or more traffic phases which cause interfering traffic movements if operated concurrently.
phase, nonconflicting	Nonconflicting phases are two or more traffic phases which do not cause interfering traffic movements if operated concurrently.
phase, pedestrian	A traffic phase allocated to pedestrian traffic which may provide a right-of-way pedestrian indication either concurrently with one or more vehicular phases, or to the exclusion of all vehicular phases.
phase, traffic	Those green, change and clearance intervals in a cycle assigned to any independent movement(s) of traffic.
phase, vehicular	A vehicular phase is a phase which is allocated to vehicular traffic movement as timed by the controller unit.
preempt dwell interval	The period of time when the track area is occupied by a tracked vehicle.
preemption	The transfer of the normal control of signals to a special signal control mode for the purpose of servicing railroad crossings, emergency vehicle passage, mass transit vehicle passage, and other special tasks, the control of which require terminating normal traffic control to provide the priority needs of the special task.
preemptor	A device or program/routine which provides preemption.
priority request	The information that describes a need for (signal) priority service based upon user-defined criteria (such as the number of minutes behind schedule, vehicle occupancy levels, vehicle class, etc.). Source: NTCIP 1211 v02.
progression	The act of various controller units providing specific green indications in accordance with a time schedule to permit continuous operation of groups of vehicles along the street at a planned speed.
red clearance interval	A clearance interval which may follow the yellow change interval during which both the terminating phase and the next phase display Red signal indications.
red revert	Provision within the controller unit to assure a minimum Red signal indication in a phase following the Yellow Change interval of that phase.
referenced lane	A lane used to define the attributes of another lane.
rest	The interval portion of a phase when present timing requirements have been completed.

Term	Definition
right-of-way transfer time	While providing preemption, the maximum amount of time needed for the worst-case condition, prior to display of the clear track green interval. This includes any railroad or traffic signal control equipment time to react to a preemption call, and any traffic signal green, pedestrian walk and clearance, yellow change and red clearance interval for conflicting traffic.
ring	A ring consists of two or more sequentially timed and individually selected conflicting phases so arranged as to occur in an established order.
Roadside Unit (RSU)	A transportation infrastructure communications device located on the roadside that provides V2X connectivity between OBUs/MUs and other parts of the transportation infrastructure including traffic control devices, traffic management systems, and back-office systems. Note: Devices that are not part of the transportation infrastructure, such as cellular base stations or satellites, are not RSUs. Source: CTI 4001
sample	A collection of data recorded over an identified period of time.
sequence, interval	The order of appearance of signal indications during successive intervals of a cycle.
service request	The information that describes a (signal) priority service to be processed by the ASC. Source: NTCIP 1211 v02
service requestor	A traveler requesting signal service or priority using a connected device. The connected device may be an OBE or a smartphone.
signal control priority strategy	Defines the phases to be serviced, phases to be omitted, and the maximum green times that can be reduced or extended to service a priority request.
Signal Monitoring Unit (SMU)	A subassembly that performs signal monitoring functions within a transportation cabinet. The signal monitoring unit is called a Malfunction Management Unit (MMU) in the NEMA TS 2 Standard and a Cabinet Monitor Unit (CMU) in the ATC Cabinet Standard.
signal plan	A unique set of parameters that define the phase / interval sequence of signal indications and control for one cycle.
signal request	A request for signal service or signal priority via an SAE J2735 Signal Request Message.
single entry	Single entry is a mode of operation (in a multi-ring CU) in which a phase in one ring can be selected and timed alone if there is no demand for service in a nonconflicting phase on the parallel ring(s).
single-ring controller unit	A single-ring CU contains two or more sequentially timed and individually selected conflicting phases so arranged as to occur in an established order.
special function	A control that may activate a device external to the controller unit.

Term	Definition
split	The segment of the cycle length allocated to each phase or interval that may occur (expressed in seconds). Note: In an actuated controller unit, split is the time in the cycle allocated to a phase.
standby mode	An operational state called by master or central command which directs the controller unit to select Pattern, Automatic Flash, or Automatic Free based on local Time Base schedule or Interconnect inputs.
stall condition	An operational state in which the ASC can no longer transmit any data to the management station. Note: The health monitor (watchdog) might or might not work in this situation, but its condition is not able to be transmitted to the management station.
TimeChangeDetail	A data frame that conveys details about the timing of a phase within a movement. The core data concept expressed is the time stamp (time mark) at which the related phase will change to the next state. Source: SAE J2735_202309
Time-based Control (TBC)	A means for the automatic selection of modes of operation of traffic signals in a manner prescribed by a predetermined time schedule.
timing pattern	See "Pattern"
timing plan	The Split times for all segments (Phase/Interval) of the coordination cycle.
track clearing interval	While providing preemption, the time assigned to clear stopped vehicles from the track area on the approach to the signalized highway intersection.
Traffic Signal Controller Broadcast Message (TSCBM)	A message defined in the V2I Hub Interface Control Document containing signal phase and timing (SPaT) information comprised of the SNMP data objects sent by the traffic signal controller to an RSU.
volume	The number of vehicles passing a given point per unit of time.
yellow change interval	The first interval following the green interval in which the signal indication for that phase is yellow.
yield	A command which permits termination of the green interval.
zone	An area in which traffic parameters can be measured and/or traffic data can be generated.

1.5 Abbreviations

The abbreviations (acronyms) used in NTCIP 1202 v04, and not defined in Section 1.4 are defined as follows:

AASHTO	American Association of State Highway and Transportation Officials

ADA	American Disability Act
AGET	Assured Green End Time
AGP	Assured Green Period
APS	Accessible Pedestrian Signals
ASC	Actuated Signal Controller
ASC WG	Actuated Signal Controller Working Group
ASN.1	Abstract Syntax Notation One
BSM	Basic Safety Message
CBR	Cars Before Reduction
CMU	Cabinet Monitor Unit
CTI	Connected Transportation Interoperability
CU	Controller Unit
CV	Connected Vehicles
CVRIA	Connected Vehicles Reference Implementation Architecture
DSRC	Dedicated Short Range Communications
ECLA	External Control Local Application
FYA	Flashing Yellow Arrow
FTRT	Features to Requirements Traceability
HOV	High Occupancy Vehicle
IO	Input/Output
ITE	Institute of Transportation Engineers
ITS	Intelligent Transportation Systems
MIB	Management Information Base
MMITSS	Multimodal Intelligent Traffic Signal System
MMU	Malfunction Management Unit
MU	Mobile Unit
NEMA	National Electrical Manufacturers Association
OBU	On-Board Unit
PRL	Protocol Requirements List
PSM	Personal Safety Message
RDZ	Red Light Violation Warning Detection Zone
RLVW	Red Light Violation Warning
RSE	Roadside Equipment
RSU	Roadside Unit

RTM	Requirements Traceability Matrix
SEP	Systems Engineering Process
SIU	Serial Interface Unit
SMU	Signal Monitoring Unit
SNMP	Simple Network Management Protocol
SPaT	Signal Phase and Timing (as defined by SAE J2735)
SRM	Signal Request Message
TBR	Time Before Reduction
TEES	Transportation Electrical Equipment Specifications
V2X	Vehicle-To-Everything
VRU	Vulnerable Road User

Section 2
Concept of Operations [Normative]

Section 2 defines the user needs addressed by subsequent sections within NTCIP 1202 v04. Accepted system engineering processes detail that requirements should only be developed to fulfill well-defined user needs. The first stage in this process is to identify the ways in which the system is intended to be used. In the case of NTCIP 1202 v04, this entails identifying the various ways in which transportation system managers may use ASC information to fulfill their duties.

This concept of operations provides the reader with:

a) a detailed description of the scope of NTCIP 1202 v04;
b) an explanation of how an ASC is expected to fit into the larger context of an ITS network;
c) a starting point in the agency procurement process; and
d) an understanding of the perspective of the designers of NTCIP 1202 v04.

Section 2 is intended for all readers of NTCIP 1202 v04, including:

a) transportation system managers
b) transportation operations personnel
c) transportation engineers
d) system integrators
e) device manufacturers

For the first three categories of readers, Section 2 is useful to understand how ASC equipment can be used in their system. For this audience, Section 2 serves as the starting point in the procurement process, and enables these readers to become familiar with each feature supported by NTCIP 1202 v04 and determine whether that feature is appropriate for their implementation. If it is, then the procurement specification needs to require support for the feature and all of the mandatory requirements related to that feature.

For the last two categories of readers, Section 2 provides a more thorough understanding as to why the more detailed requirements exist later in NTCIP 1202.

2.1 Tutorial [Informative]

A concept of operations describes a proposed system from the users' perspective. Typically, a concept of operations is used on a project to ensure that system developers understand users' needs. Within the context of NTCIP standards, a concept of operations documents the intent of each feature for which NTCIP 1202 v04 supports a communications interface. It also serves as the starting point for users to select which features may be appropriate for their project.

The concept of operations starts with a discussion of the current situation and issues that have led to the need to deploy systems covered by the scope of NTCIP 1202 v04 and to the development of NTCIP 1202 v04 itself. This discussion is presented in layman's terms such that both the potential users of the system and the system developers can understand and appreciate the situation.

The concept of operations then documents key aspects about the proposed system, including:

a) **Reference Physical Architecture.** The reference physical architecture defines the overall context of the proposed system and defines which specific interfaces are addressed by NTCIP 1202 v04. The reference physical architecture is supplemented with one or more samples that describe how the reference physical architecture may be realized in an actual deployment.

b) **Architectural Needs.** The architectural needs section discusses the issues and needs relative to the system architecture that have a direct impact on NTCIP 1202 v04.
c) **Features.** The features identify and describe the various functions that users may want components of an ASC system to perform. These features are derived from the high-level user needs identified in the problem statement but are refined and organized into a more manageable structure that forms the basis of the traceability tables contained in Section 3 and Annex A.

The architectural needs and features are collectively called user needs. Section 3 uses these user needs in the analysis of the system to define the various functional requirements of an ASC. Each user need shall be traced to one or more functional requirements, and each functional requirement shall be derived from at least one user need. This traceability is shown in the PRL as provided in Section 3.3.

While NTCIP 1202 is intended to standardize communications across a wide range of deployments, it is not intended to mandate support for every feature for every deployment. Therefore, the PRL also defines each user need and requirement as mandatory, optional, or conditional. The only items marked mandatory are those that relate to the most basic functionality of the device. To procure a device that meets specific needs, the user first identifies which optional needs are necessary for the specific project.

Each requirement identified is then presented in the RTM in Annex A, which defines how the requirement is fulfilled through standardized dialogs and data element definitions provided in Sections 4 and 5.

A conformant device may support other user needs, as long as they are conformant with the requirements of NTCIP 1202 v04 and its normative references (see Section 1.2.1). For example, a device may support data that has not been defined by NTCIP 1202; however, when exchanged via one of the NTCIP 2301 v02 protocols, the data shall be properly registered with a valid OBJECT IDENTIFIER under the Global ISO Naming Tree.

> Note: Off-the-shelf interoperability and interchangeability can only be obtained by using well-documented user needs, along with their corresponding requirements and design, that are broadly supported by the industry as a whole. Designing a system that uses environments or features not defined in a standard or not typically deployed in combination with one another inhibits the goals of interoperability and interchangeability, especially if the documentation of these user needs is not available for distribution to system integrators. NTCIP 1202 allows implementations to support additional user needs to support innovation, which is constantly needed within the industry, but users should be aware of the risks involved with using such environments or features.

The concept of operations concludes by describing the degree to which security issues have been addressed by the NTCIP 1202 v04 and by providing a description of how NTCIP 1202 v04 relates to the Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT), formerly known as the U.S. National ITS Architecture.

2.2 Current Situation and Problem Statement [Informative]

Transportation system managers use ASCs to control traffic operations on a roadway. ASCs allow different conflicting movements to travel across a roadway in a safe, orderly manner. In a roadway network, ASCs can be coordinated to improve mobility of certain movements, such as along a major arterial. Implemented correctly, ASCs can reduce:

a) the number and severity of accidents
b) delays
c) stops
d) fuel consumption
e) emission of pollutants

There are numerous factors that may affect the operation of an ASC on a roadway. Transportation system managers need to program each ASC to avoid conflicting movements. Conflicting movements are not confined to one specific mode of travel. Travel modes that have movements controllable by an ASC include:

a) Vehicles
b) Pedestrians
c) Bicycles
d) Special vehicles

Special vehicles are vehicles that have one or more characteristics so that an ASC may treat differently than "ordinary" vehicles. Special vehicles may include emergency vehicles or transit vehicles that request preferential (i.e., priority) treatment, or a high occupancy vehicle (HOV) with its own right-of-way (e.g., an HOV-only lane) through the intersection.

Each travel mode may have its own minimum clearance requirements that are satisfied to provide sufficient time for traffic to traverse the roadway before a conflicting movement is allowed to move.

Transportation system managers can also program an ASC to use inputs from other devices, such as detectors, to measure demand for a specific movement to improve mobility, so that additional time is provided for the movement where the demand exists and less time, if any, is provided for the movement where demand does not exist. An ASC also may be deployed with signal preemption or signal priority capabilities to properly manage movements in special situations. These capabilities, if implemented by the transportation system manager, may allow an emergency vehicle responding to an incident or a railroad at a railroad crossing to preempt the signal and obtain right-of-way. Similarly, signal priority may allow a transit or other fleet vehicle to request preferential treatment through a signalized intersection.

The ASC is also expected to have an important role in the connected vehicle environment. In the United States, the connected vehicle environment has three major goals, to improve safety, mobility and the environment. Many of the key applications being developed in support of these goals near signalized intersections involve the infrastructure providing signal phasing and timing information to "connected" devices, such as connected vehicles and "connected" mobile devices, such as a smartphone.

2.3 Reference Physical Architecture [Informative]

Section 2.3 represents an overview of what a complete ASC system may look like for a transportation agency, and identifies the specific information exchange paths to be addressed by NTCIP 1202 and related standards.

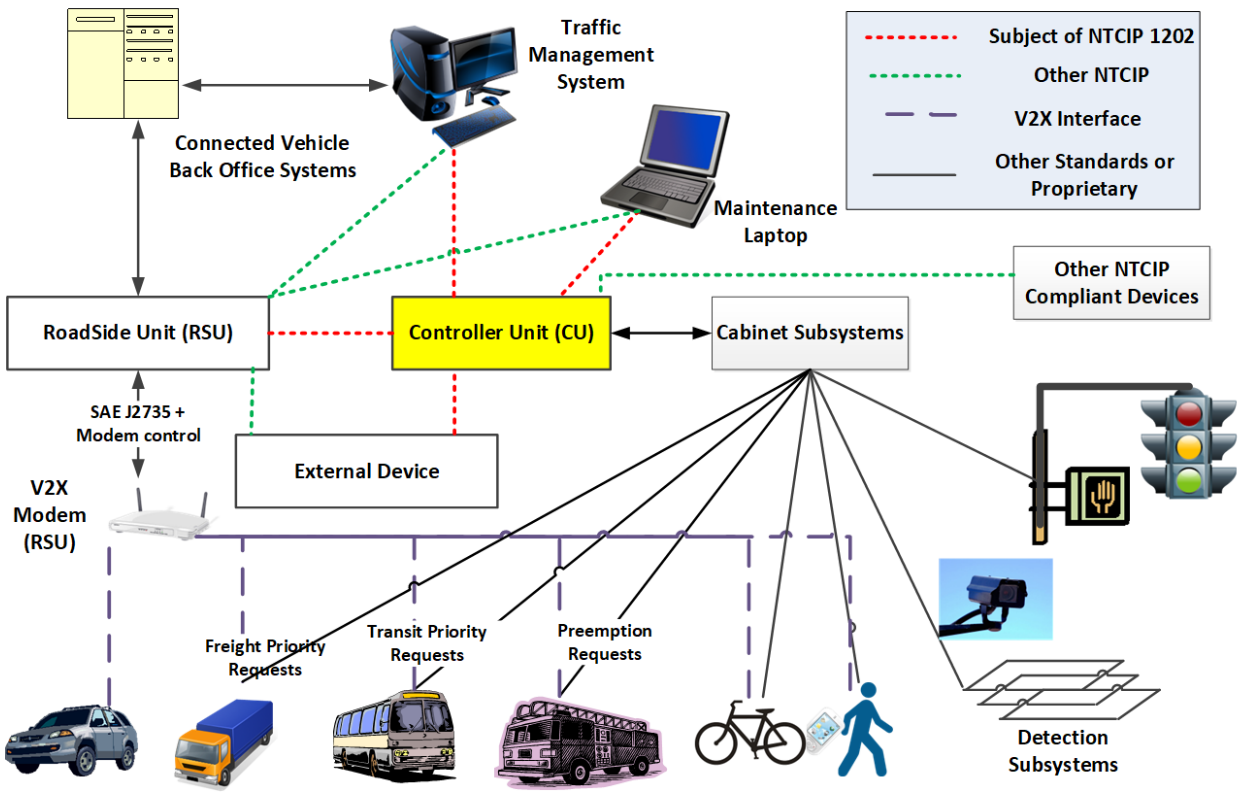

Figure 1 Reference Physical Architecture - ASC System

The physical components of the ASC system include:

a) **Controller Unit (CU).** A host computing platform that is used to manage the traffic signals at an intersection. The CU is responsible for ensuring that the proper signal indications are present on traffic signals. It communicates with the Traffic Management System and other devices through communication ports, and interfaces with the cabinet subsystems to energize the signal heads, read vehicle actuations, drive other auxiliary outputs, and read various inputs such as pedestrian push buttons. It may also communicate with other roadside devices, including other CUs.

b) **Traffic Management System.** A management station typically located in some type of management center (e.g., a Traffic Management Center (TMC)) and may be a considerable distance from the ASC. A management station is one or more host computing platforms that manage one or more NTCIP field devices, such as an ASC. The management station is responsible for configuring, monitoring, and controlling the ASC. There may be multiple management stations for a given ASC. A "manager" is a transportation system manager or maintenance person who needs to access information in the ASC through the management station.

c) **Maintenance Laptop.** A computer that a field technician may use on a trip to visit the ASC or a field processor that may be used to access the ASC. The maintenance laptop typically acts as a management station, which is a host computing platform that manage one or more NTCIP field devices, such as an ASC. The management station is used to monitor the data reported from the ASC and can command the ASC under certain conditions. The maintenance laptop typically plugs directly into the CU. A "manager" is a transportation system manager or maintenance person who needs to access information in the ASC through the management station.

d) **Roadside Unit (RSU).** A connected vehicle field device that includes a computing platform running applications and that supports secure communications with connected devices. The RSU receives messages from and transmits messages to nearby connected devices (such as vehicles via on-board units (OBUs) or mobile units (MUs) carried by vulnerable road users (VRUs) such as pedestrians, bicyclist, or road workers) using V2X communications.

e) **Detection Subsystems.** The units that provide inputs for traffic-actuated control, surveillance, or data collection systems. Detection subsystems include a wide variety of devices to detect the presence and other characteristics of travelers within the range of the intersection. In some instances, such detection devices may be connected directly to the CU and collect a variety of data such as volume, occupancy, speed, and headway or used for signal priority or preempt detection.
f) **External Device.** A physical device (equipment) that may be mounted inside the controller assembly to provide inputs to the CU, to control traffic flow, or use the outputs from the CU for other applications. Examples of external devices include traffic preemptors, signal priority equipment, accessible pedestrian signals, traffic control beacons, the V2X Hub and the MMITSS (Multimodal Intelligent Traffic Signal System) processor.
g) **Cabinet Subsystems.** The controller assembly that consists of the electrical devices in the cabinet for controlling the operation of a traffic control signal display(s). See Figure 2.

Other components shown in Figure 1 include:

h) **Connected Vehicle Back Office Systems.** Represent centers that manage and support the connected vehicle environment.
i) **Other NTCIP Compliant Devices.** Another controller unit that the CU communicates with for coordination of traffic signals. The other CU generally controls an adjacent traffic signal.

   Note: The deployment of connected vehicle equipment (such as the RSU) is currently very limited, but is expected to be widespread as more V2X equipped vehicles are delivered to the marketplace. Also, Figure 1 is only one possible architecture that might be used for the deployment of the infrastructure for connected vehicles, and other architectures are possible.

2.3.1 ASC Characteristics – Cabinet Specifications [Normative]

NTCIP 1202 is intended to address the communications interface between any management station and a CU. However, some features defined within NTCIP 1202 apply only to ASCs using a specific transportation cabinet architecture. There are five transportation cabinet architectures that are commonly used in North America.

a) **Model 332 Cabinet.** A cabinet specification defined in the Caltrans TEES.
b) **NEMA TS 1 Cabinet.** A cabinet architecture defined in NEMA TS 1.
c) **NEMA TS 2 Type 2 Cabinet.** A cabinet architecture defined in NEMA TS 2.
d) **NEMA TS 2 Type 1 Cabinet.** A cabinet architecture defined in NEMA TS 2.
e) **ATC Cabinet.** A cabinet architecture defined in ATC 5301. Previously called the ITS Cabinet.

Figure 2 shows a more detailed look at the components that may be inside a cabinet subsystem.

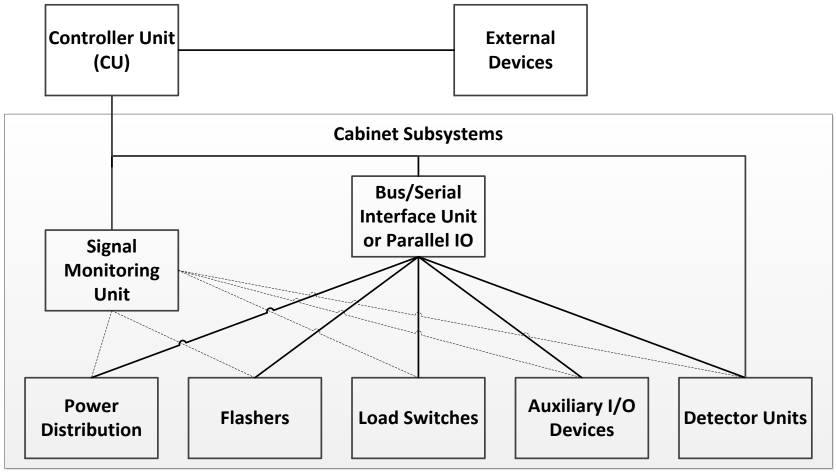

Figure 2 Controller Assembly

Cabinet subsystems include:

a) **Bus/Serial Interface Unit or Parallel IO.** The communications interface between the CU and the cabinet subsystem. It is called a Bus Interface Unit in the NEMA TS 2 Standard and a Serial Interface Unit (SIU) in the ATC Cabinet Standard. Some systems may use a Parallel Input/Output (IO) for the communications interface. It provides the means by which the CU can control the various cabinet devices, and can monitor inputs to provide analysis and data for use by the traffic management algorithms and the Traffic Management System shown in Figure 1.
b) **Power Distribution.** Provides protected power distribution to the various components and devices within the cabinet.
c) **Flashers.** Devices used to open and close signal circuits at a repetitive rate. It is typically used to provide a "fail-safe" flashing operation when the Signal Monitoring Unit determines that there is a failure within the cabinet wiring/devices such as shorted load switches, defective cabinet power supplies, or conflicting signal indications.
d) **Load Switches.** Devices used to switch power to the signal lamps/indications. This typically includes pedestrian signals, traffic signals, auxiliary signs, and other auxiliary devices.
e) **Signal Monitoring Unit (SMU).** A subassembly that performs signal monitoring functions within a transportation cabinet. The signal monitoring unit is called a Malfunction Management Unit (MMU) in the NEMA TS 2 Standard and a Cabinet Monitor Unit (CMU) in the ATC Cabinet Standard. When it detects a failure in the operation or a device, it can place the cabinet into the flashing condition using the flashers. It also monitors the power line voltage and places the cabinet into the "fail safe" condition when the operating voltage is below configured minimums and holds the cabinet in the "startup" flashing condition upon power restoration to allow the CU to boot and start normal operation.
f) **Detector Units.** Devices which support the detection of travelers (e.g., vehicles, pedestrians, bicycles, transit vehicles, emergency vehicles). In some cases, the interface allows the CU to monitor the health and gather additional information from the detection subsystems.

2.3.2 ASC Characteristics – Controller Types [Normative]

Some features defined within NTCIP 1202 may not be applicable to all ASCs - some features are dependent on whether an ASC is one of the following types of controllers.

a) **Phase-based controller.** Phase-based signal controllers refer to a device implementing non-conflicting signal indications in response to traffic conditions and the timing constraints programmed into the device. A phase controls signal indications for one or more non-conflicting traffic movements and may be actuated by those movement's traffic. In a phase-based, fully actuated system, phases without traffic present may be skipped. Green indication durations may vary between pre-set minimum and maximum values, depending on detected traffic and programmed timing information.

b) **Interval-based controller.** Interval-based signal controller refers to a device implementing a sequence of defined, discrete steps (i.e., an interval), each driving the signal indications, in a repeating cycle according to the timing constraints programmed into the device. Note that some step sequences may be displayed or skipped in response to traffic conditions.
c) **Stage-based controller.** Stage-based signal controller refers to a device implementing groups of different sequences of defined, discrete steps (i.e., an interval), each driving the signal indications in a repeated cycle. Groups of intervals may be skipped depending on traffic conditions.

Note: Some controllers can operate either as an interval-based controller or a phase-based controller (but not simultaneously). An agency (procurement) specification may include one or both of these types.

Only phase-based controllers are supported by NTCIP 1202 v04.

2.3.3 ASC Characteristics – Connected Vehicle Interface

NTCIP 1202 v04 also addresses the communications data exchange between an ASC and an RSU. It is through this communications interface with the RSU that an ASC primarily interacts with the connected vehicle environment. Before the ASC - RSU interface can be effectively addressed, an understanding of the functions that need to be performed among an ASC, an RSU, and possibly a third device, called an external device, is helpful.

Some data exchanges defined within NTCIP 1202 v04 for the connected vehicle interface are dependent on the relationship between the ASC, the RSU, and an external device, if one exists between the ASC and RSU. The National ITS Architecture, known as the Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT), implies a logical framework of applications and services that are allocated to the ASC and the RSU. These applications may have needs for information that are provided by the ASC (e.g., information needed to create signal phase and timing (SPaT) messages, status of signal priority requests) or may provide information to the ASC so the ASC may improve safety and mobility at a signalized intersection (e.g., forward a signal priority request, forward location of connected vehicle).

To better understand the connected vehicle environment around the ASC, the functions (and processes) that are needed to be performed have been allocated to one of three logical processes (See Figure 3): the ASC Process, the Connected Vehicle (CV) Application Process, and the RSU process.

NOTE: NTCIP 1202 v03 considered only two logical processes, an ASC Process and a CV Application Process (now called CV Application Process).

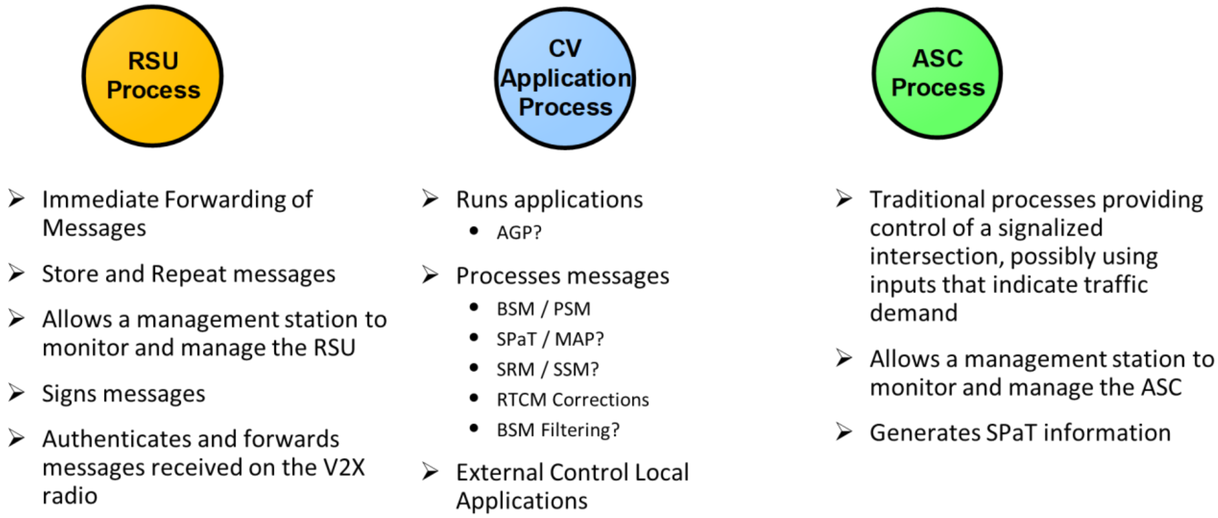

Figure 3 Connected Intersection Logical Processes

The ASC Process consists of the traditional processes providing control of a signalized intersection, possibly using inputs that indicate the traffic demand around the intersection. The source of those inputs may be detection subsystems located within or connected to the same cabinet as the traffic signal controller, or from the CV Application Process. The ASC Process allows a management station, such as a traffic management system, to monitor and manage the traffic signal controller; and generates signal phase and timing information that may be shared with the CV Application Process. These functions are addressed in this standard.

The CV Application Process consists of sub-processes that support the connected vehicle environment. From the context of an ASC, the relevant sub-processes include running intersection CV applications, receives the signal phase and timing information from the ASC Process to generate the SPAT messages to be broadcasted to connected devices, and processing Basic Safety Messages (BSMs) and Personal Safety Messages (PSMs) received from connected devices. A management station (traffic management system) may also configure a CV Application Process to use BSMs and PSMs as inputs to the ASC Process.

The RSU Process consists of sub-processes to manage broadcasting and receiving over-the-air (wireless) messages in a connected vehicle environment. The RSU Process may also perform other functions, such as send and manage security certificates, signing messages that are broadcasted, and authenticate received messages. These functions are addressed in NTCIP 1218, Object Definitions for RoadSide Units. Note: The Network Interface and Wireless Interface in NTCIP 1218 v01A is part of the RSU Process.

Figure 4 depicts the interfaces between the different entities and processes that comprise the connected vehicle environment around the ASC. The information exchanges depicted in black, specifically between the RSU Process and the CV Application Process, are expected to be in SAE J2735 format. The information exchanges in orange, specifically between the ASC Process and the CV Application Process and between the ASC Process and the Traffic Management System (management station), are addressed by NTCIP 1202 v04. The information in cyan between the RSU Process and the Traffic Management System (management station) allow a traffic management system to configure the RSU Process and are addressed by NTCIP 1218.

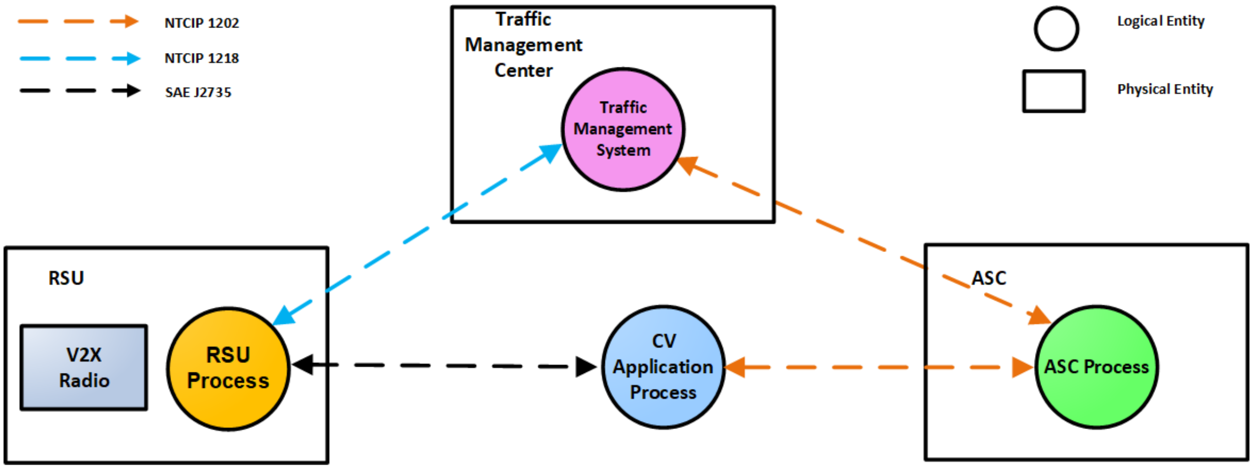

Figure 4 ASC - Connected Vehicle System Context Diagram

From a physical point of view, there are two possible physical architectures that are addressed by NTCIP 1202 v04, defined by where the CV Application Process is physically located, although there may be variations of a physical architecture.

With physical architecture 1, depicted in Figure 5, the CV Application Process is physically located within the ASC, whether it's part of the main ASC application, or a separate application (process) within the ASC. This architecture is applicable when the ASC has sufficient processing power (or a second processor) to generate CV messages, such as SPaT messages, and to process received messages, such as BSMs and PSMs. With this architecture, the ASC receives then processes BSMs and PSMs provided via the RSU, which is a nearby field-hardened computing device within the same or a separate cabinet as the ASC. The raw BSMs and PSMs received by the ASC are authenticated by the RSU, and possibly filtered by the RSU. Examples of filtered BSMs and PSMs that are forwarded by an RSU include OBUs/MUs at a specific location, vehicle type, and/or direction of travel.

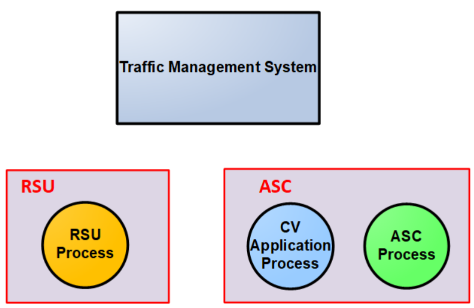

Figure 5 Physical Architecture 1

With physical architecture 1, the UPER-encoded SPaT message is generated by the (physical) ASC and sent to the RSU Process. The ASC also receives the UPER-encoded BSM and PSM messages from the RSU Process.

With physical architecture 2, depicted in Figure 6, the CV Application Process is physically located external to the ASC. The CV Application Process could be physically located (a) in the RSU, (b) an external device, or (c) the functions of the CV Application Process are shared between the RSU and an external device. The external device could be a separate processor in the same cabinet with the ASC. Examples of an external device in this context includes the V2X Hub and the MMITSS processor.

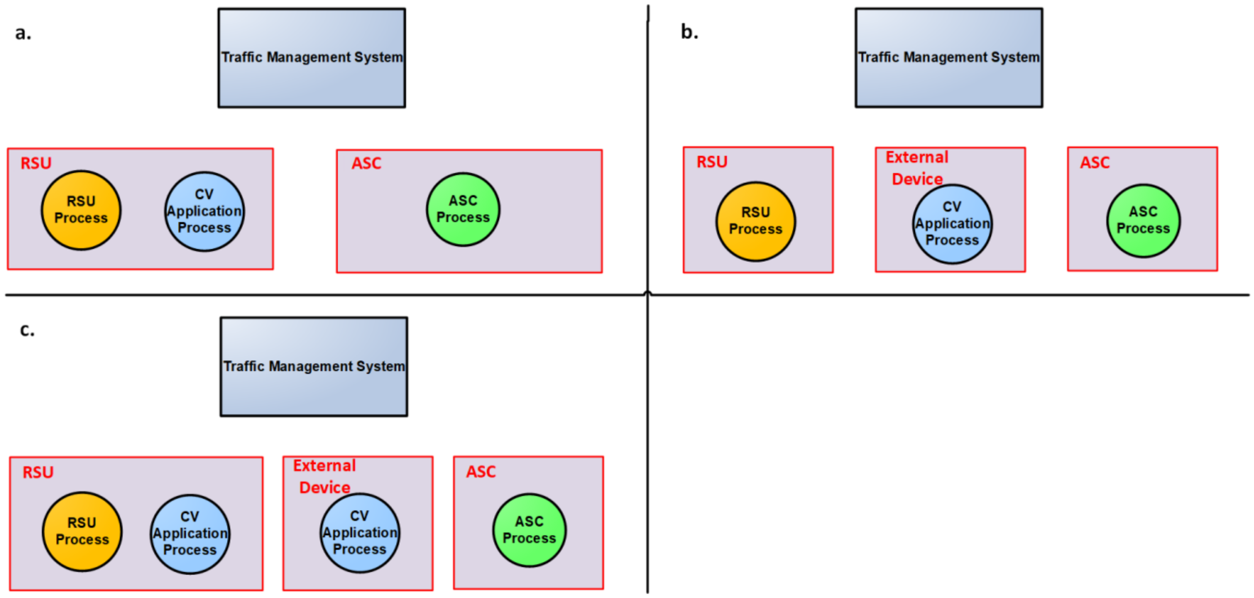

Figure 6 Physical Architecture 2

With physical architecture 2, SPaT information is generated by the (physical) ASC and sent to the CV Application Process, which then generates the SPaT message. The ASC also receives defined information about the location, kinematics, and possibly vehicle types about travelers in the vicinity of the RSU, as processed by the CV Application Process.

2.3.4 ASC Characteristics – Interface with Other Roadside Devices/Processes

NTCIP 1202 v04 also addresses the communications data exchange between an ASC and other roadside devices or processes that may exist within the Reference Physical Architecture (Figure 1) or within the controller assembly (Figure 2). The roadside devices or processes considered by NTCIP 1202 v04 include the logical interface with an:

a) **SMU.** A physical device to monitor the operations and devices in the controller assembly. The SMU is called a MMU in the NEMA TS 2 Standard and a CMU in the ATC Cabinet Standard. The SMU may inform the ASC when failures in controller assembly are detected, when the cabinet is in a "fail safe" condition (i.e., flashing), or when operating voltages are below configured minimums.
b) **External Device.** A physical device, defined above in Section 2.3.
c) **External Control Local Application (ECLA).** A logical entity that adjusts the signal timing in effect to accommodate different traffic patterns in real-time. An example of an ECLA is a traffic adaptive algorithm application external to the ASC, but asserting a higher-level control over the ASC. The ECLA may also be physically located in an external device or at a traffic management center.

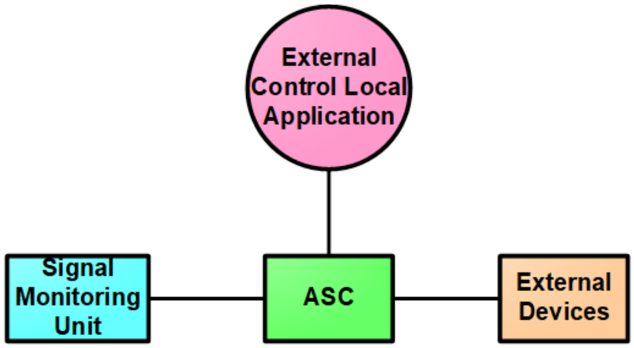

Figure 7 Roadside Process/Devices Interface

2.4 Architectural Needs

NTCIP 1202 v04 addresses the interface between an ASC and one or more management stations (e.g., central computers, laptops, RSUs, peer controller units, etc.). A management station needs to monitor the status of the ASC, manage the database in the ASC, and control the ASC. The management station also needs to retrieve data that has been collected by the ASC. After the management station has retrieved the data of interest, a transportation system manager (manager) can use the retrieved data to make decisions and initiate other events (such as changes to the ASC timing pattern) to better manage the transportation system.

NTCIP 1202 v04 also addresses the interface between an ASC with other roadside devices and processes, such as the CV Application Process. The CV Application Process needs data from the ASC about current and future signal phasing and timing information, so the CV Application Process can forward this information to connected devices. For the interfaces between an ASC and a roadside device/process, the ASC may act as a management station, and the other device/process acts as an agent/server.

To enable communications between these components, the manager needs to establish a communication system that links the ASC with a management station. For some systems, the resources required for communications may be minimal and as such the system may be designed for constant polling; other systems may require significant resources for communicating with the ASC and as such the system may be designed to minimize data exchanges. When deploying an ASC, the system designer needs to consider which of the following operational environments need to be supported.

An ASC is expected to operate in the communications environment defined as follows.

2.4.1 Provide Live Data

The typical operational environment allows a management station to monitor and control the ASC by issuing requests (e.g., requests to access information, alter information, or control the device). In this environment, the ASC responds to requests from the management station (e.g., through the provision of live data, success/failure notice of information alteration, or success/failure of the command). This environment may also be used to allow an ASC to monitor another roadside process or device by issuing requests (e.g., request to access information).

2.4.2 Provide Data Blocks

Some operational environments have limited data capacity due to limitations in the data rates of the media and/or due to multiple entities or devices sharing the same communications channel. In such environments, a manager needs to exchange sets of data together so that data can be transmitted more efficiently over networks, thereby conserving the limited data capacity of the channel. This capability reduces the upload and download times of data between a management station and an ASC, or an ASC with another device (e.g., CV Application Process). The sets of data may be standardized, pre-defined blocks of data or run-time definable sequences addressing functional areas associated with ASCs.

2.4.3 Provide for Log Data Local Storage and Retrieval

In a typical operational environment, the ASC needs to provide logged data to the management station for diagnostic purposes, and for operational environments (e.g., dial-up links) that do not have always-on connections. For example, logged data may include the time when the cabinet door is opened. The event log needs to be cleared either in a last-in last-out basis or by the management station because of limited storage space in the ASC.

2.4.4 Provide for Database Management

Traffic signal controllers are safety critical devices to manage the traffic movements for vehicles, pedestrians, bicycles, transit, and others are intersecting roadways (or railroad crossings). To ensure that the data downloaded from a traffic management system to an ASC makes logical sense, consistency checks on the downloaded need to be performed by the ASC. A manager needs to therefore be able to manage the database by being able to open the database to write data, ensure that the downloaded data was received, command the ASC to perform a verification / consistency check the downloaded data, and to close the database to make the downloaded data available to the operational aspects of the ASC. Additionally, should there be any errors, a manager needs to be able to determine the source of the error within the downloaded data.

2.4.5 Condition-based Exception Reporting

In some operational environments, it may be desirable to have the ASC automatically transmit data to the management station when certain conditions occur. Under this scenario, a manager can program the information to be automatically reported to the management station when a specified condition occurs. An example is a manager wants to know when a cabinet door is opened, when the ASC goes to an error flash condition, or when a phase becomes active; these conditions can be programmed to cause the transmission of the alarm objects thus providing the management station with information regarding the change of state at the transportation cabinet.

2.5 Features

Section 2.5 identifies and describes the various features that may be offered by the ASC. It is divided into the following:

a) Manage the ASC Configuration
b) Manage Signal Operations
c) Manage Detectors
d) Manage Connected Vehicles Interface
e) Backwards Compatibility Features

2.5.1 Manage the ASC Configuration

This section identifies and describes the various features related to managing the configuration of the ASC. It consists of the following features:

a) Manage Device Identity
b) Manage Communications
c) Manage Cabinet Environment
d) Monitor Power
e) Retrieve Operational Performance Data
f) Manage Auxiliary External Inputs/Outputs
g) Manage Database

h) Manage Peer-to-Peer with Other Devices
i) Manage Signal Monitoring Unit Information
j) Manage Interface with External Detectors
k) Manage ASC Clock
l) Manage External Local Control Application

2.5.1.1 Manage Device Identity

A manager needs to manage basic information about the ASC. This information consists of its location (latitude and longitude), and the make, model, and version of the device components. The device components can be a hardware, software, or firmware component, and could be a physical or logical entity in nature. This feature allows a manager to verify the identity of the ASC in the field and what software or firmware version is installed. This feature also allows a manager to retrieve a unique identifier of the device as provided by the device manufacturer.

2.5.1.2 Manage Communications

A manager needs to manage each communications port in the ASC. This feature consists of enabling or disabling the communications ports, and configuring or retrieving the port address (e.g., IP address). This feature allows a manager to disable an unused communications port for security purposes or to reconfigure the ASC for a new communications media.

2.5.1.3 Manage Cabinet Environment

A manager needs to monitor the transportation cabinet operating environment. This feature allows a manager to monitor for unsafe operating environments for the ASC so proper precautions can be taken. Unsafe operating environment consists of an open transportation cabinet door, high cabinet temperatures, or an indication that the cabinet fan has turned on.

2.5.1.4 Monitor Power

A manager needs to monitor the power for the ASC. This feature allows a manager to determine whether the power sources for the transportation cabinet are suspect and need maintenance or whether the intersection is operating on an alternate power source. For example, some ASCs use AC power for its battery.

2.5.1.5 Retrieve Operational Performance Data

A manager needs to retrieve operational data from the ASC for the analysis of the signal timing efficacy. The operational data consists of frequent snapshots of signal operations data and detector data and allows a manager to view the temporal relationship between signal indications and traveler arrivals.

An example of this operational data is the Indiana Traffic Signal Hi Resolution Data Logger Enumerations. This feature provides a manager with the information to evaluate the performance of signal operations, such as the quality of progression of traffic along arterials, or measuring the amount of unused green time during a cycle. A manager may wish to monitor the operational data or store the operational data in a log for retrieval at a later time.

2.5.1.6 Manage Auxiliary External Inputs/Outputs

A manager needs to monitor and control auxiliary external devices (i.e., non-signal control) through the ASC. This feature allows a manager to activate auxiliary external devices or functions that may be tied to other transportation operational needs. For example, the ASC may be co-located with a trail-blazing sign utilized for special events and not associated with traffic signal operations.

2.5.1.7 Manage Database

A manager needs to manage the configuration and version of the database in the ASC. This feature allows a manager to determine if the ASC has the correct and expected version of the database.

2.5.1.8 Manage Interface with External Detectors

A manager needs to configure an ASC to receive and use detector data for traffic signal operations. This feature allows a manager to configure the source and type of detector data received, so the ASC may use this detector data for actuated operations or a traffic adaptive algorithm. The detector data provides details about traveler demand around the signalized intersection so the ASC can efficiently manage traffic signal operations. External detector data may include queue length information on an approach, turning movement volumes, or travel time for a specific movement through the intersection.

2.5.1.9 Manage ASC Clock

A manager needs to configure the ASC clock to determine the accurate time so the ASC can coordinate traffic operations on a roadway with other signalized intersections, and provide accurate information on signal operations to travelers. This feature allows a manager to configure the time source(s), the clock status, and configure what time zone the ASC is located in.

2.5.1.10 Manage External Control Local Application

A manager needs to configure the ASC to exchange information with an External Control Local Application (ECLA) that is asserting a higher-level control over the ASC. This feature allows a manager to enable the ASC to exchange information with an ECLA so the ASC can generate accurate SPaT data to a CV Application Process. This data includes signal timing durations that are selected by the ECLA.

2.5.2 Manage Signal Operations

This section identifies and describes the various features for an ASC to monitor and control traffic signal operations. It consists of the following features:

a) Manage Signal Configuration
b) Monitor Signal Operations Status
c) Control Signal Operations

2.5.2.1 Manage Signal Configuration

This feature allows a manager to retrieve and configure the traffic signal operations of an ASC. It consists of the following sub-features.

a) Manage Controller Startup Functions
b) Manage Phase Configurations
c) Manage Coordination Configurations
d) Manage Timing Patterns
e) Manage Splits Configurations
f) Manage Ring Configurations
g) Manage Channel Configurations
h) Manage Overlap Configurations
i) Manage Preempt Configurations
j) Manage Timing Pattern Scheduler
k) Manage Action Scheduler
l) Manage I/O Mapping
m) Manage Intra-Cabinet Communications Configuration
n) Manage Pedestrian Support

2.5.2.1.1 Manage Controller Startup Functions

A manager needs to retrieve and configure the startup capabilities and functions of the ASC. This feature allows a manager to define the Start-Up states upon powerup and configure a backup timer.

2.5.2.1.2 Manage Phase Configurations

For a phase-based controller, a manager needs to retrieve and configure the phases for the ASC. This feature allows a manager to set the minimum durations, maximum durations, clearance times, allowable concurrent phases, and other phase-related features and options for all travel modes (vehicles, pedestrians, bicycles, special vehicles).

2.5.2.1.3 Manage Coordination Configurations

A manager needs to retrieve and configure the coordination modes for the ASC. This feature allows a manager to configure the allowable operational, correction and force modes, and reference points within a timing pattern to be used for signal coordination.

2.5.2.1.4 Manage Timing Patterns

A manager needs to retrieve and configure the timing patterns stored in the ASC. This feature allows a manager to configure each timing pattern, which consists of a cycle length, splits, offsets and the phase sequences.

2.5.2.1.5 Manage Splits Configurations

A manager needs to retrieve and configure the splits stored in the ASC. This feature allows a manager to configure a split, which consists of the phase assignment, the coordinated phase, the split time, and the split mode.

2.5.2.1.6 Manage Ring Configurations

A manager needs to retrieve and configure the rings in the ASC. This feature allows a manager to configure each ring, which defines the sequence of phases for that ring.

2.5.2.1.7 Manage Channel Configurations

A manager needs to retrieve and configure the channel parameters in the ASC. This feature allows a manager to configure the control source, the type of phase the channel is controlling (e.g., vehicle phase, pedestrian phase, overlap), and the flash and dimming characteristics for each channel.

2.5.2.1.8 Manage Overlap Configurations

A manager needs to retrieve and configure the overlap functions in the ASC. This feature allows a manager to configure the type of overlap operation, the included phases, the modifier phases, and any overlap extensions and clearance times for each overlap.

2.5.2.1.9 Manage Preempt Configurations

A manager needs to retrieve and configure the preempts in the ASC. Preempts are used to service special needs at an intersection, such as for a railroad crossing or emergency vehicles responding to an incident. This feature allows a manager to retrieve and configure the minimum durations, phase settings, outputs and clearance times whenever a preempt signal is detected, how the controller enters into and exits out of preemption, and to define the priority of different preempt inputs into the ASC. This feature also allows a manager to configure the ASC to enable or disable the preempt under certain conditions, such as time-of-day, or to configure the ASC to select alternate exit strategies based on input conditions.

2.5.2.1.10 Manage Timing Pattern Scheduler

A manager needs to retrieve and configure the scheduler in the ASC to implement a timing pattern based on time. This feature allows a manager to configure the ASC to implement timing patterns based on calendar days, days of the week and/or times of day.

2.5.2.1.11 Manage Action Scheduler

A manager needs to retrieve and configure the scheduler in the ASC to perform a function or a group of functions. The action scheduler allows a manager to activate an output, enable a parameter (e.g., max2), configure the ASC log, or program the condition-based exception reporting based on calendar days, days of the week and/or times of day.

For example, a manager may program the action scheduler to activate the special function output every weekday when a nearby school is in session and configure the ASC to operate in non-actuated mode during the same period of time. A manager may also configure the log not to record actuations, and to program the condition-based exception reporting not to report actuations during that same period of time.

2.5.2.1.12 Manage I/O Mapping

A manager needs to retrieve and configure the input/output mapping in the ASC. This feature allows a manager to change the input and outputs for an ASC so unused inputs or outputs, as defined by a standard specification, can be used and configured as needed. This feature also allows a manager to reset the input/output mapping to a default configuration, and configure the conditions when changes to input/output mapping can be accepted by the ASC.

2.5.2.1.13 Manage Intra-Cabinet Communications Configuration

A manager needs to retrieve and configure the ASC’s intra-cabinet communications port. For NEMA TS 2 type controllers, this is the NEMA TS 2 Port 1 in the ASC and allows a manager to indicate if a device is present on Port 1. For controllers in an ATC Cabinet, this is Serial Bus 1 (siuport1).

2.5.2.1.14 Manage Accessible Pedestrian Support

A manager needs to retrieve and configure the ASC to support Accessible Pedestrian Signals (APS). This feature enables an ASC to provide information about pedestrian signal timing to pedestrians via non-visual formats such as audible tones, verbal messages, and/or vibrating surfaces. This may include mobility impaired pedestrians. This feature also allows the ASC to exchange information with pedestrian countdown timers.

2.5.2.2 Monitor Signal Operations Status

This feature allows a manager to monitor the traffic signal operations and status of an ASC. It consists of the following sub-features.

a) Determine Controller Health
b) Determine Mode of Operation
c) Monitor Phase Status
d) Monitor Ring Status
e) Monitor Channel Status
f) Monitor Overlap Status
g) Monitor Preempt Status
h) Monitor Special Function Outputs

i) Monitor Timebase Action Status
j) Monitor Intra-Cabinet Communications Configuration
k) Monitor Signal Monitoring Unit

2.5.2.2.1 Determine Controller Health

A manager needs to monitor the health of the ASC. This feature allows a manager to determine if the essential functions and elements of the ASC are operating properly. ASC system error conditions and faults to be monitored are processor stall conditions (timeouts), memory faults, task (i.e., process) failures, communication timeouts or errors from a management station, and suspect power problems. ASC operational error conditions and faults to be monitored are conflicts, cycle failures, and coordination failures.

2.5.2.2.2 Determine Mode of Operation

A manager needs to determine the current mode of operation in the ASC. It consists of the following sub-features.

2.5.2.2.2.1 Monitor Unit-wide General Operations

A manager needs to determine if the ASC as a unit is operational and monitor operational parameters such as the coordination state, active priority calls, or if an ECLA is currently operational.

2.5.2.2.2.2 Monitor Flashing

A manager needs to determine if the ASC is in a flashing condition and the reason for the flashing condition. If a condition is detected in the controller assembly that may comprise public safety, the ASC generally reverts to a flash condition. This feature allows a manager to determine if the cause of a flash condition is normal (e.g., the ASC was commanded to flash) or if a safety critical condition was detected.

2.5.2.2.2.3 Monitor Current Timing Pattern

A manager needs to retrieve information about the timing pattern, mode of operation and its source (e.g., program entry, time base control, system interface, etc…) running in the ASC. This feature allows a manager to determine the current timing pattern and mode of operation in effect, and the programmed timing pattern and mode of operation (what should be in effect).

2.5.2.2.2.4 Monitor Current Cycle

A manager needs to retrieve information about the current timing pattern cycle in the ASC. This consists of how much time is remaining in the current cycle and the offset currently in effect.

2.5.2.2.3 Monitor Phase Status

A manager needs to retrieve the status of each phase configured in the ASC. This feature indicates if each phase is active or not (including clearance intervals) and if there is an active vehicle or pedestrian call. This feature also indicates which phases are expected to be active after the termination of an active phase. This feature allows a manager to observe and review signal operations.

2.5.2.2.4 Monitor Ring Status

A manager needs to retrieve the status of each ring output configured in the ASC. This feature allows a manager to determine what state (minimum green, extension, yellow change, red clearance, red rest, etc…) and interval the ring is currently in.

2.5.2.2.5 Monitor Channel Status

A manager needs to retrieve the status of each channel output configured in the ASC. This feature allows a manager to determine if each signal indication is red, yellow, green, flashing, or dark, and to display signal indications on a map.

2.5.2.2.6 Monitor Overlap Status

A manager needs to retrieve the status of each overlap configured in the ASC. This feature allows a manager to determine if each overlap is red, yellow or green.

2.5.2.2.7 Monitor Preempt Status

A manager needs to retrieve the status of the preempt state for each preempt input configured in the ASC. For each preempt input, this feature indicates whether an input signal is active, and if the preempt service has started, if the preempt is being delayed, is linked to another preempt sequence, is overriding another preempt sequence, is being overridden by another preempt sequence, the preempt interval (e.g., in dwell), or if the preempt is exiting out of preempt service.

2.5.2.2.8 Monitor Special Function Outputs

A manager needs to retrieve if each special function output configured in the ASC is active. For example, an ASC near a school may use its special function outputs to turn on a flashing beacon to indicate a lower speed limit when a timing pattern associated with traffic arriving and leaving the school are in effect.

2.5.2.2.9 Monitor Timebase Action Status

A manager needs to retrieve which timebase action entry is currently in effect in the ASC.

2.5.2.2.10 Monitor Intra-Cabinet Communications Configuration

A manager needs to retrieve if the ASC's intra-cabinet communications port is online. For NEMA TS 2 type controllers, this is the NEMA TS 2 Port 1 in the ASC. For traffic signal controllers in an ATC Cabinet, this is Serial Bus 1.

2.5.2.2.11 Monitor Signal Monitoring Unit

A manager needs to retrieve the data that an ASC has received from the SMU in the cabinet, including the current measured voltages and electrical current. This feature allows a manager to be aware of potential malfunctions or errors detected by the SMU so a manager may dispatch personnel to remedy the situation or perform preventative maintenance. This feature also allows the ASC to confirm the outputs between what the ASC commanded and actual field conditions, depending on the SMU type.

For example, the SMU may detect a loss of current on a signal circuit. The ASC may be configured to record the information to provide to a manager when it detects a configured loss of current on any signal circuit. A loss of current may be indicative of a potential issue that may require preventative maintenance. The SMU also may drive the cabinet to fail-flash (cabinet flash) when it detects a configured loss of current on any signal circuit.

2.5.2.3 Control Signal Operations

This feature allows a manager to control the signal operation of an ASC. It consists of the following sub-features:

a) Control ASC-wide General Operations
b) Command Timing Pattern

c) Phase Requests
d) Activate Preempt
e) Control Ring Operations
f) Activate Special Function Output
g) Control Frame 40
h) Activate Action Plan
i) Remote Manual Control

2.5.2.3.1 Control ASC-wide General Operations

A manager needs to control ASC-wide operational features within the ASC such as external minimum recalls, automatic detector calls, dimming, interconnect, and enabling/disabling remote commands to the ASC.

2.5.2.3.2 Activate Timing Pattern

A manager needs to command the ASC to a mode of operation, activate a timing pattern or activate a signal plan. This feature allows a manager to command the ASC to a standby mode, to free mode, or to flash, and to establish the system reference point. A manager may do this in the event of inclement weather or special events.

2.5.2.3.3 Phase Requests

A manager needs to control the duration and inclusion of phases for the current (signal) cycle of an ASC. This feature consists of the capability to omit phases, hold phases, force phases off, and to place calls.

2.5.2.3.4 Activate Preempt

A manager needs to activate a preempt input configured in the ASC. This feature allows a manager to force the ASC to request a preempt sequence state for diagnostic purposes or during special events.

2.5.2.3.5 Control Ring Operations

A manager needs to control ring operations of an ASC. This feature allows a manager to stop the ring timing, to activate a force off, or force the ring to rest in red.

2.5.2.3.6 Activate Special Function Output

A manager needs to activate a special function output configured in an ASC. This special function output may be used to activate other devices, such as flashing beacon or a blank out sign associated with a timing pattern.

2.5.2.3.7 Activate Action Plan

A manager needs to activate a pre-defined group of functions configured in an ASC. This feature allows a manager to command the ASC to perform a group of functions. The functions consist of allowing a manager to activate an output, configure the ASC (e.g., maximum2), configure the ASC log, or program the condition-based exception reporting.

2.5.2.3.8 Remote Manual Control

A manager needs to command the ASC to remotely advance the signal controller through the phases or intervals. This feature allows a manager to remotely and manually control a signal controller. Examples of when a manager may wish to manually control an intersection would be for special events, such as sporting events, parades and large concerts, where traffic congestion is far in excess of normal volumes.

2.5.3 Manage Detectors

This section identifies and describes the various features to monitor and control the detector inputs to the ASC. A detector may be used to identify demand for signal service. The user needs to monitor and control detector inputs consist of the following features:

a) Manage Detector Configuration
b) Monitor Detector Status
c) Monitor Detector Health
d) Control Detectors
e) Manage Detector Data
f) Monitor Detector Data from Controller

2.5.3.1 Manage Detector Configuration

A manager needs to retrieve and configure the detectors connected to the ASC. This feature allows a manager to define the travel mode being detected (vehicle, pedestrian, transit, and bicycle), select phase assignments, define capabilities, and define the criteria for detector faults. The criteria for a detector fault consist of the amount of time between detector actuations, amount of time with continuous actuations, and excessive actuations over a period of time.

2.5.3.2 Monitor Detector Status

A manager needs to monitor activations for detectors configured in the ASC. This feature allows a manager to determine the presence of vehicles, pedestrians or other travelers on the roadway.

2.5.3.3 Monitor Detector Health

A manager needs to monitor the health of the detectors configured in the ASC. This feature allows a manager to determine if the detectors are operating correctly or if a fault has been detected so maintenance personnel can be dispatched to repair the detectors if necessary.

2.5.3.4 Control Detectors

A manager needs to control a detector configured in the ASC. This feature allows a manager to clear a detector fault and place the detector back in service, and to activate a call on a detector.

2.5.3.5 Manage Detector Data Collection

A manager needs to set up the ASC to collect data from detectors configured in the ASC. This feature allows a manager to retrieve reports from the ASC on the data measured by the detectors over a user-defined period. This data consists of volumes, occupancies, and speeds as appropriate.

2.5.3.6 Monitor Detector Data from Controller

A manager needs to monitor the detector data received from external detectors. This feature allows a manager to retrieve reports from the ASC on the data reported to the ASC from external detectors.

Note: NTCIP 1209 defines a direct interface between a manager and detector.

2.5.4 Manage Connected Vehicles Interface

This section identifies and describes the various features that support the interface between an ASC and a CV Application Process in a connected vehicle environment. The connected vehicle environment is expected to use the SAE J2735 as the information standard. Several messages in SAE J2735 are pertinent to ASCs and are addressed within NTCIP 1202. These messages are:

a) **Signal Phase and Timing (SPaT) Message.** A broadcasted message providing signal phase and timing information for one or more ASC indicating the state of each permitted intersection maneuver and when an active maneuver terminates. The current signal status is also sent. This message is intended for connected devices in the broadcast vicinity of an ASC.
b) **Basic Safety Message.** A broadcasted message providing "basic" information about the location and movements of a "connected" vehicle, including its current location, speed, acceleration, and direction of travel.
c) **Personal Safety Message.** A broadcasted message providing "basic" information about the location and movements of a "connected" mobile device carried by a VRU, such as a pedestrian, bicyclist or road worker, or integrated in a device used by the VRU, such as a bicycle or wheelchair.

The features offered by an ASC to support the connected vehicle environment are organized by interface:

a) the interface between a management station and the ASC; and
b) the interface between the ASC and the CV Application Process.

2.5.4.1 Connected Vehicle Interface: Management Station – ASC Interface

The following subsections identify and describe the various features that may be offered between a management station and an ASC. These features are:

a) Manage ASC – RSU Interface
b) Manage ASC – RSU Interface Watchdog
c) Manage Signal Phase and Timing Data
d) Manage Assured Green Period

2.5.4.1.1 Manage ASC - RSU Interface

A manager needs to retrieve and configure the interface between the ASC and an RSU. This feature allows a manager to configure operational control information of how often information is exchanged between the ASC and an RSU. There may be an External Device between the ASC and the RSU, in which case this is the ASC – External Device interface.

2.5.4.1.2 Manage ASC - RSU Interface Watchdog

A manager needs to retrieve and configure an RSU interface watchdog within the ASC. This feature allows the ASC to monitor the period of time elapsed between data exchanges across the ASC and RSU interface. If the time elapsed exceeds a configured threshold, the ASC hardware is reset to clear the potential stall condition. There may be an External Device between the ASC and the RSU, in which case this is the watchdog for the ASC – External Device interface.

2.5.4.1.3 Manage Signal Phase and Timing Data

Some of the key applications that have been developed within the connected vehicle environment are related to intersection safety. For signalized intersections, this involves an RSU broadcasting SPaT messages, as defined by SAE J2735, to connected vehicles in the vicinity. Nearly all the data in the SPaT message originates from the ASC, so the ASC needs to exchange this data with the CV Application Process. However, a manager in a traffic management center needs to monitor what data is being provided to the CV Application Process to broadcast to connected devices. This feature allows a manager to manage and view the contents of the signal phase and timing data that the ASC is exchanging with the CV Application Process.

2.5.4.1.4 Manage Assured Green Period

A manager needs to configure the ASC to provide an Assured Green Period (AGP) at connected intersections. This feature allows a manager to define the parameters to calculate the AGP, establish a Red Light Violation Warning (RLVW) detection zone (RDZ), and the ability to provide an assured green end time (AGET) when the intersection is under actuated signal control.

The RLVW application decreases the likelihood that the vehicle will be in the intersection during a red signal indication. When a vehicle is approaching an intersection during a green interval, the RLVW application may provide advisories, warnings, or alerts to the driver that they may not clear the intersection before the signal turns red.

2.5.4.2 Connected Vehicle Interface: ASC – CV Application Process Interface

The following subsections identify and describe the various features that may be offered between an ASC and a CV Application Process. These features are:

a) Exchange Current and Next Movement Information
b) Exchange Next Occurrence of a Movement
c) Exchange Presence of Connected Devices
d) Exchange Roadway Geometrics Information

2.5.4.2.1 Exchange Current and Next Movement Information

An ASC needs to exchange with a CV Application Process what the current and future states of each movement is and when that state will change. This feature allows the ASC to exchange information about when each state of each movement starts and ends. The CV Application Process uses this information for its safety, mobility and environmental applications and to broadcast SPaT messages to connected vehicles and mobile devices. An ASC operating in actuated mode might only be able to provide a time period when an active movement is to terminate. An ASC also may not be able to provide about the next active movement until the end of a current active movement.

2.5.4.2.2 Exchange Next Occurrence of a Movement

An ASC needs to exchange with a CV Application Process the time when each movement will be next permitted (when it is permitted to move again). One of the applications envisioned for the connected vehicle environment is Connected Eco-Driving. This application provides customized real-time driving advice to drivers so that they can adjust their driving behavior to save fuel and reduce emissions. This may include a CV application in the CV Application Process providing recommendations for an optimal speed to equipped vehicles so vehicles arrive at the intersection when the signal indication for their desired movement is green, reducing fuel consumption and emissions created when a vehicle unnecessarily brakes and accelerates.

2.5.4.2.3 Exchange Presence of Connected Devices

An ASC needs to exchange with a CV Application Process the presence of connected devices on the roadway around the ASC in support of the AGP. This feature allows the ASC to exchange with the CV Application Process information that can be used as a call for actuated movements or to determine the demand for specific movements. The BSM and the PSM are the primary sources of presence information that are received by a CV Application Process located near the ASC and then exchanged with the ASC.

2.5.4.2.4 Exchange Roadway Geometrics Information

An ASC needs to exchange with a CV Application Process what roadway geometry plan is currently in effect at the intersection. Each roadway geometry plan defines the pathways where movements are permitted at the intersection when that roadway geometry plan is in effect. A pathway may be a vehicle lane, a pedestrian crossing, a bicycle lane, or a transit right of way. This feature allows the ASC to exchange with the CV Application Process when the roadway geometry plan in effect in the ASC has changed. The ASC uses this information to confirm that the roadway geometry plan is compatible with the signal operations timing plan in effect.

For example, an ASC may be programmed to use signal timing plans with an intersection roadway geometry with one-way approaches into the intersection. However, if the intersection roadway has been changed to two-way traffic, the ASC needs to confirm that the signal timing plan in effect is still compatible with the new roadway geometry plan that the CV Application Process is broadcasting to travelers.

2.5.4.3 ASC – ECLA Interface

An ASC needs to exchange with an ECLA on the current and future states of each movement and when the current state will change. This feature allows the ASC to receive information from the ECLA on when each state of each movement starts and ends when the ECLA is asserting a higher-level control over the ASC. The ASC uses this information to broadcast SPaT messages to connected vehicles and mobile devices.

2.5.5 Backward Compatibility Features

Prior versions of NTCIP 1202 use SNMPv1 as its application layer protocol, while NTCIP 1202 v04 uses SNMPv3 as its application layer protocol. A shortcoming of SNMPv1 is its lack of security – thus *NTCIP 9014, Infrastructure Standards Security Assessment (ISSA)*, published in 2021, recommended migrating the NTCIP center-to-field standards from SNMPv1 to SNMPv3. However, an authentication feature in SNMPv3 breaks backward compatibility for every SNMPv1 message exchanged, thus NTCIP 1202 v04 is not backward compatible with previous versions of NTCIP 1202, so this section is not applicable.

2.6 Security

Section 2.6 identifies and describes the various security features that may be offered by the ASC. It consists of the following sub-features:

a) Manage Authentication
b) Manage Accessibility
c) Manage Users
d) Log User Access
e) Manage ASC Interface Security

2.6.1 Manage Authentication

A manager needs to retrieve and configure the ASC to authenticate requests from a manager. This feature allows a manager to authenticate users and passwords in the ASC.

2.6.2 Manage Accessibility

A manager needs to retrieve and configure the ASC to limit access to specific information in the ASC based on the permissions assigned by a manager.

2.6.3 Manage Users

A manager needs to retrieve and configure a user’s profile in the ASC. Each user profile consists of a user, its password and its access rights.

2.6.4 Log User Access

A manager needs to retrieve and configure the ASC to log when and what requests were made by a manager. This feature allows a manager to track who made what changes to the ASC security configuration, or commanded the ASC to perform a security-related function. This feature is only accessible by a system administrator.

2.6.5 Manage ASC Interface Security

The following subsections identify and describe the various security features for the interfaces between the ASC and other devices and processes. These features are:

2.6.5.1 Manage Security for the ASC to RSU Interface

A manager needs to manage the security features for the ASC to the RSU communications interface as defined by other ITS standards. Other ITS standards may impose additional needs on this communications interface.

2.6.5.2 Manage Security for Other ASC Interfaces

A manager needs to manage the security features for the communications interface between the ASC and other devices or processes. These security features may be defined by other ITS standards.

2.7 Operational Policies and Constraints

The ASC WG recognize that the following constraints may apply.

- Public agencies may implement operational policies, rules, or regulations that takes precedence over the use of this standard. No governmental operational policies, rules or regulation shall be violated when applying NTCIP 1202.
- It is the operational policy of some agencies that authorized personnel is/are present at the physical location of the ASC, before an ASC accepts a change to the configuration of the ASC. This operational policy is usually enforced by requiring that the door of the transportation cabinet containing the ASC is open.
- The operation and maintenance of the connected signalized intersection uses the traffic signal timing principles and practices that have guided signal timing operations for many decades. Many of these principles and practices have been studied, researched, and time tested. Significant changes to these principles and practices may require additional studies and research before they can be adopted and deployed.

2.8 Relationship to the ITS National Architecture [Informative]

Architecture Reference for Cooperative and Intelligent Transportation, known as ARC-IT, combines the National ITS Architecture and the Connected Vehicle Reference Implementation Architecture (CVRIA). NTCIP 1202 v04 addresses many ARC-IT flows associated with the operation of an ASC.

NTCIP 1202 v04 addresses fourteen (14) ARC-IT flows between a Traffic Management Center (TMC) and a Traffic Signal Controller (ITS Roadway Equipment (IRE)) that are associated with the operation of an ASC. These flows are:

a) **Rail Crossing Control Data.** Data required for Highway-Rail Intersection (HRI) information transmitted at railroad grade crossings and within railroad operations.
b) **Rail Crossing Request.** A request for highway-rail intersection status or a specific control request intended to modify HRI operation.
c) **Rail Crossing Status.** Status of the highway-rail intersection equipment including both the current state or mode of operation and the current equipment condition.

d) **Right-of-Way Request Notification.** Notice that a request has occurred for signal prioritization, signal preemption, pedestrian call, multi-modal crossing activation, or other sources for right-of-way requests.
e) **Signal Control Commands.** Control of traffic signal controllers or field masters including clock synchronization.
f) **Signal Control Coordination.** The direct flow of information between field equipment. This includes configuration and control of traffic signal controllers and field masters. Configuration data and operational status of traffic signal control equipment including operating condition and current indications are returned.
g) **Signal Control Device Configuration.** Data used to configure traffic signal control equipment including local controllers and system masters.
h) **Signal Control Plans.** Traffic signal timing parameters including minimum green time and interval durations for basic operation and cycle length, splits, offset, phase sequence, etc. for coordinated systems.
i) **Signal Control Status.** Operational and status data of traffic signal control equipment including operating conditions and current indications.
j) **Signal Fault Data.** Faults from traffic signal control equipment.
k) **Signal System Configuration.** Data used to configure traffic signal systems including configuring control sections and mode of operation (time-based or traffic responsive).
l) **Traffic Detector Data.** Raw and/or processed traffic detector data which allows derivation of traffic flow variables (e.g., speed, volume, and density measures) and associated information (e.g., congestion, potential incidents). This flow includes the traffic data and the operational status of the traffic detectors.
m) **Traffic Detector Control.** Information used to configure and control traffic detector systems such as inductive loop detectors and machine vision sensors.

NTCIP 1202 v04 also addresses fifteen (15) ARC-IT flows between a Traffic Signal Controller, represented as an ITS Roadway Equipment (IRE), and an RSU, represented as a Connected Vehicle Roadside Equipment (CVRE). These flows are:

a) **Arriving Train Information.** Information for a train approaching a highway-rail intersection that may include direction and allow calculation of approximate arrival time and closure duration.
b) **Conflict Monitor Status.** A control flow that supports failsafe operation in the event that a conflict is detected that requires the RSE to enter a failsafe operating mode.
c) **Intersection Control Status.** Status data provided by the traffic signal controller including phase information, alarm status, and priority/preempt status.
d) **Intersection Infringement Info.** Vehicle path information sent by a vehicle that is violating the stop bar at an intersection. This flow includes the vehicle's position, heading, speed, acceleration, transmission, steering-wheel angle, braking status, size information, and trajectory.
e) **Intersection Status Monitoring.** Current signal phase and timing information for all lanes at a signalized intersection. This flow identifies monitoring of communications by a receiver at the intersection to support monitoring for conflicts between actual signal states and RSE communications about those states.
f) **Mixed Use Crossing Status.** Current pedestrian and other non-motorized user locations including an indication of whether the call button has been activated, the current state of the mixed-use crossing signal, and information indicating whether non-motorized users are currently occupying the cross walk.
g) **Personal Location Information.** Pedestrian, bicyclist, and other non-motorized user locations at an intersection as detected and reported by an RSE.
h) **Signal Preemption Request.** Direct request for preemption to a traffic signal controller that results in preemption of the current control plan and grants right-of-way to the requesting vehicle. This flow identifies the required phase and timing of the preemption. This flow may also cancel the preemption request (e.g., when the requesting vehicle clears the intersection).
i) **Signal Service Request.** A call for service or extension for a signal control phase that is issued by the RSE for connected vehicles approaching an intersection and/or pedestrians at a crosswalk. This flow identifies the desired phase and service time.

j) **Track Status.** Current status of the wayside equipment and notification of an arriving train.
k) **Traffic Situation Data.** Current, aggregate traffic data collected from connected vehicles that can be used to supplement or replace information collected by roadside traffic detectors. It includes raw and/or processed reported vehicle speeds, counts, and other derived measures. Raw and/or filtered vehicle control events may also be included to support incident detection.
l) **Vehicle Entries and Exit.** Information exchanged between an RSE and ITS Roadway Equipment (ASC) that supports detection of non-equipped vehicles in an automated lane, low emissions zone, or other facility where V2I communications is used to monitor vehicles at entry or exit points. This exchange also supports identification of non-equipped vehicles where an RSE is used for payment collection. This generic exchange can be implemented by any approach that compares vehicle detections with V2I communications by the RSE to identify vehicles that are not equipped or are otherwise unable to communicate with the RSE.

Section 3
Functional Requirements [Normative]

Section 3 defines the Functional Requirements based on the user needs identified in the Concept of Operations (Section 2). Section 3 includes:

a) A tutorial
b) Protocol Requirements List (PRL). A Functional Requirement is a requirement of a given function and therefore is only required to be implemented if the associated functionality (e.g., user need) is selected through the use of the PRL. The PRL also indicates which of the items are mandatory, conditional, or optional. The PRL can be used by procurement personnel to specify the desired features of an ASC system or can be used by a manufacturer to document the features supported by their implementation.
c) Architectural Requirements. These are requirements related to the architectural needs defined in Section 2.4.
d) Data Exchange and Operational Environment Requirements. These are requirements related to the features identified in Section 2.5 that can be realized through a data exchange. For example, the requirement to be able to monitor what signal indications are active.
e) Supplemental Non-communications Requirements. These are additional requirements derived from the Concept of Operations that do not fall into one of the above two categories. For example, they include requirements related to performance requirements.
f) Generic Requirements. There are requirements that are generic to all NTCIP field devices. For example, clock synchronization of devices is a requirement that is considered generic to all NTCIP devices. These requirements can be found in Annex G.

Section 3 is intended for all readers, including:

a) Transportation operations managers
b) Transportation operations personnel
c) Transportation engineers
d) System integrators
e) Device manufacturers

For the first three categories of readers, Section 3 is useful in understanding the details that NTCIP 1202 v04 requires of an ASC. For these readers, Section 3.3.3 is particularly useful in preparing procurement specifications and assist in mapping the various rows of this table to the more detailed text contained within the other sections.

For the last two categories of readers, this section is useful to fully understand what is required of equipment meeting this interface standard. The table in Section 3.3.3 may be used to document the capabilities of their implementations.

3.1 Tutorial [Informative]

This Functional Requirements section defines the formal requirements that are intended to satisfy the user needs identified in Section 2. This is achieved through the development of a PRL that traces each user need to one or more requirements defined in this section. The details of each requirement are then presented following the PRL. The functional requirements are presented in three broad categories as follows:

a) Architectural Requirements. These requirements define the required behavior of the system in exchanging data across the communications interface, including any restrictions to general architectural requirements, based upon the architectural needs identified in the Concept of Operations.
b) Data Exchange Requirements. These requirements define the required behavior of the system in exchanging data across the communications interface based upon the features identified in the Concept of Operations.
c) Supplemental Requirements. These requirements define additional requirements of the system that are derived from the architectural and/or data exchange requirements, but are not themselves architectural or data exchange requirements. A given supplemental requirement may relate to multiple architectural and/or data exchange requirements. Supplemental requirements include capabilities of the equipment (e.g., service processing or clearing expired priority requests).

3.2 Scope Of the Interface [Informative]

<In the opinion of the responsible NTCIP working group, this section does not apply in the context of NTCIP 1202 v04.>

3.3 Protocol Requirements List (PRL)

The PRL, provided in Table 5

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.3	Reference Physical Architecture [Informative]					
2.3.1	ASC Characteristics – Cabinet Specifications [Normative]			M	Yes	
2.3.1.a (332)	Model 332 Cabinet			O.1 (1)	Yes / No	
2.3.1.b (TS1)	NEMA TS 1 Cabinet			O.1 (1)	Yes / No	
2.3.1.c (TS2-2)	NEMA TS 2 Type 2 Cabinet			O.1 (1)	Yes / No	
2.3.1.d (TS2-1)	NEMA TS 2 Type 1 Cabinet			O.1 (1)	Yes / No	
2.3.1.e (ATC)	ATC Cabinet			O.1 (1)	Yes / No	
2.3.2	ASC Characteristics – Controller Types [Normative]			M	Yes	
2.3.2.a	Phase-based controller			M	Yes	
2.3.2.b	Interval-based controller			NA	NA	Interval-based controllers are not supported by NTCIP 1202 v04
2.3.2.c	Stage-based controller			NA	NA	Stage-based controllers are not supported by NTCIP 1202 v04
2.4	Architectural Needs					
2.4.1	Provide Live Data			M	Yes	
		3.4.1.1	Retrieve Data	M	Yes	
		3.4.1.2	Deliver Data	M	Yes	
		3.4.1.3	Explore Data	M	Yes	
		3.4.1.4.1	Monitor SNMP Information	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.5.2.6	Determine basic capabilities of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.7	Determine SNMP capabilities of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.8	Determine the SNMP engine identifier	M	Yes	
		ISO 26048-1§ 8.5.2.9	Monitor when the SNMP capabilities last changed	M	Yes	
		ISO 26048-1§ 8.5.4	Controller performance requirements	M	Yes	
		3.6.1	Response Time for Requests	M	Yes	The Response Time for all requests shall be ___ milliseconds (5-500: Default=25). Note: Redundant with the previous requirement.
		~~3.6.2~~	Atomic Operations	M	Yes	Note: should 3.6.2 be deleted?
2.4.2	Provide Data Blocks			O	Yes / No	
		3.4.2.1	Store Pre-defined Compressed Data Blocks	M	Yes	
		3.5.2.1.14.1.1	Configure Block Object Get Control - Phase Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.2	Configure Block Object Get Control - Vehicle Detector Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.3	Configure Block Object Get Control - Pedestrian Detector Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.4	Configure Block Object Get Control - Pattern Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.5	Configure Block Object Get Control - Split Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.6	Configure Block Object Get Control - Overlap Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.7	Configure Block Object Get Control - Preempt Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.8	Configure Block Object Get Control - Sequence Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.9	Configure Block Object Get Control - Channel Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.2	Monitor Block Error Status - Error-causing Data Element	M	Yes	
		ISO 26048-1§ 8.7	Dynamic object feature	O	Yes / No	All requirements that trace to this feature are also inherited. See the Features To Requirements

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						Traceability (FTRT) in ISO 26048-1.
2.4.3	Provide for Log Data Local Storage and Retrieval			O.3 (1..*)	Yes / No	Note: Either Logging or Exception Reporting is Mandatory in ISO 26048-1.
		ISO 26048-1§ 8.9	File Feature	M	Yes	
		ISO 26048-1§ 8.10	Logging feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.4.4	Provide for Database Management			M	Yes	
		ISO 26048-1§ 8.21	Transaction feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.4.5	Condition-based Exception Reporting			O.3 (1..*)	Yes / No	Note: Either Logging or Exception Reporting is Mandatory in ISO 26048-1.
		ISO 26048-1§ 8.1	Action feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.11	Notification feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.16	SNMP target feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5	Features					
2.5.1	Manage the ASC Configuration			M	Yes	
2.5.1.1	Manage Device Identity			M	Yes	
		3.5.1.1.1	Configure ASC Location - Antenna Offset	C	Yes / NA	Mandatory if an external GNSS

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						device is attached to the ASC.
		ISO 26048-1§ 8.5.2.1	Configure the controller's identity	M	Yes	
		ISO 26048-1§ 8.5.2.2	Configure the default language	M	Yes	
		ISO 26048-1§ 8.5.2.5	Control remote reset of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.14	Obtain the field device configuration identifier	M	Yes	
		ISO 26048-1§ 8.5.3.1	Support maximum message size	M	Yes	The maximum SNMP message size shall be _____ bytes (Default: 484 bytes).
		ISO 26048-1§ 8.5.3.2	Support total memory	O	Yes / No	The controller shall support a minimum of _____ kilobytes of total memory.
		ISO 26048-1§ 8.5.3.3	Support changeable memory	O	Yes / No	The controller shall support a minimum of _____ kilobytes of changeable memory.
		ISO 26048-1§ 8.8.2.1.1	Configure the field device's location by providing coordinates	M	Yes	
		ISO 26048-1§ 8.8.2.1.2	Configure the field device's location by copying coordinates	O	Yes / No	
		ISO 26048-1§ 8.8.2.1.3	Configure the field device's location by GNSS reading	O	Yes / No	
		ISO 26048-1§ 8.8.2.1.4	Configure the field device's physical components	M	Yes	
		ISO 26048-1§ 8.12	Owner feature	M	Yes	
2.5.1.2	Manage Communications			O	Yes / No	Propose to delete.
		3.5.1.2.1.1	Enable/Disable Communications Port	M	Yes	The ASC shall not be allowed to enable/disable the following ports numbers: _________
		3.5.1.2.2.1	Determine Number of ASC Communications Ports	M	Yes	
2.5.1.3	Manage Cabinet Environment			O	Yes / No	
		ISO 26048-1§ 8.8.2.1.4	Configure the cabinet's physical components	M	Yes	
		ISO 26048-1§ 8.8.12	Field device doors	M	Yes	All requirements that trace to this feature are also

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						inherited. See the FTRT in ISO 26048-1. Minimum number of doors to be supported (Default: 2) ____
		ISO 26048-1§ 8.8.13	Field device fans	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of fans supported (Default: 2) _____
		ISO 26048-1§ 8.8.15	Field device heaters	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of heaters supported (Default: 1) _____
		ISO 26048-1§ 8.8.16	Field device humidity	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of humidity sensors supported (Default: 1) _____
		ISO 26048-1§ 8.8.21	Field device temperature	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of temperature sensors supported (Default: 1) _____
2.5.1.4 (Power)	Monitor Power			O	Yes / No	
		ISO 26048-1§ 8.8.2.1.9	Monitor current power source	M	Yes	
		ISO 26048-1§ 8.8.2.2.1.a	Support power sources – mainline (alternating current) power	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.8.2.2.1.b	Support power sources – battery power	UPS:M	Yes / NA	
		ISO 26048-1§ 8.8.2.2.1.c (Generator)	Support power sources – generator power	O	Yes / No	
		ISO 26048-1§ 8.2.2.2.1.d (Solar)	Support power sources – solar power	O	Yes / No	
		ISO 26048-1§ 8.8.2.2.2 (UPS)	Support UPS power	O	Yes / No	
		ISO 26048-1§ 8.8.10	Field device battery	UPS:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet battery supported (Default: 1) _____
		ISO 26048-1§ 8.8.14	Field device generator	Generator:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet generator supported (Default: 1) _____
		ISO 26048-1§ 8.8.17	Field device mains power	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet mains power supported (Default: 1) _____
		ISO 26048-1§ 8.8.20	Field device solar power	Solar:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						the cabinet solar power system supported (Default: 1) ____
2.5.1.5	Retrieve Operational Performance Data			O	Yes / No	
		ISO 26048-1§ 8.13	Recording feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.1.6	Manage Auxiliary External Inputs/Outputs			O	Yes / No	
		ISO 26048-1§ 8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Number of additional SRSA ports to support: ____. This value is exclusive of ports used to monitor cabinet power, cabinet doors, fans, heaters, temperature sensor and humidity sensors.
2.5.1.7	Manage Database			M	Yes	
		3.5.1.7.1	Determine Configuration Identifier Parameter Content	M	Yes	
		ISO 26048-1§ 8.5.2.14	Obtain the field device configuration identifier	M	Yes	
2.5.1.8	Manage Interface with External Detectors			O	Yes / No	See 2.5.3.5 for detector data collected by the ASC through external inputs. Requirements to view advanced detector data are found in NTCIP 1209.
2.5.1.9	Manage ASC Clock			M	Yes	
		ISO 26048-1§ 8.2.1	UTC clock	M	Yes	All requirements that trace to this feature are also inherited. See the

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						FTRT in ISO 26048-1.
		ISO 26048-1§ 8.2.2	Local clock	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.2.3	Daylight saving time	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.1.10	Manage External Control Local Application State			O	Yes / No	
		3.5.1.10.1.1	Enable ECLA Communications	M	Yes	
		3.5.1.10.1.2	Disable ECLA Communications	M	Yes	
		3.5.1.10.2	Monitor ECLA Data Input Time	M	Yes	
2.5.2	Manage Signal Operations			M	Yes	
2.5.2.1	Manage Signal Configuration			M	Yes	
2.5.2.1.1	Manage Controller Startup Functions			M	Yes	
		3.5.2.1.1.1.1	Configure Start-Up Flash Mode	O	Yes / No	
		3.5.2.1.1.1.2	Configure Start-Up Flash Time	M	Yes	
		3.5.2.1.1.2	Configure Backup Time	M	Yes	
2.5.2.1.2	Manage Phase Configurations			M	Yes	
		3.5.2.1.2.1.1	Enable/Disable Phase	M	Yes	
		3.5.2.1.2.1.2	Configure Phase Minimum Green Time	M	Yes	
		3.5.2.1.2.1.3	Configure Phase Passage Time	M	Yes	
		3.5.2.1.2.1.4	Configure Two Fixed Phase Maximum Green Times	M	Yes	
		3.5.2.1.2.1.5	Configure Three Fixed Phase Maximum Green Times	O	Yes / No	
		3.5.2.1.2.1.6	Configure Phase Yellow Change Time	M	Yes	
		3.5.2.1.2.1.7	Configure Phase Red Clearance Time	M	Yes	
		3.5.2.1.2.1.8	Configure Phase Red Revert Time	O	Yes / No	
		3.5.2.1.2.1.9	Configure Unit Red Revert Time	Unit:M	Yes / NA	
		3.5.2.1.2.1.10	Configure Phase Added Initial Time	M	Yes	
		3.5.2.1.2.1.11	Configure Phase Maximum Initial Time	M	Yes	
		3.5.2.1.2.1.12	Configure Phase Time Before Reduction	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.2.1.13	Configure Phase Time to Reduce	M	Yes	
		3.5.2.1.2.1.14	Configure Phase Cars Before Reduction	O	Yes / No	
		3.5.2.1.2.1.15	Configure Phase Reduce By Time	O	Yes / No	
		3.5.2.1.2.1.16	Configure Phase Minimum Gap Time	M	Yes	
		3.5.2.1.2.1.17	Configure Phase Dynamic Maximum Limit	O	Yes / No	
		3.5.2.1.2.1.18	Configure Phase Dynamic Maximum Step	O	Yes / No	
		3.5.2.1.2.1.19	Configure Phase Start-Up State	M	Yes	
		3.5.2.1.2.1.20	Configure Automatic Flash Entry Phase	O	Yes / No	
		3.5.2.1.2.1.21	Configure Automatic Flash Exit Phase	O	Yes / No	
		3.5.2.1.2.1.22	Configure Call to Non-Actuated 1	O	Yes / No	
		3.5.2.1.2.1.23	Configure Call to Non-Actuated 2	O	Yes / No	
		3.5.2.1.2.1.24	Configure Non-Lock Detector Memory	O	Yes / No	
		3.5.2.1.2.1.25	Configure Phase Minimum Vehicle Recall	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.26	Configure Phase Maximum Vehicle Recall	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.27	Configure Phase Soft Vehicle Recall	O	Yes / No	
		3.5.2.1.2.1.28	Configure Dual Phase Entry	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.29	Configure Simultaneous Gap Disable	O	Yes / No	
		3.5.2.1.2.1.30	Configure Guaranteed Passage	O	Yes / No	
		3.5.2.1.2.1.31	Configure Actuated Rest-in-Walk	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.32	Configure Conditional Service Enable	O	Yes / No	
		3.5.2.1.2.1.33	Configure Added Initial Calculation	O	Yes / No	
		3.5.2.1.2.1.34	Configure Phase-to-Ring Association	M	Yes	
		3.5.2.1.2.1.35	Configure Phase Concurrency	M	Yes	
		3.5.2.1.2.1.36	Configure Pedestrian Clearance Time Allowed During Vehicle Clearance	O	Yes / No	
		3.5.2.1.2.1.37	Configure Pedestrian Walk Time	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.2.1.38	Configure Pedestrian Clearance Time	M	Yes	
		3.5.2.1.2.1.39	Configure Pedestrian Phase Walk Service Limit	M	Yes	
		3.5.2.1.2.1.40	Configure Pedestrian Phase Don't Walk Revert Time	M	Yes	
		3.5.2.1.2.1.41	Configure Non-Lock Ped Detector Memory	M	Yes	
		3.5.2.1.2.1.42	Configure Pedestrian Phase Recall	M	Yes	
		3.5.2.1.2.1.43	Configure Phase Alternate Pedestrian Clearance Time	O	Yes / No	
		3.5.2.1.2.1.44	Configure Phase Alternate Pedestrian Walk Time	O	Yes / No	
		3.5.2.1.2.1.45	Configure Pedestrian Phase Advanced Walk Time	O	Yes / No	
		3.5.2.1.2.1.46	Configure Pedestrian Phase Delayed Walk Time	O	Yes / No	
		3.5.2.1.2.1.47	Configure Phase Advance Warning Green	O	Yes / No	
		3.5.2.1.2.1.48	Configure Phase Advance Warning Red	O	Yes / No	
		3.5.2.1.2.1.49	Configure Flashing Yellow Arrow Associated Vehicle Phase	O	Yes / No	
		3.5.2.1.2.1.50	Configure Flashing Red Arrow Associated Vehicle Phase	O	Yes / No	
		3.5.2.1.2.1.51	Configure Alternate Minimum Green Time during Transition	O	Yes / No	
		3.5.2.1.2.1.52	Configure Alternate Minimum Walk Time during Transition	O	Yes / No	
		3.5.2.1.2.1.53	Configure Alternate Minimum Pedestrian Clearance Time during Transition	O	Yes / No	
		3.5.2.1.2.2(PhaseSet)	Configure Multiple Phase Sets	O	Yes / No	
		3.5.2.1.2.3.1	Determine Maximum Number of Phases	M	Yes	The ASC shall support at least _____ phases.
		3.5.2.1.2.3.2	Determine Maximum Number of Phase Sets	PhaseSet:M	Yes / NA	The ASC shall support at least _____ phase sets.
2.5.2.1.3 (Coord)	Manage Coordination Configurations			O	Yes / No	
		3.5.2.1.3.1	Configure Operational Mode for Coordination	M	Yes	The ASC shall support the following values: ____ automatic ____ pattern ____ manual free ____ manual flash

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.3.2	Configure Correction Mode for Coordination	M	Yes	The ASC shall support the following values: __ dwell ___ shortway __ add only ___ subtract only
		3.5.2.1.3.3	Configure Maximum Mode for Coordination	M	Yes	The ASC shall support the following values: ___ maxInhibit ___maximum1 ___ maximum2 ___maximum3
		3.5.2.1.3.4	Configure Unit-level Force Mode for Coordination	O	Yes / No	
		3.5.2.1.3.5	Configure Phase-level Force Mode for Coordination	O	Yes / No	
		3.5.2.1.3.6	Configure Pattern Reference Phase	M	Yes	
		3.5.2.1.3.7	Configure Pattern Reference Point	M	Yes	
		3.5.2.1.3.9	Configure Pattern Synchronization Time	M	Yes	
2.5.2.1.4	Manage Timing Patterns			Coord:M	Yes / NA	
		3.5.2.1.4.1.1	Configure Pattern to Run Free	M	Yes	
		3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	O	Yes / No	
		3.5.2.1.4.1.3	Configure Pattern Cycle Time for Coordination	M	Yes	
		3.5.2.1.4.1.4	Configure Pattern Offset Time	M	Yes	
		3.5.2.1.4.1.5	Configure Pattern Split Association	M	Yes	
		3.5.2.1.4.1.6	Configure Pattern Sequence Association	M	Yes	
		3.5.2.1.4.1.7	Configure Pattern Maximum Mode	O	Yes / No	
		3.5.2.1.4.1.8	Configure Pattern Phase Set	O	Yes / No	
		3.5.2.1.4.1.9	Configure Pattern Overlap Set	O	Yes / No	
		3.5.2.1.4.1.10	Configure Pattern Vehicle Detector Set	O	Yes / No	
		3.5.2.1.4.1.11	Configure Pattern Pedestrian Detector Set	O	Yes / No	
		3.5.2.1.4.1.12	Configure Pattern Special Functions	O	Yes / No	
		3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	M	Yes	The ASC shall support at least

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						____ timing patterns.
2.5.2.1.5	Manage Splits Configurations			O	Yes / No	
		3.5.2.1.3.8	Configure Omit Phases During Transitions	O	Yes / No	
		3.5.2.1.5.1.1	Configure Phase Split Time	M	Yes	
		3.5.2.1.5.1.2	Configure Phase Split Mode	M	Yes	
		3.5.2.1.5.1.3	Configure Split Coordination Phase	M	Yes	
		3.5.2.1.5.2	Determine Maximum Number of Phase Splits	M	Yes	The ASC shall support at least ____ splits
2.5.2.1.6 (Ring)	Manage Ring Configurations			O	Yes / No	
		3.5.2.1.6.1	Configure Sequence Data	M	Yes	
		3.5.2.1.6.2	Determine Maximum Number of Rings	M	Yes	The ASC shall support at least ____ rings
		3.5.2.1.6.3	Determine Maximum Number of Sequences	M	Yes	The ASC shall support at least ____ sequences
2.5.2.1.7 (Channel)	Manage Channel Configurations			O	Yes / No	
		3.5.2.1.7.1.1	Configure Channel Control Source	M	Yes	
		3.5.2.1.7.1.2	Configure Channel Control Type	M	Yes	
		3.5.2.1.7.1.3.1	Configure Channel Flash Yellow	M	Yes	
		3.5.2.1.7.1.3.2	Configure Channel Flash Red	M	Yes	
		3.5.2.1.7.1.3.3	Configure Channel Flash Alternate Half Hertz	O	Yes / No	
		3.5.2.1.7.1.3.4	Configure Channel Flash Alternate First or Second	O	Yes / No	
		3.5.2.1.7.2	Determine Maximum Number of Channels	M	Yes	The ASC shall support at least ____ channels (See appropriate hardware specification such as NEMA TS 2 to determine maximum number of supported channels)
2.5.2.1.8 (Overlap)	Manage Overlap Configurations			O	Yes / No	
		3.5.2.1.8.1.1.1	Configure Overlap Type - Vehicle Normal	O.4 (1..*)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.8.1.1.2	Configure Overlap Type - Vehicle Minus Green and Yellow	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.3	Configure Overlap Type - Pedestrian Normal	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.4	Configure Overlap Type - Flashing Yellow Arrow - 3 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.5	Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.6	Configure Overlap Type - Flashing Red Arrow - 3 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.7	Configure Overlap Type - Flashing Red Arrow - 4 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.8	Configure Overlap Type - 2 Section Transit Specific Signal Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.9	Configure Overlap Type - Minus Green Yellow Alternate	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.2	Configure Overlap Included Phases	M	Yes	
		3.5.2.1.8.1.3	Configure Overlap Modifier Phases	M	Yes	
		3.5.2.1.8.1.4	Configure Pedestrian Modifier Phases	O	Yes / No	
		3.5.2.1.8.1.5	Configure Overlap Trailing Green	M	Yes	
		3.5.2.1.8.1.6	Configure Overlap Trailing Yellow	M	Yes	
		3.5.2.1.8.1.7	Configure Overlap Trailing Red Clearance	M	Yes	
		3.5.2.1.8.1.8	Configure Overlap Walk	O	Yes / No	
		3.5.2.1.8.1.9	Configure Overlap Pedestrian Clearance	O	Yes / No	
		3.5.2.1.8.2 (OverlapSet)	Configure Multiple Overlap Sets	O	Yes / No	
		3.5.2.1.8.3.1	Determine Maximum Number of Overlaps	M	Yes	The ASC shall support at least ____ overlaps
		3.5.2.1.8.3.2	Determine Maximum Number of Overlap Sets	Overlap Set:M	Yes / NA	The ASC shall support at least ____ overlap sets.
2.5.2.1.9 (Preempt)	Manage Preempt Configurations			O	Yes / No	
		3.5.2.1.9.1.1	Enable/Disable Preempt Inputs	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.9.1.2	Configure Preempt Control - Non-Locking Memory	O	Yes / No	
		3.5.2.1.9.1.3	Configure Preempt Control - Override Automatic Flash	O	Yes / No	
		3.5.2.1.9.1.4	Configure Preempt Control - Override Preempt	O	Yes / No	
		3.5.2.1.9.1.5	Configure Preempt Control - Flash Dwell	O	Yes / No	
		3.5.2.1.9.1.6	Configure Preempt Control - All Red Entry	O	Yes / No	
		3.5.2.1.9.1.7	Configure Preempt Link	M	Yes	
		3.5.2.1.9.1.8	Configure Preempt Delay	M	Yes	
		3.5.2.1.9.1.9	Configure Preempt Minimum Duration	M	Yes	
		3.5.2.1.9.1.10.1	Configure Preempt Enter Minimum Green Time	O	Yes / No	
		3.5.2.1.9.1.10.2	Configure Preempt Enter Minimum Walk Time	O	Yes / No	
		3.5.2.1.9.1.10.3	Configure Preempt Enter Pedestrian Clearance Time	O	Yes / No	
		3.5.2.1.9.1.10.4	Configure Preempt Enter Yellow Change Time	O	Yes / No	
		3.5.2.1.9.1.10.5	Configure Preempt Enter Red Clearance Time	O	Yes / No	
		3.5.2.1.9.1.11.1	Configure Preempt Track Clearance Green Time	M	Yes	
		3.5.2.1.9.1.11.2	Configure Preempt Track Clearance Yellow Change Time	O	Yes / No	
		3.5.2.1.9.1.11.3	Configure Preempt Track Red Clearance Time	O	Yes / No	
		3.5.2.1.9.1.11.4	Configure Preempt Track Clearance Phases	M	Yes	
		3.5.2.1.9.1.11.5	Configure Preempt Track Clearance Overlaps	O	Yes / No	
		3.5.2.1.9.1.12.1	Configure Preempt Minimum Green Dwell Time	M	Yes	
		3.5.2.1.9.1.12.2	Configure Preempt Dwell Phases	M	Yes	
		3.5.2.1.9.1.12.3	Configure Preempt Dwell Pedestrian Movements	O	Yes / No	
		3.5.2.1.9.1.12.4	Configure Preempt Dwell Overlaps	O	Yes / No	
		3.5.2.1.9.1.12.5	Configure Preempt Cycling Phases	O	Yes / No	
		3.5.2.1.9.1.12.6	Configure Preempt Cycling Pedestrian Movements	O	Yes / No	
		3.5.2.1.9.1.12.7	Configure Preempt Cycling Phases Sequence	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.9.1.12.8	Configure Preempt Cycling Overlaps	O	Yes	
		3.5.2.1.9.1.13.1	Configure Preempt Exit Phases	O	Yes / No	
		3.5.2.1.9.1.13.2	Configure Preempt Exit Phase Strategy	O	Yes / No	
		3.5.2.1.9.1.13.3	Configure Preempt Exit Priority Levels	C	Yes / No	Mandatory for Conformance if 'Exit to Queue Delay Recovery' is supported as a Preempt Exit Phase (See 3.5.2.1.9.1.13.2)
		3.5.2.1.9.1.14.1	Configure Preempt Maximum Presence Time	M	Yes	
		3.5.2.1.9.1.14.2	Configure Preempt Maximum Presence Action	M	Yes	
		3.5.2.1.9.1.15	Configure Preempt Gate Description	O	Yes / No	
		3.5.2.1.9.2	Determine Maximum Number of Preempts	M	Yes	The ASC shall support at least ____ preempts
2.5.2.1.10	Manage Timing Pattern Scheduler			M	Yes / No	
		ISO 26048-1§ 8.1.2.3	Configure an action owner	M	Yes	
		ISO 26048-1§ 8.1.2.4	Configure an action group	M	Yes	
		ISO 26048-1§ 8.1.2.5	Configure an action	M	Yes	
		ISO 26048-1§ 8.6	Day plan feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.2.1.11	Manage Action Scheduler			M	Yes / No	
		ISO 26048-1§ 8.1.2.3	Configure an action owner	M	Yes	
		ISO 26048-1§ 8.1.2.4	Configure an action group	M	Yes	
		ISO 26048-1§ 8.1.2.5	Configure an action	M	Yes	
		ISO 26048-1§ 8.6	Day plan feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.2.1.4.1.1	Configure Pattern to Run Free	patternFunction:M	Yes / NA	
		3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	patternFunction:M	Yes / NA	
		3.5.2.1.4.1.12(patternFunction)	Configure Pattern Special Functions	O.5 (1..*)	Yes / No	
		3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	patternFunction:M	Yes	
		3.5.2.1.10.1.1	Configure Timebased Action - Pattern	actionFunction:M	Yes / NA	
		3.5.2.1.10.1.2(actionFunction)	Configure Timebased Action - Special Functions	O.5 (1..*)	Yes / No	
		3.5.2.1.10.1.3	Determine Maximum Number of Timebased Actions	actionFunction:M	Yes / NA	
2.5.2.1.12	Manage I/O Mapping			O	Yes / No	
		3.5.2.1.11.1.1	Set Active I/O Map	M	Yes	
		3.5.2.1.11.1.2.1	Configure I/O Map Description	M	Yes	
		3.5.2.1.11.1.2.2.1	Configure I/O Map Input Device	M	Yes	
		3.5.2.1.11.1.2.2.2	Configure I/O Map Input Device Pin	M	Yes	
		3.5.2.1.11.1.2.2.3	Configure I/O Map Input Function	M	Yes	
		3.5.2.1.11.1.2.3.1	Configure I/O Map Output Device	M	Yes	
		3.5.2.1.11.1.2.3.2	Configure I/O Map Output Device Pin	M	Yes	
		3.5.2.1.11.1.2.3.3	Configure I/O Map Output Function	M	Yes	
		3.5.2.1.11.2.1	Retrieve Maximum Number of I/O Maps	M	Yes	
		3.5.2.1.11.2.2	Retrieve Maximum Number of I/O Map Inputs	M	Yes	
		3.5.2.1.11.2.3	Retrieve Maximum Number of I/O Map Outputs	M	Yes	
		3.5.2.1.11.2.4	Retrieve I/O Mapping Activate Conditions	M	Yes	The following conditions shall be satisfied before a new I/O map can be activated: ___ Cabinet Door Open ___ in any flash state

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
						___ programmed all red flash ___ in CVM flash ___ ASC restart
		3.5.2.1.11.2.5	Retrieve I/O Mapping Input Functions	M	Yes	
		3.5.2.1.11.2.6	Retrieve I/O Mapping Output Functions	M	Yes	
		3.5.2.1.11.2.7	Retrieve I/O Map Input Device Pin Status	M	Yes	
		3.5.2.1.11.2.8	Retrieve I/O Map Output Device Pin Status	M	Yes	
		3.5.2.1.11.2.9.1	Enumerate I/O Map - FIO Inputs	332:M	Yes / NA	
		3.5.2.1.11.2.9.2	Enumerate I/O Map - FIO Outputs	332:M	Yes / NA	
		3.5.2.1.11.2.9.3	Enumerate I/O Map - TS1 Inputs	TS1, TS2-2:M	Yes / NA	
		3.5.2.1.11.2.9.4	Enumerate I/O Map - TS1 Outputs	TS1, TS2-2:M	Yes / NA	
		3.5.2.1.11.2.9.5	Enumerate I/O Map - TS2 BIU Inputs	TS2-1:M	Yes / NA	
		3.5.2.1.11.2.9.6	Enumerate I/O Map - TS2 BIU Outputs	TS2-1:M	Yes / NA	
		3.5.2.1.11.2.9.7	Enumerate I/O Map - ATC Cabinet SIU Inputs	ATC:M	Yes / NA	
		3.5.2.1.11.2.9.8	Enumerate I/O Map - ATC Cabinet SIU Outputs	ATC:M	Yes / NA	
		3.5.2.1.11.2.9.9	Enumerate I/O Map - Auxiliary Device Inputs	O	Yes / No	
		3.5.2.1.11.2.9.10	Enumerate I/O Map - Auxiliary Device Outputs	O	Yes / No	
2.5.2.1.13	Manage Intra-Cabinet Communications Configuration			O	Yes / No	
		3.5.2.1.12.1.1	Determine Serial Bus 1 Device Present	ATC:M	Yes / NA	The ASC shall support at least ____ Serial Bus 1 Addresses (between 1 and 255).
		3.5.2.1.12.2.1	Determine TS2 Port 1 Device Present	TS2-2:M	Yes / NA	The ASC shall support at least ____ TS2 Port1 Addresses (between 1 and 255).
		3.5.2.1.12.2.2	Enable/Disable TS2 Port 1 Frame 40 Messages	TS2-2:M	Yes / NA	
2.5.2.1.14	Manage Accessible Pedestrian Support			O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Confor mance**	**Supp ort**	**Additional Specifications**
		3.5.2.1.13.1.1	Configure APS Push Button Minimum Press Time	M	Yes	
		3.5.2.1.13.1.2	Configure APS Push Button to Phase Association	M	Yes	
		3.5.2.1.13.1.3	Configure APS Extra Crossing Time	M	Yes	
		3.5.2.1.13.1.4	Configure Pedestrian Detector for Alternate Pedestrian Timing	M	Yes	
2.5.2.2	Monitor Signal Operations Status					
2.5.2.2.1	Determine Controller Health			M	Yes	
		ISO 26048-1§ 8.5.2.9	Monitor when the SNMP capabilities last changed	M	Yes	
		ISO 26048-1§ 8.5.2.10	Monitor controller communications	M	Yes	
		ISO 26048-1§ 8.5.2.11	Monitor controller operational status	M	Yes	
		ISO 26048-1§ 8.5.2.12	Monitor controller up time	M	Yes	
		ISO 26048-1§ 8.5.2.13	Monitor watchdog failure count	M	Yes	
		3.5.2.2.1.1	Monitor External Alarm Input States	M	Yes	The ASC shall support at least ____ Alarm Groups (between 1 and 255).
		3.5.2.2.1.2	Monitor External Alarm Active	M	Yes	
		3.5.2.2.1.3	Monitor Flash Status	M	Yes	
		3.5.2.2.1.4	Monitor Local Override	M	Yes	
		3.5.2.2.1.5	Monitor Coordination Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.6	Monitor Detector Fault	Detector: M	Yes / NA	
		3.5.2.2.1.7	Monitor Stop Time Input Alarm	M	Yes	
		3.5.2.2.1.8	Monitor Cycle Fault Alarm	M	Yes	
		3.5.2.2.1.9	Monitor Coordination Fault	Coord:M	Yes / NA	
		3.5.2.2.1.10	Monitor Coordination Fail Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.11	Monitor Cycle Fail Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.12	Monitor Cabinet IO Link Alarm	M	Yes	
		3.5.2.2.1.13	Monitor SMU Communications Error	O	Yes / No	
		3.5.2.2.1.14	Monitor Preempt Maximum Presence Alarm	Preempt: O	Yes / No / NA	
2.5.2.2.2	Determine Mode of Operation					
2.5.2.2.2.1 (Unit)	Monitor Unit-wide General Operations			O	Yes / No	
		3.5.2.2.2.1	Monitor Unit Control Status	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.2.2.2.2	Monitor Preempt Active	Preempt: M	Yes / NA	
		3.5.2.2.2.3	Monitor Offset Transitioning	Coord:M	Yes / NA	
		3.5.2.2.2.4	Monitor Priority Call Active	O	Yes / No	
		3.5.2.2.2.5	Monitor Local Free Status	M	Yes	
		3.5.2.2.2.6	Monitor Coordination Active	Coord:M	Yes / NA	
		3.5.2.2.2.7	Monitor ECLA Control Active	O	Yes / No	
2.5.2.2.2.2	Monitor Flashing			M	Yes	
		3.5.2.2.1.3	Monitor Flash Status	M	Yes	
2.5.2.2.2.3	Monitor Current Timing Pattern			Coord:M	Yes / NA	
		3.5.2.2.2.5	Monitor Local Free Status	M	Yes	
		3.5.2.2.2.8.1	Monitor Current Pattern Status	M	Yes	
		3.5.2.2.2.8.2	Monitor Current Pattern Command Source	O	Yes / No	
		3.5.2.2.2.8.3	Monitor Current Pattern Fault Status	O	Yes / No	
2.5.2.2.2.4	Monitor Current Cycle			Coord:M	Yes / NA	
		3.5.2.2.2.9.1	Monitor Coordination Cycle Status	M	Yes	
		3.5.2.2.2.9.2	Monitor Coordination Synchronization Status	M	Yes	
		3.5.2.2.2.9.3	Monitor Current Offset	M	Yes	
2.5.2.2.3	Monitor Phase Status			M	Yes	
		3.5.2.2.3.1	Monitor Active Red Phases	M	Yes	
		3.5.2.2.3.2	Monitor Active Yellow Phases	M	Yes	
		3.5.2.2.3.3	Monitor Active Green Phases	M	Yes	
		3.5.2.2.3.4	Monitor Active Don't Walk Phases	M	Yes	
		3.5.2.2.3.5	Monitor Active Pedestrian Clearance Phases	M	Yes	
		3.5.2.2.3.6	Monitor Active Walk Phases	M	Yes	
		3.5.2.2.3.7	Monitor Active On Phases	M	Yes	
		3.5.2.2.3.8	Monitor Next Phases	M	Yes	
		3.5.2.2.3.9	Monitor Phase Vehicle Calls	M	Yes	
		3.5.2.2.3.10	Monitor Phase Pedestrian Calls	M	Yes	
		3.5.2.2.7.4	Monitor Current Gate Status	O	Yes / No	
2.5.2.2.4	Monitor Ring Status			Ring:M	Yes / NA	
		3.5.2.2.4.1	Monitor Ring Status	M	Yes	
		3.5.2.2.4.2	Monitor Ring Termination Cause	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.4.3	Monitor Current Phase On Time	M	Yes	
2.5.2.2.5	Monitor Channel Status			Channel: M	Yes / NA	
		3.5.2.2.5.1	Monitor Active Red Channels	M	Yes	
		3.5.2.2.5.2	Monitor Active Yellow Channels	M	Yes	
		3.5.2.2.5.3	Monitor Active Green Channels	M	Yes	
2.5.2.2.6	Monitor Overlap Status			Overlap: M	Yes / NA	
		3.5.2.2.6.1	Monitor Active Red Overlaps	M	Yes	
		3.5.2.2.6.2	Monitor Active Yellow Overlaps	M	Yes	
		3.5.2.2.6.3	Monitor Active Green Overlaps	M	Yes	
		3.5.2.2.6.4	Monitor Active Flashing Yellow Arrow Overlaps	O	Yes / No	
		3.5.2.2.6.5	Monitor Active Flashing Red Arrow Overlaps	O	Yes / No	
2.5.2.2.7	Monitor Preempt Status			Preempt: M	Yes / NA	
		3.5.2.2.7.1	Monitor Currently Active Preempt	M	Yes	
		3.5.2.2.7.2	Monitor Current Preempt Inputs	M	Yes	
		3.5.2.2.7.3	Monitor Current Preempt State	M	Yes	
		3.5.2.2.7.4	Monitor Current Gate Status	O	Yes / No	
2.5.2.2.8 (SpecialFunc)	Monitor Special Function Outputs			O	Yes / No	
		3.5.2.2.8.1	Determine Maximum Number of Special Functions	M	Yes	The ASC shall support at least ____ Special Functions (between 1 and 255).
		3.5.2.2.8.2	Monitor Special Function Status	M	Yes	
		3.5.2.2.8.3	Monitor Special Function Control Source	O	Yes / No	
2.5.2.2.9	Monitor Timebase Action Status			M	Yes	
		3.5.2.2.2.8.1	Monitor Current Pattern Status	patternFunction: M	Yes / NA	
		3.5.2.1.10.1.4	Determine Action in Effect	actionFunction:M	Yes / NA	
2.5.2.2.10	Monitor Intra-Cabinet Communications Configuration			O	Yes / No	
		3.5.2.2.9.1	Monitor TS2 Port 1 Status	TS2-2:M	Yes / NA	
		3.5.2.2.9.2	Monitor TS2 Port 1 Fault Frame	TS2-2:M	Yes / NA	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.9.3	Monitor Serial Bus 1 Status	ATC:M	Yes / NA	
2.5.2.2.11	Monitor Signal Monitoring Unit			O	Yes / No	
		3.5.2.2.10.1	Monitor Signal Monitoring Unit Channel Voltage	ATC:M	Yes	It's optional for any cabinet, but format is standard on an ATC.
		3.5.2.2.10.2	Monitor Signal Monitoring Unit Channel Current	ATC:M	Yes	
2.5.2.3	Control Signal Operations			M	Yes	
2.5.2.3.1	Control ASC-wide General Operations			M	Yes	
		3.5.2.3.1.1	Enable/Disable Manual Backup	O	Yes / No	
		3.5.2.3.1.2	Control Global Minimum Recall	M	Yes	
		3.5.2.3.1.3	Control Call to Non-Actuated 1	M	Yes	
		3.5.2.3.1.4	Control Call to Non-Actuated 2	M	Yes	
		3.5.2.3.1.5	Control Walk Rest Modifier	M	Yes	
		3.5.2.3.1.6	Control Interconnect	O	Yes / No	
2.5.2.3.2	Activate Timing Pattern			Coord:M	Yes / NA	
		3.5.2.3.2.1	Activate System Timing Pattern Remotely	M	Yes	
		3.5.2.3.2.2	Control System Reference Point	M	Yes	
2.5.2.3.3	Phase Requests			O	Yes / No	
		3.5.2.3.3.1	Control Phase Omits	M	Yes	
		3.5.2.3.3.2	Control Pedestrian Phase Omits	M	Yes	
		3.5.2.3.3.3	Control Phase Holds	M	Yes	
		3.5.2.3.3.4	Control Phase Force Offs	O	Yes / No	
		3.5.2.3.3.5	Control Phase Vehicle Calls	M	Yes	
		3.5.2.3.3.6	Control Phase Pedestrian Calls	M	Yes	
2.5.2.3.4	Activate Preempt			Preempt: O	Yes / No / NA	
		3.5.2.3.4	Activate Preempt Remotely	M	Yes	
2.5.2.3.5	Control Ring Operations			Ring:O	Yes / No / NA	
		3.5.2.3.5.1	Control Ring Stop Time	M	Yes	
		3.5.2.3.5.2	Control Ring Force Offs	M	Yes	
		3.5.2.3.5.3	Control Ring Maximum 2 Settings	M	Yes	
		3.5.2.3.5.4	Control Ring Maximum 3 Settings	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.3.5.5	Control Ring Maximum Inhibit Settings	M	Yes	
		3.5.2.3.5.6	Control Ring Pedestrian Recycle Settings	M	Yes	
		3.5.2.3.5.7	Control Ring Red Rest Settings	M	Yes	
		3.5.2.3.5.8	Control Ring Red Clearance Omit Settings	M	Yes	
2.5.2.3.6	Activate Special Function Output			SpecialFunc:O	Yes / No / NA	
		3.5.2.3.6	Activate Special Function Remotely	M	Yes	
2.5.2.3.7	Activate Action Plan			O	Yes / No	
		3.5.2.1.10.5	Activate Action Plan Remotely	actionFunction:M	Yes / NA	
		3.5.2.3.2.1	Activate System Timing Pattern Remotely	patternFunction:M	Yes / NA	
2.5.2.3.8	Remote Manual Control			O	Yes / No	
		3.5.2.3.7.1	Enable Remote Manual Control	M	Yes	
		3.5.2.3.7.2	Advance Interval During Remote Manual Control	M	Yes	
		3.5.2.3.7.3	Configure Manual Control Timeout	M	Yes	
		3.5.2.3.7.4	Enable/Disable Automatic Pedestrian Clearance Setting	M	Yes	
2.5.3	Manage Detectors					
2.5.3.1 (Detector)	Manage Detector Configuration			M	Yes	
		3.5.3.1.1.1	Configure Vehicle Travel Mode	O	Yes / No	
		3.5.3.1.1.2	Configure Vehicle Detector Description	O	Yes / No	
		3.5.3.1.1.3	Configure Vehicle Detector Yellow Lock Call Enabled	O	Yes / No	
		3.5.3.1.1.4	Configure Vehicle Detector Red Lock Call Enabled	O	Yes / No	
		3.5.3.1.1.5	Configure Vehicle Detector Passage Enabled	O	Yes / No	
		3.5.3.1.1.6	Configure Vehicle Detector Added Initial Time Enabled	O	Yes / No	
		3.5.3.1.1.7	Configure Vehicle Detector Queue Enabled	O	Yes / No	
		3.5.3.1.1.8	Configure Vehicle Detector Call Enabled	M	Yes	
		3.5.3.1.1.9	Configure Vehicle Detector Call Phase	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.3.1.1.10	Configure Vehicle Detector Switch Phase	M	Yes	
		3.5.3.1.1.11	Configure Vehicle Detector Delay Time	M	Yes	
		3.5.3.1.1.12	Configure Vehicle Detector Extend Time	M	Yes	
		3.5.3.1.1.13	Configure Vehicle Detector Queue Limit Time	O	Yes / No	
		3.5.3.1.1.14	Configure Vehicle Detector No Activity Fault Time	M	Yes	
		3.5.3.1.1.15	Configure Vehicle Detector Maximum Presence Fault Time	M	Yes	
		3.5.3.1.1.16	Configure Vehicle Detector Erratic Counts	M	Yes	
		3.5.3.1.1.17	Configure Vehicle Detector Fail Time	O	Yes / No	
		3.5.3.1.2 (VehDetectSet)	Configure Multiple Vehicle Detector Sets for Actuation	O	Yes / No	
		3.5.3.1.3.1	Configure Pedestrian Detector Description	O	Yes / No	
		3.5.3.1.3.2	Configure Pedestrian Detector Call Phase	M	Yes	
		3.5.3.1.3.3	Configure Pedestrian Detector No Activity Fault Time	M	Yes	
		3.5.3.1.3.4	Configure Pedestrian Detector Maximum Presence Fault Time	M	Yes	
		3.5.3.1.3.5	Configure Pedestrian Detector Erratic Counts	M	Yes	
		3.5.3.1.3.6	Configure Pedestrian Detector Non-Lock Calls	O	Yes / No	
		3.5.3.1.3.7	Configure Pedestrian Detector for Presence Detection	O	Yes / No	
		3.5.3.1.3.8	Configure Pedestrian Detector for Delayed Walk	O	Yes / No	
		3.5.3.1.3.9	Configure Pedestrian Detector for Advanced Walk	O	Yes / No	
		3.5.3.1.4 (PedDetectSet)	Configure Multiple Pedestrian Detector Sets for Actuations	O	Yes / No	
		3.5.3.1.5.1	Determine Maximum Number of Vehicle Detectors	M	Yes	The ASC shall support at least ____ vehicle detectors (between 1 and 255).
		3.5.3.1.5.2	Determine Maximum Number of Vehicle Detector Sets	VehDetectSet:M	Yes / NA	The ASC shall support at least ____ vehicle detector sets.
		3.5.3.1.5.3	Determine Maximum Number of Pedestrian Detectors	M	Yes	The ASC shall support at least

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						_____ pedestrian detectors (between 1 and 255).
		3.5.3.1.5.4	Determine Maximum Number of Pedestrian Detector Sets	PedDetectSet:M	Yes / NA	The ASC shall support at least _____ pedestrian detector sets.
2.5.3.2	Monitor Detector Status			O	Yes / No	
		3.5.3.2.1	Monitor Active Vehicle Detector Actuations	M	Yes	
		3.5.3.2.2	Monitor Active Pedestrian Detector Actuations	M	Yes	
2.5.3.3	Monitor Detector Health			O	Yes / No	
		3.5.3.3.1.1	Monitor Vehicle Detector Alarm Status	M	Yes	
		3.5.3.3.1.2	Monitor Vehicle Detector Faults from Controller	M	Yes	
		3.5.3.3.1.3	Monitor Vehicle Detector Faults from Detector	O	Yes / No	
		3.5.3.3.3.1	Monitor Pedestrian Detector Alarm Status	M	Yes	
		3.5.3.3.3.2	Monitor Pedestrian Detector Faults	M	Yes	
2.5.3.4	Control Detectors			O	Yes / No	
		3.5.3.4.1	Control Vehicle Detector Reset	M	Yes	
		3.5.3.4.2	Control Pedestrian Detector Reset	M	Yes	
		3.5.3.4.3	Control Detector Diagnostic Reset	O	Yes / No	
		3.5.3.4.4	Control Vehicle Detector Actuation	O	Yes / No	
		3.5.3.4.5	Control Pedestrian Detector Actuation	O	Yes / No	
2.5.3.5	Manage Detector Data Collection			O	Yes / No	
		3.5.3.5.3.1	Configure Vehicle Detector Data Sample Period	M	Yes	
		3.5.3.5.3.2	Configure Pedestrian Detector Sample Period	M	Yes	
		3.5.3.5.3.3 (Speed)	Configure Vehicle Speed Detectors	O	Yes / No	
		3.5.3.5.3.4	Configure Single Detector Speed Mode	Speed:M	Yes / NA	
		3.5.3.5.3.5	Configure Paired Detector	Speed:M	Yes / NA	
		3.5.3.5.3.6	Configure Paired Detector Placement	Speed:M	Yes / NA	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.3.5.3.7	Configure Paired Detector Spacing	Speed:M	Yes / NA	
		3.5.3.5.3.8	Configure Average Vehicle Length	O	Yes / No	
		3.5.3.5.3.9	Configure Vehicle Detection Zone Length	O	Yes / No	
		3.3.3.5.4	Configure Multiple Vehicle Detector Sets for Data Collection	O	Yes / No	
2.5.3.6	Monitor Detector Data from Controller			O	Yes / No	
		3.5.3.5.1.1	Monitor Vehicle Detector Data Sequence	M	Yes	
		3.5.3.5.1.2	Monitor Vehicle Volume Data	O	Yes / No	
		3.5.3.5.1.3	Monitor Vehicle Occupancy Data	O	Yes / No	
		3.5.3.5.1.4	Monitor Vehicle Average Speed	Speed:M	Yes / NA	
		3.5.3.5.1.5	Monitor Vehicle Detector Data Sample Time	O	Yes / No	
		3.5.3.5.2.1	Monitor Pedestrian Detector Data Sequence	O	Yes / No	
		3.5.3.5.2.2	Monitor Pedestrian Counts	O	Yes / No	
		3.5.3.5.2.3	Monitor Pedestrian Actuations	O	Yes / No	
		3.5.3.5.2.4	Monitor Pedestrian Services	O	Yes / No	
2.5.4 (CV)	Manage Connected Vehicles Interface			O	Yes / No	
2.5.4.1	Connected Vehicle Interface: Management Station – ASC Interface			M	Yes / No	
2.5.4.1.1	Manage ASC - RSU Interface			M	Yes	
		3.5.4.1.1	Configure ASC Communications Port for RSU	M	Yes	
		3.5.4.1.2	Configure Logical RSU Ports and Address	M	Yes	
		3.5.4.1.3	Configure RSU Interface Polling Period	O	Yes / No	
2.5.4.1.2	Manage ASC - RSU Interface Watchdog			O	Yes / No	
		3.5.4.1.4	Configure RSU Interface Watchdog	M	Yes	
		3.5.4.1.5	Monitor RSU Interface Watchdog Timer	M	Yes	
		3.5.4.1.6	Monitor RSU Interface Watchdog Alarm	M	Yes	
2.5.4.1.3	Manage Signal Phase and Timing Data			M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.4.2.1.1	Enable Signal Phase and Timing Data	M	Yes	
		3.5.4.2.1.2	Retrieve Signal Phase and Timing Generation Time	O	Yes / No	
		3.5.4.2.1.3.1.1	Monitor Movement Minimum End Time	M	Yes	
		3.5.4.2.1.3.1.2	Monitor Movement Maximum End Time	M	Yes	
		3.5.4.2.1.3.1.3	Monitor Movement Likely End Time	O	Yes / No	
		3.5.4.2.1.3.1.4	Monitor Movement Likely End Time Confidence	O	Yes / No	
		3.5.4.2.1.3.1.5	Monitor Movement Next Occurrence	M	Yes	
		3.5.4.2.1.3.1.6	Monitor Movement Start Time	M	Yes	
		3.5.4.2.1.3.1.7	Monitor Next Movement Minimum End Time	M	Yes	
		3.5.4.2.1.3.1.8	Monitor Next Movement Maximum End Time	M	Yes	
		3.5.4.2.1.3.1.9	Monitor Next Movement Start Time	M	Yes	
		3.5.4.2.1.3.1.10	Determine Maximum Number of Movement Events			
		3.5.4.2.1.3.2.1	Configure Queue Detectors for Movement Assistance	MvtQueue:M	Yes / NA	
		3.5.4.2.1.3.2.2	Configure Pedestrian Detectors for Movement Conflict Assistance	MvtConflict:O.4 (1..*)	Yes / No / NA	
		3.5.4.2.1.3.2.3	Configure Bicycle Detectors for Movement Conflict Assistance	MvtConflict:O.4 (1..*)	Yes / No / NA	
		3.5.4.2.1.3.3.1 (MvtQueue)	Monitor Lane Connection Queue Length	O	Yes / No	
		3.5.4.2.1.3.3.2 (MvtConflict)	Monitor Lane Connection Vulnerable Road User Detection	O	Yes / No	
		3.5.4.2.1.3.4.1 (SpdAdvice)	Configure Advisory Speed Type	O	Yes / No	
		3.5.4.2.1.3.4.2	Configure Advisory Speed	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.4.3	Configure Advisory Speed Zone	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.4.4	Configure Advisory Speed Vehicle Type	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.5	Monitor Movement State	M	Yes	
		3.5.4.2.1.3.6	Monitor Next Movement State	M	Yes	
		3.5.4.2.1.3.7	Monitor Movement Status	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.4.2.1.4.1	Configure Concurrent Enabled Lanes	M	Yes	
		3.5.4.2.1.4.2	Configure Enabled Lanes by Time of Day	M	Yes	
		3.5.4.2.1.4.3	Determine Lanes Enabled	M	Yes	
		3.5.4.2.1.4.4	Command Enabled Lanes	M	Yes	
		3.5.4.2.1.5	Enable Signal Phase and Timing Exchange	M	Yes	
		3.5.4.2.1.6	Configure Road Authority Identifier	M	Yes	
		3.5.4.2.1.7.1	Monitor Manual Control Indication	M	Yes	
		3.5.4.2.1.7.2	Monitor Stop Indication	M	Yes	
		3.5.4.2.1.7.3	Monitor Failure Flash Indication	M	Yes	
		3.5.4.2.1.7.4	Monitor Preemption Operation Indication	M	Yes	
		3.5.4.2.1.7.5	Monitor Priority Operation Indication	M	Yes	
		3.5.4.2.1.7.6	Monitor Fixed Time Control Indication	M	Yes	
		3.5.4.2.1.7.7	Monitor Non-Fixed Time Control Indication	M	Yes	
		3.5.4.2.1.7.8	Monitor Standby Operation Indication	M	Yes	
		3.5.4.2.1.7.9	Monitor Controller Failure	M	Yes	
		3.5.4.2.1.7.10	Monitor MAP Message Validity	M	Yes	
		3.5.4.2.1.7.11	Monitor SPaT Message Validity	M	Yes	
		3.5.4.2.1.8	Mark SPaT Invalid - Controller	M	Yes	
		3.5.4.2.1.9	Mark SPaT Invalid - Port	O	Yes / No	
		3.5.4.2.1.10	Mark MAP Message Invalid - Controller	M	Yes	
		3.5.4.2.1.11	Mark MAP Message Invalid - Port	O	Yes / No	
		3.5.4.2.1.12.1	Determine Maximum Number of Signal Groups	M	Yes	
		3.5.4.2.1.12.2	Configure Signal Groups Intersection Mapping	M	Yes	
		3.5.4.2.1.12.3	Configure Signal Group Control Source	M	Yes	
		3.5.4.2.1.12.4	Configure Signal Group Indication Types	M	Yes	
		3.5.4.2.1.12.5	Configure Signal Group Protected or Permissive State	M	Yes	
		3.5.4.2.1.12.6	Configure Signal Group Revocable Lanes	M	Yes	
		3.5.4.2.1.12.7	Determine Maximum Number of Signal State Entries	M	Yes	
		3.5.4.2.1.12.8	Configure Customized Signal State Parameters	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.4.2.1.13	Retrieve Signal Phase and Timing Time Point	M	Yes	
2.5.4.1.4	Manage Assured Green Period			O	Yes / No	
		3.5.4.2.2.1	Enabled Connected Device Connection	M	Yes	
		3.5.4.2.2.2	Configure Vehicle Detector for Connected Vehicle Application	M	Yes	
		3.5.4.2.2.3	Configure Connected Vehicle Detector Assigned Input	M	Yes	
		3.5.4.2.2.4	Configure Connected Vehicle Detector Port Assignment	O	Yes / No	
		3.5.4.2.2.5	Configure Assured Green Period Duration	M	Yes	
		3.5.4.2.2.6	Configure Red Light Violation Warning Parameters	O	Yes / No	
2.5.4.2	Connected Vehicle Interface: ASC – CV Application Process Interface			M	Yes	
2.5.4.2.1	Exchange Current and Next Movement Information			M	Yes	
		3.5.4.2.1.1.1	Provide Movement Time Point	M	Yes	
		3.5.4.2.1.1.2	Provide Movement State	M	Yes	
		3.5.4.2.1.1.3	Provide Movement Minimum End Time	M	Yes	
		3.5.4.2.1.1.4	Provide Movement Maximum End Time	M	Yes	
		3.5.4.2.1.1.5	Provide Movement Likely End Time	O	Yes / No	
		3.5.4.2.1.1.6	Provide Movement Likely End Time Confidence	O	Yes / No	
		3.5.4.2.1.1.7	Provide Next Movement State	M	Yes	
		3.5.4.2.1.1.8	Provide Next Movement Minimum End Time	M	Yes	
		3.5.4.2.1.1.9	Provide Next Movement Maximum End Time	M	Yes	
		3.5.4.2.1.1.10	Provide Next Movement Start Time	M	Yes	
		3.5.4.2.1.2.1	Provide Lane Connection Queue Length	O	Yes / No	
		3.5.4.2.1.2.2	Provide Lane Connection Traveler Detection	O	Yes / No	
		3.5.4.2.1.3.1	Provide Advisory Speed Type	O	Yes / No	
		3.5.4.2.1.3.2	Provide Advisory Speed	O	Yes / No	
		3.5.4.2.1.3.3	Provide Advisory Speed Zone	O	Yes / No	
		3.5.4.2.1.3.4	Provide Advisory Speed Vehicle Type	O	Yes / No	
		3.5.4.2.1.4	Provide Road Authority ID	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.4.2.1.5	Provide Signal Phase and Timing Intersection Status	M	Yes	
		3.5.4.3.1.6	Provide Compressed SPaT Information to External CV Application Process	M	Yes	
		3.5.4.3.2.1	Provide UPER-encoded SPaT Message	M	Yes	
		3.6.5.1	SPaT Maximum Transmission Start Time	M	Yes	The Maximum Transmission Start Time for all SPAT data shall be ___ milliseconds (Default=10).
		3.6.5.2	Movement Time Point Minimum Transmission Rate	M	Yes	The Movement Time Point Minimum Transmission Rate shall be once per _____ milliseconds (Default=100).
		3.6.5.3	SPaT Maximum Transmission Rate	M	Yes	
		3.6.5.4	SPaT Time Accuracy	M	Yes	
2.5.4.2.2	Exchange Next Occurrence of a Movement			M	Yes	
		3.5.4.3.1.1.11	Provide Movement Next Occurrence	M	Yes	
		3.5.4.3.4	Monitor CV Certificate Faults	O	Yes / No	
		3.6.5.1	SPaT Maximum Transmission Start Time	M	Yes	The Maximum Transmission Start Time for all SPAT data that changed shall be ___ milliseconds (Default=10).
		3.6.5.2	Movement Time Point Minimum Transmission Rate	M	Yes	The Movement Time Point Minimum Transmission Rate shall be once per _____ milliseconds (Default=100).
		3.6.5.3	SPaT Maximum Transmission Rate	M	Yes	
		3.6.5.4	SPaT Time Accuracy	M	Yes	
2.5.4.2.3	Exchange Presence of Connected Devices			O	Yes / No	
		3.5.4.3.2.2	Retrieve BSMs	O.12(1..*)	Yes / No	
		3.5.4.3.2.3	Retrieve PSMs	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.4.3.2.4	Retrieve Actuation Report	O.12(1..*)	Yes / No	
		3.5.4.3.2.5	Retrieve Detection Report	O	Yes / No	
2.5.4.2.4	Exchange Roadway Geometrics Information			O	Yes / No	
		3.5.4.3.3.1	Retrieve MAP Plan in Effect	M	Yes	
		3.5.4.3.3.2	Confirm MAP Plan Compatibility	M	Yes	
2.5.4.3	ASC – ECLA Interface			O	Yes / No	
		3.5.4.4.1	Receive Current Phase Minimum End Time from an ECLA	M	Yes	
		3.5.4.4.2	Receive Current Phase Maximum End Time from an ECLA	M	Yes	
		3.5.4.4.3	Receive Current Phase Likely End Time from an ECLA	O	Yes / No	
		3.5.4.4.4	Receive Current Phase Likely End Time Confidence from an ECLA	O	Yes / No	
		3.5.4.4.5	Receive Next Phase from an ECLA	M	Yes	
		3.5.4.4.6	Receive Compressed ECLA Input Data	M	Yes	
2.5.5	Backward Compatibility Features					
2.6	Security			M	Yes	
2.6.1	Manage Authentication			M	Yes	
2.6.2	Manage Accessibility			M	Yes	
2.6.3	Manage Users			M	Yes	
		ISO 26048-1§ 8.22	View-based access control model (VACM) feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.6.4	Log User Access			O	Yes / No	
		ISO 26048-1§ 8.1.3.1	Validate access upon action activation	M	Yes	
		ISO 26048-1§ 8.1.3.2	Validate access upon action being called	M	Yes	
		3.6.1	Response Time for Requests	M	Yes	The Response Time for all requests shall

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
						be ___ milliseconds (5-500: Default=25).
2.6.5	Manage ASC Interface Security			O	Yes / No	
2.6.5.1	Manage Security for the ASC to RSU Interface					
2.6.5.2	Manage Security for Other ASC Interfaces					

in Section 3.3.3, maps the user needs defined in Section 2 to the requirements defined in Section 3. The PRL can be used by:

a) A user or specification writer to indicate which requirements are to be implemented in a project-specific implementation.
b) The protocol implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight.
c) The supplier and user, as a detailed indication of the capabilities of the implementation.
d) The user, as a basis for initially checking the potential interoperability with another implementation.

3.3.1 Notation [Informative]

The following notations and symbols are used to indicate status and conditional status in the PRL within all NTCIP standards. Not all of these notations and symbols may be used within NTCIP 1202 v04.

3.3.1.1 Conformance Symbols

The symbols in Table 1 are used to indicate status under the Conformance column in the PRL.

Table 1 Conformance Symbols

Symbol	Status
M	Mandatory
M.#	Support of every item of the group labeled by the same numeral # is required, but only one is active at a time
O	Optional
O.# (range)	Part of an option group. Support of the number of items indicated by the '(range)' is required from all options labeled with the same numeral #
C	Conditional
NA	Not-applicable (i.e., logically impossible in the scope of the standard)
X	Excluded or prohibited

The O.# (range) notation is used to show a set of selectable options (e.g., O.2 (1..*) would indicate that one or more of the option group 2 options shall be implemented). Two character combinations are used for dynamic requirements. In this case, the first character refers to the static (implementation) status, and the second refers to the dynamic (use); thus, "MO" means "mandatory to be implemented, optional to be used."

3.3.1.2 Conditional Status Notation

The predicate notations in Table 2 may be used.

Table 2 Conditional Status Notation

Predicate	Notation
<predicate>:	This notation introduces a single item that is conditional on the <predicate>.
<predicate>::	This notation introduces a table or a group of tables, all of which are conditional on the <predicate>.
(predicate)	This notation introduces the first occurrence of the predicate. The feature associated with this notation is the base feature for all options that have this predicate in their conformance column.

The <predicate>: notation means that the status following it applies only when the PRL states that the feature or features identified by the predicate are supported. In the simplest case, <predicate> is the identifying tag of a single PRL item. The <predicate> notation may precede a table or group of tables in a section or subsection. When the group predicate is true then the associated section shall be completed. The symbol <predicate> also may be a Boolean expression composed of several indices. "AND," "OR," and "NOT" shall be used to indicate the Boolean logical operations.

The predicates used in NTCIP 1202 v04 map to the sections indicated in Table 3.

Table 3 Predicate Mapping to NTCIP 1202 v04 Section

Predicate	**Section**
332	2.3.1.a
ATC	2.3.1.e
actionFunction	3.5.2.1.10.1.2
Coord	2.5.2.1.3
Channel	2.5.2.1.7
CV	2.5.4
Detector	2.5.3.1
Generator	ISO 26048-1§ 8.2.2.2.1.c
Overlap	2.5.2.1.8
MvtConflict	3.5.4.2.1.3.3.2
MvtQueue	3.5.4.2.1.3.3.1
OverlapSet	3.5.2.1.8.2
patternFunction	3.5.2.1.4.1.12
PedDetectSet	3.5.3.1.4
PhaseSet	3.5.2.1.2.2
Power	2.5.1.4
Preempt	2.5.2.1.9
Queue	ISO 26048-1§ 8.10.1.3.3.2
Ring	2.5.2.1.6
Solar	ISO 26048-1§ 8.2.2.2.1.d
SpdAdvice	3.5.4.2.1.3.4.1
SpecialFunc	2.5.2.2.8
TS1	2.3.1.b
TS2-1	2.3.1.d
TS2-2	2.3.1.c
TwoStep	ISO 26048-1§ 8.8.3.6
Unit	2.5.2.2.2.1
UPS	ISO 26048-1§ 8.2.2.2.2
VehDetectSet	3.5.3.1.2

3.3.1.3 Support Column Symbols

The Support column in the PRL can be used by a procurement specification to identify the required features for the given procurement or by an implementer to identify which features have been implemented. In either case, the user circles the appropriate answer (Yes, No, or N/A) in the support column:

Table 4 Support Column Entries

Entry	Identifier
Yes	Supported by the implementation.
No	Not supported by the implementation.
N/A	Not applicable

3.3.2 Instructions for Completing the PRL [Informative]

In the 'Support' column, each response shall be selected either from the indicated set of responses (for example: Yes / No / NA), or it shall reference additional items that are to be attached (for example, list of traffic signal controllers to be supported by an implementation).

If a conditional requirement is inapplicable, use the Not Applicable (NA) choice. If a mandatory requirement is not satisfied, exception information shall be supplied by entering a reference Xi, where i is a unique identifier, to an accompanying rationale for the non-conformance. When the status is expressed as a two-character combination (as defined in 3.3.1.1 above), the response shall address each element of the requirement; e.g., for the requirement "mo," the possible compliant responses are "yy" or "yn."

> Note: A specification can allow for flexibility in a deliverable by leaving the selection in the Support column blank for a given row.

3.3.2.1 Conformance Definition

To claim "Conformance" to NTCIP 1202 v04, the vendor shall minimally fulfill the mandatory requirements as identified in the PRL table (see Table 5).

> Note: The reader and user of NTCIP 1202 v04 are advised that 'conformance' to NTCIP 1202 v04 should not be confused with 'compliance' to a specification. NTCIP 1202 v04 is as broad as possible to allow a very simple ASC implementation to be 'conformant' to NTCIP 1202 v04. An agency specification needs to identify the requirements of a particular project and needs to require the support of those requirements. A specification writer is advised to match the requirements of a project with the corresponding standardized requirements defined in NTCIP 1202 v04 to achieve interoperability. This means that functions and requirements defined as 'optional' in NTCIP 1202 v04 might need to be selected in a specification (in effect made 'mandatory' for the project-specific specification).

A conformant device may offer additional (optional) features, as long as they are conformant with the requirements of NTCIP 1202 v04 and the standards it references (e.g., ISO 26048-1 and NTCIP 2301 v02). For example, to claim conformance to additional features, an implementation shall conform to all of the mandatory and selected optional requirements that trace to the subject user needs in the PRL, AND shall fulfill the requirement by using all of the dialogs and data elements traced to the subject requirement in the Requirements Traceability Matrix (RTM) in Annex A.

A device may also support data that has not been defined by NTCIP 1202 v04; however, when exchanged via one of the NTCIP 2301 v02 protocols, the data shall be properly registered with a valid OBJECT IDENTIFIER under the Global ISO Naming Tree.

Note: Off-the-shelf interoperability and interchangeability can only be obtained through well-documented features broadly supported by the industry as a whole. Designing a system that uses features not defined in a standard or not typically deployed in combination with one another inhibits the goals of interoperability and interchangeability, especially if the documentation of these features is not available for distribution to system integrators. Standards allow the use of additional features to support innovation, which is constantly needed within the industry; but users should be aware of the risks involved with using such features.

To claim "Conformance" to NTCIP 1202 v04, an ASC device shall be provided with a MIB that contains all non-NTCIP-standardized (including custom, proprietary and vendor-, agency-, or implementation-specific) object and block definitions. Object and block definitions contained in the MIB shall:

a) use the ASN.1 notation and conventions used in NTCIP 1202 v04 standardized object and block definitions,
b) include non-NTCIP-standardized enumerations, and
c) include meaningful, human-understandable, English language DESCRIPTION fields including descriptions of the object and all supported values.

In addition, to claim "Conformance" to NTCIP 1202 v04, an ASC device shall use the NTCIP 1202 v04 standardized objects to manage NTCIP 1202 v04 functionality. Non-NTCIP-standardized objects may be used to manage NTCIP 1202 v04 functionality only if NTCIP 1202 v04 standardized objects for the same functions are also supported. ASC devices or systems attempting to manage, configure, or monitor an NTCIP 1202 v04 standardized object shall not be required to use proprietary objects for NTCIP 1202 v04 functionality.

3.3.3 Protocol Requirements List (PRL) Table

In addition to the Conformance column and the Support column, which were discussed in Sections 3.3.1 and 3.3.2, the additional columns in the PRL table are the User Need ID and User Need columns, FR ID and Functional Requirements columns and the Additional Specifications column.

a) User Need ID. The number assigned to the user need statement. The user needs are defined within Section 2, and the PRL is based upon the user need sections within that Section.
b) User Need. A short descriptive title identifying the user need.
c) FR ID. The number assigned to the functional requirement statement. The requirements are defined within Section 3, and the PRL references the traces from user needs to these requirements.
d) Functional Requirement. A short descriptive title identifying the functional requirement.
e) Additional Specifications. Identify other requirements to satisfy, including user selectable range values. The "Additional Specifications" column may (and should) be used by a procurement specification to provide additional notes and requirements for the product to be procured or may be used by an implementer to provide any additional details about the implementation. In some cases, default text already exists in this field, which the user should complete to fully specify the equipment. However, additional text can be added to this field as needed to fully specify a feature.

Note: Visit www.ntcip.org for information on the availability of electronic copies of the PRL.

Table 5 Protocol Requirements List (PRL)

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.3	Reference Physical Architecture [Informative]					
2.3.1	ASC Characteristics – Cabinet Specifications [Normative]			M	Yes	
2.3.1.a (332)	Model 332 Cabinet			O.1 (1)	Yes / No	
2.3.1.b (TS1)	NEMA TS 1 Cabinet			O.1 (1)	Yes / No	
2.3.1.c (TS2-2)	NEMA TS 2 Type 2 Cabinet			O.1 (1)	Yes / No	
2.3.1.d (TS2-1)	NEMA TS 2 Type 1 Cabinet			O.1 (1)	Yes / No	
2.3.1.e (ATC)	ATC Cabinet			O.1 (1)	Yes / No	
2.3.2	ASC Characteristics – Controller Types [Normative]			M	Yes	
2.3.2.a	Phase-based controller			M	Yes	
2.3.2.b	Interval-based controller			NA	NA	Interval-based controllers are not supported by NTCIP 1202 v04
2.3.2.c	Stage-based controller			NA	NA	Stage-based controllers are not supported by NTCIP 1202 v04
2.4	Architectural Needs					
2.4.1	Provide Live Data			M	Yes	
		3.4.1.1	Retrieve Data	M	Yes	
		3.4.1.2	Deliver Data	M	Yes	
		3.4.1.3	Explore Data	M	Yes	
		3.4.1.4.1	Monitor SNMP Information	M	Yes	
		ISO 26048-1§ 8.5.2.6	Determine basic capabilities of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.7	Determine SNMP capabilities of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.8	Determine the SNMP engine identifier	M	Yes	
		ISO 26048-1§ 8.5.2.9	Monitor when the SNMP capabilities last changed	M	Yes	
		ISO 26048-1§ 8.5.4	Controller performance requirements	M	Yes	
		3.6.1	Response Time for Requests	M	Yes	The Response Time for all requests shall be ___ milliseconds (5-500: Default=25). Note: Redundant with the previous requirement.
		~~3.6.2~~	Atomic Operations	M	Yes	Note: should 3.6.2 be deleted?
2.4.2	Provide Data Blocks			O	Yes / No	
		3.4.2.1	Store Pre-defined Compressed Data Blocks	M	Yes	
		3.5.2.1.14.1.1	Configure Block Object Get Control - Phase Data	O.2 (1..*)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.14.1.2	Configure Block Object Get Control - Vehicle Detector Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.3	Configure Block Object Get Control - Pedestrian Detector Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.4	Configure Block Object Get Control - Pattern Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.5	Configure Block Object Get Control - Split Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.6	Configure Block Object Get Control - Overlap Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.7	Configure Block Object Get Control - Preempt Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.8	Configure Block Object Get Control - Sequence Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.9	Configure Block Object Get Control - Channel Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.2	Monitor Block Error Status - Error-causing Data Element	M	Yes	
		ISO 26048-1§ 8.7	Dynamic object feature	O	Yes / No	All requirements that trace to this feature are also inherited. See the Features To Requirements Traceability (FTRT) in ISO 26048-1.
2.4.3	Provide for Log Data Local Storage and Retrieval			O.3 (1..*)	Yes / No	Note: Either Logging or Exception Reporting is Mandatory in ISO 26048-1.
		ISO 26048-1§ 8.9	File Feature	M	Yes	
		ISO 26048-1§ 8.10	Logging feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.4.4	Provide for Database Management			M	Yes	
		ISO 26048-1§ 8.21	Transaction feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.4.5	Condition-based Exception Reporting			O.3 (1..*)	Yes / No	Note: Either Logging or Exception Reporting is Mandatory in ISO 26048-1.
		ISO 26048-1§ 8.1	Action feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.11	Notification feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.16	SNMP target feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5	Features					
2.5.1	Manage the ASC Configuration			M	Yes	
2.5.1.1	Manage Device Identity			M	Yes	
		3.5.1.1.1	Configure ASC Location - Antenna Offset	C	Yes / NA	Mandatory if an external GNSS device is attached to the ASC.
		ISO 26048-1§ 8.5.2.1	Configure the controller's identity	M	Yes	
		ISO 26048-1§ 8.5.2.2	Configure the default language	M	Yes	
		ISO 26048-1§ 8.5.2.5	Control remote reset of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.14	Obtain the field device configuration identifier	M	Yes	
		ISO 26048-1§ 8.5.3.1	Support maximum message size	M	Yes	The maximum SNMP message size shall be _____ bytes (Default: 484 bytes).
		ISO 26048-1§ 8.5.3.2	Support total memory	O	Yes / No	The controller shall support a minimum of _____ kilobytes of total memory.
		ISO 26048-1§ 8.5.3.3	Support changeable memory	O	Yes / No	The controller shall support a minimum of _____ kilobytes of changeable memory.
		ISO 26048-1§ 8.8.2.1.1	Configure the field device's location by providing coordinates	M	Yes	
		ISO 26048-1§ 8.8.2.1.2	Configure the field device's location by copying coordinates	O	Yes / No	
		ISO 26048-1§ 8.8.2.1.3	Configure the field device's location by GNSS reading	O	Yes / No	
		ISO 26048-1§ 8.8.2.1.4	Configure the field device's physical components	M	Yes	
		ISO 26048-1§ 8.12	Owner feature	M	Yes	
2.5.1.2	Manage Communications			O	Yes / No	Propose to delete.
		3.5.1.2.1.1	Enable/Disable Communications Port	M	Yes	The ASC shall not be allowed to enable/disable the following ports numbers: _________
		3.5.1.2.2.1	Determine Number of ASC Communications Ports	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.1.3	Manage Cabinet Environment			O	Yes / No	
		ISO 26048-1§ 8.8.2.1.4	Configure the cabinet's physical components	M	Yes	
		ISO 26048-1§ 8.8.12	Field device doors	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of doors to be supported (Default: 2) ___
		ISO 26048-1§ 8.8.13	Field device fans	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of fans supported (Default: 2) ____
		ISO 26048-1§ 8.8.15	Field device heaters	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of heaters supported (Default: 1) ____
		ISO 26048-1§ 8.8.16	Field device humidity	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of humidity sensors supported (Default: 1) ____
		ISO 26048-1§ 8.8.21	Field device temperature	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of temperature sensors supported (Default: 1) ____
2.5.1.4 (Power)	Monitor Power			O	Yes / No	
		ISO 26048-1§ 8.8.2.1.9	Monitor current power source	M	Yes	
		ISO 26048-1§ 8.8.2.2.1.a	Support power sources – mainline (alternating current) power	M	Yes	
		ISO 26048-1§ 8.8.2.2.1.b	Support power sources – battery power	UPS:M	Yes / NA	
		ISO 26048-1§ 8.8.2.2.1.c (Generator)	Support power sources – generator power	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		ISO 26048-1§ 8.2.2.2.1.d (Solar)	Support power sources – solar power	O	Yes / No	
		ISO 26048-1§ 8.8.2.2.2 (UPS)	Support UPS power	O	Yes / No	
		ISO 26048-1§ 8.8.10	Field device battery	UPS:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet battery supported (Default: 1) _____
		ISO 26048-1§ 8.8.14	Field device generator	Generator:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet generator supported (Default: 1) _____
		ISO 26048-1§ 8.8.17	Field device mains power	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet mains power supported (Default: 1) _____
		ISO 26048-1§ 8.8.20	Field device solar power	Solar:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet solar power system supported (Default: 1) _____
2.5.1.5	Retrieve Operational Performance Data			O	Yes / No	
		ISO 26048-1§ 8.13	Recording feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.1.6	Manage Auxiliary External Inputs/Outputs			O	Yes / No	
		ISO 26048-1§ 8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
						Number of additional SRSA ports to support: ____. This value is exclusive of ports used to monitor cabinet power, cabinet doors, fans, heaters, temperature sensor and humidity sensors.
2.5.1.7	Manage Database			M	Yes	
		3.5.1.7.1	Determine Configuration Identifier Parameter Content	M	Yes	
		ISO 26048-1§ 8.5.2.14	Obtain the field device configuration identifier	M	Yes	
2.5.1.8	Manage Interface with External Detectors			O	Yes / No	See 2.5.3.5 for detector data collected by the ASC through external inputs. Requirements to view advanced detector data are found in NTCIP 1209.
2.5.1.9	Manage ASC Clock			M	Yes	
		ISO 26048-1§ 8.2.1	UTC clock	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.2.2	Local clock	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.2.3	Daylight saving time	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.1.10	Manage External Control Local Application State			O	Yes / No	
		3.5.1.10.1.1	Enable ECLA Communications	M	Yes	
		3.5.1.10.1.2	Disable ECLA Communications	M	Yes	
		3.5.1.10.2	Monitor ECLA Data Input Time	M	Yes	
2.5.2	Manage Signal Operations			M	Yes	
2.5.2.1	Manage Signal Configuration			M	Yes	
2.5.2.1.1	Manage Controller Startup Functions			M	Yes	
		3.5.2.1.1.1.1	Configure Start-Up Flash Mode	O	Yes / No	
		3.5.2.1.1.1.2	Configure Start-Up Flash Time	M	Yes	
		3.5.2.1.1.2	Configure Backup Time	M	Yes	
2.5.2.1.2	Manage Phase Configurations			M	Yes	
		3.5.2.1.2.1.1	Enable/Disable Phase	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.2.1.2.1.2	Configure Phase Minimum Green Time	M	Yes	
		3.5.2.1.2.1.3	Configure Phase Passage Time	M	Yes	
		3.5.2.1.2.1.4	Configure Two Fixed Phase Maximum Green Times	M	Yes	
		3.5.2.1.2.1.5	Configure Three Fixed Phase Maximum Green Times	O	Yes / No	
		3.5.2.1.2.1.6	Configure Phase Yellow Change Time	M	Yes	
		3.5.2.1.2.1.7	Configure Phase Red Clearance Time	M	Yes	
		3.5.2.1.2.1.8	Configure Phase Red Revert Time	O	Yes / No	
		3.5.2.1.2.1.9	Configure Unit Red Revert Time	Unit:M	Yes / NA	
		3.5.2.1.2.1.10	Configure Phase Added Initial Time	M	Yes	
		3.5.2.1.2.1.11	Configure Phase Maximum Initial Time	M	Yes	
		3.5.2.1.2.1.12	Configure Phase Time Before Reduction	M	Yes	
		3.5.2.1.2.1.13	Configure Phase Time to Reduce	M	Yes	
		3.5.2.1.2.1.14	Configure Phase Cars Before Reduction	O	Yes / No	
		3.5.2.1.2.1.15	Configure Phase Reduce By Time	O	Yes / No	
		3.5.2.1.2.1.16	Configure Phase Minimum Gap Time	M	Yes	
		3.5.2.1.2.1.17	Configure Phase Dynamic Maximum Limit	O	Yes / No	
		3.5.2.1.2.1.18	Configure Phase Dynamic Maximum Step	O	Yes / No	
		3.5.2.1.2.1.19	Configure Phase Start-Up State	M	Yes	
		3.5.2.1.2.1.20	Configure Automatic Flash Entry Phase	O	Yes / No	
		3.5.2.1.2.1.21	Configure Automatic Flash Exit Phase	O	Yes / No	
		3.5.2.1.2.1.22	Configure Call to Non-Actuated 1	O	Yes / No	
		3.5.2.1.2.1.23	Configure Call to Non-Actuated 2	O	Yes / No	
		3.5.2.1.2.1.24	Configure Non-Lock Detector Memory	O	Yes / No	
		3.5.2.1.2.1.25	Configure Phase Minimum Vehicle Recall	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.26	Configure Phase Maximum Vehicle Recall	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.27	Configure Phase Soft Vehicle Recall	O	Yes / No	
		3.5.2.1.2.1.28	Configure Dual Phase Entry	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.29	Configure Simultaneous Gap Disable	O	Yes / No	
		3.5.2.1.2.1.30	Configure Guaranteed Passage	O	Yes / No	
		3.5.2.1.2.1.31	Configure Actuated Rest-in-Walk	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.32	Configure Conditional Service Enable	O	Yes / No	
		3.5.2.1.2.1.33	Configure Added Initial Calculation	O	Yes / No	
		3.5.2.1.2.1.34	Configure Phase-to-Ring Association	M	Yes	
		3.5.2.1.2.1.35	Configure Phase Concurrency	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.2.1.36	Configure Pedestrian Clearance Time Allowed During Vehicle Clearance	O	Yes / No	
		3.5.2.1.2.1.37	Configure Pedestrian Walk Time	M	Yes	
		3.5.2.1.2.1.38	Configure Pedestrian Clearance Time	M	Yes	
		3.5.2.1.2.1.39	Configure Pedestrian Phase Walk Service Limit	M	Yes	
		3.5.2.1.2.1.40	Configure Pedestrian Phase Don't Walk Revert Time	M	Yes	
		3.5.2.1.2.1.41	Configure Non-Lock Ped Detector Memory	M	Yes	
		3.5.2.1.2.1.42	Configure Pedestrian Phase Recall	M	Yes	
		3.5.2.1.2.1.43	Configure Phase Alternate Pedestrian Clearance Time	O	Yes / No	
		3.5.2.1.2.1.44	Configure Phase Alternate Pedestrian Walk Time	O	Yes / No	
		3.5.2.1.2.1.45	Configure Pedestrian Phase Advanced Walk Time	O	Yes / No	
		3.5.2.1.2.1.46	Configure Pedestrian Phase Delayed Walk Time	O	Yes / No	
		3.5.2.1.2.1.47	Configure Phase Advance Warning Green	O	Yes / No	
		3.5.2.1.2.1.48	Configure Phase Advance Warning Red	O	Yes / No	
		3.5.2.1.2.1.49	Configure Flashing Yellow Arrow Associated Vehicle Phase	O	Yes / No	
		3.5.2.1.2.1.50	Configure Flashing Red Arrow Associated Vehicle Phase	O	Yes / No	
		3.5.2.1.2.1.51	Configure Alternate Minimum Green Time during Transition	O	Yes / No	
		3.5.2.1.2.1.52	Configure Alternate Minimum Walk Time during Transition	O	Yes / No	
		3.5.2.1.2.1.53	Configure Alternate Minimum Pedestrian Clearance Time during Transition	O	Yes / No	
		3.5.2.1.2.2(PhaseSet)	Configure Multiple Phase Sets	O	Yes / No	
		3.5.2.1.2.3.1	Determine Maximum Number of Phases	M	Yes	The ASC shall support at least ____ phases.
		3.5.2.1.2.3.2	Determine Maximum Number of Phase Sets	PhaseSet:M	Yes / NA	The ASC shall support at least ____ phase sets.
2.5.2.1.3 (Coord)	Manage Coordination Configurations			O	Yes / No	
		3.5.2.1.3.1	Configure Operational Mode for Coordination	M	Yes	The ASC shall support the following values: ___ automatic ___ pattern

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						___ manual free ___ manual flash
		3.5.2.1.3.2	Configure Correction Mode for Coordination	M	Yes	The ASC shall support the following values: __ dwell ___ shortway __ add only ___ subtract only
		3.5.2.1.3.3	Configure Maximum Mode for Coordination	M	Yes	The ASC shall support the following values: ___ maxInhibit ___maximum1 ___ maximum2 ___maximum3
		3.5.2.1.3.4	Configure Unit-level Force Mode for Coordination	O	Yes / No	
		3.5.2.1.3.5	Configure Phase-level Force Mode for Coordination	O	Yes / No	
		3.5.2.1.3.6	Configure Pattern Reference Phase	M	Yes	
		3.5.2.1.3.7	Configure Pattern Reference Point	M	Yes	
		3.5.2.1.3.9	Configure Pattern Synchronization Time	M	Yes	
2.5.2.1.4	Manage Timing Patterns			Coord:M	Yes / NA	
		3.5.2.1.4.1.1	Configure Pattern to Run Free	M	Yes	
		3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	O	Yes / No	
		3.5.2.1.4.1.3	Configure Pattern Cycle Time for Coordination	M	Yes	
		3.5.2.1.4.1.4	Configure Pattern Offset Time	M	Yes	
		3.5.2.1.4.1.5	Configure Pattern Split Association	M	Yes	
		3.5.2.1.4.1.6	Configure Pattern Sequence Association	M	Yes	
		3.5.2.1.4.1.7	Configure Pattern Maximum Mode	O	Yes / No	
		3.5.2.1.4.1.8	Configure Pattern Phase Set	O	Yes / No	
		3.5.2.1.4.1.9	Configure Pattern Overlap Set	O	Yes / No	
		3.5.2.1.4.1.10	Configure Pattern Vehicle Detector Set	O	Yes / No	
		3.5.2.1.4.1.11	Configure Pattern Pedestrian Detector Set	O	Yes / No	
		3.5.2.1.4.1.12	Configure Pattern Special Functions	O	Yes / No	
		3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	M	Yes	The ASC shall support at least ____ timing patterns.
2.5.2.1.5	Manage Splits Configurations			O	Yes / No	
		3.5.2.1.3.8	Configure Omit Phases During Transitions	O	Yes / No	
		3.5.2.1.5.1.1	Configure Phase Split Time	M	Yes	
		3.5.2.1.5.1.2	Configure Phase Split Mode	M	Yes	
		3.5.2.1.5.1.3	Configure Split Coordination Phase	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.5.2	Determine Maximum Number of Phase Splits	M	Yes	The ASC shall support at least ____ splits
2.5.2.1.6 (Ring)	Manage Ring Configurations			O	Yes / No	
		3.5.2.1.6.1	Configure Sequence Data	M	Yes	
		3.5.2.1.6.2	Determine Maximum Number of Rings	M	Yes	The ASC shall support at least ____ rings
		3.5.2.1.6.3	Determine Maximum Number of Sequences	M	Yes	The ASC shall support at least ____ sequences
2.5.2.1.7 (Channel)	Manage Channel Configurations			O	Yes / No	
		3.5.2.1.7.1.1	Configure Channel Control Source	M	Yes	
		3.5.2.1.7.1.2	Configure Channel Control Type	M	Yes	
		3.5.2.1.7.1.3.1	Configure Channel Flash Yellow	M	Yes	
		3.5.2.1.7.1.3.2	Configure Channel Flash Red	M	Yes	
		3.5.2.1.7.1.3.3	Configure Channel Flash Alternate Half Hertz	O	Yes / No	
		3.5.2.1.7.1.3.4	Configure Channel Flash Alternate First or Second	O	Yes / No	
		3.5.2.1.7.2	Determine Maximum Number of Channels	M	Yes	The ASC shall support at least ____ channels (See appropriate hardware specification such as NEMA TS 2 to determine maximum number of supported channels)
2.5.2.1.8 (Overlap)	Manage Overlap Configurations			O	Yes / No	
		3.5.2.1.8.1.1.1	Configure Overlap Type - Vehicle Normal	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.2	Configure Overlap Type - Vehicle Minus Green and Yellow	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.3	Configure Overlap Type - Pedestrian Normal	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.4	Configure Overlap Type - Flashing Yellow Arrow - 3 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.5	Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.6	Configure Overlap Type - Flashing Red Arrow - 3 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.7	Configure Overlap Type - Flashing Red Arrow - 4 Section Head	O.4 (1..*)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.8.1.1.8	Configure Overlap Type - 2 Section Transit Specific Signal Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.9	Configure Overlap Type - Minus Green Yellow Alternate	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.2	Configure Overlap Included Phases	M	Yes	
		3.5.2.1.8.1.3	Configure Overlap Modifier Phases	M	Yes	
		3.5.2.1.8.1.4	Configure Pedestrian Modifier Phases	O	Yes / No	
		3.5.2.1.8.1.5	Configure Overlap Trailing Green	M	Yes	
		3.5.2.1.8.1.6	Configure Overlap Trailing Yellow	M	Yes	
		3.5.2.1.8.1.7	Configure Overlap Trailing Red Clearance	M	Yes	
		3.5.2.1.8.1.8	Configure Overlap Walk	O	Yes / No	
		3.5.2.1.8.1.9	Configure Overlap Pedestrian Clearance	O	Yes / No	
		3.5.2.1.8.2 (OverlapSet)	Configure Multiple Overlap Sets	O	Yes / No	
		3.5.2.1.8.3.1	Determine Maximum Number of Overlaps	M	Yes	The ASC shall support at least ____ overlaps
		3.5.2.1.8.3.2	Determine Maximum Number of Overlap Sets	OverlapSet:M	Yes / NA	The ASC shall support at least ____ overlap sets.
2.5.2.1.9 (Preempt)	Manage Preempt Configurations			O	Yes / No	
		3.5.2.1.9.1.1	Enable/Disable Preempt Inputs	M	Yes	
		3.5.2.1.9.1.2	Configure Preempt Control - Non-Locking Memory	O	Yes / No	
		3.5.2.1.9.1.3	Configure Preempt Control - Override Automatic Flash	O	Yes / No	
		3.5.2.1.9.1.4	Configure Preempt Control - Override Preempt	O	Yes / No	
		3.5.2.1.9.1.5	Configure Preempt Control - Flash Dwell	O	Yes / No	
		3.5.2.1.9.1.6	Configure Preempt Control - All Red Entry	O	Yes / No	
		3.5.2.1.9.1.7	Configure Preempt Link	M	Yes	
		3.5.2.1.9.1.8	Configure Preempt Delay	M	Yes	
		3.5.2.1.9.1.9	Configure Preempt Minimum Duration	M	Yes	
		3.5.2.1.9.1.10.1	Configure Preempt Enter Minimum Green Time	O	Yes / No	
		3.5.2.1.9.1.10.2	Configure Preempt Enter Minimum Walk Time	O	Yes / No	
		3.5.2.1.9.1.10.3	Configure Preempt Enter Pedestrian Clearance Time	O	Yes / No	
		3.5.2.1.9.1.10.4	Configure Preempt Enter Yellow Change Time	O	Yes / No	
		3.5.2.1.9.1.10.5	Configure Preempt Enter Red Clearance Time	O	Yes / No	
		3.5.2.1.9.1.11.1	Configure Preempt Track Clearance Green Time	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.9.1.11.2	Configure Preempt Track Clearance Yellow Change Time	O	Yes / No	
		3.5.2.1.9.1.11.3	Configure Preempt Track Red Clearance Time	O	Yes / No	
		3.5.2.1.9.1.11.4	Configure Preempt Track Clearance Phases	M	Yes	
		3.5.2.1.9.1.11.5	Configure Preempt Track Clearance Overlaps	O	Yes / No	
		3.5.2.1.9.1.12.1	Configure Preempt Minimum Green Dwell Time	M	Yes	
		3.5.2.1.9.1.12.2	Configure Preempt Dwell Phases	M	Yes	
		3.5.2.1.9.1.12.3	Configure Preempt Dwell Pedestrian Movements	O	Yes / No	
		3.5.2.1.9.1.12.4	Configure Preempt Dwell Overlaps	O	Yes / No	
		3.5.2.1.9.1.12.5	Configure Preempt Cycling Phases	O	Yes / No	
		3.5.2.1.9.1.12.6	Configure Preempt Cycling Pedestrian Movements	O	Yes / No	
		3.5.2.1.9.1.12.7	Configure Preempt Cycling Phases Sequence	O	Yes / No	
		3.5.2.1.9.1.12.8	Configure Preempt Cycling Overlaps	O	Yes	
		3.5.2.1.9.1.13.1	Configure Preempt Exit Phases	O	Yes / No	
		3.5.2.1.9.1.13.2	Configure Preempt Exit Phase Strategy	O	Yes / No	
		3.5.2.1.9.1.13.3	Configure Preempt Exit Priority Levels	C	Yes / No	Mandatory for Conformance if 'Exit to Queue Delay Recovery' is supported as a Preempt Exit Phase (See 3.5.2.1.9.1.13.2)
		3.5.2.1.9.1.14.1	Configure Preempt Maximum Presence Time	M	Yes	
		3.5.2.1.9.1.14.2	Configure Preempt Maximum Presence Action	M	Yes	
		3.5.2.1.9.1.15	Configure Preempt Gate Description	O	Yes / No	
		3.5.2.1.9.2	Determine Maximum Number of Preempts	M	Yes	The ASC shall support at least ____ preempts
2.5.2.1.10	Manage Timing Pattern Scheduler			M	Yes / No	
		ISO 26048-1§ 8.1.2.3	Configure an action owner	M	Yes	
		ISO 26048-1§ 8.1.2.4	Configure an action group	M	Yes	
		ISO 26048-1§ 8.1.2.5	Configure an action	M	Yes	
		ISO 26048-1§ 8.6	Day plan feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.2.1.11	Manage Action Scheduler			M	Yes / No	
		ISO 26048-1§ 8.1.2.3	Configure an action owner	M	Yes	
		ISO 26048-1§ 8.1.2.4	Configure an action group	M	Yes	
		ISO 26048-1§ 8.1.2.5	Configure an action	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.6	Day plan feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		3.5.2.1.4.1.1	Configure Pattern to Run Free	patternFunction:M	Yes / NA	
		3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	patternFunction:M	Yes / NA	
		3.5.2.1.4.1.12(patternFunction)	Configure Pattern Special Functions	O.5 (1..*)	Yes / No	
		3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	patternFunction:M	Yes	
		3.5.2.1.10.1.1	Configure Timebased Action - Pattern	actionFunction:M	Yes / NA	
		3.5.2.1.10.1.2(actionFunction)	Configure Timebased Action - Special Functions	O.5 (1..*)	Yes / No	
		3.5.2.1.10.1.3	Determine Maximum Number of Timebased Actions	actionFunction:M	Yes / NA	
2.5.2.1.12	Manage I/O Mapping			O	Yes / No	
		3.5.2.1.11.1.1	Set Active I/O Map	M	Yes	
		3.5.2.1.11.1.2.1	Configure I/O Map Description	M	Yes	
		3.5.2.1.11.1.2.2.1	Configure I/O Map Input Device	M	Yes	
		3.5.2.1.11.1.2.2.2	Configure I/O Map Input Device Pin	M	Yes	
		3.5.2.1.11.1.2.2.3	Configure I/O Map Input Function	M	Yes	
		3.5.2.1.11.1.2.3.1	Configure I/O Map Output Device	M	Yes	
		3.5.2.1.11.1.2.3.2	Configure I/O Map Output Device Pin	M	Yes	
		3.5.2.1.11.1.2.3.3	Configure I/O Map Output Function	M	Yes	
		3.5.2.1.11.2.1	Retrieve Maximum Number of I/O Maps	M	Yes	
		3.5.2.1.11.2.2	Retrieve Maximum Number of I/O Map Inputs	M	Yes	
		3.5.2.1.11.2.3	Retrieve Maximum Number of I/O Map Outputs	M	Yes	
		3.5.2.1.11.2.4	Retrieve I/O Mapping Activate Conditions	M	Yes	The following conditions shall be satisfied before a new I/O map can be activated: ___ Cabinet Door Open ___ in any flash state ___ programmed all red flash ___ in CVM flash ___ ASC restart
		3.5.2.1.11.2.5	Retrieve I/O Mapping Input Functions	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.11.2.6	Retrieve I/O Mapping Output Functions	M	Yes	
		3.5.2.1.11.2.7	Retrieve I/O Map Input Device Pin Status	M	Yes	
		3.5.2.1.11.2.8	Retrieve I/O Map Output Device Pin Status	M	Yes	
		3.5.2.1.11.2.9.1	Enumerate I/O Map - FIO Inputs	332:M	Yes / NA	
		3.5.2.1.11.2.9.2	Enumerate I/O Map - FIO Outputs	332:M	Yes / NA	
		3.5.2.1.11.2.9.3	Enumerate I/O Map - TS1 Inputs	TS1, TS2-2:M	Yes / NA	
		3.5.2.1.11.2.9.4	Enumerate I/O Map - TS1 Outputs	TS1, TS2-2:M	Yes / NA	
		3.5.2.1.11.2.9.5	Enumerate I/O Map - TS2 BIU Inputs	TS2-1:M	Yes / NA	
		3.5.2.1.11.2.9.6	Enumerate I/O Map - TS2 BIU Outputs	TS2-1:M	Yes / NA	
		3.5.2.1.11.2.9.7	Enumerate I/O Map - ATC Cabinet SIU Inputs	ATC:M	Yes / NA	
		3.5.2.1.11.2.9.8	Enumerate I/O Map - ATC Cabinet SIU Outputs	ATC:M	Yes / NA	
		3.5.2.1.11.2.9.9	Enumerate I/O Map - Auxiliary Device Inputs	O	Yes / No	
		3.5.2.1.11.2.9.10	Enumerate I/O Map - Auxiliary Device Outputs	O	Yes / No	
2.5.2.1.13	Manage Intra-Cabinet Communications Configuration			O	Yes / No	
		3.5.2.1.12.1.1	Determine Serial Bus 1 Device Present	ATC:M	Yes / NA	The ASC shall support at least ____ Serial Bus 1 Addresses (between 1 and 255).
		3.5.2.1.12.2.1	Determine TS2 Port 1 Device Present	TS2-2:M	Yes / NA	The ASC shall support at least ____ TS2 Port1 Addresses (between 1 and 255).
		3.5.2.1.12.2.2	Enable/Disable TS2 Port 1 Frame 40 Messages	TS2-2:M	Yes / NA	
2.5.2.1.14	Manage Accessible Pedestrian Support			O	Yes / No	
		3.5.2.1.13.1.1	Configure APS Push Button Minimum Press Time	M	Yes	
		3.5.2.1.13.1.2	Configure APS Push Button to Phase Association	M	Yes	
		3.5.2.1.13.1.3	Configure APS Extra Crossing Time	M	Yes	
		3.5.2.1.13.1.4	Configure Pedestrian Detector for Alternate Pedestrian Timing	M	Yes	
2.5.2.2	Monitor Signal Operations Status					
2.5.2.2.1	Determine Controller Health			M	Yes	
		ISO 26048-1§ 8.5.2.9	Monitor when the SNMP capabilities last changed	M	Yes	
		ISO 26048-1§ 8.5.2.10	Monitor controller communications	M	Yes	
		ISO 26048-1§ 8.5.2.11	Monitor controller operational status	M	Yes	
		ISO 26048-1§ 8.5.2.12	Monitor controller up time	M	Yes	
		ISO 26048-1§ 8.5.2.13	Monitor watchdog failure count	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.1.1	Monitor External Alarm Input States	M	Yes	The ASC shall support at least ____ Alarm Groups (between 1 and 255).
		3.5.2.2.1.2	Monitor External Alarm Active	M	Yes	
		3.5.2.2.1.3	Monitor Flash Status	M	Yes	
		3.5.2.2.1.4	Monitor Local Override	M	Yes	
		3.5.2.2.1.5	Monitor Coordination Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.6	Monitor Detector Fault	Detector:M	Yes / NA	
		3.5.2.2.1.7	Monitor Stop Time Input Alarm	M	Yes	
		3.5.2.2.1.8	Monitor Cycle Fault Alarm	M	Yes	
		3.5.2.2.1.9	Monitor Coordination Fault	Coord:M	Yes / NA	
		3.5.2.2.1.10	Monitor Coordination Fail Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.11	Monitor Cycle Fail Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.12	Monitor Cabinet IO Link Alarm	M	Yes	
		3.5.2.2.1.13	Monitor SMU Communications Error	O	Yes / No	
		3.5.2.2.1.14	Monitor Preempt Maximum Presence Alarm	Preempt:O	Yes / No / NA	
2.5.2.2.2	Determine Mode of Operation					
2.5.2.2.2.1 (Unit)	Monitor Unit-wide General Operations			O	Yes / No	
		3.5.2.2.2.1	Monitor Unit Control Status	M	Yes	
		3.5.2.2.2.2	Monitor Preempt Active	Preempt:M	Yes / NA	
		3.5.2.2.2.3	Monitor Offset Transitioning	Coord:M	Yes / NA	
		3.5.2.2.2.4	Monitor Priority Call Active	O	Yes / No	
		3.5.2.2.2.5	Monitor Local Free Status	M	Yes	
		3.5.2.2.2.6	Monitor Coordination Active	Coord:M	Yes / NA	
		3.5.2.2.2.7	Monitor ECLA Control Active	O	Yes / No	
2.5.2.2.2.2	Monitor Flashing			M	Yes	
		3.5.2.2.1.3	Monitor Flash Status	M	Yes	
2.5.2.2.2.3	Monitor Current Timing Pattern			Coord:M	Yes / NA	
		3.5.2.2.2.5	Monitor Local Free Status	M	Yes	
		3.5.2.2.2.8.1	Monitor Current Pattern Status	M	Yes	
		3.5.2.2.2.8.2	Monitor Current Pattern Command Source	O	Yes / No	
		3.5.2.2.2.8.3	Monitor Current Pattern Fault Status	O	Yes / No	
2.5.2.2.2.4	Monitor Current Cycle			Coord:M	Yes / NA	
		3.5.2.2.2.9.1	Monitor Coordination Cycle Status	M	Yes	
		3.5.2.2.2.9.2	Monitor Coordination Synchronization Status	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.9.3	Monitor Current Offset	M	Yes	
2.5.2.2.3	Monitor Phase Status			M	Yes	
		3.5.2.2.3.1	Monitor Active Red Phases	M	Yes	
		3.5.2.2.3.2	Monitor Active Yellow Phases	M	Yes	
		3.5.2.2.3.3	Monitor Active Green Phases	M	Yes	
		3.5.2.2.3.4	Monitor Active Don't Walk Phases	M	Yes	
		3.5.2.2.3.5	Monitor Active Pedestrian Clearance Phases	M	Yes	
		3.5.2.2.3.6	Monitor Active Walk Phases	M	Yes	
		3.5.2.2.3.7	Monitor Active On Phases	M	Yes	
		3.5.2.2.3.8	Monitor Next Phases	M	Yes	
		3.5.2.2.3.9	Monitor Phase Vehicle Calls	M	Yes	
		3.5.2.2.3.10	Monitor Phase Pedestrian Calls	M	Yes	
		3.5.2.2.7.4	Monitor Current Gate Status	O	Yes / No	
2.5.2.2.4	Monitor Ring Status			Ring:M	Yes / NA	
		3.5.2.2.4.1	Monitor Ring Status	M	Yes	
		3.5.2.2.4.2	Monitor Ring Termination Cause	M	Yes	
		3.5.2.2.4.3	Monitor Current Phase On Time	M	Yes	
2.5.2.2.5	Monitor Channel Status			Channel:M	Yes / NA	
		3.5.2.2.5.1	Monitor Active Red Channels	M	Yes	
		3.5.2.2.5.2	Monitor Active Yellow Channels	M	Yes	
		3.5.2.2.5.3	Monitor Active Green Channels	M	Yes	
2.5.2.2.6	Monitor Overlap Status			Overlap:M	Yes / NA	
		3.5.2.2.6.1	Monitor Active Red Overlaps	M	Yes	
		3.5.2.2.6.2	Monitor Active Yellow Overlaps	M	Yes	
		3.5.2.2.6.3	Monitor Active Green Overlaps	M	Yes	
		3.5.2.2.6.4	Monitor Active Flashing Yellow Arrow Overlaps	O	Yes / No	
		3.5.2.2.6.5	Monitor Active Flashing Red Arrow Overlaps	O	Yes / No	
2.5.2.2.7	Monitor Preempt Status			Preempt:M	Yes / NA	
		3.5.2.2.7.1	Monitor Currently Active Preempt	M	Yes	
		3.5.2.2.7.2	Monitor Current Preempt Inputs	M	Yes	
		3.5.2.2.7.3	Monitor Current Preempt State	M	Yes	
		3.5.2.2.7.4	Monitor Current Gate Status	O	Yes / No	
2.5.2.2.8 (SpecialFunc)	Monitor Special Function Outputs			O	Yes / No	
		3.5.2.2.8.1	Determine Maximum Number of Special Functions	M	Yes	The ASC shall support at least ____ Special Functions (between 1 and 255).

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.8.2	Monitor Special Function Status	M	Yes	
		3.5.2.2.8.3	Monitor Special Function Control Source	O	Yes / No	
2.5.2.2.9	Monitor Timebase Action Status			M	Yes	
		3.5.2.2.2.8.1	Monitor Current Pattern Status	patternFunction:M	Yes / NA	
		3.5.2.1.10.1.4	Determine Action in Effect	actionFunction:M	Yes / NA	
2.5.2.2.10	Monitor Intra-Cabinet Communications Configuration			O	Yes / No	
		3.5.2.2.9.1	Monitor TS2 Port 1 Status	TS2-2:M	Yes / NA	
		3.5.2.2.9.2	Monitor TS2 Port 1 Fault Frame	TS2-2:M	Yes / NA	
		3.5.2.2.9.3	Monitor Serial Bus 1 Status	ATC:M	Yes / NA	
2.5.2.2.11	Monitor Signal Monitoring Unit			O	Yes / No	
		3.5.2.2.10.1	Monitor Signal Monitoring Unit Channel Voltage	ATC:M	Yes	It's optional for any cabinet, but format is standard on an ATC.
		3.5.2.2.10.2	Monitor Signal Monitoring Unit Channel Current	ATC:M	Yes	
2.5.2.3	Control Signal Operations			M	Yes	
2.5.2.3.1	Control ASC-wide General Operations			M	Yes	
		3.5.2.3.1.1	Enable/Disable Manual Backup	O	Yes / No	
		3.5.2.3.1.2	Control Global Minimum Recall	M	Yes	
		3.5.2.3.1.3	Control Call to Non-Actuated 1	M	Yes	
		3.5.2.3.1.4	Control Call to Non-Actuated 2	M	Yes	
		3.5.2.3.1.5	Control Walk Rest Modifier	M	Yes	
		3.5.2.3.1.6	Control Interconnect	O	Yes / No	
2.5.2.3.2	Activate Timing Pattern			Coord:M	Yes / NA	
		3.5.2.3.2.1	Activate System Timing Pattern Remotely	M	Yes	
		3.5.2.3.2.2	Control System Reference Point	M	Yes	
2.5.2.3.3	Phase Requests			O	Yes / No	
		3.5.2.3.3.1	Control Phase Omits	M	Yes	
		3.5.2.3.3.2	Control Pedestrian Phase Omits	M	Yes	
		3.5.2.3.3.3	Control Phase Holds	M	Yes	
		3.5.2.3.3.4	Control Phase Force Offs	O	Yes / No	
		3.5.2.3.3.5	Control Phase Vehicle Calls	M	Yes	
		3.5.2.3.3.6	Control Phase Pedestrian Calls	M	Yes	
2.5.2.3.4	Activate Preempt			Preempt:O	Yes / No / NA	
		3.5.2.3.4	Activate Preempt Remotely	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.2.3.5	Control Ring Operations			Ring:O	Yes / No / NA	
		3.5.2.3.5.1	Control Ring Stop Time	M	Yes	
		3.5.2.3.5.2	Control Ring Force Offs	M	Yes	
		3.5.2.3.5.3	Control Ring Maximum 2 Settings	M	Yes	
		3.5.2.3.5.4	Control Ring Maximum 3 Settings	O	Yes / No	
		3.5.2.3.5.5	Control Ring Maximum Inhibit Settings	M	Yes	
		3.5.2.3.5.6	Control Ring Pedestrian Recycle Settings	M	Yes	
		3.5.2.3.5.7	Control Ring Red Rest Settings	M	Yes	
		3.5.2.3.5.8	Control Ring Red Clearance Omit Settings	M	Yes	
2.5.2.3.6	Activate Special Function Output			SpecialFunc: O	Yes / No / NA	
		3.5.2.3.6	Activate Special Function Remotely	M	Yes	
2.5.2.3.7	Activate Action Plan			O	Yes / No	
		3.5.2.1.10.5	Activate Action Plan Remotely	actionFunction:M	Yes / NA	
		3.5.2.3.2.1	Activate System Timing Pattern Remotely	patternFunction:M	Yes / NA	
2.5.2.3.8	Remote Manual Control			O	Yes / No	
		3.5.2.3.7.1	Enable Remote Manual Control	M	Yes	
		3.5.2.3.7.2	Advance Interval During Remote Manual Control	M	Yes	
		3.5.2.3.7.3	Configure Manual Control Timeout	M	Yes	
		3.5.2.3.7.4	Enable/Disable Automatic Pedestrian Clearance Setting	M	Yes	
2.5.3	Manage Detectors					
2.5.3.1 (Detector)	Manage Detector Configuration			M	Yes	
		3.5.3.1.1.1	Configure Vehicle Travel Mode	O	Yes / No	
		3.5.3.1.1.2	Configure Vehicle Detector Description	O	Yes / No	
		3.5.3.1.1.3	Configure Vehicle Detector Yellow Lock Call Enabled	O	Yes / No	
		3.5.3.1.1.4	Configure Vehicle Detector Red Lock Call Enabled	O	Yes / No	
		3.5.3.1.1.5	Configure Vehicle Detector Passage Enabled	O	Yes / No	
		3.5.3.1.1.6	Configure Vehicle Detector Added Initial Time Enabled	O	Yes / No	
		3.5.3.1.1.7	Configure Vehicle Detector Queue Enabled	O	Yes / No	
		3.5.3.1.1.8	Configure Vehicle Detector Call Enabled	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.3.1.1.9	Configure Vehicle Detector Call Phase	M	Yes	
		3.5.3.1.1.10	Configure Vehicle Detector Switch Phase	M	Yes	
		3.5.3.1.1.11	Configure Vehicle Detector Delay Time	M	Yes	
		3.5.3.1.1.12	Configure Vehicle Detector Extend Time	M	Yes	
		3.5.3.1.1.13	Configure Vehicle Detector Queue Limit Time	O	Yes / No	
		3.5.3.1.1.14	Configure Vehicle Detector No Activity Fault Time	M	Yes	
		3.5.3.1.1.15	Configure Vehicle Detector Maximum Presence Fault Time	M	Yes	
		3.5.3.1.1.16	Configure Vehicle Detector Erratic Counts	M	Yes	
		3.5.3.1.1.17	Configure Vehicle Detector Fail Time	O	Yes / No	
		3.5.3.1.2 (VehDetectSet)	Configure Multiple Vehicle Detector Sets for Actuation	O	Yes / No	
		3.5.3.1.3.1	Configure Pedestrian Detector Description	O	Yes / No	
		3.5.3.1.3.2	Configure Pedestrian Detector Call Phase	M	Yes	
		3.5.3.1.3.3	Configure Pedestrian Detector No Activity Fault Time	M	Yes	
		3.5.3.1.3.4	Configure Pedestrian Detector Maximum Presence Fault Time	M	Yes	
		3.5.3.1.3.5	Configure Pedestrian Detector Erratic Counts	M	Yes	
		3.5.3.1.3.6	Configure Pedestrian Detector Non-Lock Calls	O	Yes / No	
		3.5.3.1.3.7	Configure Pedestrian Detector for Presence Detection	O	Yes / No	
		3.5.3.1.3.8	Configure Pedestrian Detector for Delayed Walk	O	Yes / No	
		3.5.3.1.3.9	Configure Pedestrian Detector for Advanced Walk	O	Yes / No	
		3.5.3.1.4 (PedDetectSet)	Configure Multiple Pedestrian Detector Sets for Actuations	O	Yes / No	
		3.5.3.1.5.1	Determine Maximum Number of Vehicle Detectors	M	Yes	The ASC shall support at least ____ vehicle detectors (between 1 and 255).
		3.5.3.1.5.2	Determine Maximum Number of Vehicle Detector Sets	VehDetectSet: M	Yes / NA	The ASC shall support at least ____ vehicle detector sets.
		3.5.3.1.5.3	Determine Maximum Number of Pedestrian Detectors	M	Yes	The ASC shall support at least ____ pedestrian detectors (between 1 and 255).
		3.5.3.1.5.4	Determine Maximum Number of Pedestrian Detector Sets	PedDetectSet: M	Yes / NA	The ASC shall support at least ____ pedestrian detector sets.

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.3.2	Monitor Detector Status			O	Yes / No	
		3.5.3.2.1	Monitor Active Vehicle Detector Actuations	M	Yes	
		3.5.3.2.2	Monitor Active Pedestrian Detector Actuations	M	Yes	
2.5.3.3	Monitor Detector Health			O	Yes / No	
		3.5.3.3.1.1	Monitor Vehicle Detector Alarm Status	M	Yes	
		3.5.3.3.1.2	Monitor Vehicle Detector Faults from Controller	M	Yes	
		3.5.3.3.1.3	Monitor Vehicle Detector Faults from Detector	O	Yes / No	
		3.5.3.3.3.1	Monitor Pedestrian Detector Alarm Status	M	Yes	
		3.5.3.3.3.2	Monitor Pedestrian Detector Faults	M	Yes	
2.5.3.4	Control Detectors			O	Yes / No	
		3.5.3.4.1	Control Vehicle Detector Reset	M	Yes	
		3.5.3.4.2	Control Pedestrian Detector Reset	M	Yes	
		3.5.3.4.3	Control Detector Diagnostic Reset	O	Yes / No	
		3.5.3.4.4	Control Vehicle Detector Actuation	O	Yes / No	
		3.5.3.4.5	Control Pedestrian Detector Actuation	O	Yes / No	
2.5.3.5	Manage Detector Data Collection			O	Yes / No	
		3.5.3.5.3.1	Configure Vehicle Detector Data Sample Period	M	Yes	
		3.5.3.5.3.2	Configure Pedestrian Detector Sample Period	M	Yes	
		3.5.3.5.3.3 (Speed)	Configure Vehicle Speed Detectors	O	Yes / No	
		3.5.3.5.3.4	Configure Single Detector Speed Mode	Speed:M	Yes / NA	
		3.5.3.5.3.5	Configure Paired Detector	Speed:M	Yes / NA	
		3.5.3.5.3.6	Configure Paired Detector Placement	Speed:M	Yes / NA	
		3.5.3.5.3.7	Configure Paired Detector Spacing	Speed:M	Yes / NA	
		3.5.3.5.3.8	Configure Average Vehicle Length	O	Yes / No	
		3.5.3.5.3.9	Configure Vehicle Detection Zone Length	O	Yes / No	
		3.3.3.5.4	Configure Multiple Vehicle Detector Sets for Data Collection	O	Yes / No	
2.5.3.6	Monitor Detector Data from Controller			O	Yes / No	
		3.5.3.5.1.1	Monitor Vehicle Detector Data Sequence	M	Yes	
		3.5.3.5.1.2	Monitor Vehicle Volume Data	O	Yes / No	
		3.5.3.5.1.3	Monitor Vehicle Occupancy Data	O	Yes / No	
		3.5.3.5.1.4	Monitor Vehicle Average Speed	Speed:M	Yes / NA	
		3.5.3.5.1.5	Monitor Vehicle Detector Data Sample Time	O	Yes / No	
		3.5.3.5.2.1	Monitor Pedestrian Detector Data Sequence	O	Yes / No	
		3.5.3.5.2.2	Monitor Pedestrian Counts	O	Yes / No	
		3.5.3.5.2.3	Monitor Pedestrian Actuations	O	Yes / No	
		3.5.3.5.2.4	Monitor Pedestrian Services	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.4 (CV)	Manage Connected Vehicles Interface			O	Yes / No	
2.5.4.1	Connected Vehicle Interface: Management Station – ASC Interface			M	Yes / No	
2.5.4.1.1	Manage ASC - RSU Interface			M	Yes	
		3.5.4.1.1	Configure ASC Communications Port for RSU	M	Yes	
		3.5.4.1.2	Configure Logical RSU Ports and Address	M	Yes	
		3.5.4.1.3	Configure RSU Interface Polling Period	O	Yes / No	
2.5.4.1.2	Manage ASC - RSU Interface Watchdog			O	Yes / No	
		3.5.4.1.4	Configure RSU Interface Watchdog	M	Yes	
		3.5.4.1.5	Monitor RSU Interface Watchdog Timer	M	Yes	
		3.5.4.1.6	Monitor RSU Interface Watchdog Alarm	M	Yes	
2.5.4.1.3	Manage Signal Phase and Timing Data			M	Yes	
		3.5.4.2.1.1	Enable Signal Phase and Timing Data	M	Yes	
		3.5.4.2.1.2	Retrieve Signal Phase and Timing Generation Time	O	Yes / No	
		3.5.4.2.1.3.1.1	Monitor Movement Minimum End Time	M	Yes	
		3.5.4.2.1.3.1.2	Monitor Movement Maximum End Time	M	Yes	
		3.5.4.2.1.3.1.3	Monitor Movement Likely End Time	O	Yes / No	
		3.5.4.2.1.3.1.4	Monitor Movement Likely End Time Confidence	O	Yes / No	
		3.5.4.2.1.3.1.5	Monitor Movement Next Occurrence	M	Yes	
		3.5.4.2.1.3.1.6	Monitor Movement Start Time	M	Yes	
		3.5.4.2.1.3.1.7	Monitor Next Movement Minimum End Time	M	Yes	
		3.5.4.2.1.3.1.8	Monitor Next Movement Maximum End Time	M	Yes	
		3.5.4.2.1.3.1.9	Monitor Next Movement Start Time	M	Yes	
		3.5.4.2.1.3.1.10	Determine Maximum Number of Movement Events			
		3.5.4.2.1.3.2.1	Configure Queue Detectors for Movement Assistance	MvtQueue:M	Yes / NA	
		3.5.4.2.1.3.2.2	Configure Pedestrian Detectors for Movement Conflict Assistance	MvtConflict:O.4 (1..*)	Yes / No / NA	
		3.5.4.2.1.3.2.3	Configure Bicycle Detectors for Movement Conflict Assistance	MvtConflict:O.4 (1..*)	Yes / No / NA	
		3.5.4.2.1.3.3.1 (MvtQueue)	Monitor Lane Connection Queue Length	O	Yes / No	
		3.5.4.2.1.3.3.2 (MvtConflict)	Monitor Lane Connection Vulnerable Road User Detection	O	Yes / No	
		3.5.4.2.1.3.4.1 (SpdAdvice)	Configure Advisory Speed Type	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.4.2.1.3.4.2	Configure Advisory Speed	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.4.3	Configure Advisory Speed Zone	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.4.4	Configure Advisory Speed Vehicle Type	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.5	Monitor Movement State	M	Yes	
		3.5.4.2.1.3.6	Monitor Next Movement State	M	Yes	
		3.5.4.2.1.3.7	Monitor Movement Status	M	Yes	
		3.5.4.2.1.4.1	Configure Concurrent Enabled Lanes	M	Yes	
		3.5.4.2.1.4.2	Configure Enabled Lanes by Time of Day	M	Yes	
		3.5.4.2.1.4.3	Determine Lanes Enabled	M	Yes	
		3.5.4.2.1.4.4	Command Enabled Lanes	M	Yes	
		3.5.4.2.1.5	Enable Signal Phase and Timing Exchange	M	Yes	
		3.5.4.2.1.6	Configure Road Authority Identifier	M	Yes	
		3.5.4.2.1.7.1	Monitor Manual Control Indication	M	Yes	
		3.5.4.2.1.7.2	Monitor Stop Indication	M	Yes	
		3.5.4.2.1.7.3	Monitor Failure Flash Indication	M	Yes	
		3.5.4.2.1.7.4	Monitor Preemption Operation Indication	M	Yes	
		3.5.4.2.1.7.5	Monitor Priority Operation Indication	M	Yes	
		3.5.4.2.1.7.6	Monitor Fixed Time Control Indication	M	Yes	
		3.5.4.2.1.7.7	Monitor Non-Fixed Time Control Indication	M	Yes	
		3.5.4.2.1.7.8	Monitor Standby Operation Indication	M	Yes	
		3.5.4.2.1.7.9	Monitor Controller Failure	M	Yes	
		3.5.4.2.1.7.10	Monitor MAP Message Validity	M	Yes	
		3.5.4.2.1.7.11	Monitor SPaT Message Validity	M	Yes	
		3.5.4.2.1.8	Mark SPaT Invalid - Controller	M	Yes	
		3.5.4.2.1.9	Mark SPaT Invalid - Port	O	Yes / No	
		3.5.4.2.1.10	Mark MAP Message Invalid - Controller	M	Yes	
		3.5.4.2.1.11	Mark MAP Message Invalid - Port	O	Yes / No	
		3.5.4.2.1.12.1	Determine Maximum Number of Signal Groups	M	Yes	
		3.5.4.2.1.12.2	Configure Signal Groups Intersection Mapping	M	Yes	
		3.5.4.2.1.12.3	Configure Signal Group Control Source	M	Yes	
		3.5.4.2.1.12.4	Configure Signal Group Indication Types	M	Yes	
		3.5.4.2.1.12.5	Configure Signal Group Protected or Permissive State	M	Yes	
		3.5.4.2.1.12.6	Configure Signal Group Revocable Lanes	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.4.2.1.12.7	Determine Maximum Number of Signal State Entries	M	Yes	
		3.5.4.2.1.12.8	Configure Customized Signal State Parameters	M	Yes	
		3.5.4.2.1.13	Retrieve Signal Phase and Timing Time Point	M	Yes	
2.5.4.1.4	Manage Assured Green Period			O	Yes / No	
		3.5.4.2.2.1	Enabled Connected Device Connection	M	Yes	
		3.5.4.2.2.2	Configure Vehicle Detector for Connected Vehicle Application	M	Yes	
		3.5.4.2.2.3	Configure Connected Vehicle Detector Assigned Input	M	Yes	
		3.5.4.2.2.4	Configure Connected Vehicle Detector Port Assignment	O	Yes / No	
		3.5.4.2.2.5	Configure Assured Green Period Duration	M	Yes	
		3.5.4.2.2.6	Configure Red Light Violation Warning Parameters	O	Yes / No	
2.5.4.2	Connected Vehicle Interface: ASC – CV Application Process Interface			M	Yes	
2.5.4.2.1	Exchange Current and Next Movement Information			M	Yes	
		3.5.4.2.1.1.1	Provide Movement Time Point	M	Yes	
		3.5.4.2.1.1.2	Provide Movement State	M	Yes	
		3.5.4.2.1.1.3	Provide Movement Minimum End Time	M	Yes	
		3.5.4.2.1.1.4	Provide Movement Maximum End Time	M	Yes	
		3.5.4.2.1.1.5	Provide Movement Likely End Time	O	Yes / No	
		3.5.4.2.1.1.6	Provide Movement Likely End Time Confidence	O	Yes / No	
		3.5.4.2.1.1.7	Provide Next Movement State	M	Yes	
		3.5.4.2.1.1.8	Provide Next Movement Minimum End Time	M	Yes	
		3.5.4.2.1.1.9	Provide Next Movement Maximum End Time	M	Yes	
		3.5.4.2.1.1.10	Provide Next Movement Start Time	M	Yes	
		3.5.4.2.1.2.1	Provide Lane Connection Queue Length	O	Yes / No	
		3.5.4.2.1.2.2	Provide Lane Connection Traveler Detection	O	Yes / No	
		3.5.4.2.1.3.1	Provide Advisory Speed Type	O	Yes / No	
		3.5.4.2.1.3.2	Provide Advisory Speed	O	Yes / No	
		3.5.4.2.1.3.3	Provide Advisory Speed Zone	O	Yes / No	
		3.5.4.2.1.3.4	Provide Advisory Speed Vehicle Type	O	Yes / No	
		3.5.4.2.1.4	Provide Road Authority ID	M	Yes	
		3.5.4.2.1.5	Provide Signal Phase and Timing Intersection Status	M	Yes	
		3.5.4.3.1.6	Provide Compressed SPaT Information to External CV Application Process	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.4.3.2.1	Provide UPER-encoded SPaT Message	M	Yes	
		3.6.5.1	SPaT Maximum Transmission Start Time	M	Yes	The Maximum Transmission Start Time for all SPAT data shall be ___ milliseconds (Default=10).
		3.6.5.2	Movement Time Point Minimum Transmission Rate	M	Yes	The Movement Time Point Minimum Transmission Rate shall be once per _____ milliseconds (Default=100).
		3.6.5.3	SPaT Maximum Transmission Rate	M	Yes	
		3.6.5.4	SPaT Time Accuracy	M	Yes	
2.5.4.2.2	Exchange Next Occurrence of a Movement			M	Yes	
		3.5.4.3.1.1.11	Provide Movement Next Occurrence	M	Yes	
		3.5.4.3.4	Monitor CV Certificate Faults	O	Yes / No	
		3.6.5.1	SPaT Maximum Transmission Start Time	M	Yes	The Maximum Transmission Start Time for all SPAT data that changed shall be ___ milliseconds (Default=10).
		3.6.5.2	Movement Time Point Minimum Transmission Rate	M	Yes	The Movement Time Point Minimum Transmission Rate shall be once per _____ milliseconds (Default=100).
		3.6.5.3	SPaT Maximum Transmission Rate	M	Yes	
		3.6.5.4	SPaT Time Accuracy	M	Yes	
2.5.4.2.3	Exchange Presence of Connected Devices			O	Yes / No	
		3.5.4.3.2.2	Retrieve BSMs	O.12(1..*)	Yes / No	
		3.5.4.3.2.3	Retrieve PSMs	O	Yes / No	
		3.5.4.3.2.4	Retrieve Actuation Report	O.12(1..*)	Yes / No	
		3.5.4.3.2.5	Retrieve Detection Report	O	Yes / No	
2.5.4.2.4	Exchange Roadway Geometrics Information			O	Yes / No	
		3.5.4.3.3.1	Retrieve MAP Plan in Effect	M	Yes	
		3.5.4.3.3.2	Confirm MAP Plan Compatibility	M	Yes	
2.5.4.3	ASC – ECLA Interface			O	Yes / No	
		3.5.4.4.1	Receive Current Phase Minimum End Time from an ECLA	M	Yes	
		3.5.4.4.2	Receive Current Phase Maximum End Time from an ECLA	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.4.4.3	Receive Current Phase Likely End Time from an ECLA	O	Yes / No	
		3.5.4.4.4	Receive Current Phase Likely End Time Confidence from an ECLA	O	Yes / No	
		3.5.4.4.5	Receive Next Phase from an ECLA	M	Yes	
		3.5.4.4.6	Receive Compressed ECLA Input Data	M	Yes	
2.5.5	Backward Compatibility Features					
2.6	Security			M	Yes	
2.6.1	Manage Authentication			M	Yes	
2.6.2	Manage Accessibility			M	Yes	
2.6.3	Manage Users			M	Yes	
		ISO 26048-1§ 8.22	View-based access control model (VACM) feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.6.4	Log User Access			O	Yes / No	
		ISO 26048-1§ 8.1.3.1	Validate access upon action activation	M	Yes	
		ISO 26048-1§ 8.1.3.2	Validate access upon action being called	M	Yes	
		3.6.1	Response Time for Requests	M	Yes	The Response Time for all requests shall be ___ milliseconds (5-500: Default=25).
2.6.5	Manage ASC Interface Security			O	Yes / No	
2.6.5.1	Manage Security for the ASC to RSU Interface					
2.6.5.2	Manage Security for Other ASC Interfaces					

3.4 Architectural Requirements

Requirements for communication capabilities follow.

3.4.1 Support Basic Communications Requirements

The requirements to respond to requests follow. In addition, some requirements to determine the communications capabilities are found in Sections 8.5 of ISO 26048-1 – the specific requirements can be found under User Need 2.4.1, Provide Live Data in Table 5.

3.4.1.1 Retrieve Data

Upon request from a management station, the ASC shall provide the data requested.

3.4.1.2 Deliver Data

Upon request from a management station, the ASC shall receive the data (e.g., configuration data, commands, etc.) provided.

3.4.1.3 Explore Data

Upon request from a management station, the ASC shall allow dynamic discovery of the data concepts and data instances supported by the ASC.

3.4.1.4 Monitor SNMP Requirements

The requirements for SNMP monitoring conformance follow. In addition, some requirements to monitor and control the ASC are found in Section 8.5 of ISO 26048-1.

3.4.1.4.1 Monitor SNMP Information

Upon request from a management station, the ASC shall return basic instrumentation and control information of the SNMP device.

3.4.2 Manage Data Blocks Requirements

The requirements to manage sets of data so that data may be exchanged more efficiently over communications networks follow. In addition, some requirements to manage sets of data are found in Section 8.7 of ISO 26048-1 – the specific requirements can be found under User Need 2.4.2, Provide Data Blocks in Table 5.

3.4.2.1 Store Pre-defined Compressed Data Blocks

Upon request from a management station, the ASC shall store pre-defined sequences (blocks) of configuration data. Each pre-defined block contains configuration parameters for a functional area associated with the operation and management of an ASC. The pre-defined blocks are used to reduce the upload and download times of data between a management station and the ASC or between an ASC and another device. Several pre-defined blocks are defined by this standard.

Each block contains configuration parameters for a functional area associated with the operation and management of an ASC. These blocks allow a transportation manager to select what sets of configuration data need to be exchanged with the ASC in a bandwidth efficient manner, either because the data is exchanged often, or the set otherwise requires a large amount of bandwidth.

For example, NTCIP 1202 may define a single block that is exchanged to fulfill all the Configure Phases Requirements found in Section 6, in addition to the individual design content defined by NTCIP 1202 to fulfill each requirement in the same section. It is assumed that the single block uses less bandwidth than the bandwidth required to fulfill each requirement in the same section separately.

3.4.3 Support Logged Data Requirements

The requirements to download logged data from an ASC follow. The logged data may be used for different purposes and by different groups, so to provide the logged data in an organized manner, the logged data may be stored in different event logs (classes) managed by different owners. Each event type defines the condition(s) when an event is logged. The requirements for logging event data with the ASC are found in Sections 8.9 and 8.10 of ISO 26048-1 – the specific requirements can be found under User Need 2.4.3, Provide for Log Data Local Storage and Retrieval in Table 5. This log is not intended for the high-resolution performance data defined in the "Indiana Traffic Signal Hi Resolution Data Logger Enumerations."

3.4.4 Support Database Management Requirements

The requirements to manage the ASC database follow. The requirements to manage the ASC database are found in Section 8.21 of ISO 26048-1 – the specific requirements can be found under User Need 2.4.4, Provide for Database Management in Table 5.

3.4.5 Support Condition-based Exception Reporting Requirements

An ASC system may wish to transmit data to another system when specific events occur. The requirements to configure notifications and acknowledgements in an ASC and to notify other systems are found in Sections 8.1, 8.11 and 8.16 of ISO 26048-1 – the specific requirements can be found under User Need 2.4.5, Condition-based Exception Reporting in Table 5.

3.5 Data Exchange and Operational Environment Requirements

The data exchange requirements for an ASC follows.

To reduce the number of requirements stated, it is assumed that if a requirement to configure or store a value exists, an implicit requirement exists that when a management station requests confirmation of a value, the ASC will return that value.

3.5.1 ASC Configuration Management Requirements

The requirements for managing an ASC configuration follow.

3.5.1.1 Manage ASC Information Requirements

The requirements to manage information about the ASC follow. The requirements to manage the physical location and other identification details of an ASC are found in Sections 8.5, 8.8, and 8.12 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.1.1, Manage Device Identity in Table 5.

3.5.1.1.1 Configure ASC Location - Antenna Offset

If an external GNSS or similar geopositioning device is attached to the ASC, upon request from the management station, the ASC shall store the offset in elevation, in meters, between the antenna used by a GNSS or similar geopositioning device and the base of the structure for a permanent ASC. A geographic position provided by a GNSS (or similar) device is usually based on the location of the antenna. Generally, the longitude and latitude of the antenna is the same location as the ASC, but the height of the antenna will normally be higher than the base of the structure. This requirement corrects the GNSS reading, which includes the elevation of the antenna, for the base of the structure.

3.5.1.2 Manage Communications Requirements

An ASC typically has several communications ports for exchanging information with the cabinet subsystems and other external devices in the cabinet. There are different potential ports configurations that might be used including Ethernet, RS-232, and dial-up, as well as ports to connect to wireless infrastructures such as Wi-Fi, cellular, or others. Additionally, an ASC is likely to have at least two ports for external communications (one for communications with a central management system and one to connect to a local port to be used by authorized persons interacting with the ASC directly), but additional ports might be present, particularly for interconnecting neighboring controllers and/or when connecting to a roadside unit.

3.5.1.2.1 Configure Communications Requirements

The requirements for configuring the communications ports in the ASC follow.

3.5.1.2.1.1 Enable/Disable Communications Port

Upon request from a management station, an ASC shall enable or disable a communications port on the ASC.

3.5.1.2.2 Retrieve Communications Requirements

The requirements for retrieving information about the communications ports in the ASC follow.

3.5.1.2.2.1 Determine Number of ASC Communications Ports

Upon request from a management station, the ASC shall return the number of communications ports supported by the ASC.

3.5.1.3 Manage Cabinet Environment Requirements

An ASC system includes a cabinet within which the controller subsystems reside including the controller unit. Each cabinet has at least one door, and many cabinets have fans that turn on when temperature thresholds have been reached. The requirements to manage the cabinet environment of the ASC are found in Section 8.8 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.1.3, Manage Cabinet Environment in Table 5.

3.5.1.4 Monitor Power Source Requirements

The requirements to monitor the status of the power-provision-related equipment associated with the ASC cabinet are found in Section 8.8 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.1.4, Monitor Power in Table 5. The requirements address the source of power for the ASC and the cabinet that houses the ASC, not the cabinet power supplies that may be in the cabinet.

3.5.1.5 Manage Operational Performance Data Requirements

Operational performance data consists of frequent snapshots of signal operations data and detector data measured by the ASC and allows the management station to view the temporal relationship between signal indications and traveler arrivals. The requirements to manage the collection and retrieval of high-resolution operational data are found in Section 8.13 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.1.5, Retrieve Operational Performance Data in Table 5.

3.5.1.6 Manage Auxiliary External Inputs/Outputs Requirements

The requirements to support the monitoring and control of auxiliary external devices or functions, also called supplemental roadside sensors and actuator (SRSA) ports, to satisfy other transportation operational needs follow. SRSA ports may interface with sensors or actuators such as float switches that indicate if there is flooding in and around the ASC cabinet. The requirements to monitor and control SRSA ports are found in Section 8.18 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.1.6, Manage Auxiliary External Inputs/Outputs in Table 5. ISO 26048-1 defines separate requirements for cabinet door sensors, temperature sensors, humidity sensors, fans, and heater, so these requirements do not consider those sensors or devices.

3.5.1.7 Manage Database Operations

The requirements to manage an ASC database, which is required for consistency checks of certain objects, follow. The requirement to identify any change to the field device's configuration is found in Section 8.5 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.1.7, Manage Database in Table 5.

3.5.1.7.1 Determine Configuration Identifier Parameter Content

Upon request from a management station, the ASC shall return the configuration parameters being used to create configuration parameter identifier by listing all configuration parameters based on their SNMP Object Identifiers (Object OIDs) including scalar and instance indicators and starting with configuration parameter OIDs defined in the NTCIP standards and followed by any manufacturer-specific configuration parameter OIDs. See Section 8.5.2.14 of ISO 26048-1.

3.5.1.8 Manage Interface with External Detectors Requirements

See 2.5.3.5 for detector data collected by the ASC through external inputs. Requirements to view advanced detector data are found in NTCIP 1209.

3.5.1.9 Manage ASC Clock Requirements

The requirements to manage the ASC clock are found in Section 8.2 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.1.9, Manage ASC Clock in Table 5.

3.5.1.10 Manage External Control Local Application State Requirements

In some scenarios, the signal timing is determined by an external device called the ECLA (See Section 2.3.4.c), which sends commands to the ASC to advance the phase at specific times determined by the ECLA. The requirements that allow a management station to configure retrieving signal timing durations and movement states from an ECLA follow.

3.5.1.10.1 Manage ECLA Interface Requirements

The requirements to manage the communications interface between an ASC and an ECLA follow.

3.5.1.10.1.1 Enable ECLA Communications

Upon request from a management station, the ASC shall enable exchanging information with an ECLA.

3.5.1.10.1.2 Disable ECLA Communications

Upon request from a management station, the ASC shall disable exchanging information

3.5.1.10.2 Monitor ECLA Data Input Time

Upon request from a management station, the ASC shall return the most recent time that an ECLA provided data to the ASC. This may be used to verify that the ECLA is operating normally.

3.5.2 Manage Signal Operations Management Requirements

The requirements for managing the signal operations of an ASC follow.

3.5.2.1 Manage Signal Configuration Requirements

The requirements to manage the traffic signal configurations are defined in the following paragraphs.

3.5.2.1.1 Manage Unit Configuration Requirements

The requirements to manage the unit configurations of the ASC follow.

3.5.2.1.1.1 Manage Start-Up Flash Requirements

The requirements to manage the ASC Start-Up Flash state follow.

Note: The Fault Monitor and Voltage Monitor outputs shall be inactive during the Start-Up Flash state (See NEMA TS-2).

3.5.2.1.1.1.1 Configure Start-Up Flash Mode

Upon request from a management station, the ASC shall store the Start-Up Flash mode for the controller after restoration of a defined power interruption or activation of the external start input. Possible Start-Up Flash modes are:

a) Automatic Flash. Start-Up Flash is based on the ASC channel settings.
b) Cabinet Flash. Start-Up Flash is based on the cabinet configuration.
c) All Red Controller Flash. Start-Up Flash is all red regardless of the ASC channel settings.
d) Other. A state not defined in this standard.

Note: MUTCD states that "Changes from (all-red) flashing mode to steady (stop-and-go) mode shall be made by changing the flashing red indications to steady red indications followed by appropriate green indications to begin the steady mode cycle" (see FHWA MUTCD 2023 Edition, Section 4G.04.A); and "The steady red clearance interval provided during the change from red-red flashing mode to steady (stop-and-go) mode should have a duration of 6 seconds." (see FHWA MUTCD 2023 Edition, Section 4G.04 02).

3.5.2.1.1.1.2 Configure Start-Up Flash Time

Upon request from a management station, the ASC shall store the period of time, in seconds, the ASC remains in the Start-Up flash state after the power is restored following a power interruption. The period of time the ASC is allowed to be in the start-up state is 0 to 255 seconds.

3.5.2.1.1.2 Configure Backup Time

Upon request from a management station, the ASC shall store the backup time, in seconds, as provided in the request. The backup time is a value from 0 to 16777216 seconds, with a value of 0 indicating this feature is disabled.

The backup time defines the period of time to be exceeded when no SET operation to any of the system control parameters as defined in Section 5.4.3, Backup Time Parameter, after which the ASC reverts to

Backup Mode and clears and system control parameters. This allows the ASC to revert to its previous settings in the event that communication from the management station is lost.

3.5.2.1.2 Manage Phase Configuration Requirements

The requirements to manage the phase configurations of the ASC follow.

Unless otherwise noted, such as a pedestrian phase, a phase may be assigned to a vehicular, transit vehicle, or bicycle movement.

3.5.2.1.2.1 Configure Phase Requirements

To manage a phase-based controller, the requirements to allow a management system to configure each defined phase follow.

3.5.2.1.2.1.1 Enable/Disable Phase

Upon request from a management station, the ASC shall store if a phase is enabled or disabled for the current configuration. A disabled phase does not provide any outputs nor respond to any phase inputs.

3.5.2.1.2.1.2 Configure Phase Minimum Green Time

Upon request from a management station, the ASC shall store the minimum amount of time, from 0 and 255 seconds, that the Green indication is to be displayed for a phase.

3.5.2.1.2.1.3 Configure Phase Passage Time

Upon request from a management station, the ASC shall store the extensible time of the Green indication for a phase in tenths of a second, between 0 and 25.5 seconds. The extensible time of the Green indication is the amount of time that the Green indication is extended after a vehicle actuation. The Green indication is extended until the passage timer is timed out.

3.5.2.1.2.1.4 Configure Two Fixed Phase Maximum Green Times

Upon request from a management station, the ASC shall store a default and one additional user-defined value for the maximum amount of time, from 0 and 999 seconds, for which the vehicle phase shows a Green indication. In the absence of a serviceable conflicting call, the ASC holds the Maximum Green timer in rest, which may be overridden by external input, coordMaximumMode, or another method defined in NTCIP 1202.

3.5.2.1.2.1.5 Configure Three Fixed Phase Maximum Green Times

Upon request from a management station, the ASC shall store a default and two additional user-defined values for the maximum amount of time, from 0 and 999 seconds, for which the vehicle phase shows a Green indication. In the absence of a serviceable conflicting call, the ASC holds the Maximum Green timer in rest, which may be overridden by external input, coordMaximumMode, or another method defined in NTCIP 1202.

3.5.2.1.2.1.6 Configure Phase Yellow Change Time

Upon request from a management station, the ASC shall store the amount of time the Yellow Change indication is to be displayed for a phase in tenths of a second from 0 to 25.5 seconds.

3.5.2.1.2.1.7 Configure Phase Red Clearance Time

Upon request from a management station, the ASC shall store the Red Clearance interval for a phase in tenths of a second, from 0 to 25.5 seconds.

3.5.2.1.2.1.8 Configure Phase Red Revert Time

Upon request from a management station, the ASC shall store the minimum amount of time a Red indication is to be displayed following a yellow change interval, prior to the next Green Interval for the same phase as provided in the request. The minimum red indication for this phase is in tenths of a second, from 0 to 25.5 seconds.

3.5.2.1.2.1.9 Configure Unit Red Revert Time

Upon request from a management station, the ASC shall store the minimum amount of time a Red indication is to be displayed following a yellow change interval, prior to the next Green Interval for the same phase as provided in the request. The minimum red indication for all phases defined in the ASC is in tenths of a second, from 0.0 to 25.5 seconds.

3.5.2.1.2.1.10 Configure Phase Added Initial Time

Upon request from a management station, the ASC shall store the amount of time for a phase, in tenths of a second, by which the ASC is to increase the variable green time period (initial time period) based on the vehicle actuations detected during the associated phase's yellow and red indications. The possible amount of added initial time is between 0 to 25.5 seconds. The value is used in conjunction with the Volume Density operation that might be used within the ASC.

3.5.2.1.2.1.11 Configure Phase Maximum Initial Time

Upon request from a management station, the ASC shall store the maximum amount of time, from 0 to 255 seconds, that the variable green time period (initial time period) of a phase can be increased, which cannot be less than the minimum green time of the phase.

3.5.2.1.2.1.12 Configure Phase Time Before Reduction

Upon request from a management station, the ASC shall store the Time Before Reduction period for a phase from 0 to 255 seconds. The Time Before Reduction (TBR) period begins when the phase is Green, and there is a serviceable conflicting call. The linear reduction of the allowable gap from the Passage Time begins when the TBR period or the Cars Before Reduction (CBR) is satisfied, whatever occurs first. If the serviceable conflicting call is removed while timing the TBR period, the associated internal ASC timer is reset.

3.5.2.1.2.1.13 Configure Phase Time to Reduce

Upon request from a management station, the ASC shall store the time to reduce for a phase from 0 to 255 seconds. The time to reduce is used to control the linear rate of reduction between the Passage Time and the minimum gap, as defined by NEMA TS 1 and NEMA TS 2.

3.5.2.1.2.1.14 Configure Phase Cars Before Reduction

Upon request from a management station, the ASC shall store the Cars Before Reduction (CBR) parameter for a phase in number of vehicles, from 0 to 255. The CBR begins counting when the phase is Green, and there is a serviceable conflicting call. The linear reduction of the allowable gap from the Passage Time begins when the CBR or Time Before Reduction (TBR) period is satisfied, whatever occurs first.

3.5.2.1.2.1.15 Configure Phase Reduce By Time

Upon request from a management station, the ASC shall store a parameter to control the rate of reduction for a phase in tenths of a second, from 0 to 25.5 seconds. This parameter allows the use of an alternate time to reduce algorithm other than the linear reduction defined by NEMA TS 1 and NEMA TS 2. The Phase Time to Reduce remains the same.

3.5.2.1.2.1.16 Configure Phase Minimum Gap Time

Upon request from a management station, the ASC shall store the minimum amount of time in tenths of seconds, from 0 to 25.5 seconds, to which the gap between vehicles can be reduced with the purpose that the phase can be terminated, if the detected gap between subsequent detector actuations is greater than this value.

3.5.2.1.2.1.17 Configure Phase Dynamic Maximum Limit

Upon request from a management station, the ASC shall store the upper limit or lower limit of the maximum allowable time of the Green indication for a phase, from 0 to 255 seconds. If the Dynamic Maximum Limit is larger than the normal maximum time of the Green Indication, it becomes the upper limit. If the Dynamic Maximum Limit is lower than the normal maximum time of the Green indication, it becomes the lower limit. The ASC disables the use of this function if the maximum recall time or a failed detector associated with the phase is active.

3.5.2.1.2.1.18 Configure Phase Dynamic Maximum Step

Upon request from a management station, the ASC shall store the step value for increasing or decreasing the allowable maximum time of the Green indication in tenths of a second, from 0 to 25.5 seconds. How the ASC decides when to adjust allowable time is not defined in this standard.

3.5.2.1.2.1.19 Configure Phase Start-Up State

Upon request from a management station, the ASC shall store the state for a phase after the termination of the Start-Up flash state (See Section 3.5.2.1.1.1.1). The valid phase states define by this standard are:

a) phaseNotOn. The phase is not active and no intervals are timing.
b) greenWalk. The phase starts at the beginning of the green and walk timing intervals.
c) greenNoWalk. The phase starts at the beginning of the green interval with no walks.
d) yellowChange. The phase starts at the beginning of the yellow change interval.
e) redClear. The phase starts at the beginning of the red clearance interval.
f) other. The phase is not enabled or starts in a state not defined by this standard.

Note: MUTCD states that "Changes from (all-red) flashing mode to steady (stop-and-go) mode shall be made by changing the flashing red indications to steady red indications followed by appropriate green indications to begin the steady mode cycle" (see FHWA MUTCD 2023 Edition, Section 4G.04.A); and "The steady red clearance interval provided during the change from red-red flashing mode to steady (stop-and-go) mode should have a duration of 6 seconds." (see FHWA MUTCD 2023 Edition, Section 4G.04 02). Thus, if the Start-Up flash mode is all-red, then the phase Start-Up state must be redClear.

3.5.2.1.2.1.20 Configure Automatic Flash Entry Phase

Upon request from a management station, the ASC shall store which phases are serviced before initiating Automatic Flash when Automatic Flash is called.

3.5.2.1.2.1.21 Configure Automatic Flash Exit Phase

Upon request from a management station, the ASC shall store which phases are serviced when Automatic Flash terminates.

3.5.2.1.2.1.22 Configure Call to Non-Actuated 1

Upon request from a management station, the ASC shall store which phases respond if the Call to Non-Actuated 1 input is active.

3.5.2.1.2.1.23 Configure Call to Non-Actuated 2

Upon request from a management station, the ASC shall store which phases respond if the Call to Non-Actuated 2 input is active.

3.5.2.1.2.1.24 Configure Non-Lock Detector Memory

Upon request from a management station, the ASC shall store whether a call present at the beginning of a phase's yellow time is locked.

3.5.2.1.2.1.25 Configure Phase Minimum Vehicle Recall

Upon request from a management station, the ASC shall store if a recurring call for vehicle service exists for a phase to be served for at least the phase’s minimum Green time.

3.5.2.1.2.1.26 Configure Phase Maximum Vehicle Recall

Upon request from a management station, the ASC shall store if a recurring call for service exists for a phase to be served for the maximum time that may be allocated to the phase.

3.5.2.1.2.1.27 Configure Phase Soft Vehicle Recall

Upon request from a management station, the ASC shall store if a call is to be placed on a phase when all conflicting phases are in resting in green or red, and there are no serviceable conflicting calls.

3.5.2.1.2.1.28 Configure Dual Phase Entry

Upon request from a management station, the ASC shall store if a phase is to become active upon entry into a concurrency group, when no calls exist in its ring within its concurrency group. This is valid for multi-ring configurations only.

3.5.2.1.2.1.29 Configure Simultaneous Gap Disable

Upon request from a management station, the ASC shall store if a gapped-out phase is allowed to revert to the extensible portion of the phase. This is valid for multi-ring configurations only.

3.5.2.1.2.1.30 Configure Guaranteed Passage

Upon request from a management station, the ASC shall store if the phase operates in volume density mode. The volume density mode uses gap reduction to retain the right of way for the unexpired portion of the Passage time following the decision to terminate the green due to a reduced gap.

3.5.2.1.2.1.31 Configure Actuated Rest-in-Walk

Upon request from a management station, the ASC shall store if an actuated phase rests in Walk if there is no serviceable conflicting call at the end of the Walk time. This setting is also used to enable Rest-in-Walk if Maximum Vehicle Recall is enabled for this phase.

3.5.2.1.2.1.32 Configure Conditional Service Enable

Upon request from a management station, the ASC shall store if conditional service, as defined in NEMA TS 2 Section 3.5.3.9, is allowed. Conditional service provides an optional method for phase selection in multi-ring configurations.

3.5.2.1.2.1.33 Configure Added Initial Calculation

Upon request from a management station, the ASC shall store what detector values to use for the calculation of the variable portion of the green time (added initial time):

a) the largest count value from all associated detectors; or
b) the sum from all associated detectors.

3.5.2.1.2.1.34 Configure Phase-to-Ring Association

Upon request from a management station, the ASC shall store the ring (number), with which the phase is associated with or if the phase is disabled.

3.5.2.1.2.1.35 Configure Phase Concurrency

Upon request from a management station, the ASC shall store the phase numbers allowed to run concurrently with the phase. Phases within the same ring cannot run concurrently.

3.5.2.1.2.1.36 Configure Pedestrian Clearance Time Allowed During Vehicle Clearance

Upon request from a management station, an ASC shall store the amount of time in tenths of a second, from 0 to 25.5 seconds, the pedestrian clearance may extend into the vehicle clearance time (yellow change and red clearance) for a phase.

3.5.2.1.2.1.37 Configure Pedestrian Walk Time

Upon request from a management station, the ASC shall store the amount of time the pedestrian WALK indication is to be displayed for a pedestrian phase, between 0 and 255 seconds.

Note: MUTCD states that the WALK indication should be at least 4 seconds with a normal minimum duration of 7 seconds (see FHWA MUTCD 2023 Edition, Figure 4I-4 and Section 4I-06 12).

3.5.2.1.2.1.38 Configure Pedestrian Clearance Time

Upon request from a management station, the ASC shall store the amount of time the first pedestrian clearance indication is to be displayed for a pedestrian phase in seconds, between 0 and 255 seconds. The first pedestrian clearance indication is the interval following a pedestrian WALK indication and is normally a Flashing Don’t Walk.

3.5.2.1.2.1.39 Configure Pedestrian Phase Walk Service Limit

Upon request from a management station, an ASC shall store if the pedestrian WALK indication is allowed to be shown again within the same pedestrian phase (after the initial pedestrian Walk, Flashing Don't Walk, and minimum Don't Walk time).

3.5.2.1.2.1.40 Configure Pedestrian Phase Don't Walk Revert Time

Upon request from a management station, the ASC shall store the minimum amount of time a pedestrian Don't Walk indication is to be displayed following a Flashing Don't Walk time before it may indicate a WALK indication again. The minimum pedestrian Don't Walk time indication for this pedestrian phase is in tenths of a second, from 0 to 25.5 seconds.

3.5.2.1.2.1.41 Configure Non-Lock Ped Detector Memory

Upon request from a management station, the ASC shall store if a pedestrian call present at the beginning of the phase's pedestrian clearance interval (Flashing Don’t Walk) is locked.

3.5.2.1.2.1.42 Configure Pedestrian Phase Recall

Upon request from a management station, the ASC shall store if a recurring call for pedestrian service exists for a pedestrian phase when that phase is not in its Walk interval. The ASC does not recycle the pedestrian service until a conflicting phase is serviced.

3.5.2.1.2.1.43 Configure Phase Alternate Pedestrian Clearance Time

Upon request from a management station, the ASC shall store an alternate pedestrian clearance time for a pedestrian phase, from 0 to 255 seconds. This alternate time may be used to support an extended pedestrian clearance time.

3.5.2.1.2.1.44 Configure Phase Alternate Pedestrian Walk Time

Upon request from a management station, the ASC shall store the amount of time for a pedestrian phase, in seconds from 0 to 255 seconds. This alternate time may be used to support an extended Walk time period based on a pedestrian detector input such as an extended push button press.

3.5.2.1.2.1.45 Configure Pedestrian Phase Advanced Walk Time

Upon request from a management station, an ASC shall store the amount of time, in tenths of a second, that a parallel pedestrian WALK indication starts before the start of the of the vehicle Green indication for the phase. This is known as a leading pedestrian interval. If configured, this may occur in response to a pedestrian detector input or if Pedestrian Recall is enabled.

3.5.2.1.2.1.46 Configure Pedestrian Phase Delayed Walk Time

Upon request from a management station, an ASC shall store the amount of time, in tenths of a second, that a parallel pedestrian WALK indication starts after the start of the vehicle Green indication for the phase. If configured, this may occur in response to a pedestrian detector input such as an extended push button press.

3.5.2.1.2.1.47 Configure Phase Advance Warning Green

Upon request from a management station, an ASC shall store the amount of time in tenths of a second, from 0.0 to 25.5 seconds, that an Advance Warning Green indication is displayed before the start of the phase Green. The warning signal indication, which may be a separate signal indication or device, is placed upstream of the phase's approach and indicates that the phase Green indication is about to start or has started.

3.5.2.1.2.1.48 Configure Phase Advance Warning Red

Upon request from a management station, an ASC shall store the amount of time in tenths of a second, from 0.0 to 25.5 seconds, that an Advance Warning Red indication is to be displayed before the start of the phase Red. The warning signal, which may be a separate signal indication or device, is placed upstream of the phase's approach and indicates that the phase Red is about to start or has started.

3.5.2.1.2.1.49 Configure Flashing Yellow Arrow Associated Vehicle Phase

Upon request from a management station, the ASC shall store the associated phase for which the Flashing Yellow Arrow indication is displayed.

3.5.2.1.2.1.50 Configure Flashing Red Arrow Associated Vehicle Phase

Upon request from a management station, the ASC shall store the associated phase for which the Flashing Red Arrow indication is displayed.

3.5.2.1.2.1.51 Configure Alternate Minimum Green Time during Transition

Upon request from a management station, the ASC shall store the alternate minimum Green time for a phase, from 1 to 255 seconds, that is to be used if the correction mode has been set to Shortway or Subtract Only. The alternate minimum Green time cannot be greater than the minimum Green time for this phase - see 3.5.2.1.2.1.2 Configure Phase Minimum Green Time.

3.5.2.1.2.1.52 Configure Alternate Minimum Walk Time during Transition

Upon request from a management station, the ASC shall store the alternate minimum Walk time, from 1 to 255 seconds, that is to be used for a pedestrian phase if the correction mode has been set to the Shortway or Subtract Only. The alternate minimum Walk time cannot be greater than the Walk time for this phase - see 3.5.2.1.2.1.37 Configure Pedestrian Walk Time.

3.5.2.1.2.1.53 Configure Alternate Minimum Pedestrian Clearance Time during Transition

Upon request from a management station, the ASC shall store the alternate minimum pedestrian clearance time, from 1 to 255 seconds, that is to be used for a pedestrian phase if the correction mode has been set to the Shortway or Subtract Only. The alternate minimum Walk time cannot be greater than the Walk time for this phase - see 3.5.2.1.2.1.38 Configure Pedestrian Clearance Time.

3.5.2.1.2.2 Configure Multiple Phase Sets

Upon request from a management station, the ASC shall store phase parameters for multiple sets of phases in a manner consistent with the following requirements as selected under User Need 2.5.2.1.2, Manage Phase Configurations in Table 5 Protocol Requirements List (PRL).

a) 3.5.2.1.2.1.2 Configure Phase Minimum Green Time
b) 3.5.2.1.2.1.3 Configure Phase Passage Time
c) 3.5.2.1.2.1.4 Configure Two Fixed Phase Maximum Green Times
d) 3.5.2.1.2.1.5 Configure Three Fixed Phase Maximum Green Times
e) 3.5.2.1.2.1.6 Configure Phase Yellow Change Time
f) 3.5.2.1.2.1.7 Configure Phase Red Clearance Time
g) 3.5.2.1.2.1.8 Configure Phase Red Revert Time
h) 3.5.2.1.2.1.10 Configure Phase Added Initial Time
i) 3.5.2.1.2.1.11 Configure Phase Maximum Initial Time
j) 3.5.2.1.2.1.12 Configure Phase Time Before Reduction
k) 3.5.2.1.2.1.13 Configure Phase Time to Reduce
l) 3.5.2.1.2.1.14 Configure Phase Cars Before Reduction

m) 3.5.2.1.2.1.15 Configure Phase Reduce By Time
n) 3.5.2.1.2.1.16 Configure Phase Minimum Gap Time
o) 3.5.2.1.2.1.17 Configure Phase Dynamic Maximum Limit
p) 3.5.2.1.2.1.18 Configure Phase Dynamic Maximum Step
q) 3.5.2.1.2.1.22 Configure Call to Non-Actuated 1
r) 3.5.2.1.2.1.23 Configure Call to Non-Actuated 2
s) 3.5.2.1.2.1.24 Configure Non-Lock Detector Memory
t) 3.5.2.1.2.1.25 Configure Phase Minimum Vehicle Recall
u) 3.5.2.1.2.1.26 Configure Phase Maximum Vehicle Recall
v) 3.5.2.1.2.1.27 Configure Phase Soft Vehicle Recall
w) 3.5.2.1.2.1.28 Configure Dual Phase Entry
x) 3.5.2.1.2.1.29 Configure Simultaneous Gap Disable
y) 3.5.2.1.2.1.31 Configure Actuated Rest-in-Walk
z) 3.5.2.1.2.1.32 Configure Conditional Service Enable
aa) 3.5.2.1.2.1.33 Configure Added Initial Calculation
bb) 3.5.2.1.2.1.36 Configure Phase-to-Ring Association
cc) 3.5.2.1.2.1.37 Configure Pedestrian Walk Time
dd) 3.5.2.1.2.1.38 Configure Pedestrian Clearance Time
ee) 3.5.2.1.2.1.39 Configure Pedestrian Phase Walk Service Limit
ff) 3.5.2.1.2.1.40 Configure Pedestrian Phase Don't Walk Revert Time
gg) 3.5.2.1.2.1.41 Configure Non-Lock Ped Detector Memory
hh) 3.5.2.1.2.1.42 Configure Pedestrian Phase Recall
ii) 3.5.2.1.2.1.43 Configure Phase Alternate Pedestrian Clearance Time
jj) 3.5.2.1.2.1.44 Configure Phase Alternate Pedestrian Walk Time
kk) 3.5.2.1.2.1.45 Configure Pedestrian Phase Advanced Walk Time
ll) 3.5.2.1.2.1.46 Configure Pedestrian Phase Delayed Walk Time
mm) 3.5.2.1.2.1.47 Configure Phase Advance Warning Green
nn) 3.5.2.1.2.1.48 Configure Phase Advance Warning Red
oo) 3.5.2.1.2.1.51 Configure Alternate Minimum Green Time during Transition
pp) 3.5.2.1.2.1.52 Configure Alternate Minimum Walk Time during Transition
qq) 3.5.2.1.2.1.53 Configure Alternate Minimum Pedestrian Clearance Time during Transition

3.5.2.1.2.3 Retrieve Phase Configuration Requirements

The requirements to return the configuration parameters associated with each defined phase follow.

3.5.2.1.2.3.1 Determine Maximum Number of Phases

Upon request from a management station, the ASC shall return the maximum number of phases in a single phase set as a number from 1 to 255 phases that can be configured within the ASC.

3.5.2.1.2.3.2 Determine Maximum Number of Phase Sets

Upon request from a management station, the ASC shall return the maximum number of phase sets as a number from 1 to 255 phase sets that can be configured within the ASC.

3.5.2.1.3 Manage Coordination Configuration Requirements

The requirements to configure the traffic signal coordination parameters of the ASC follow.

3.5.2.1.3.1 Configure Operational Mode for Coordination

Upon request from a management station, the ASC shall set the operational mode for coordination. Valid values for operational modes are:

a) Automatic. The ASC runs coordinated operation, free and flash as determined automatically by the possible sources (i.e., system command, timebase schedule, or interconnect inputs).
b) Manual Pattern. The ASC coordinated operation using a manually selected pattern.
c) Manual Free. The ASC runs in Free Mode.
d) Manual Flash. The ASC flashes based on channel settings as it would in Automatic Flash.

The operational mode for coordination remains in effect until it is set again by the management station.

3.5.2.1.3.2 Configure Correction Mode for Coordination

Upon request from a management station, the ASC shall set the coordination correction mode, which changes the offset for the coordination algorithm. Coordination corrections may be done using vendor-specific methods. Valid correction modes are:

a) Dwell. The ASC dwells in the coordinated phase until the new offset is reached.
b) Shortway. The ASC adds to or subtracts from the timing in a manner that limits the cycle change.
c) Add Only. The ASC adds to the timing in a manner that limits the cycle change.
d) Subtract Only. The ASC subtracts from the timing in a manner that limits the cycle change.

3.5.2.1.3.3 Configure Maximum Mode for Coordination

Upon request from a management station, the ASC shall store the default (unit) maximum mode for coordination. The valid maximum modes are:

a) maxInhibit. The maximum timing settings are not used while coordination is running a pattern.
b) maximum1. The Maximum 1 timing is used while coordination is running a pattern.
c) maximum2. The Maximum 2 timing is used while coordination is running a pattern.
d) maximum3. The Maximum 3 timing is used while coordination is running a pattern.

A user may define a Maximum Mode for Coordination for each pattern separately (See 3.5.2.1.4.1.7 Configure Pattern Maximum Mode). If a maximum mode for coordination is defined for a pattern, the pattern maximum mode is used.

3.5.2.1.3.4 Configure Unit-level Force Mode for Coordination

Upon request from a management station, the ASC shall set the default (unit) coordination process force mode. This setting of the unit-level mode is overridden if a force mode for coordination is defined for a phase (See 3.5.2.1.3.5 Configure Phase-level Force Mode for Coordination). Valid force modes are:

a) Fixed. Each non-coordinated phase is limited to its split time value, allowing unused split time to be reallocated to the coordinated phase.
b) Floating. Each non-coordinated phase is forced off at a fixed position in the cycle, allowing unused split time to be reallocated to the next phase.

The force mode does not apply to phases operating with Maximum Vehicle Recall, a non-Actuated split mode, or a Call to Non-Actuated input.

3.5.2.1.3.5 Configure Phase-level Force Mode for Coordination

Upon request from a management station, the ASC shall set the coordination process force mode for each non coordinated phase in a pattern. Valid force modes are:

a) unitCoordForceMode. The phase uses the default (unit) coordination process force mode
b) Fixed. The phase is limited to its split time value, allowing unused split time phase to be reallocated to the next phase that is either coordinated or fixed.

c) Floating. The phase is forced off at a fixed position in the cycle. This allows unused split time from the phase to be reallocated to the next phase that is either coordinated or fixed. Additionally, this allows unused split time to be reallocated to the phase from a previous non-coordinated phase.

The force mode does not apply to phases operating with Maximum Vehicle Recall, a non-Actuated split mode, or a Call to Non-Actuated input.

3.5.2.1.3.6 Configure Pattern Reference Phase

Upon request from a management station, an ASC shall store the phase number for a timing pattern that contains the reference point where the timing pattern cycle begins and ends. The reference phase must be a coordinated phase. The reference phase is not used if the ASC is running Free.

3.5.2.1.3.7 Configure Pattern Reference Point

Upon request from a management station, an ASC shall store the specific reference point for a timing pattern. The reference point is start or end of an interval in the reference phase defined in 3.5.2.1.3.6 Configure Pattern Reference Phase. Valid reference points are:

a) greenBegin. The start of the green interval. If the phase has a leading pedestrian interval, then this is the start of the leading pedestrian interval.
b) yellowBegin. The start of the yellow change interval.
c) redBegin. Thes start of the red clearance interval.
d) redEnd. The end of the red clearance interval.

The reference point is not used if the ASC is running Free.

3.5.2.1.3.8 Configure Omit Phases During Transitions

Upon request from a management station, an ASC shall store if a phase can be omitted from currently the selected pattern when the controller is in transition.

3.5.2.1.3.9 Configure Pattern Synchronization Time

Upon request from a management station, the ASC shall store the pattern synchronization reference time, in minutes past midnight, from 0 to 65534 minutes. This is the time that the master cycle begins and the local cycle offset is relative to (See 3.5.2.1.4.1.4 Configure Pattern Offset Time).

If this value is 65535, the start or activation time of the timebase pattern is used as the synchronization reference by the ASC.

3.5.2.1.4 Manage Timing Patterns Requirements

The requirements to manage the traffic signal timing pattern parameters of the ASC follow.

3.5.2.1.4.1 Configure Timing Patterns Requirements

The requirements to configure the traffic signal timing patterns stored within an ASC follow.

3.5.2.1.4.1.1 Configure Pattern to Run Free

Upon request from a management station, the ASC shall store that the ASC is to run in Free mode when this pattern is called.

3.5.2.1.4.1.2 Configure Pattern for Automatic Flash

Upon request from a management station, the ASC shall store that the ASC is to go into Automatic Flash when this pattern is called.

3.5.2.1.4.1.3 Configure Pattern Cycle Time for Coordination

Upon request from a management station, the ASC shall store the length of the pattern cycle as long as 999 seconds. A cycle length of 0 seconds is used for timing patterns where the ASC runs in Free mode. Otherwise, the pattern is configured for coordination.

If the pattern cycle time is of insufficient length to service the minimum timing parameters (Minimum Green, Walk, Pedestrian Clearance, Yellow Clearance, Minimum Red, etc.) of all phases, the ASC automatically implements Free Mode and indicates this in the ASC's alarm and pattern status objects.

Note: The NEMA TS-2 Range is 30 – 255 seconds.

3.5.2.1.4.1.4 Configure Pattern Offset Time

Upon request from a management station, the ASC shall store the time in seconds, from 0 to 998 seconds that the local time zero lags the system time zero for this pattern. This is the time difference between the local cycle and the master cycle.

3.5.2.1.4.1.5 Configure Pattern Split Association

Upon request from a management station, the ASC shall store the split associated with a traffic signal timing pattern.

3.5.2.1.4.1.6 Configure Pattern Sequence Association

Upon request from a management station, the ASC shall store the sequence associated with a traffic signal timing plan. If there is no sequence specified, the ASC goes into Automatic Flash.

3.5.2.1.4.1.7 Configure Pattern Maximum Mode

Upon request from a management station, the ASC shall store the maximum mode for a traffic signal timing pattern. The valid maximum modes are:

a) coordMaximumMode. The default maximum mode defined by the ASC is used for this pattern.
b) maxInhibit. The maximum timing settings are not used for pattern.
c) maximum1. The Maximum 1 timing is used for this pattern.
d) maximum2. The Maximum 2 timing is used for this pattern.
e) maximum3. The Maximum 3 timing is used for this pattern.

3.5.2.1.4.1.8 Configure Pattern Phase Set

Upon request from a management station, the ASC shall store the phase set to be used for a traffic signal timing pattern.

3.5.2.1.4.1.9 Configure Pattern Overlap Set

Upon request from a management station, the ASC shall store the overlap set to be used for a traffic signal timing pattern.

3.5.2.1.4.1.10 Configure Pattern Vehicle Detector Set

Upon request from a management station, the ASC shall store the vehicle detector set to be used for a traffic signal timing pattern.

3.5.2.1.4.1.11 Configure Pattern Pedestrian Detector Set

Upon request from a management station, the ASC shall store the pedestrian detector set to be used for a traffic signal timing pattern.

3.5.2.1.4.1.12 Configure Pattern Special Functions

Upon request from a management station, the ASC shall configure up to 32 special functions to be activated when a traffic signal timing pattern is active.

3.5.2.1.4.1.13 Determine Maximum Number of Timing Patterns

Upon request from a management station, the ASC shall return the maximum number of traffic signal plans / patterns that can be configured in the ASC.

3.5.2.1.5 Manage Splits Configuration Requirements

The requirements to manage the phase splits within traffic signal timing plans / patterns parameters of the ASC follow.

3.5.2.1.5.1 Configure Split Requirements

The requirements to configure the phase splits to be used within the traffic signal timing plans / patterns stored within the ASC follow.

3.5.2.1.5.1.1 Configure Phase Split Time

Upon request from a management station, the ASC shall store the time, in seconds from 0 to 999 seconds, which the split phase is allowed to receive, before the phase is terminated / forced off, when constant demand exists on all phases. The split time includes all phase clearance times for the associated phase.

The ASC operates differently depending on the configuration of other parameters as follows:

a) If the phase’s force-off point is floating, the split time parameter is equal to the maximum amount a time a non-coordinated parameter may receive.
b) If the phase’s force-off point is fixed, the allowed time may be longer, if a previous phase gapped out.
c) If the cycle time for a pattern is zero (i.e., the ASC is in Manual Free Mode), then the split time is used as a maximum time for the phase as long as the split time is not zero.
d) If the sum of split times for all phases of a pattern is less than the cycle time, the ASC allocates any extra time to the coordinated phase. If the sum of split times for all phases of a pattern is greater than the defined cycle time for a pattern, then the ASC places itself into the Manual Free mode.

3.5.2.1.5.1.2 Configure Phase Split Mode

Upon request from a management station, the ASC shall store that the operational phase split mode of a phase. The valid split modes are:

a) None. The split mode of the phase is not operated under split mode control.

b) Minimum Recall. The split mode of the phase is operated using the minimum recall setting, where demand is placed for the phase when the phase is not in its Green interval.
c) Maximum Recall. The split mode of the phase is operated using the maximum recall setting, where a constant demand is placed for the phase during all phases. This setting applies for phases with vehicle, transit-only or bicycle movements.
d) Pedestrian Recall. The split mode of the phase is operated with a pedestrian recall, or a constant demand for pedestrian service during all other phases. The minimum recall settings also apply during pedestrian recall.
e) Maximum and Pedestrian Recall. The split mode of the phase is operated using the larger of maximum vehicle recall setting or the pedestrian recall setting.
f) Phase Omitted. The split mode of the phase is operated with this phase omitted.
g) Non-Actuated. The split mode of the phase is operated using a fixed split time for this phase.

3.5.2.1.5.1.3 Configure Split Coordination Phase

Upon request from a management station, the ASC shall store if a given phase is designated as a coordinated phase.

3.5.2.1.5.2 Determine Maximum Number of Phase Splits

Upon request from a management station, the ASC shall return the maximum number of phase splits, as a number from 1 to 255 splits, that can be configured in the ASC.

3.5.2.1.6 Manage Ring Configuration Requirements

The requirements to manage the traffic signal timing rings of the ASC follow.

3.5.2.1.6.1 Configure Sequence Data

Upon request from a management station, the ASC shall store the sequential listing of valid phases to be included in a sequence plan.

3.5.2.1.6.2 Determine Maximum Number of Rings

Upon request from a management station, the ASC shall return the maximum number of rings, as a number from 1 to 255 rings, that can be configured in the ASC.

3.5.2.1.6.3 Determine Maximum Number of Sequences

Upon request from a management station, the ASC shall return the maximum number of sequences, as a number from 1 to 255 sequences, that can be configured in the ASC.

3.5.2.1.7 Manage Channel Configuration Requirements

The requirements to manage the channels of the ASC follow.

3.5.2.1.7.1 Configure Channel Requirements

The requirements to configure the channels within the ASC follow.

3.5.2.1.7.1.1 Configure Channel Control Source

Upon request from a management station, the ASC shall store the phase or overlap that controls each channel.

3.5.2.1.7.1.2 Configure Channel Control Type

Upon request from a management station, the ASC shall store what type of signal head is controlled by the channel. Valid types are:

a) phaseVehicle. The channel controls a vehicle signal head using a phase. This includes bicycle and transit signal heads.
b) phasePedestrian. The channel controls a pedestrian signal head using a phase.
c) Overlap. The channel controls a vehicle signal head using an overlap. This includes flashing yellow arrows, flashing red arrows, as well as bicycle and transit signal head.
d) pedOverlap. The channel controls a pedestrian signal head using an overlap.
e) queueJump. The channel controls a signal head for a queue jump, typically used in transit signal priority.

3.5.2.1.7.1.3 Configure Channel Flash Requirements

The requirements to enable or disable the state of a channel during Automatic Flash or Manual Flash within the ASC follow.

3.5.2.1.7.1.3.1 Configure Channel Flash Yellow

Upon request from a management station, the ASC shall store if the Yellow indication is flashing during Automatic Flash.

3.5.2.1.7.1.3.2 Configure Channel Flash Red

Upon request from a management station, the ASC shall store if the Red indication is flashing during Automatic Flash.

Note: A channel cannot flash both yellow and red simultaneously.

3.5.2.1.7.1.3.3 Configure Channel Flash Alternate Half Hertz

Upon request from a management station, the ASC shall store if the flash alternate Half Hertz is on.

3.5.2.1.7.1.3.4 Configure Channel Flash Alternate First or Second

Upon request from a management station, the ASC shall store if a channel is to flash first or second in an alternating manner. This is known as a Wig-Wag.

3.5.2.1.7.2 Determine Maximum Number of Channels

Upon request from a management station, the ASC shall return the maximum number of channels supported by the ASC.

Note: See the appropriate hardware reference such as NEMA TS2, Caltrans TEES, ATC 5201, or other to determine the hardware's maximum number of channels.

3.5.2.1.8 Manage Overlap Configuration Requirements

The requirements to manage overlaps within the ASC follow.

3.5.2.1.8.1 Configure Overlap Requirements

The requirements to configure the overlaps within the ASC follow.

3.5.2.1.8.1.1 Configure Overlap Type Requirements

The requirements to configure the overlap types used within the ASC follow.

3.5.2.1.8.1.1.1 Configure Overlap Type - Vehicle Normal

Upon request from a management station, the ASC shall store that the Overlap control type is 'Normal'. The overlap may be assigned to vehicular, transit-only or bicycle movements.

a) The ASC sets the overlap output to be Green, when an included overlap phase is green, and when an included overlap phase is yellow (or in the Red Clearance interval), and simultaneously another included overlap phrase is next in the sequence.
b) The overlap is yellow when an included overlap phase is yellow and simultaneously another included overlap phase is not next in the associated phase sequence.
c) Otherwise, the overlap output is red.

3.5.2.1.8.1.1.2 Configure Overlap Type - Vehicle Minus Green and Yellow

Upon request from a management station, the ASC shall store that the Overlap control type is 'Vehicle Minus Green and Yellow'.

a) The overlap output is green if an included overlap phase is green and an overlap modifier phase is not green, or if an included overlap phase is yellow (or in the Red Clearance interval) and simultaneously another included overlap phase is next and while an overlap modifier phase is not green.
b) The overlap is yellow when an included overlap phase is yellow, and an overlap modifier phase is not yellow, and another included overlap phase is not next in the associated phase sequence.
c) Otherwise, the overlap output is red.

3.5.2.1.8.1.1.3 Configure Overlap Type - Pedestrian Normal

Upon request from a management station, the ASC shall store that the Overlap control type is 'Pedestrian Normal'.

a) The overlap output is Walk when an included overlap phase is in a walk interval OR when an included overlap phase is in a pedestrian clearance interval and an overlap included phase is next.
b) Upon completion of the Walk interval, the overlap enters the pedestrian clearance interval if another included overlap phase is not next in the associated phase sequence.
c) The overlap remains in the pedestrian clearance interval or steady don't-walk when an included overlap phase is yellow and simultaneously another included overlap phase is not next in the associated phase sequence.
d) Otherwise, the overlap output is steady Don’t Walk.

3.5.2.1.8.1.1.4 Configure Overlap Type - Flashing Yellow Arrow - 3 Section Head

Upon request from a management station, the ASC shall store that the Overlap control type is 'fYA3-1'. This overlap control type is used with a 3-section signal head where the overlap output drives the green arrow, combined yellow/flashing yellow arrow, and red arrow.

a) The overlap output is flashing yellow arrow when an overlap included phase is green and an overlap modifier phase is NOT green; or when an overlap included phase is yellow (or red clearance), an overlap included phase is next or an overlap modifier phase is next, and a modifier phase is NOT green.

b) The overlap output is yellow when an overlap included phase is yellow, an overlap included phase is not next, and an overlap modifier phase is NOT green; or when an overlap modifier phase is yellow.
c) The overlap output is red when an overlap included phase is red, an overlap modifier phase is NOT green, and an overlap modifier phase is NOT yellow; or when an overlap modifier phase is timing a red-clearance interval.
d) The overlap output is green when an overlap modifier phase is green.

3.5.2.1.8.1.1.5 Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head

Upon request from a management station, the ASC shall store that the Overlap control type is 'fYA4-1'. This overlap control type is used with a 4-section signal head where the overlap output drives the flashing yellow arrow, yellow and red.

a) The overlap output is flashing yellow arrow when an overlap included phase is green and an overlap modifier phase is NOT green; or when an overlap included phase is yellow (or red clearance), an overlap included phase or an overlap modifier phase is next and an overlap modifier phase is NOT green.
b) The overlap output is yellow when an overlap included phase is yellow, an overlap included phase is not next, and an overlap modifier phase is NOT green; and when an overlap modifier phase is yellow.
c) The overlap output is red when an overlap included phase is red, an overlap modifier phase is NOT green, and an overlap modifier phase is NOT yellow; or when an overlap modifier phase is timing a red-clearance interval.
d) The overlap output is blank/dark when an overlap modifier phase is green.

3.5.2.1.8.1.1.6 Configure Overlap Type - Flashing Red Arrow - 3 Section Head

Upon request from a management station, the ASC shall store that the Overlap control type is 'fRA3'. This overlap control type drives the green arrow, yellow arrow, and combined red/flashing red arrow.

a) The overlap output is green when an overlap modifier phase is green.
b) The overlap output is yellow when an overlap modifier phase is yellow; or when an overlap modifier phase is red and an overlap included phase is yellow.
c) The overlap output is red when the overlap modifier and included phases are red.
d) The overlap output is flashing red when an overlap included phase is green and an overlap modifier phase is red.

3.5.2.1.8.1.1.7 Configure Overlap Type - Flashing Red Arrow - 4 Section Head

Upon request from a management station, the ASC shall store that the Overlap control type is 'fRA4'. The overlap control type drives the yellow arrow, red arrow, and flashing red arrow.

a) The overlap outputs are blank when the overlap modifier phase is green.
b) The overlap output is yellow when an overlap modifier phase is yellow; or when an overlap modifier phase is red and an overlap included phase is yellow.
c) The overlap output is red when an overlap modifier phase and an overlap included phase are red.
d) The overlap output is flashing red when an overlap included phase is green and an overlap modifier phase is red.

3.5.2.1.8.1.1.8 Configure Overlap Type - 2 Section Transit Specific Signal Head

Upon request from a management station, the ASC shall store that the Overlap control type is 'transit-2'. The overlap control type drives a 2-section bar signal for transit vehicles using overlap green (vertical bar) and red (horizontal bar) outputs.

a) The overlap output is green (vertical bar) when an overlap included phase is green.
b) The overlap output is a flashing vertical bar when an overlap included phase is yellow and an overlap included phase is not next.
c) The overlap output is red (horizontal bar) when an overlap included phase is red.

3.5.2.1.8.1.1.9 Configure Overlap Type - Minus Green Yellow Alternate

Upon request from a management station, the ASC shall store that the Overlap control type is 'Vehicle Minus Green and Yellow Alternate'.

a) The overlap output is green when an overlap included phase is green and an overlap modifier phase is NOT green; OR when an overlap included phase is yellow (or red clearance) and an overlap included phase is next and an overlap modifier phase is NOT green and an overlap modifier phase is not next.
b) The overlap output is yellow when an overlap included phase is yellow and an overlap modifier phase is NOT yellow and an overlap included phase is not next.
c) The overlap output is red when the overlap green and yellow are not ON.

3.5.2.1.8.1.2 Configure Overlap Included Phases

Upon request from a management station, the ASC shall store the phase numbers that are 'included phases' for the overlap.

3.5.2.1.8.1.3 Configure Overlap Modifier Phases

Upon request from a management station, the ASC shall store the phase numbers that are modifier phases for a vehicle overlap. The modifier phase, when present, affects how the overlap responds, based on the overlap control type. For example, an active modifier phase may exclude the operation of the overlap.

3.5.2.1.8.1.4 Configure Pedestrian Modifier Phases

Upon request from a management station, the ASC shall store the phase numbers that are pedestrian modifier phases for a vehicle overlap. The pedestrian modifier phase, when active, affects how the overlap responds, based on the overlap type.

3.5.2.1.8.1.5 Configure Overlap Trailing Green

Upon request from a management station, the ASC shall store the trailing green time, from 0 to 255 seconds, which is the time that an overlap green that would normally terminate might be extended.

Note: this requirement also covers the use of a Flashing Yellow Arrow in lieu of or in addition to a Green.

3.5.2.1.8.1.6 Configure Overlap Trailing Yellow

Upon request from a management station, the ASC shall store the trailing yellow time in tenths of a second, from 0 to 25.5 seconds. When the overlap green time has been extended (See Section 3.5.2.1.8.1.5 Configure Overlap Trailing Green), then this value determines the overlap's yellow duration.

Note: this requirement also covers the use of a Flashing Yellow Arrow in lieu of or in addition to a Yellow.

3.5.2.1.8.1.7 Configure Overlap Trailing Red Clearance

Upon request from a management station, the ASC shall store the trailing red time in tenths of a second, from 0 to 25.5 seconds. When the overlap green time has been extended (See Section 3.5.2.1.8.1.5 Configure Overlap Trailing Green), then this value determines the overlap's red clearance duration.

Note: this requirement also covers the use of a Flashing Red Arrow in lieu of or in addition to a red clearance indication.

3.5.2.1.8.1.8 Configure Overlap Walk

Upon request from a management station, the ASC shall store the walk time for a pedestrian overlap from 0 to 255 seconds.

3.5.2.1.8.1.9 Configure Overlap Pedestrian Clearance

Upon request from a management station, the ASC shall store the duration of the pedestrian clearance from 0 to 255 seconds.

3.5.2.1.8.2 Configure Multiple Overlap Sets

Upon request from a management station, the ASC shall store overlap parameters for multiple sets of overlaps in a manner consistent with the following requirements as selected under User Need 2.5.2.1.8, Manage Overlap Configurations in Table 5 Protocol Requirements List (PRL).

a) 3.5.2.1.8.1.2 Configure Overlap Included Phases
b) 3.5.2.1.8.1.3 Configure Overlap Modifier Phases
c) 3.5.2.1.8.1.4 Configure Pedestrian Modifier Phases
d) 3.5.2.1.8.1.5 Configure Overlap Trailing Green
e) 3.5.2.1.8.1.6 Configure Overlap Trailing Yellow
f) 3.5.2.1.8.1.7 Configure Overlap Trailing Red Clearance
g) 3.5.2.1.8.1.8 Configure Overlap Walk
h) 3.5.2.1.8.1.9 Configure Overlap Pedestrian Clearance

3.5.2.1.8.3 Retrieve Overlaps Requirements

The requirements to retrieve the overlaps within the ASC follow.

3.5.2.1.8.3.1 Determine Maximum Number of Overlaps

Upon request from a management station, the ASC shall return the maximum number of overlaps that can be configured within the ASC within a single set, from 1 to 255 overlaps.

3.5.2.1.8.3.2 Determine Maximum Number of Overlap Sets

Upon request from a management station, the ASC shall return the maximum number of overlap sets, from 1 to 255 overlap sets, that can be configured within the ASC.

3.5.2.1.9 Manage Preempt Configuration Requirements

The requirements to manage the preemptions within the ASC follow.

3.5.2.1.9.1 Configure Preempt Requirements

The requirements to configure preempts within an ASC follow.

3.5.2.1.9.1.1 Enable/Disable Preempt Inputs

Upon request from a management station, an ASC shall store the enabling or disabling of a preemption input within the ASC. Disabling preempts should be done with extreme caution.

3.5.2.1.9.1.2 Configure Preempt Control - Non-Locking Memory

Upon request from a management station, the ASC shall store if operation is enabled that does not require detector memory, meaning that the preempt does not occur if the preempt request terminates prior to the expiration of the preempt delay time.

3.5.2.1.9.1.3 Configure Preempt Control - Override Automatic Flash

Upon request from a management station, the ASC shall store if a preempt is allowed to override automatic flash.

3.5.2.1.9.1.4 Configure Preempt Control - Override Preempt

Upon request from a management station, the ASC shall store if a preempt is allowed to override the next higher numbered preempt definition. Normally, a lower number preempt may override a higher number preempt, e.g., preempt number 1 may override preempt 2. This requirement prevents the lower number preempt from over-riding the next higher numbered preempt. For example, if preempt 1 is NOT allowed to override the next higher numbered preempt, in this case 2, then a call on preempt 1 will not override a call on preempt 2.

3.5.2.1.9.1.5 Configure Preempt Control - Flash Dwell

Upon request from a management station, the ASC shall store if the phases identified as preempt dwell phases and the overlaps identified as preempt dwell overlaps will Flash Yellow during the Preempt Dwell interval. All other phases and overlaps flash a red indication with this setting.

3.5.2.1.9.1.6 Configure Preempt Control - All Red Entry

Upon request from a management station, the ASC shall store if all phases show a red indication at the start of the preempt. Just to clarify – it deliberately does not state, "all phases and overlaps"

3.5.2.1.9.1.7 Configure Preempt Link

Upon request from a management station, the ASC shall store the identity of a higher priority preempt (lower preempt number) to be combined with the current preempt. At the end of the preempt's Dwell Green time, the ASC automatically calls the linked preempt, which remains active until the preempt signal for the current preempt is removed. The ASC does not link a lower priority preempt (higher preempt number) or a non-valid preempt with the current preempt.

3.5.2.1.9.1.8 Configure Preempt Delay

Upon request from a management station, the ASC shall store the time, from 0 to 999 seconds, that a preempt input might be active prior to initiating a preempt sequence. If a call for a non-locking preempt is removed prior to completion of this time, the ASC does not initiate the preempt sequence.

3.5.2.1.9.1.9 Configure Preempt Minimum Duration

Upon request from a management station, the ASC shall store the minimum duration, from 0 to 65535 seconds, that a preempt is active. The timing begins at the end of the preempt's delay time (See 3.5.2.1.9.1.8 Configure Preempt Delay), if one is defined, otherwise the timing begins when the preempt input goes active and prevents an exit from the preempt dwell interval until this time has elapsed.

3.5.2.1.9.1.10 Preempt Entry Configuration Requirements

The requirements to configure a preempt when the ASC enters preemption follow. The period of time is sometimes called the right-of-way transfer time.

3.5.2.1.9.1.10.1 Configure Preempt Enter Minimum Green Time

Upon request from a management station, the ASC shall store the minimum green duration for a preempt initiated transition, from 0 to 255 seconds. A preempt initiated transition does not cause the termination of an existing GREEN display prior to the lesser of the phase's Minimum Green Time or this preempt minimum green time. If the preempt minimum green time is set to zero, when the ASC immediately terminates the phase's GREEN display.

3.5.2.1.9.1.10.2 Configure Preempt Enter Minimum Walk Time

Upon request from a management station, the ASC shall store the minimum walk time for a preempt initiated transition from 0 to 255 seconds. A preempt initiated transition does not cause the termination of an existing WALK display prior to the lesser of the phase's Pedestrian Green Time or this preempt minimum walk time. If the preempt minimum walk time is set to zero, the ASC immediately terminates the phase's Walk display.

3.5.2.1.9.1.10.3 Configure Preempt Enter Pedestrian Clearance Time

Upon request from a management station, the ASC shall store the pedestrian clearance time for a normal WALK display terminated by a preempt initiated transition, from 0 to 255 seconds. A preempt initiated transition does not cause the termination of an existing pedestrian clearance display prior to the lesser of the phase's pedestrian clearance time or this preempt pedestrian clearance time. If the preempt enter pedestrian clearance time is set to zero, the ASC immediately terminates the phase's pedestrian clearance (Flashing Don't Walk) display.

3.5.2.1.9.1.10.4 Configure Preempt Enter Yellow Change Time

Upon request from a management station, the ASC shall store the duration in tenths of a second, from 0 to 25.5 seconds, of the Enter Yellow interval terminated by a preempt initiated transition. A preempt initiated transition does not cause the termination of an existing YELLOW indication prior to the lesser of the phase's Yellow time or this preempt Enter Yellow time. If the preempt enter Yellow Change time is set to zero, the ASC immediately terminates the phase's YELLOW indication.

3.5.2.1.9.1.10.5 Configure Preempt Enter Red Clearance Time

Upon request from a management station, the ASC shall store the duration in tenths of a second, from 0 to 25.5 seconds, of the Enter Red Clearance interval for a normal Red Clearance interval terminated by a preempt initiated transition. A preempt initiated transition does not cause the termination of the Red Clearance time prior to the lesser of the phase's Red Clearance time or this preempt Enter Red Clearance time. If the preempt Enter Red Clearance time is set to zero, the ASC immediately terminates the phase's Red Clearance time.

3.5.2.1.9.1.11 Configure Preempt Track Clearance Requirements

The requirements to configure a preempt during the track clearance interval follow.

3.5.2.1.9.1.11.1 Configure Preempt Track Clearance Green Time

Upon request from a management station, the ASC shall store the track clearance time for the defined preempt track phases, from 0 to 255 seconds. During this time, the selected Preempt Track Clearance

Phases are green along with any overlaps driven by these phases. If the preempt track clearance time is set to zero, the ASC omits the track clearance movement.

3.5.2.1.9.1.11.2 Configure Preempt Track Clearance Yellow Change Time

Upon request from a management station, the ASC shall store the duration of the Track Yellow change interval in tenths of a second, from 0 to 25.5 seconds. The lesser of the phase's Yellow Change time or this preempt Track Yellow Change time controls the yellow timing for the track clearance movement.

3.5.2.1.9.1.11.3 Configure Preempt Track Red Clearance Time

Upon request from a management station, the ASC shall store the duration of the Track Red Clearance interval in tenths of a second, from 0 to 25.5 seconds. The lesser of the phase's Red Clearance time or this preempt Track Red Clearance time controls the Red Clearance timing for the track clearance movement.

3.5.2.1.9.1.11.4 Configure Preempt Track Clearance Phases

Upon request from a management station, the ASC shall store the phases to be active for the preempt during the preempt track clearance interval.

3.5.2.1.9.1.11.5 Configure Preempt Track Clearance Overlaps

Upon request from a management station, the ASC shall store the overlaps to be active for a preempt during the preempt track clearance interval. Any overlap not specified will not be active during the preempt even if its parent phases are active.

3.5.2.1.9.1.12 Configure Preempt Dwell Requirements

The requirements to configure a preempt during the dwell interval follow.

3.5.2.1.9.1.12.1 Configure Preempt Minimum Green Dwell Time

Upon request from a management station, the ASC shall store the minimum green time for a preempt to remain in a preempt dwell interval, from 0 to 255 seconds. Dwell phases may dwell for this amount of time before any phases cycle during the preempt. The ASC determines the phases that are active during the dwell interval green time based on the settings for the preempt dwell phases. The preempt dwell interval green duration does not terminate prior to the completion of the preempt minimum duration and the preempt minimum green dwell time, and if the preempt call is no longer present / active.

3.5.2.1.9.1.12.2 Configure Preempt Dwell Phases

Upon request from a management station, the ASC shall store the phases to remain in green during the preempt dwell interval.

3.5.2.1.9.1.12.3 Configure Preempt Dwell Pedestrian Movements

Upon request from a management station, the ASC shall store the phases that may have their pedestrian movement(s) rest in walk during the preempt dwell interval, which is followed by the pedestrian movements defined in the preempt cycling pedestrian list.

3.5.2.1.9.1.12.4 Configure Preempt Dwell Overlaps

Upon request from a management station, the ASC shall store the overlaps to be active for a preempt during the preempt dwell interval. Any overlap not specified will not be active during the preempt even if its parent phases are active.

3.5.2.1.9.1.12.5 Configure Preempt Cycling Phases

Upon request from a management station, the ASC shall store the phases to be allowed to cycle during the preempt dwell interval.

3.5.2.1.9.1.12.6 Configure Preempt Cycling Pedestrian Movements

Upon request from a management station, the ASC shall store the phases with pedestrian movements to be allowed to cycle during the preempt dwell interval.

3.5.2.1.9.1.12.7 Configure Preempt Cycling Phases Sequence

Upon request from a management station, an ASC shall store the sequence of the phases selected for cycling during the preempt dwell interval. If no sequence is specified, then the sequence is the one that was in effect when the preempt was activated.

3.5.2.1.9.1.12.8 Configure Preempt Cycling Overlaps

Upon request from a management station, the ASC shall store the overlaps to be allowed to cycle during the preempt dwell interval. Any overlap not specified will not be active during the preempt even if its parent phases are active.

3.5.2.1.9.1.13 Configure Preempt Exit Requirements

The requirements to configure a preempt when the preempt dwell interval ends follow.

3.5.2.1.9.1.13.1 Configure Preempt Exit Phases

Upon request from a management station, the ASC shall store the phases that are allowed to be active following the preempt dwell interval.

3.5.2.1.9.1.13.2 Configure Preempt Exit Phase Strategy

Upon request from a management station, an ASC shall store the exit strategy to be used following the end of the preempt dwell interval. The valid exit strategies are:

a) Exit to Normal Operations. The ASC immediately enters the exit phases to be active as configured following the preempt dwell interval.
b) Exit to Queue Delay Recovery. The ASC enters the phase with the highest demand or longest wait time, as determined by the Preempt Exit Priority Level.
c) Return to Short Service Phase. The ASC goes to the first "short service phase", which is a phase where only the preempt minimum green time was serviced during the advanced preemption time or right-of-way transfer time (preemption entry intervals).
d) Return to Coordination. The ASC immediately returns to the place in the coordinated cycle where the ASC would have been had there been no preempt.

3.5.2.1.9.1.13.3 Configure Preempt Exit Priority Levels

Upon request from a management station, an ASC shall store the relative weights for the priority level for each phase when the Queue Delay Recovery exit strategy is used following the end of the preempt dwell interval. The relative weights are in integers, and a higher number indicates a larger weight for the demand and wait time for that phase.

3.5.2.1.9.1.14 Configure Preempt Max Presence Exceeded Requirements

The requirements to configure the actions to be taken if the preempt maximum presence time has been exceeded within the ASC follow.

3.5.2.1.9.1.14.1 Configure Preempt Maximum Presence Time

Upon request from a management station, the ASC shall store the maximum presence time, from 0 to 65535 seconds, for which a preempt can remain active. If this preempt presence time has elapsed, the call for preemption is considered invalid until a change of the preempt's state occurs (such as the preempt is no longer active). The timing begins at the end of the preempt's delay time, if one is defined, otherwise the timing begins when the preempt input goes active. If the preempt maximum presence time is set to zero, the ASC is to disable the preempt maximum presence time.

3.5.2.1.9.1.14.2 Configure Preempt Maximum Presence Action

Upon request from a management station, an ASC shall store how to exit preemption if the preempt maximum presence time has been exceeded. Valid actions when the maximum presence time has been exceeded are:

a) Preempt Exit phases. The ASC will use the exit strategy following the end of the preempt dwell interval (See 3.5.2.1.9.1.13.2 Configure Preempt Exit Phase Strategy).
b) All-Red Flash. The ASC will go to all-red flash.

3.5.2.1.9.1.15 Configure Preempt Gate Description

Upon request from a management station, the ASC shall store a description for a gate that activates during preemption.

3.5.2.1.9.2 Determine Maximum Number of Preempts

Upon request from a management station, the ASC shall return the maximum number of preempts, as a number from 1 to 255 preempts, that can be configured in the ASC.

3.5.2.1.10 Manage ASC Scheduler Requirements

The ASC may be configured to use different traffic signal timing patterns or special functions based on a time-of-day schedule. The requirements to manage the ASC scheduler functions are found in Section 8.1 and Section 8.6 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.2.1.10, Manage Timing Pattern Scheduler and User Need 2.5.2.1.11, Manage Action Scheduler in Table 5.

3.5.2.1.10.1 Configure ASC Timebased Action Requirements

The ASC scheduler functions may use timebased actions to implement different traffic signal timing patterns and special functions. The following requirements are used to implement such actions.

Note: All the functionality of these timebased action requirements are also supported by timing pattern requirements found in 3.5.2.1.4 Manage Timing Patterns Requirements. Previous versions of NTCIP 1202 only supported use of timebased actions to implement scheduled special functions. NTCIP 1202 v04 recommends using timing patterns to implement scheduled special functions but implementations that use timebased actions are still conformant.

3.5.2.1.10.1.1 Configure Timebased Action - Pattern

Upon request from a management station, the ASC shall store the identity of the timing pattern that is active when the timebased action is active. If the timebased action pattern is set to zero, the ASC reverts to a lower priority entity such as 'interconnect' (if available).

3.5.2.1.10.1.2 Configure Timebased Action - Special Functions

Upon request from a management station, the ASC shall configure up to 32 special functions to be activated when the Action is active.

3.5.2.1.10.1.3 Determine Maximum Number of Timebased Actions

Upon request from a management station, the ASC shall return the maximum number of timebased actions that can be configured in the ASC.

3.5.2.1.10.1.4 Determine Action in Effect

Upon request from a management station, the ASC shall return what action plan is currently in effect.

3.5.2.1.10.1.5 Activate Action Plan Remotely

Upon request from a management station, the ASC shall activate a configured action plan. This requirement allows a management station to activate or override a timebased action plan entry, even if the timing pattern the action plan is associated with is not in effect.

3.5.2.1.11 Manage I/O Mapping Requirements

The ASC communicates with different Field I/O Devices in the cabinet. The number and types of Field I/O Devices depend on the transportation cabinet architecture used. The types of Field I/O Devices supported for each transportation cabinet architecture are indicated in Table 6.

Table 6 Field I/O Devices Supported

Cabinet Architecture	Field I/O Devices Supported
Model 332 Cabinet	Model 2070-2A (or equivalent) - Defined in ATC 5202 - Model 2070 Controller Standard Version 3.
NEMA TS 1 Cabinet	Model 2070-8 (or equivalent) - Defined in ATC 5202 - Model 2070 Controller Standard Version 3.
NEMA TS 2 Type 1 Cabinet	Terminal & Facilities (T&F) Bus Interface Unit (BIU) - Defined in NEMA TS 2-2021. Detector Bus Interface Unit (BIU) - Defined in NEMA TS 2-2021.
NEMA TS 2 Type 2 Cabinet	Model 2070-8 (or equivalent) - Defined in ATC 5202 - Model 2070 Controller Standard Version 3.
ATC Cabinet	Serial Interface Unit - Defined in ATC 5301 v02

In addition to these the ASC may communicate with Auxiliary I/O devices (such as the front panel AUX switch on the 170 and 2070 controllers) and manufacturer specific custom I/O (such as 'D' connectors on TS1 controllers).

The ASC I/O mapping supports all these types of I/O devices and combinations of them to create a mapping of all I/O active at any one time. Multiple I/O mappings are supported which allow default I/O maps for different situations and cabinet configurations to be pre-loaded.

The Active I/O map is the I/O map that the ASC is currently using for signal operations. Changing the Active I/O map requires a database transaction. A new Active I/O map will only take effect if the database transaction data is successfully verified and the I/O Map Activate Conditions are satisfied (See Section 3.5.2.1.11.2.4 Retrieve I/O Mapping Activate Conditions).

The requirements to manage the Input/Output (I/O) Mapping within the ASC follow.

3.5.2.1.11.1 Configure I/O Mapping Requirements

The requirements to configure the I/O Mapping within the ASC follow.

3.5.2.1.11.1.1 Set Active I/O Map

Upon request from a management station, an ASC shall change the Active I/O map currently being used. This change is required to be made as part of a database transaction, and only if the Activate Requirements specified in section 3.5.2.1.11.2.4 are satisfied for the new I/O map to take effect.

3.5.2.1.11.1.2 Configure I/O Map Requirements

The requirements to configure an I/O Map within the ASC follow.

3.5.2.1.11.1.2.1 Configure I/O Map Description

Upon request from a management station, an ASC shall store the description for an I/O map. This description may be any text describing the I/O map such as the intended cabinet type, the intended intersection, etc.

3.5.2.1.11.1.2.2 Configure I/O Map Input Requirements

The requirements to configure the inputs for an I/O Map within the ASC follow.

3.5.2.1.11.1.2.2.1 Configure I/O Map Input Device

Upon request from a management station, an ASC shall store a value indicating the device for each input pin in an I/O map.

3.5.2.1.11.1.2.2.2 Configure I/O Map Input Device Pin

Upon request from a management station, an ASC shall store a value indicating the device pin number for each input pin in an I/O map.

3.5.2.1.11.1.2.2.3 Configure I/O Map Input Function

Upon request from a management station, an ASC shall store a value indicating the input function to be mapped to each input pin in an I/O map.

3.5.2.1.11.1.2.3 Configure I/O Map Output Requirements

The requirements to configure the outputs for an I/O Map within the ASC follow.

3.5.2.1.11.1.2.3.1 Configure I/O Map Output Device

Upon request from a management station, an ASC shall store a value indicating the device for each output pin in an I/O map.

3.5.2.1.11.1.2.3.2 Configure I/O Map Output Device Pin

Upon request from a management station, an ASC shall store a value indicating the device pin number for each output pin in an I/O map.

3.5.2.1.11.1.2.3.3 Configure I/O Map Output Function

Upon request from a management station, an ASC shall store a value indicating the output function to be mapped to each output pin in an I/O map.

3.5.2.1.11.2 Determine I/O Mapping Requirements

The requirements to retrieve the I/O Mapping within the ASC follow.

3.5.2.1.11.2.1 Retrieve Maximum Number of I/O Maps

Upon request from a management station, an ASC shall return the maximum number of I/O maps supported by the ASC.

3.5.2.1.11.2.2 Retrieve Maximum Number of I/O Map Inputs

Upon request from a management station, an ASC shall return the maximum number of I/O map inputs supported by the ASC. This is the number of inputs that the ASC can support at any one time from all input devices.

3.5.2.1.11.2.3 Retrieve Maximum Number of I/O Map Outputs

Upon request from a management station, an ASC shall return the maximum number of I/O map outputs supported by the ASC. This is the number of inputs that the ASC can support at any one time from all input devices.

3.5.2.1.11.2.4 Retrieve I/O Mapping Activate Conditions

Upon request from a management station, an ASC shall return requirements to be fulfilled for a new I/O map to take effect. The values defined by this standard are:

a) a cabinet door be open (indicating that a technician is at the cabinet)
b) the cabinet be in any flash state
c) the cabinet be in all red flash
d) the cabinet be in cabinet (CVM) flash
e) the ASC must be restarted

3.5.2.1.11.2.5 Retrieve I/O Mapping Input Functions

Upon request from a management station, an ASC shall return a listing of the input functions that the ASC supports for I/O mapping.

3.5.2.1.11.2.6 Retrieve I/O Mapping Output Functions

Upon request from a management station, an ASC shall return a listing of the output function that the ASC supports for I/O mapping.

3.5.2.1.11.2.7 Retrieve I/O Map Input Device Pin Status

Upon request from a management station, an ASC shall return the status of each input in an I/O map.

3.5.2.1.11.2.8 Retrieve I/O Map Output Device Pin Status

Upon request from a management station, an ASC shall return the status of each output in an I/O map.

3.5.2.1.11.2.9 Enumerate I/O Mapping Device Pin Requirements

The ASC MIB shall contain enumerations of the standard devices and their device pins that an ASC shall support. These enumerations are:

3.5.2.1.11.2.9.1 Enumerate I/O Map - FIO Inputs

The ASC shall support a default mapping of the device input pins for a Model 332 cabinet.

3.5.2.1.11.2.9.2 Enumerate I/O Map - FIO Outputs

The ASC shall support a default mapping of the device output pins for a Model 332 cabinet.

3.5.2.1.11.2.9.3 Enumerate I/O Map - TS1 Inputs

The ASC shall support a default mapping of the device input pins for a NEMA TS 1 or a NEMA TS 2 Type 2 cabinet.

3.5.2.1.11.2.9.4 Enumerate I/O Map - TS1 Outputs

The ASC shall support a default mapping of the device output pins for a NEMA TS 1 or a NEMA TS 2 Type 2 cabinet.

3.5.2.1.11.2.9.5 Enumerate I/O Map - TS2 BIU Inputs

The ASC shall support a default mapping of the device input pins for a NEMA TS 2 Type 1 cabinet.

3.5.2.1.11.2.9.6 Enumerate I/O Map - TS2 BIU Outputs

The ASC shall support a default mapping of the device output pins for a NEMA TS 2 Type 1 cabinet.

3.5.2.1.11.2.9.7 Enumerate I/O Map - ATC Cabinet SIU Inputs

The ASC shall support a default mapping of the device input pins for an ATC cabinet.

3.5.2.1.11.2.9.8 Enumerate I/O Map - ATC Cabinet SIU Outputs

The ASC shall support a default mapping of the device output pins for an ATC cabinet.

3.5.2.1.11.2.9.9 Enumerate I/O Map - Auxiliary Device Inputs

The ASC shall support a default mapping for the auxiliary input pins.

3.5.2.1.11.2.9.10 Enumerate I/O Map - Auxiliary Device Outputs

The ASC shall support a default mapping for the auxiliary output pins.

3.5.2.1.12 Manage Intra-Cabinet Communications Requirements

The requirements to manage intra-cabinet communications within the ASC follow.

3.5.2.1.12.1 Manage Intra-Cabinet Communications Requirements - ATC

The requirements to retrieve intra-cabinet communications configuration in the ASC in ATC Cabinets follow.

3.5.2.1.12.1.1 Determine Serial Bus 1 Device Present

Upon request from a management station, the ASC shall return if a device is present for a Serial Bus 1 address. The ASC only transmits command frames to those devices that are present as determined by this value.

3.5.2.1.12.2 Manage Intra-Cabinet Communications Requirements - TS2

The requirements to retrieve intra-cabinet communications configuration in the ASC in NEMA TS2 Cabinets follow.

3.5.2.1.12.2.1 Determine TS2 Port 1 Device Present

Upon request from a management station, the ASC shall return if a device is present for a TS2 Port 1 address. The ASC only transmits command frames to those devices that are present as determined by this value.

3.5.2.1.12.2.2 Enable/Disable TS2 Port 1 Frame 40 Messages

Upon request from a management station, the ASC shall enable or disable the Frame 40 messages for each Port 1 address. Frame 40 is used to poll the secondary stations for a secondary-to-secondary message exchange. The ASC only transmits Command 40 series frames to those devices that are enabled, as determined by this value.

3.5.2.1.13 Manage ADA Support Requirements

The requirements to manage American Disability Act (ADA) Support in the ASC follow.

3.5.2.1.13.1 Configure ADA Support Requirements

The requirements to configure the ASC to support ADA in the ASC follow.

3.5.2.1.13.1.1 Configure APS Push Button Minimum Press Time

Upon request from a management station, the ASC shall store the time in tenths of seconds, from 0.0 to 25.5 seconds, that an Accessible Pedestrian Signal (APS) Push Button needs to be pressed as a minimum to actuate any APS features. This requirement enables the ASC to receive inputs from installed APS push buttons to actuate any APS features. FHWA MUTCD 2023 Edition Section 4K.05, item 02 states that the push button should be pressed for 1.0 seconds or greater to actuate any APS features. A value of 0.0 indicates that the APS features are disabled.

3.5.2.1.13.1.2 Configure APS Push Button to Phase Association

Upon request from a management station, the ASC shall which phase an APS push button is associated with.

3.5.2.1.13.1.3 Configure APS Extra Crossing Time

Upon request from a management station, the ASC shall store the time in tenths of seconds, from 0.0 to 25.5 seconds, that the pedestrian walk and clearance time is extended, if an APS push button has been pressed for equal to or greater than the APS push button minimum press time. A value of 0 indicates no additional crossing time.

3.5.2.1.13.1.4 Configure Pedestrian Detector for Alternate Pedestrian Timing

Upon request from a management station, the ASC shall store if a pedestrian detector is used to place calls for alternate pedestrian timing.

3.5.2.1.14 Manage Block Object Requirements

The requirements to manage the compressed data blocks within the ASC follow.

3.5.2.1.14.1 Configure Block Object Get Control Requirements

The requirements to define the ASC-specific compressed data block objects for configuring ASC parameters follow.

3.5.2.1.14.1.1 Configure Block Object Get Control - Phase Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the phases sets within the ASC.

3.5.2.1.14.1.2 Configure Block Object Get Control - Vehicle Detector Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the vehicle detector sets within the ASC.

3.5.2.1.14.1.3 Configure Block Object Get Control - Pedestrian Detector Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the pedestrian detectors within the ASC.

3.5.2.1.14.1.4 Configure Block Object Get Control - Pattern Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the timing patterns within the ASC.

3.5.2.1.14.1.5 Configure Block Object Get Control - Split Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the phase split definitions within the ASC.

3.5.2.1.14.1.6 Configure Block Object Get Control - Overlap Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the phase overlap sets within the ASC.

3.5.2.1.14.1.7 Configure Block Object Get Control - Preempt Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the preempts within the ASC.

3.5.2.1.14.1.8 Configure Block Object Get Control - Sequence Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the phase sequences within the ASC.

3.5.2.1.14.1.9 Configure Block Object Get Control - Channel Data

Upon request from a management station, the ASC shall store the compressed data block object reference parameters needed to configure the channels within the ASC.

3.5.2.1.14.2 Monitor Block Error Status - Error-causing Data Element

When a SET request from a management station is received that contains invalid values and/or values not supported by the ASC, the ASC shall return the identification of the data element that caused the error.

3.5.2.2 Monitor Signal Operations Requirements

The requirements to monitor signal operations within the ASC follow.

3.5.2.2.1 Determine Controller Health Requirements

The requirements to determine the ASC's health status follow. In addition, some requirements to monitor the status of the ASC are found in Section 8.5 of ISO 26048-1 – the specific requirements can be found under User Need 2.5.2.2.1, Determine Controller Health in Table 5.

3.5.2.2.1.1 Monitor External Alarm Input States

Upon request from a management station, the ASC shall return the active state of all user configured alarm inputs.

3.5.2.2.1.2 Monitor External Alarm Active

Upon request from a management station, the ASC shall return an alarm value when any of the user configured alarm input is active.

3.5.2.2.1.3 Monitor Flash Status

Upon request from a management station, the ASC shall return an alarm value when the intersection is in Flash and why the intersection is in Flash. Valid reasons defined by this standard are:

a) Automatic. The ASC is programmed to run Automatic Flash.
b) Local Manual. The local flash input is active, SMU Flash is not active, and the flash is not commanded by the central system.
c) Controller Fault Flash. The flash is the result of the ASC detecting a fault.
d) SMU. The Signal Monitoring Unit Flash input is active.
e) Start-Up. The ASC is currently in the Start-Up flash period.
f) Preempt. The flash is the result of a preempt.
g) Remote Manual. The flash was commanded by a local management station or the central system.
h) Other. Flash for a reason not defined by this standard.

The ASC can only report one state at a time.

3.5.2.2.1.4 Monitor Local Override

Upon request from a management station, the ASC shall return an alarm value when any external input or ASC programming has prevented the device from responding to a system pattern command (See 3.5.2.3.2.1 Activate System Timing Pattern Remotely).

3.5.2.2.1.5 Monitor Coordination Alarm

Upon request from a management station, the ASC shall return an alarm value when the ASC is not running the called pattern without offset correction within a user-specified number of cycles from receiving the command (default = three cycles). The ASC does not cause an alarm to be set if an offset correction requires less than the user-specified number of cycles (default = three) due to cycle overrun caused by servicing a pedestrian call.

3.5.2.2.1.6 Monitor Detector Fault

Upon request from a management station, the ASC shall return an alarm value when a vehicle or pedestrian detector fault occurs.

3.5.2.2.1.7 Monitor Stop Time Input Alarm

Upon request from a management station, the ASC shall return an alarm value when the stop time input is active.

3.5.2.2.1.8 Monitor Cycle Fault Alarm

Upon request from a management station, the ASC shall return an alarm value when the ASC is operating in the coordinated mode and cycling diagnostics indicate that a serviceable call exists that has not been serviced for two cycles.

3.5.2.2.1.9 Monitor Coordination Fault

Upon request from a management station, the ASC shall return an alarm value when a cycle fault is in effect and the serviceable call has been serviced within two cycles after the cycle fault alarm activating.

3.5.2.2.1.10 Monitor Coordination Fail Alarm

Upon request from a management station, the ASC shall return an alarm value when a Coordination Fault is in effect and a Cycle Fault occurs again within two cycles of the coordination retry.

3.5.2.2.1.11 Monitor Cycle Fail Alarm

Upon request from a management station, the ASC shall return an alarm value when the ASC is operating in non-coordinated mode as the result of either a Cycle Fault or the ASC operating in Free mode, and cycling diagnostics indicate that a serviceable call exists that has not been serviced for two cycles.

3.5.2.2.1.12 Monitor Cabinet IO Link Alarm

Upon request from a management station, the ASC shall return an alarm value when there is a communication error with an IO device in the cabinet, such as a NEMA TS2 Port 1 response frame fault.

3.5.2.2.1.13 Monitor SMU Communications Error

Upon request from a management station, an ASC shall return an alarm value when there is communication error with the SMU because the communication link has failed or because the SMU has been removed.

3.5.2.2.1.14 Monitor Preempt Maximum Presence Alarm

Upon request from a management station, the ASC shall return if the preempt maximum presence timer has been exceeded. This fault indicates that a preempt call has remained active for a time period greater than the maximum time configured.

3.5.2.2.2 Retrieve Current Operation Requirements

The requirements to determine the ASC's mode of operations within the ASC follow.

3.5.2.2.2.1 Monitor Unit Control Status

Upon request from a management station, the ASC shall return the control mode for the ASC. Valid ASC unit control states defined by this standard are:

a) Other. The ASC is controlled by a source not specified by the standard.
b) System Control. The ASC is controlled by master or central commands.
c) System Standby. The ASC is controlled locally based on master or central command to use local control.
d) Backup Mode. The ASC is in backup mode.
e) Manual. The ASC is controlled by a manual selection of a timing pattern, manual free or manual flash.
f) Local Timebase. The ASC is controlled by the local time base.
g) Interconnect. The ASC is controlled by the local interconnect inputs.
h) Interconnect Backup. The ASC is controlled by the local TBC due to invalid Interconnect inputs or loss of sync.
i) Remote Manual Control. The ASC is controlled by central command by issuing Holds on a Green Rest point in each phase or interval and then issues a Remote Interval Advance Control command to advance to the next interval.
j) Local Manual Control. The ASC is controlled by manual interval advances at the cabinet, such as from a Police Panel.

3.5.2.2.2.2 Monitor Preempt Active

Upon request from a management station, the ASC shall return an alarm value when any of the preemption inputs is active.

3.5.2.2.2.3 Monitor Offset Transitioning

Upon request from a management station, the ASC shall return an alarm value when it is performing an offset transition.

3.5.2.2.2.4 Monitor Priority Call Active

Upon request from a management station, the ASC shall return an alarm value when it is responding to a priority call using NTCIP 1211 or another method.

3.5.2.2.2.5 Monitor Local Free Status

Upon request from a management station, the ASC shall return an alarm value when any of the ASC's inputs and/or programming cause the ASC not to run coordination and why the ASC is running in Free Mode. Valid reasons are:

a) The ASC is running in Free mode because there is no called pattern.
b) The ASC is running in Free mode but is cycling to a point to begin coordination.
c) The ASC is not responding to coordination due to one of the ASC inputs.
d) The ASC programming for the called pattern is to operate in the Free mode.
e) The ASC is running in Free mode because the called timing pattern is invalid.
f) The ASC is running in Free mode due to a request by the ASC's internal cycling diagnostics.
g) The ASC is running in Free mode for a reason not specified by the standard.

The ASC can only report one state at a time.

3.5.2.2.2.6 Monitor Coordination Active

Upon request from a management station, the ASC shall return an alarm value when coordination is active and not preempted or overridden.

3.5.2.2.2.7 Monitor ECLA Control Active

Upon request from a management station, the ASC shall return an alarm value when it is currently controlled by an ECLA.

3.5.2.2.2.8 Monitor Current Timing Pattern Requirements

The requirements to monitor the ASC's current timing pattern follow.

3.5.2.2.2.8.1 Monitor Current Pattern Status

Upon request from a management station, the ASC shall return the coordination pattern or mode currently in effect, regardless of programmed settings. Valid values defined by this standard are:

a) The signal timing pattern operating in Coordination mode.
b) 254. The ASC is operating in Manual Free mode.
c) 255. The ASC is operating in Manual Flash mode.

The ASC can report only one state at a time.

3.5.2.2.2.8.2 Monitor Current Pattern Command Source

Upon request from a management station, the ASC shall return the source that activated the mode of operation that is current in effect. Valid values defined by this standard are:

a) Remote. The pattern / mode was commanded remotely through NTCIP or another remote interface.
b) Timebased. The pattern / mode was commanded from a schedule.
c) Front Panel. The pattern / mode was commanded from the front panel.
d) Backup Mode. The pattern / mode was caused by the ASC going into Backup mode or a failure.
e) Other. The pattern / mode was commanded from a source not defined by this standard.

3.5.2.2.2.8.3 Monitor Current Pattern Fault Status

Upon request from a management station, the ASC shall return if a timing signal timing pattern is running or why the pattern is invalid. Valid reasons defined in this standard on why the pattern is invalid are:

a) Bad Cycle Time. The pattern cycle time is not adequate to service the minimum requirements of all phases.
b) Split Overrun. The sum of all split times (3.5.2.1.5.1.1 Configure Phase Split Time) exceed the pattern’s cycle time.
c) Invalid Offset. The pattern is configured for coordination and the programmed pattern offset time value is not less than programmed cycle time.
d) Invalid Reference Point. The pattern is configured for coordination and does not have a reference point that within a coordinated phase within the programmed sequence.
e) Other. The pattern is invalid due to a reason not specified in the standard.

3.5.2.2.2.9 Monitor Current Cycle Requirements

The requirements to monitor the current cycle information follow.

3.5.2.2.2.9.1 Monitor Coordination Cycle Status

Upon request from a management station, the ASC shall return the current position in the local coordination cycle of the running pattern in seconds, from 0 to 2x the maximum cycle length. This value counts down from the current pattern's cycle time to zero. This value may be greater than the current pattern's cycle time during a coordination cycle with offset correction by the amount of the correction.

3.5.2.2.2.9.2 Monitor Coordination Synchronization Status

Upon request from a management station, the ASC shall return the time into the master cycle. This value counts up from zero to current pattern's cycle time, but may exceed the current pattern's cycle time if the system reference point has changed (See 3.5.2.1.3.9 Configure Pattern Synchronization Time).

3.5.2.2.2.9.3 Monitor Current Offset

Upon request from a management station, an ASC shall return the offset currently in effect, including during transition.

Note: The offset may change during the current cycle as a result of a priority call or the ASC coordination process.

3.5.2.2.3 Monitor Phase Status Requirements

The requirements to monitor the phases within the ASC follow.

3.5.2.2.3.1 Monitor Active Red Phases

Upon request from a management station, the ASC shall return which phases currently have an active RED indication.

This does not include phases that are disabled.

3.5.2.2.3.2 Monitor Active Yellow Phases

Upon request from a management station, the ASC shall return which phases currently have an active YELLOW indication.

3.5.2.2.3.3 Monitor Active Green Phases

Upon request from a management station, the ASC shall return which phases currently have an active GREEN indication.

3.5.2.2.3.4 Monitor Active Don't Walk Phases

Upon request from a management station, the ASC shall return which pedestrian phases currently have an active DON’T WALK indication.

3.5.2.2.3.5 Monitor Active Pedestrian Clearance Phases

Upon request from a management station, the ASC shall return which pedestrian phases currently have an active Flashing DON’T WALK indication.

3.5.2.2.3.6 Monitor Active Walk Phases

Upon request from a management station, the ASC shall return which pedestrian phases currently have an active WALK indication.

3.5.2.2.3.7 Monitor Active On Phases

Upon request from a management station, the ASC shall return which phases are currently active. A phase is active during its Green, Yellow, Red Clearance, Walk, or Pedestrian Clearance intervals.

3.5.2.2.3.8 Monitor Next Phases

Upon request from a management station, the ASC shall return which phases are currently committed to be active next (after the current Active Phase terminates). The ASC determines the next phase to be serviced at the end of the Green interval of the terminating phase, if possible. If the next phase to be serviced cannot be determined at the end of the Green interval, the ASC makes the determination after the end of all vehicle change and clearance intervals.

3.5.2.2.3.9 Monitor Phase Vehicle Calls

Upon request from a management station, the ASC shall return which phases currently have an active vehicle call. This may include bicycles and transit vehicles. A call may be placed remotely, such as by external detection of a vehicle or by a CV Application Process.

3.5.2.2.3.10 Monitor Phase Pedestrian Calls

Upon request from a management station, the ASC shall return which phases currently have an active pedestrian call. A call may be placed remotely, such as through an external input or by a CV Application Process.

3.5.2.2.4 Retrieve Current Ring Requirements

The requirements to monitor the ring control status within the ASC follow.

3.5.2.2.4.1 Monitor Ring Status

Upon request from a management station, the ASC shall return all of the current status indications, which are valid at the time the request was issued, for each configured ring. Valid ring states defined by this standard are:

a) Minimum Green
b) Extension
c) Maximum
d) Green Rest
e) Yellow Change
f) Red Clearance
g) Red Rest
h) Advanced Walk
i) Walk
j) Delayed Pedestrian Waiting
k) Rest-in-Walk
l) Flashing Don’t Walk
m) Don’t Walk
n) Undefined

3.5.2.2.4.2 Monitor Ring Termination Cause

Upon request from a management station, the ASC shall return if the active phase in the ring was terminated by force off, maximum time out or gap out.

3.5.2.2.4.3 Monitor Current Phase On Time

Upon request from a management station, an ASC shall return the time into the currently active phase, in tenths of seconds, for each ring.

3.5.2.2.5 Retrieve Current Channel Status Requirements

The requirements to retrieve the current status of the channels (organized as channel status groups) within the ASC follow.

3.5.2.2.5.1 Monitor Active Red Channels

Upon request from a management station, the ASC shall return which channels currently have an active RED indication.

3.5.2.2.5.2 Monitor Active Yellow Channels

Upon request from a management station, the ASC shall return which channels currently have an active YELLOW indication.

3.5.2.2.5.3 Monitor Active Green Channels

Upon request from a management station, the ASC shall return which channels currently have an active GREEN indication.

3.5.2.2.6 Retrieve Current Overlap Status Requirements

The requirements to retrieve the current status of the overlaps within the ASC follow.

3.5.2.2.6.1 Monitor Active Red Overlaps

Upon request from a management station, the ASC shall return which overlaps currently have an active RED indication.

This does not include overlaps that are not in use.

3.5.2.2.6.2 Monitor Active Yellow Overlaps

Upon request from a management station, the ASC shall return which overlaps currently have an active YELLOW indication.

3.5.2.2.6.3 Monitor Active Green Overlaps

Upon request from a management station, the ASC shall return which overlaps currently have an active GREEN indication.

3.5.2.2.6.4 Monitor Active Flashing Yellow Arrow Overlaps

Upon request from a management station, the ASC shall return which overlaps currently have an active Flashing Yellow Arrow indication.

3.5.2.2.6.5 Monitor Active Flashing Red Arrow Overlaps

Upon request from a management station, the ASC shall return which overlaps currently have an active Flashing Red Arrow indication.

3.5.2.2.7 Retrieve Current Preempt Status Requirements

The requirements to retrieve the current status of the preempts within the ASC follow.

3.5.2.2.7.1 Monitor Currently Active Preempt

Upon request from a management station, the ASC shall return the identifier of the preempts that are currently being serviced, if any.

3.5.2.2.7.2 Monitor Current Preempt Inputs

Upon request from a management station, an ASC shall return the input state for each preempt input configured in the ASC. Valid input states defined by this standard are:

a) no preempt input signal detected
b) preempt input signal is detected

3.5.2.2.7.3 Monitor Current Preempt State

Upon request from a management station, the ASC shall return the preempt status of the current active preempt. Valid preempt states defined by this standard are:

a) Not Active. The preemption input is not active, and this preemption is not active.
b) Not Active with Call. The preemption input is active, but the preemption service has not initiated (Delay Interval or higher preempt service). This state is mutually exclusive to the 'Advanced Preemption' status.
c) Entry Started. The preemption service is timing the entry intervals.
d) Track Service. The preemption service is timing the track clearance intervals.
e) Dwell. The preemption service is timing the dwell intervals.
f) Link Active. The preemption service is performing the linked operation.
g) Exit Strategy in Effect. The preemption service is timing the exit strategy.
h) Maximum Presence. The preempt input has exceeded the preempt's maximum presence time.
i) Advanced Preemption. The preemption service is timing the advanced preemption time. This state is mutually exclusive to the 'Not Active with Call' status.

j) Other. Preempt service is not defined in NTCIP 1202.

Each preempt input can be only in one state at a time.

3.5.2.2.7.4 Monitor Current Gate Status

Upon request from a management station, the ASC shall return whether each of the gates are fully lowered.

3.5.2.2.8 Retrieve Special Function Outputs Requirements

The requirements to retrieve the special functions within the ASC follow.

3.5.2.2.8.1 Determine Maximum Number of Special Functions

Upon request from a management station, the ASC shall return the maximum number of special functions, as a number from 1 to 255 special functions, which can be configured in the ASC.

3.5.2.2.8.2 Monitor Special Function Status

Upon request from a management station, the ASC shall return an indication whether a special function, regardless if it is a physical or logical function, is on or off.

3.5.2.2.8.3 Monitor Special Function Control Source

Upon request from a management station, the ASC shall return the source that activated a special function, regardless if it is a physical or logical function. Valid values defined by this standard are:

a) Remote. A management station activated the special function.
b) Timebased. The Action Scheduler activated the special function.
c) Front Panel. The special function was activated via the front panel.
d) Other. The special function was activated by a source not specified by the standard.

3.5.2.2.9 Monitor Intra-Cabinet Communications Requirements

The requirements to monitor the intra-cabinet communications within the ASC follow.

3.5.2.2.9.1 Monitor TS2 Port 1 Status

Upon request from a management station, the ASC shall return the communications status with the device on a TS2 Port 1 address. Valid TS2 Port 1 states defined by this standard are:

a) Online. Indicates that at least five of the most recent ten response transfers were received correctly.
b) Response Fault. Indicates that more than five of the most recent ten response transfers were received incorrectly.
c) Other. Indicates a state not specified by this standard.

3.5.2.2.9.2 Monitor TS2 Port 1 Fault Frame

Upon request from a management station, the ASC shall return the frame number that caused the most recent fault for a TS2 Port 1 address.

3.5.2.2.9.3 Monitor ATC Serial Bus 1 Status

Upon request from a management station, the ASC shall return the communications status with the device on a Serial Bus 1 address. Valid Serial Bus 1 States are:

a) Online: Indicates that at least five of the most recent ten response transfers were received correctly.
b) Response Fault: Indicates that more than five of the most recent ten response transfers were received incorrectly.
c) Other: Indicates a state not specified by the standard.

3.5.2.2.10 Monitor Signal Monitoring Unit Requirements

The requirements to monitor SMU outputs for diagnostic purposes follow.

3.5.2.2.10.1 Monitor Signal Monitoring Unit Channel Voltage

Upon request from a management station, the ASC shall return the signal circuit voltage, from 0 to 254 VAC RMS or from 0 to 50 VDC RMS for DC cabinets, on a channel as provided by the SMU. A value of 255 is used to indicate unknown, or no measurement.

3.5.2.2.10.2 Monitor Signal Monitoring Unit Channel Current

Upon request from a management station, the ASC shall return the current, in milliAmps, from 0 to 65000 mA, on a channel as provided by the SMU. A value of 65535 indicates unknown or no measurement.

3.5.2.3 Manage Signal Operations Control Requirements

The requirements to manage the control of the signal operations within the ASC follow.

3.5.2.3.1 Control ASC Function Requirements

The requirements to activate functions within the ASC follow.

3.5.2.3.1.1 Enable/Disable Manual Backup

Upon request from a management station, the ASC shall enable Manual Backup Mode, regardless of how much time is remaining in the Backup Timer (See Section 3.5.2.1.1.2 Configure Backup Time) and clear all system control parameters as defined in Section 5.4.3 Backup Time Parameter. The system control parameters cannot be used until this setting is disabled.

3.5.2.3.1.2 Control Global Minimum Recall

Upon request from a management station, the ASC shall store if a recurring demand is to exist on all phases for minimum vehicle service.

3.5.2.3.1.3 Control Call to Non-Actuated 1

Upon request from a management station, the ASC shall store if all phases programmed to respond to a Call to Non-Actuated 1 input should operate in non-actuated mode.

3.5.2.3.1.4 Control Call to Non-Actuated 2

Upon request from a management station, the ASC shall store if all phases programmed to respond to a Call to Non-Actuated 2 input should operate in non-actuated mode.

3.5.2.3.1.5 Control Walk Rest Modifier

Upon request from a management station, the ASC shall store if any non-actuated phases remain in the timed-out Walk state (Rest in Walk) in the absence of a serviceable conflicting call. This only applies to phases programmed to respond to a Call to Non-Actuated input or a split running in non-actuated mode.

3.5.2.3.1.6 Control Interconnect

Upon request from a management station, the ASC shall store if the interconnect inputs operate at a higher priority than the timebase control.

3.5.2.3.2 Command Timing Plan Requirements

The requirements to select the mode of operation or a timing pattern within the ASC follow.

3.5.2.3.2.1 Activate System Timing Pattern Remotely

Upon request from a management station, the ASC shall allow a management station to select the coordinated timing pattern or operational mode for the ASC while allowing the setting to be overridden by Backup Mode. This allows the ASC to revert to its previous coordinated timing pattern or operational mode in the event that communication from the management station is lost. Valid patterns/modes that can be commanded and defined by this standard are:

a) Standby. Allows the ASC to select the pattern or mode based on the local timebase schedule or interconnect inputs.
b) Pattern Number. Commands the ASC to a specific timing pattern. Timing patterns are identified by an identifier from 1 to 253.
c) Manual Free. Commands the ASC to operate in free mode without coordination.
d) Manual Flash. Commands the ASC to flash based on channel settings as in Automatic Flash.

3.5.2.3.2.2 Control System Reference Point

Upon request from a management station, the ASC shall store the System Reference Point for the called system pattern by defining a point in the system pattern cycle in seconds, from 0 to 998 seconds. This System Reference Point is established to the next System Reference Point. If the System Reference Point is set to 65535, the ASC references the system reference point to the local timebase.

3.5.2.3.3 Control Phases Requirements

The requirements to control the phases within the ASC follow.

3.5.2.3.3.1 Control Phase Omits

Upon request from a management station, the ASC shall store which phases are to be prevented from being active. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.3.2 Control Pedestrian Phase Omits

Upon request from a management station, the ASC shall store which phases are to have their corresponding pedestrian movements prevented from being active. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.3.3 Control Phase Holds

Upon request from a management station, the ASC shall activate/deactivate a hold command for a phase, which is equivalent to energizing the Hold input (See NEMA TS-2, Section 3.5.3.11.1). This setting is cleared when the ASC goes into backup mode.

3.5.2.3.3.4 Control Phase Force Offs

Upon request from a management station, the ASC shall activate/deactivate a Force Off command for a phase, which is equivalent to energizing the Force Off input (See NEMA TS-2, Section 3.5.4.1.1). This setting is cleared when the ASC goes into backup mode.

3.5.2.3.3.5 Control Phase Vehicle Calls

Upon request from a management station, the ASC shall place vehicle calls on a phase. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.3.6 Control Phase Pedestrian Calls

Upon request from a management station, the ASC shall place pedestrian calls on a phase. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.4 Activate Preempt Remotely

Upon request from a management station, the ASC shall manually activate a preempt. If the preemption action has already been started by a preemption input, the ASC keeps that already-started preemption action. When manually activating a preempt, the ASC remains in preemption until it completes the preemption sequence or until the management station removes the preempt.

This setting is cleared when the ASC goes into backup mode.

3.5.2.3.5 Control Ring Requirements

The requirements to activate the ring control functions within the ASC follow.

3.5.2.3.5.1 Control Ring Stop Time

Upon request from a management station, the ASC shall activate/deactivate the stop timing input for a ring. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.5.2 Control Ring Force Offs

Upon request from a management station, the ASC shall activate/deactivate a Force Off input for a ring. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.5.3 Control Ring Maximum 2 Time Settings

Upon request from a management station, the ASC shall store if the Maximum 2 Time setting is enabled for a ring. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.5.4 Control Ring Maximum 3 Time Settings

Upon request from a management station, the ASC shall store if the Maximum 3 Time setting is enabled for a ring. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.5.5 Control Ring Maximum Inhibit Settings

Upon request from a management station, the ASC shall store if the Maximum time setting is inhibited for a ring. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.5.6 Control Ring Pedestrian Recycle Settings

Upon request from a management station, the ASC shall store if the pedestrian recycle setting is active for a ring. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.5.7 Control Ring Red Rest Settings

Upon request from a management station, the ASC shall store if the Red Rest setting is active for a ring. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.5.8 Control Ring Red Clearance Omit Settings

Upon request from a management station, the ASC shall store if the Red Clearance setting is omitted for each ring. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.6 Activate Special Function Remotely

Upon request from a management station, the ASC shall store if a special function, regardless if it is a physical or logical function, is turned on or off. This setting is cleared when the ASC goes into backup mode.

3.5.2.3.7 Remote Manual Control Requirements

The requirements to remotely advance the ASC to the next interval follow.

3.5.2.3.7.1 Enable Remote Manual Control

Upon request from a management station, the ASC shall enable or disable remote manual control mode. While in remote manual control mode, the ASC advances to the next interval only upon receiving an advance command from a management station.

3.5.2.3.7.2 Advance Interval During Remote Manual Control

Upon request from a management station, the ASC shall allow a management station to command the signal controller to advance to the next interval. Under remote manual control mode, the ASC behaves as if the manual control input was active. The ASC will not time phases, such as when using a coordinated timing pattern, but instead will advance to the next interval when remotely commanded to by a management station.

NOTE: NEMA TS-2 does not allow a Yellow Change interval to be terminated by the Interval Advance input.

3.5.2.3.7.3 Configure Manual Control Timeout

Upon request from a management station, the ASC shall store a timeout value, from 1 to 65535 seconds, as a failsafe in case of a loss of communications. When the ASC is in remote manual control mode, the remote manual control timer will decrement once per second until it reaches zero, at which time the ASC will disable remote manual control and revert back to normal signal operation. This forces a management station to continually reset the remote manual control timer to maintain remote manual control.

3.5.2.3.7.4 Enable/Disable Automatic Pedestrian Clearance Setting

Upon request from a management station, the ASC shall set the Pedestrian Clearance interval to be protected from being terminated by an Interval Advance input when Manual Control Enable is active.

3.5.3 Detector Management Requirements

The requirements for managing the detectors of an ASC follow.

3.5.3.1 Manage Detector Configuration Requirements

The requirements to manage the detector configurations of an ASC are defined in the following paragraphs.

3.5.3.1.1 Configure Vehicle Detector Requirements

To manage the traffic-actuated operations of an ASC controller, the ASC shall allow a management system to configure vehicle detectors, including bicycle detectors and transit vehicle detectors. Each detector may be a physical detector (such as a loop detector) or a virtual detector (e.g., via video processing or basic safety messages). The requirements to configure a vehicle detector of an ASC follow.

Pedestrian detectors are managed separately (See Section 3.5.3.1.3 Configure Pedestrian Detector Requirements).

3.5.3.1.1.1 Configure Vehicle Detector Travel Mode

Upon request from a management station, the ASC shall store the travel mode identified for the detector. The travel mode shall be one of vehicle (not otherwise assigned), transit, or bicycle.

3.5.3.1.1.2 Configure Vehicle Detector Description

Upon request from a management station, the ASC shall store a user description for a vehicle detector.

3.5.3.1.1.3 Configure Vehicle Detector Yellow Lock Call Enabled

Upon request from a management station, the ASC shall store if a vehicle detector is instructed to lock a call to the assigned phase if an actuation occurs while the phase is not timing the Green interval. If the Yellow Lock Call and Red Lock Call are both enabled for a given phase, the ASC shall keep the yellow lock call enabled.

3.5.3.1.1.4 Configure Vehicle Detector Red Lock Call Enabled

Upon request from a management station, the ASC shall store if a vehicle detector is instructed to lock a call to the assigned phase if an actuation occurs while the phase is not timing Green or Yellow intervals. If the Yellow Lock Call and Red Lock Call are both enabled for a given phase, the ASC shall disable the red lock call.

3.5.3.1.1.5 Configure Vehicle Detector Passage Enabled

Upon request from a management station, the ASC shall store if the associated phase passage timer remains reset for the duration of a vehicle detector actuation if the phase is in the Green interval.

3.5.3.1.1.6 Configure Vehicle Detector Added Initial Time Enabled

Upon request from a management station, the ASC shall store if detector actuation counts for a vehicle detector are accumulated for use in the added initial calculations. If enabled, counts are accumulated starting at the beginning of the Yellow interval and terminating at the beginning of the Green interval.

3.5.3.1.1.7 Configure Vehicle Detector Queue Enabled

Upon request from a management station, the ASC shall store if the Green interval of the assigned phase for a vehicle detector is extended upon actuation until either a gap occurs or the Green has been active longer than the Vehicle Detector Queue Limit Time.

3.5.3.1.1.8 Configure Vehicle Detector Call Enabled

Upon request from a management station, the ASC shall store if a call is placed for vehicle service upon actuation of a vehicle detector while the phase is not timing the Green interval.

3.5.3.1.1.9 Configure Vehicle Detector Call Phase

Upon request from a management station, the ASC shall store the assigned call phase associated with a vehicle detector. If no phase is assigned, the ASC disables the ability of the detector to call a phase.

3.5.3.1.1.10 Configure Vehicle Detector Switch Phase

Upon request from a management station, the ASC shall store the programmed (switch) phase to which actuation of a vehicle detector is switched when the current (call) phase is Yellow or Red and the programmed (switch) phase is Green. Detector switching allows the detector to call and extend the current phase and send calls to the switch phase once the current phase ends.

3.5.3.1.1.11 Configure Vehicle Detector Delay Time

Upon request from a management station, the ASC shall store the delay time, in tenths of a second, from 0 to 255.0 seconds, before an ASC recognizes an actuation from a vehicle detector when the phase is not Green.

3.5.3.1.1.12 Configure Vehicle Detector Extend Time

Upon request from a management station, the ASC shall store the time, in tenths of a second, from 0 to 25.5 seconds, that an actuation for a vehicle detector is extended from the point of termination, when the phase is Green.

3.5.3.1.1.13 Configure Vehicle Detector Queue Limit Time

Upon request from a management station, the ASC shall store the length of time in seconds, from 0 to 255 seconds, that an actuation from a vehicle queue detector may continue into the Green phase. This time commences when the phase becomes Green and when the time expires, the ASC ignores any associated actuations / detector inputs. The ASC might shorten this time due to other overriding parameters such as Maximum Green time or Force Off commands.

3.5.3.1.1.14 Configure Vehicle Detector No Activity Fault Time

Upon request from a management station, the ASC shall store the time period in minutes, from 0 to 65535 minutes, before the ASC declares the absence of any actuations for a vehicle detector to be a fault and the vehicle detector is classified as failed. The ASC disables the diagnostics for this detector if the No Activity Time value for this vehicle detector is set to zero.

3.5.3.1.1.15 Configure Vehicle Detector Maximum Presence Fault Time

Upon request from a management station, the ASC shall store the time period in minutes, from 0 to 255 minutes, before the ASC declares the presence of a continuous actuation of a vehicle detector to be a fault and the vehicle detector is classified as failed. The ASC disables the diagnostics for this detector if the Maximum Presence Time value for this vehicle detector is set to zero.

3.5.3.1.1.16 Configure Vehicle Detector Erratic Counts

Upon request from a management station, the ASC shall store the number of actuations for a vehicle detector, from 0 to 255 counts per minute, above which the ASC declares the vehicle detector to be a fault and the vehicle detector is classified as failed. The ASC disables the diagnostics for this detector if the Erratic Count value for this vehicle detector is set to zero.

3.5.3.1.1.17 Configure Vehicle Detector Fail Time

Upon request from a management station, the ASC shall store the amount of time, in seconds, that the ASC holds a call for the associated phase during all non-Green intervals for a failed vehicle detector. The ASC places a constant call on the phase (maximum recall) if the vehicle detector fail time is set to the maximum of 255 seconds. The ASC does not place a call on this detector if the Fail Time value for this vehicle detector is set to zero.

3.5.3.1.2 Configure Multiple Vehicle Detector Sets for Actuation

Upon request from a management station, the ASC shall store parameters for multiple sets of vehicle detectors to be used actuations in a manner consistent with the following requirements as selected under User Need 2.5.3.1, Manage Detector Configuration in Table 5 Protocol Requirements List (PRL).

a) 3.5.3.1.1.1 Configure Vehicle Detector Travel Mode
b) 3.5.3.1.1.3 Configure Vehicle Detector Yellow Lock Call Enabled
c) 3.5.3.1.1.4 Configure Vehicle Detector Red Lock Call Enabled
d) 3.5.3.1.1.5 Configure Vehicle Detector Passage Enabled
e) 3.5.3.1.1.6 Configure Vehicle Detector Added Initial Time Enabled
f) 3.5.3.1.1.7 Configure Vehicle Detector Queue Enabled
g) 3.5.3.1.1.8 Configure Vehicle Detector Call Enabled
h) 3.5.3.1.1.9 Configure Vehicle Detector Call Phase
i) 3.5.3.1.1.10 Configure Vehicle Detector Switch Phase
j) 3.5.3.1.1.11 Configure Vehicle Detector Delay Time
k) 3.5.3.1.1.12 Configure Vehicle Detector Extend Time
l) 3.5.3.1.1.13 Configure Vehicle Detector Queue Limit Time
m) 3.5.3.1.1.14 Configure Vehicle Detector No Activity Fault Time
n) 3.5.3.1.1.15 Configure Vehicle Detector Maximum Presence Fault Time
o) 3.5.3.1.1.16 Configure Vehicle Detector Erratic Counts
p) 3.5.3.1.1.17 Configure Vehicle Detector Fail Time

3.5.3.1.3 Configure Pedestrian Detector Requirements

MUTCD (FHWA MUTCD 2023 Edition, Section 4I.05) defines, "Pedestrian detectors may be pushbuttons or passive detection devices. Passive detection devices register the presence of a pedestrian in a position indicative of a desire to cross, without requiring the pedestrian to push a button. Some passive detection devices are capable of tracking the progress of a pedestrian as the pedestrian crosses the roadway for the purpose of extending or shortening the duration of certain pedestrian timing intervals".

The requirements to manage the pedestrian detector configurations of an ASC follow.

3.5.3.1.3.1 Configure Pedestrian Detector Description

Upon request from a management station, the ASC shall store a user description for a pedestrian detector.

3.5.3.1.3.2 Configure Pedestrian Detector Call Phase

Upon request from a management station, the ASC shall store the assigned phase associated with a pedestrian detector. If no phase is assigned, the ASC disables the ability of the pedestrian detector to call a phase.

3.5.3.1.3.3 Configure Pedestrian Detector No Activity Fault Time

Upon request from a management station, the ASC shall store the time period in minutes, from 0 to 65535 minutes, when the ASC declares the absence of any actuations for a pedestrian detector to be a fault, and the pedestrian detector is classified as failed. The ASC disables the diagnostics for this detector, if the No Activity Time for this pedestrian detector is set to zero.

3.5.3.1.3.4 Configure Pedestrian Detector Maximum Presence Fault Time

Upon request from a management station, the ASC shall store the time period in minutes, from 0 to 255 minutes, when the ASC will declare the presence of a continuous actuation of a pedestrian detector to be a fault, and the pedestrian detector is classified as failed. The ASC disables the diagnostics for this detector if the Maximum Presence Time for this pedestrian detector is set to zero.

3.5.3.1.3.5 Configure Pedestrian Detector Erratic Counts

Upon request from a management station, the ASC shall store the number of actuations for a pedestrian detector, from 0 to 255 counts per minute, above which the ASC declares the pedestrian detector to be a fault and the pedestrian detector is classified as failed. The ASC disables the diagnostics for this detector, if the Erratic Count value for this pedestrian detector is set to zero.

3.5.3.1.3.6 Configure Pedestrian Detector Non-Lock Calls

Upon request from a management station, the ASC shall store if a pedestrian detector is used to place non-locked calls for pedestrian timings.

3.5.3.1.3.7 Configure Pedestrian Detector for Presence Detection

Upon request from a management station, the ASC shall store if a pedestrian detector is used to detect the presence of a pedestrian in the pedestrian crosswalk instead of detecting a pedestrian call for service.

3.5.3.1.3.8 Configure Pedestrian Detector for Delayed Walk

Upon request from a management station, the ASC shall store if a pedestrian detector is used to place a call for pedestrian service from the assigned phase with the WALK indication delayed (See 3.5.2.1.2.1.46 Configure Pedestrian Phase Delayed Walk Time). If this setting is disabled, there is no delay for the walk indication.

3.5.3.1.3.9 Configure Pedestrian Detector for Advanced Walk

Upon request from a management station, the ASC shall store if a pedestrian detector is used to place a call for pedestrian service from the assigned phase with a leading pedestrian interval configured (See 3.5.2.1.2.1.45 Configure Pedestrian Phase Advanced Walk Time). If this setting is disabled, the assigned phase does not time a leading pedestrian interval unless Pedestrian Recall is enabled.

3.5.3.1.4 Configure Multiple Pedestrian Detector Sets for Actuation

Upon request from a management station, the ASC shall store parameters for multiple sets of pedestrian detectors to be used actuations in a manner consistent with the following requirements as selected under User Need 2.5.3.1, Manage Detector Configuration in Table 5 Protocol Requirements List (PRL).

a) 3.5.2.1.13.1.4 Configure Pedestrian Detector for Alternate Pedestrian Timing
b) 3.5.3.1.3.2 Configure Pedestrian Detector Call Phase
c) 3.5.3.1.3.3 Configure Pedestrian Detector No Activity Fault Time
d) 3.5.3.1.3.4 Configure Pedestrian Detector Maximum Presence Fault Time
e) 3.5.3.1.3.5 Configure Pedestrian Detector Erratic Counts
f) 3.5.3.1.3.6 Configure Pedestrian Detector Non-Lock Calls
g) 3.5.3.1.3.7 Configure Pedestrian Detector for Presence Detection
h) 3.5.3.1.3.8 Configure Pedestrian Detector for Delayed Walk
i) 3.5.3.1.3.9 Configure Pedestrian Detector for Advanced Walk

3.5.3.1.5 Retrieve Detector Configuration Requirements

The requirements to retrieve the detector configuration settings including vehicle and pedestrian detectors from the ASC follow.

3.5.3.1.5.1 Determine Maximum Number of Vehicle Detectors

Upon request from a management station, the ASC shall return the maximum number of vehicle detectors that can be configured within the ASC.

3.5.3.1.5.2 Determine Maximum Number of Vehicle Detector Sets

Upon request from a management station, the ASC shall return the maximum number of vehicle detector sets that can be configured within the ASC.

3.5.3.1.5.3 Determine Maximum Number of Pedestrian Detectors

Upon request from a management station, the ASC shall return the maximum number of pedestrian detectors that can be configured within the ASC.

3.5.3.1.5.4 Determine Maximum Number of Pedestrian Detector Sets

Upon request from a management station, the ASC shall return the maximum number of vehicle detector sets that can be configured within the ASC.

3.5.3.2 Retrieve Detector Status Requirements

The requirements to monitor the status of the detectors connected to an ASC controller follow.

3.5.3.2.1 Monitor Active Vehicle Detector Actuations

Upon request from a management station, the ASC shall return which vehicle detectors are currently actuated (vehicle presence detected). An actuation may be placed remotely, such as through an external detection of a vehicle or by a CV Application Process.

3.5.3.2.2 Monitor Active Pedestrian Detector Actuations

Upon request from a management station, the ASC shall return which pedestrian detectors are currently actuated. An actuation may be placed remotely, such as by automatic detection of a pedestrian, by pressing a push button, or by a CV Application Process.

3.5.3.3 Retrieve Detector Health Requirements

The requirements to monitor the health status of the detectors connected to an ASC controller follow.

3.5.3.3.1 Retrieve Vehicle Detector Health Requirements

The requirements to monitor the health status of vehicle detectors connected to an ASC controller follow.

3.5.3.3.1.1 Monitor Vehicle Detector Alarm Status

Upon request from a management station, the ASC shall return which vehicle detectors currently have an active alarm. The ASC clears any alarm that is not currently active.

3.5.3.3.1.2 Monitor Vehicle Detector Faults from Controller

Upon request from a management station, the ASC shall return if a vehicle detector has been flagged as non-operational / failed and why. Valid reasons defined by this standard are:

a) No Activity Fault. There has been no actuation for a user-defined period of time (See Section 3.5.3.1.1.14 Configure Vehicle Detector No Activity Fault Time).
b) Maximum Presence Fault. There has been a continuous actuation for a user-defined period of time (See Section 3.5.3.1.1.15 Configure Vehicle Detector Maximum Presence Fault Time).
c) Erratic Output Fault. There has been a higher number of actuations per minute than a user-defined threshold (See Section 3.5.3.1.1.16 Configure Vehicle Detector Erratic Counts)
d) Communications Fault. Communications with a vehicle detector have failed.
e) Configuration Fault. A vehicle detector is assigned but not supported.
f) Other Fault. A vehicle detector has failed due to some other cause not defined by this standard.

These faults are detected by the ASC.

3.5.3.3.1.3 Monitor Vehicle Detector Faults from Detector

Upon request from a management station, the ASC shall return if a vehicle detector has been flagged by the detector as non-operational / failed and why. Valid reasons defined by this standard are:

a) Watchdog Fault. A vehicle detector has reported a watchdog error.
b) Open Loop Fault. A vehicle detector has reported an open loop / broken wire.
c) Shorted Loop Fault. A vehicle detector has reported a shorted loop wire.
d) Excessive Change Faults. A vehicle detector has reported an excessive inductance change.
e) Other Fault. A vehicle detector has failed due to some other cause not defined by this standard.

These faults are reported to the ASC from a vehicle detector and are typically from loop detectors.

3.5.3.3.2 Retrieve Pedestrian Detector Health Requirements

The requirements to monitor the health status of pedestrian detectors connected to an ASC follow.

3.5.3.3.2.1 Monitor Pedestrian Detector Alarm Status

Upon request from a management station, the ASC shall return which pedestrian detectors have an active alarm. The ASC clears any alarm that is not currently active.

3.5.3.3.2.2 Monitor Pedestrian Detector Faults

Upon request from a management station, the ASC shall return if a vehicle detector has been flagged as non-operational / failed and why. Valid reasons defined by this standard are:

a) No Activity Fault. There has been no actuation for a user-defined period of time (See Section 3.5.3.1.3.3 Configure Pedestrian Detector No Activity Fault Time).
b) Maximum Presence Fault. There has been a continuous actuation for a user-defined period of time (See 3.5.3.1.3.4 Configure Pedestrian Detector Maximum Presence Fault Time).
c) Erratic Output Fault. There has been a higher number of actuations per minute than a user-defined threshold (See Section 3.5.3.1.3.5 Configure Pedestrian Detector Erratic Counts).
d) Communications Fault. Communications with a pedestrian detector have failed.
e) Configuration Fault. A pedestrian detector is assigned not supported.
f) Other Fault. A pedestrian detector has failed due to some other cause not defined by this standard.

These faults are detected by the ASC.

3.5.3.4 Control Detector Requirements

The requirements to control detectors connected to an ASC follow.

3.5.3.4.1 Control Vehicle Detector Reset

Upon request from a management station, the ASC shall reset a vehicle detector. The ASC automatically returns a detector reset to a non-reset state after the ASC has executed the reset command.

3.5.3.4.2 Control Pedestrian Detector Reset

Upon request from a management station, the ASC shall reset a pedestrian detector. The ASC automatically returns a detector reset to a non-reset state after the ASC has executed the reset command.

3.5.3.4.3 Control Detector Diagnostic Reset

Upon request from a management station, the ASC shall clear all current detector faults and reset all diagnostic counters for all vehicle and pedestrian detectors. The diagnostic thresholds remain unchanged.

3.5.3.4.4 Control Vehicle Detector Actuation

Upon request from a management station, the ASC shall place a remote actuation on a vehicle detector. This setting is cleared when the ASC goes into Backup mode.

3.5.3.4.5 Control Pedestrian Detector Actuation

Upon request from a management station, the ASC shall place a remote actuation on a pedestrian detector. This setting is cleared when the ASC goes into Backup mode.

3.5.3.5 Manage Detector Data Collection Requirements

The requirements to manage the data obtainable from the detectors connected to an ASC through contact closures in the cabinet follow.

Requirements to obtain data from detectors through a network interface are found in NTCIP 1209.

3.5.3.5.1 Monitor Vehicle Detector Data Requirements

The requirements to retrieve the data collected from vehicle detectors stored within the ASC follow.

3.5.3.5.1.1 Monitor Vehicle Detector Data Sequence

Upon request from a management station, the ASC shall return a sequence number, from 0 to 255, for detector data reported. The ASC increments the detector data sequence number by 1 at the end of the sample period. The sequence number is used by the management station to determine if the detector data reported is duplicated or if there is detector data missing.

3.5.3.5.1.2 Monitor Vehicle Volume Data

Upon request from a management station, the ASC shall return the vehicle count, in numbers of vehicles from 0 to 65535 vehicles, measured by each of those vehicle detectors assigned to collect volume data during the sample period. The ASC resets the volume count number at the end of the sample period and restarts the count at the beginning of the new sample period.

This is a measurement of vehicles physically detected. This does not include actuations placed remotely, such as from a management station or from a CV Application Process.

3.5.3.5.1.3 Monitor Vehicle Occupancy Data

Upon request from a management station, the ASC shall return occupancy rates in 0.5% increments, from 0 to 100%, from those detectors assigned to collect occupancy data during the sample period. The ASC resets the occupancy rate number at the end of the sample period and restarts the occupancy calculation at the beginning of each new sample period.

3.5.3.5.1.4 Monitor Vehicle Average Speed

Upon request from a management station, the ASC shall return the average speed, in kilometers per hour, measured by each of those vehicle detectors assigned to collect average speed data during the sample period. The ASC resets the average speed value at the end of the sample period. Valid average speed values are from 0 to 255 kilometers per hour.

3.5.3.5.1.5 Monitor Vehicle Detector Data Sample Time

Upon request from a management station, the ASC shall return the end time in controller local time of the vehicle detector data collection period (sample period).

3.5.3.5.1.6 Monitor Vehicle Detector Data Sample Duration

Upon request from a management station, the ASC shall return the duration of the data collection period in effect, from 1 to 3600 seconds. There are various ways to configure the data collection period (i.e., a duration specifically set by the user or a duration set to that of the cycle time). This requirement refers to the sample period that is in effect when the data is collected.

3.5.3.5.2 Monitor Pedestrian Detector Data Requirements

The requirements to retrieve the data collected from pedestrian detectors stored within the ASC follow.

3.5.3.5.2.1 Monitor Pedestrian Detector Data Sequence

Upon request from a management station, the ASC shall return a sequence number, from 0 to 255, for pedestrian detector data reported. The ASC increments the pedestrian detector data sequence number by 1 at the end of the sample period. The sequence number is used by the management station to determine if the pedestrian detector data reported is duplicated or if there is pedestrian detector data missing.

3.5.3.5.2.2 Monitor Pedestrian Counts

Upon request from a management station, the ASC shall return the number of pedestrians currently detected within the detection zone during the defined sample period.

3.5.3.5.2.3 Monitor Pedestrian Detector Actuations

Upon request from a management station, the ASC shall return the number of pedestrian actuations during the defined sample period.

3.5.3.5.2.4 Monitor Pedestrian Services

Upon request from a management station, the ASC shall return the number of pedestrian services (the number of times the pedestrian transitioned from don't walk to walk) during the defined sample period.

3.5.3.5.2.5 Monitor Pedestrian Detector Data Sample Time

Upon request from a management station, the ASC shall return the end time in controller local time of the pedestrian detector data collection period (sample period).

3.5.3.5.2.6 Monitor Pedestrian Detector Data Sample Duration

Upon request from a management station, the ASC shall return the duration of the data collection period in effect, from 1 to 3600 seconds, for the pedestrian detectors. There are various ways to configure the data collection period (i.e., a duration specifically set by the user, a duration set to the vehicle data collection period, or a duration set to that of the cycle time). This requirement refers to the sample period that is in effect when the data is collected.

3.5.3.5.3 Configure Detector Data Collection Requirements

The requirements to configure the data collection parameters for vehicle detectors connected to an ASC through contact closure communications interface follow.

3.5.3.5.3.1 Configure Vehicle Detector Data Sample Period

Upon request from a management station, the ASC shall store the sample period for collecting vehicle detector data as follows: a value of 0 indicates that no sampling is to be performed, a value of 1 to 3600 is the number of seconds for the sample period, and a value of 65535 indicates that the sample period is equal to the cycle length if the ASC is running coordination.

The ASC stores the collected detector data in the ASC's database at the end of the sample period and resets the detector data timer.

3.5.3.5.3.2 Configure Pedestrian Detector Data Sample Period

Upon request from a management station, the ASC shall store the sample period for collecting pedestrian detector data as follows: a value of 0 indicates that no sampling is to be performed, a value of 1 to 3600 is the number of seconds for the sample period, a value of 65534 indicates that the sample period should be the same as the sample period for the vehicle detectors, and a value of 65535 indicates that the sample period is equal to the cycle length if the ASC is running coordination.

The ASC stores the collected detector data in the ASC's database at the end of the sample period and resets the detector data timer.

3.5.3.5.3.3 Configure Vehicle Speed Detectors

Upon request from a management station, the ASC shall store if a vehicle detector is instructed to collect speed data.

3.5.3.5.3.4 Configure Single Detector Speed Mode

Upon request from a management station, the ASC shall store the single detector speed mode. It identifies how the ASC should calculate speed without a paired detector. If the speed detector is a paired detector, this option is used when there is an error on one or more of the paired detectors.

3.5.3.5.3.5 Configure Paired Detector

Upon request from a management station, the ASC shall store the vehicle detector identifier of the paired detector. A value of 0 is the default, indicating that the detector is not paired. Paired detectors may be used for calculating speed, wrong way travel, or other conditions.

Note: It is the responsibility of the implementers of this feature to ensure that the detector pairs make logical sense, they are located in the same lane, the paired detectors reference each other, and they are properly identified detector placements.

3.5.3.5.3.6 Configure Paired Detector Placement

Upon request from a management station, the ASC shall store whether a paired detector is the leading or trailing detector of the detector pair.

3.5.3.5.3.7 Configure Paired Detector Spacing

Upon request from a management station, the ASC shall store the distance between the detector pair measured from leading edge to leading edge of each of the two vehicle detectors measured in centimeters from 0 to 65,535 centimeters.

3.5.3.5.3.8 Configure Average Vehicle Length

Upon request from a management station, the ASC shall store the average vehicle length for the detection zone in a range from 1 to 4000 centimeters.

3.5.3.5.3.9 Configure Vehicle Detection Zone Length

Upon request from a management station, the ASC shall store the vehicle detector's detection zone length measured from leading edge to trailing edge of the detection zone in centimeters from 0 to 4000 centimeters.

3.5.3.5.4 Configure Multiple Vehicle Detector Sets for Data Collection

Upon request from a management station, the ASC shall store parameters for multiple sets of vehicle detectors to be used for data collection over a serial communications interface in a manner consistent with the following requirements as selected under User Need 2.5.3.5, Manage Detector Data Collection in Table 5 Protocol Requirements List (PRL).

a) 3.5.3.5.3.3 Configure Vehicle Speed Detectors
b) 3.5.3.5.3.4 Configure Single Detector Speed Mode
c) 3.5.3.5.3.8 Configure Average Vehicle Length

3.5.4 Connected Vehicles Interface Management

The requirements for managing the interfaces for an ASC in a connected vehicle environment are categorized as follows:

a) Manage ASC – RSU Interface Requirements
b) Manage ASC Process Requirements
c) Manage ASC – CV Application Process Requirements
d) Manage ASC – ECLA Interface Requirements

The requirements for managing the connected vehicles interface of an ASC follow.

3.5.4.1 Manage ASC - RSU Interface Requirements

The following requirements are to manage connecting an RSU to the ASC. These requirements may also apply to an intermediate device between the RSU and the ASC.

3.5.4.1.1 Configure ASC Communications Port for RSU

Upon request from a management station, the ASC shall store which communications port is used to exchange data with an RSU.

3.5.4.1.2 Configure Logical RSU Ports and Address

Upon request from a management station, the ASC shall store the name and network address of each RSU that the ASC will exchange data with. An ASC may communicate with up to 16 RSUs as part of the connected vehicle environment.

3.5.4.1.3 Configure RSU Interface Polling Period

Upon request from a management station, the ASC shall store the polling period, from 1 to 65535 milliseconds, for the ASC to exchange connected vehicle data with an RSU.

3.5.4.1.4 Configure RSU Interface Watchdog

Upon request from a management station, the ASC shall store the maximum time duration, in milliseconds, for an RSU watchdog timer in the ASC. The RSU watchdog timer is used to track activity across a RSU interface. If no activity is detected across the RSU interface for a period longer than the maximum time duration a RSU watchdog no activity fault is reported. The RSU watchdog timer is a value from 1 to 65535 milliseconds and includes a value to disable the watchdog timer.

3.5.4.1.5 Monitor RSU Interface Watchdog Timer

Upon request from a management station, the ASC shall return the RSU watchdog time, from 0 to 65535 milliseconds, for a specific logical RSU port as provided in the request. The RSU watchdog time represents the amount of time that has elapsed since activity was last detected across the specified logical RSU port interface.

3.5.4.1.6 Monitor RSU Interface Watchdog Alarm

Upon request from a management station, an ASC shall return an alarm if any RSU watchdog no activity timer fault is detected. This fault indicates that no activity has been detected across any the RSU interface for a period longer than a stored threshold.

3.5.4.2 Manage ASC Process Requirements

The management station for these requirements is NOT an RSU, it represents a computing device at a traffic management center, such as a traffic management system, or could be a field maintenance laptop. The requirements to manage the data exchanges between a management station and the ASC are:

a) Manage Signal Phase and Timing Requirements
b) Manage Connected Vehicle Detector Requirements

The requirements for a management station to manage the ASC process for a connected vehicle environment follow.

3.5.4.2.1 Manage Signal Phase and Timing Requirements

Some of the key applications that have been developed for the connected vehicle environment are related to intersection safety. For signalized intersections, this involves an RSU broadcasting SPaT (Signal Phase and Timing) messages that are processed by a CV Application Process, as defined by SAE J2735, to connected vehicles in the vicinity. Such a CV Application Process may reside on the RSU or on the ASC directly (See 2.3.3 ASC Characteristics – Connected Vehicle Interface).

The source of the SPaT data broadcasted by an RSU comes from a CV Application Process that receives data from the ASC Process. However, a management station, such as one in a traffic management center, needs to monitor what data is being broadcasted to connected vehicles. The requirements that allow a management station to configure and retrieve signal phase and timing data from an ASC follow.

3.5.4.2.1.1 Enable Signal Phase and Timing Data

Upon request from a management station, the ASC shall store if the controller is to generate signal phase and timing data for the intersection(s).

3.5.4.2.1.2 Retrieve Signal Phase and Timing Generation Time

Upon request from a management station, the ASC shall return the time when the signal phase and timing data was generated by the ASC. This timestamp is represented in hours, minutes, seconds and milliseconds of the time of day. This requirement is for testing purposes only. This requirement allows the operator at a TMC to view when the controller generates the SPaT data. Note it is not used by the RSU to generate the SAE J2735 SPaT message, though it may be a need in the future. The timestamp in the SAE J2735 SPaT message is the time the message is generated and signed.

3.5.4.2.1.3 Exchange Movement Status Requirements

The SPaT message that is broadcasted by an RSU to connected vehicles includes information about what vehicle (and pedestrian) movements are permitted at a signalized intersection. To provide this information the RSU needs movement data from the ASC. These requirements allow a management station to monitor the movement data that an ASC is exchanging with the CV Application Process. The requirements to retrieve the movement data that an ASC is exchanging with a CV Application Process are defined as follows.

3.5.4.2.1.3.1 Retrieve Movement Timing Requirements

The requirements to provide the timing of a movement at the intersection are defined as follows.

NOTE: it is required that an ASC system time be synchronized with an accurate and reliable UTC time source before providing these time points.

3.5.4.2.1.3.1.1 Monitor Movement Minimum End Time

Upon request from a management station, the ASC shall return the time point of earliest end time for the current movement state (e.g., at the end of a permissive green or at the end of a permissive yellow) of a movement at an intersection. If the duration of the current state of a movement is fixed, this value indicates the end time (and is equal to the Movement Maximum End Time). This value can be viewed as the earliest possible time point at which the current interval could change, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour.

3.5.4.2.1.3.1.2 Monitor Movement Maximum End Time

Upon request from a management station, the ASC shall return the latest possible end time point of the current movement state (e.g., at the end of a protected green or end of a steady red) of a movement at an intersection. This value can be viewed as the latest possible time point at which the current interval could change, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour. If the duration of the current state of a movement is fixed, this value indicates the end time (and is equal to the Movement Minimum End Time).

3.5.4.2.1.3.1.3 Monitor Movement Likely End Time

Upon request from a management station, the ASC shall return the time point when the current movement state of a movement will most likely end (e.g., at the end of a protected green or end of a steady red) at an intersection. The likely end time point may be predicted based on data available to the ASC. The time point is measured in tenths of a second in the current or next hour. A value of undefined is returned when the time point cannot be estimated with a high level of confidence.

3.5.4.2.1.3.1.4 Monitor Movement Likely End Time Confidence

Upon request from a management station, the ASC shall return the statistical confidence that the reported likely end time for the current movement state (e.g., at the end of a protected green or end of a permissive clearance time) of a movement at an intersection is accurate. The confidence value is measured as a probability class, as defined by DE_TimeIntervalConfidence in SAE J2735.

3.5.4.2.1.3.1.5 Monitor Movement Next Occurrence

Upon request from a management station, the ASC shall return the estimated time point when a movement at an intersection is next allowed to proceed (i.e., the movement phase state will be permissive-Movement-Allowed or protected-Movement-Allowed). The time point is measured in tenths of a second in the current or next hour. This value can be viewed as the estimated time point at which the movement is next allowed to proceed, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. A value of undefined is returned when the time point cannot be estimated with a high level of confidence. This requirement is used to support ECO-driving applications.

3.5.4.2.1.3.1.6 Monitor Movement Start Time

Upon request from a management station, the ASC shall return a value of unknown for the start (time) of the current movement state.

Note: Start time is always a future time (See CTI 4501, No Current Movement State Start Time). Thus, the start time of the current interval is always unknown.

3.5.4.2.1.3.1.7 Monitor Next Movement Minimum End Time

Upon request from a management station, the ASC shall return the time point of the earliest possible end of the movement state immediately after the current movement state at the intersection. If the end time of the current movement state is known (maximum end time equals minimum end time) and the interval duration of the next movement state is fixed, this value indicates the end time point of the next movement state. This value can be viewed as the earliest possible time point at which the next interval could end, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour. If the next movement state is unknown, the time point of the earliest possible end time is also unknown.

NOTE: The minimum end time of the next movement will be the minimum end time of the current movement plus the minimum amount of time that the next movement state could time.

3.5.4.2.1.3.1.8 Monitor Next Movement Maximum End Time

Upon request from a management station, the ASC shall return the time point of the latest possible end of the movement state immediately after the current movement state at the intersection. This value can be viewed as the latest possible time point at which the next movement state could end, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour. If the next movement state is unknown, the time point of the latest possible end time is also unknown. When the end time of the current movement state is known (maximum end time equals minimum end time) and the duration of the next movement state is fixed, such as when the ASC is operating in fixed time or the yellow interval duration, the minimum end time of the next movement state will equal the maximum end time of the next movement state.

3.5.4.2.1.3.1.9 Monitor Next Movement Start Time

Upon request from a management station, the ASC return the time point, in tenths of a second in the current or next hour, of the start time of the next (future) movement state to follow the current movement state for a signal group. If the start time is unknown, a value of unknown is used. If the next movement state is unknown, the start time will also be unknown. If the next movement state is known, the start time is equal to the movement minimum end time of the current interval.

3.5.4.2.1.3.1.10 Determine Maximum Number of Movement Events

Upon request from a management station, the ASC return the number of movement events for each signal group supported. CTI 4501 requires that at least two movement events be supported – the first movement event represents the current interval and the second movement event represents the next interval (after the current interval). The SAE J2735 SPaT message supports up to 16 movement events for each signal group.

3.5.4.2.1.3.2 Configure Movement Assistance Requirements

The SPaT message in SAE J2735 can also provide potential pedestrian or bicyclist conflicts and queuing information to travelers. The requirements to configure detectors to provide this information to travelers wishing to traverse through the intersection are defined as follows.

3.5.4.2.1.3.2.1 Configure Queue Detectors for Movement Assistance

Upon request from a management station, the ASC shall store the identifiers of the vehicle detectors that provide queue information for a specific movement through the intersection. This queue information, measured in meters, is provided so connected vehicles are aware of how many vehicles are queued, if any, for a specific movement through the intersection.

3.5.4.2.1.3.2.2 Configure Pedestrian Detectors for Movement Conflict Assistance

Upon request from a management station, the ASC shall store the identifiers of the pedestrian (presence) detectors indicating the potential presence of pedestrians that conflict with a specific vehicle movement through the intersection. This information is provided so connected vehicles are aware of the potential presence of a pedestrian may conflict with its movement through the intersection.

3.5.4.2.1.3.2.3 Configure Bicycle Detectors for Movement Conflict Assistance

Upon request from a management station, the ASC shall store the identifiers of the detectors that determine the presence of bicyclists that conflict with a specific vehicle movement through the intersection. This information is provided so connected vehicles are aware of the potential presence of a bicyclist may conflict with its movement through the intersection.

3.5.4.2.1.3.3 Retrieve Movement Assistance Requirements

The requirements to provide potential pedestrian or bicyclist conflicts and queuing information to assist connected vehicles traversing through the intersection are defined as follows.

3.5.4.2.1.3.3.1 Monitor Lane Connection Queue Length

Upon request from a management station, the ASC shall return the distance, in meters, from the stop line of the approach movement to the back edge of the last vehicle in the queue, as measured along the center line of the lane for a specific movement maneuver through the intersection. Valid values are 0 to 10000 meters, where 0 indicates no queue or the queue distance is unknown, and 10000 represents all distance ≥ 10000 meters. The detectors that provide this queue information is configured in Section 3.5.4.2.1.3.2.1.

3.5.4.2.1.3.3.2 Monitor Lane Connection Vulnerable Road User Detection

Upon request from a management station, the ASC shall return if any conflicting pedestrians or bicycles are detected for a specific movement maneuver through the intersection. This value is either on or off, with off indicating a high certainty that there is no pedestrian or bicycle present. The presence inputs that indicate if a conflicting pedestrian or bicyclist may be present is configured in Sections 3.5.4.2.1.3.2.2 and 3.5.4.2.1.3.2.3.

3.5.4.2.1.3.4 Manage Advisory Speed Requirements

The SPaT message in SAE J2735 can also provide speed advisories for specific movements and specific vehicle types. The requirements to provide advisory speed information for a movement through the intersection are defined as follows.

3.5.4.2.1.3.4.1 Configure Advisory Speed Type

Upon request from a management station, the ASC shall store the type of speed advisory for a specific movement traversing the intersection. Valid types of speed advisories are defined by DE_AdvisorySpeedType in SAE J2735. Speed advisories may also be configured for specific vehicle types as defined in the MAP message for the intersection.

3.5.4.2.1.3.4.2 Configure Advisory Speed

Upon request from a management station, the ASC shall store the advisory speed, in tenths of a meter per second, provided for a specific movement traversing the intersection. Speed advisories may be configured for specific advisory speed types (See 3.5.4.2.1.3.4.1) or vehicle types as defined in the MAP message for the intersection.

3.5.4.2.1.3.4.3 Configure Advisory Speed Zone

Upon request from a management station, the ASC shall store the distance, in meters, upstream from the stop bar that a speed advisory is recommended for a movement traversing the intersection. A value of 10000 indicates that the distance is 10,000 meters or greater. A value of 0 represents unknown.

3.5.4.2.1.3.4.4 Configure Advisory Speed Vehicle Type

Upon request from a management station, the ASC shall store the vehicle type that a speed advisory is recommended for a specific movement traversing the intersection. The vehicle type(s) is identified in the associated MAP message for the intersection. If no vehicle type is identified, then the advisory speed applies to all vehicles.

3.5.4.2.1.3.5 Monitor Movement State

Upon request from a management station, the ASC shall return the current movement phase state of a movement at the intersection. The valid states are defined by DE_MovementPhaseState in SAE J2735. If the movement state is not known with a high level of confidence, a value of unavailable is used.

3.5.4.2.1.3.6 Monitor Next Movement State

Upon request from a management station, the ASC shall return the movement state immediately after the current movement state of a movement at the intersection. The valid states are defined by DE_MovementPhaseState in SAE J2735. If the next movement state is not known with a high level of confidence, a value of unavailable is used.

3.5.4.2.1.3.7 Monitor Movement Status

Upon request from a management station, the ASC shall return the movement data containing what vehicle (or pedestrian) movements are permitted and when at an intersection in a compressed manner. The connected vehicle environment is expected to have limitations in the data rates and data capacity. This requirement allows the ASC to group sets of data so that the data can be transmitted more efficiently.

3.5.4.2.1.4 Manage Enabled Lane Requirements

The SPaT message in SAE J2735 can also indicate to travelers traversing across the intersection which revocable lanes at the intersection are currently active (enabled). Each lane defined for a roadway geometry plan can be defined as a revocable lane—that is, the lane is not always active for a specific use.

For example, a shoulder lane may be used by vehicles during rush hours and closed to vehicle traffic during all other times. In the roadway geometry (MAP) plan for the intersection, that shoulder lane can be defined as a vehicle lane and as revocable. During rush hours, the SPAT message would then indicate that the shoulder lane is active (Enabled) by including the lane identifier (of the shoulder lane). During non-rush hours, the SPAT message would not include the lane identifier of the shoulder lane, indicating that the shoulder lane is not active (enabled).

The requirements to configure and command enabled (revocable) lanes are defined as follows.

3.5.4.2.1.4.1 Configure Concurrent Enabled Lanes

Upon request from a management station, the ASC shall store what revocable lanes are allowed to be active (enabled) concurrently. This requirement allows the management station to set which revocable lane(s) may be active (enabled) at the same time, thereby preventing the enabling of conflicting revocable lanes.

3.5.4.2.1.4.2 Configure Enabled Lanes by Time of Day

Upon request from a management station, the ASC shall store if a revocable lane is active (enabled) or inactive based on a time base schedule. This requirement allows a management station to set a group of enabled lanes based on a schedule.

3.5.4.2.1.4.3 Determine Lanes Enabled

Upon request from a management station, the ASC shall return the revocable lanes that are currently enabled (active).

3.5.4.2.1.4.4 Command Enabled Lanes

Upon request from a management station, the ASC shall store if a set of revocable lanes is active (enabled) or inactive. This requirement allows a management station to remotely command if the signal phase and timing data provided to a CV Application Process to indicate which revocable lane(s) are enabled or not. The set of revocable lanes is defined in Section 3.5.4.2.1.4.1, Configure Concurrent Enabled Lanes. This command may override the set of enabled lanes in the schedule.

3.5.4.2.1.5 Enable Signal Phase and Timing Data Exchange

Upon request from a management station, the ASC shall store if the controller unit can exchange signal phase and timing data for the intersection(s) with an RSU port. An ASC may provide SPAT data to more than one RSU (or CV Application Process). This requirement allows a management station to control which RSU port(s) can the ASC share SPAT data with.

3.5.4.2.1.6 Configure Road Authority Identifier

Upon request from a management station, the ASC shall store the unique identifier of the agency that operates and/or maintains the intersection or the ASC. The unique identifier is an object identifier as defined by DF_RoadAuthorityID in SAE J2735. In the US, this object identifier allows use consisting of the Geographic Names Information System (GNIS) codes maintained by the US Geological Survey (USGS).

3.5.4.2.1.7 Retrieve Signal Phase and Timing Intersection Status Requirements

Upon request from a management station, the ASC shall return the status of the ASC as part of the signal phase and timing data broadcasted to connected devices. The intersection status values are defined by DE_IntersectionStatusObject in SAE J2735.

3.5.4.2.1.7.1 Monitor Manual Control Indication

Upon request from a management station, the ASC shall return if the intersection is operating under manual control. Manual Control indicates to the receiver of this information that the controller is "not in control", thus the controller generally does not know when the movement phase state will change. The ASC is under manual control when it receives a Manual Control Enabled (MCE) command or Manual Advance input (e.g., from a police pickle). Whether the ASC is under manual control is updated accordingly to be consistent with the current set of possible dwell/advance options. The Minimum End Time and Maximum End Time are adjusted accordingly.

3.5.4.2.1.7.2 Monitor Stop Indication

Upon request from a management station, the ASC shall return if the intersection is operating under stop time. Stop Time indicates to the receiver of this information that all counting/timing by the controller has stopped, thus most TimeChangeDetails are unknown. The ASC is under stop time when it receives a stop time input. The active (frozen) controller state on the affected (stopped) ring remains the basis to determine the MovementPhaseState for each movement and the TimeChangeDetails. When in stop time, the stop time duration is unknown, however, the TimeChangeDetails will remain consistent to the current phase timers. If the form of stop time input allows continuation of countdown timers for clearance intervals or other timers, the TimeChangeDetails are adjusted accordingly.

3.5.4.2.1.7.3 Monitor Failure Flash Indication

Upon request from a management station, the ASC shall return if the intersection is in a failure flash condition. Stop Time indicates to the receiver of this information that all counting/timing by the controller has stopped, thus most TimeChangeDetails are unknown. A failure flash condition is any type of flash that must be terminated by a source external to the controller. There are three types of failure flash.

- "Monitor Flash" is a Flash controlled by the monitor in the following two scenarios:
  - When resuming operation after a power loss or interruption, the monitor keeps the Flash Bus energized for a minimum of 6 seconds before energizing the Signal Bus and transferring control to the controller.
  - When a fault is detected by the monitor, it energizes the Flash Bus until either the fault is cleared by pressing the Reset button on the monitor (latching fault) or, with certain types of faults, when the condition that caused the fault is no longer present (non-latching fault).
- "Fault Monitor Flash" occurs when the controller detects an anomaly in the TSC Infrastructure, such as a mismatch between its copy of the permissive channels and the monitor's copy of the permissive channels.
- "Local Flash" is a flash controlled by human-operated switches in the cabinet, typically labeled "AUTO/FLASH", that is used by technicians to flash the signals when performing maintenance on the controller (Tech Flash) or by police during unusual traffic conditions or situations (Police Flash).

The end of Failure Flash is indeterminate because the ASC does not know when the monitor is going to be reset or when the flash switch is going to be moved from "FLASH" to "AUTO". Therefore, the TimeChangeDetails during Failure Flash cannot be supplied.

> Note: It is more important for SPaT to match the roadway signalization rather than internal controller logic. Some cabinet variations do not have flash sense inputs. The ASC Working Group recommends agencies to ensure flash sense inputs are mapped into the ASC. ASCs that cannot determine if cabinet flash is active are PROHIBITED from providing any SPaT data to the RSU. See Section F.3.3.4.3 for additional notes about a Failure Flash Indication.

3.5.4.2.1.7.4 Monitor Preemption Operation Indication

Upon request from a management station, the ASC shall return if the intersection is operating in Active Preemption Control. Preempt Control indicates to the receiver of this information that the controller is actively servicing a preemption request. The TimeChangeDetails are generated under assumption that any inactive preemption, NTCIP 1211, or SAE J2735 Signal Request Message requests remain inactive and any active priority requests remain active until served.

3.5.4.2.1.7.5 Monitor Priority Operation Indication

Upon request from a management station, the ASC shall return if the intersection is operating in Active Priority Control. Priority Control indicates to the receiver of this information that the controller is actively servicing a priority request and thus, the TimeChangeDetails for the intersection may suddenly change as the controller changes from "normal" operations to servicing a priority request as the controller transitions to serve the movement requested. The TimeChangeDetails are generated under assumption that any inactive priority, NTCIP 1211, or SAE J2735 Signal Request Message requests remain inactive and any active priority requests remain active until served.

3.5.4.2.1.7.6 Monitor Fixed Time Control Indication

Upon request from a management station, the ASC shall return if the intersection is operating under fixed time control. Fixed Time Operation indicates to the receiver of this information that the controller is operating in fixed time and thus the values in TimeChangeDetails are not dynamically changing cycle by cycle.

The controller is operating under fixed time control if all phases in the active sequence are running with maximum vehicle recall, running a non-actuated split, or is responding to a call to non-actuated input.

3.5.4.2.1.7.7 Monitor Non-Fixed Time Control Indication

Upon request from a management station, the ASC shall return if the intersection is operating under Active Demand. Active Demand, also called traffic dependent operation indicates to the receiver of this information that the controller is operating based on different levels of traffic parameters (requests, duration of gaps or more complex parameters), and not a fixed time mode.

The TimeChangeDetails are generated under assumption of any active demand for phase or pedestrian service. This demand can be sourced from cabinet inputs, central system commands, controller configuration, internal logic, or various other means. The TimeChangeDetails are generated upon current demand (Phase/Pedestrian calls) and updated within one second upon changes to this demand.

NOTE: Demand need not be actively serviceable to be considered in the SPaT message. For example, a coordinator may omit a phase until its permissive window opens, however SPaT is to consider this phase to have serviceable demand with expectation of future opening of this permissive window.

3.5.4.2.1.7.8 Monitor Standby Operation Indication

Upon request from a management station, the ASC shall return if the intersection is operating in a soft flash operation (flash through phase loadswitch), and the end of the flash condition is determinate at some point by the controller. StandbyOperation indicates that the controller is in a flashing condition but the controller will be aware of when the controller will exit the flashing condition. Standby operations include startup flash, automatic flash, and preempt flash. See Section F.3.3.4.8 for additional notes about a Standby Operation Indication.

3.5.4.2.1.7.9 Monitor Controller Failure

Upon request from a management station, the ASC shall return if the intersection is in a failure mode. Failure Mode indicates to the receiver of this information that the controller is not operating properly or has failed. Examples of a controller failure mode includes a process in the controller that is not be operating properly, a memory data error was detected, or an internal voltage failure was detected. Most failures will be fatal and would not result in this bit being SET, but for non-catastrophic failures, this bit can be set and the controller would likely activate cabinet flash.

3.5.4.2.1.7.10 Monitor MAP Message Validity

Upon request from a management station, the ASC shall return if the controller is indicating that the MAP message for this intersection should be marked as valid or invalid. The controller may allow an authorized user to indicate that the MAP message for the intersection is invalid because the signalized intersection is under test, or because there is temporary work at the signalized intersection, such as lane closures, that may make the MAP message invalid.

3.5.4.2.1.7.11 Monitor SPaT Data Validity

Upon request from a management station, the ASC shall return if the controller is providing valid or invalid SPaT data for the intersection. The controller may be providing invalid SPaT data that is not to be used if the signalized intersection is under test, or if there is temporary work at the signalized intersection, such as lane closures, that may make the SPaT data invalid.

3.5.4.2.1.8 Mark SPaT Invalid - Controller

Upon request from a management station, the ASC shall store an indication if the SPaT data being provided by the controller is valid or not, i.e., whether the SPaT data to be used or not. Situations when the SPaT data may be invalid include when the signalized intersection is under test, or if there is temporary construction at the signalized intersection such that the SPaT data should be ignored.

3.5.4.2.1.9 Mark SPaT Invalid - Port

Upon request from a management station, the ASC shall store an indication if the SPaT data being provided by the controller to a specific RSU port is valid or not, i.e., whether the SPaT data to be used or not. Situations when the SPaT data may be invalid include when the signalized intersection is under test. Unlike 3.5.4.2.1.8, Mark SPaT Invalid - Controller, this requirement allows the SPaT data to be marked invalid for a specific CV Application Process (e.g., RSU).

3.5.4.2.1.10 Mark MAP Message Invalid - Controller

Upon request from a management station, the ASC shall store an indication if the MAP message associated with the signalized intersection controlled by the controller is valid or not, i.e., whether the MAP message to be used or not. Situations when the MAP message may be invalid include when the signalized intersection is under test, or if there is temporary construction at the signalized intersection such that the MAP message should be ignored.

The MAP message is generally broadcasted by the CV Application Process but the indication on whether a MAP message is valid is contained in the SPaT message and the controller is the primary source of the data for the SPaT message. From a field maintenance perspective, the field maintenance personnel generally have easier access to the traffic controller than the RSU, and are likely to indicate the SPaT data is valid or invalid also. Thus, this requirement allows the field maintenance personnel to indicate whether the SPaT and MAP messages as valid or invalid from the same device (the controller).

3.5.4.2.1.11 Mark MAP Message Invalid - Port

Upon request from a management station, the ASC shall store an indication if the MAP message broadcasted by a specific CV Application Process data is valid or invalid, i.e., whether the MAP message is to be used or not. Situations when the MAP message may be invalid include when the signalized intersection is under test. Unlike 3.5.4.2.1.10, Mark MAP Message Invalid - Controller, this requirement allows the MAP message to be marked invalid for a specific CV Application Process (e.g., RSU).

3.5.4.2.1.12 Manage Signal Group Requirements

The following requirements are for managing signal groups for use in connected intersections. The signalGroupID is used in SAE J2735 to associate every active movement in a given intersection to a designated collection of one or more lanes of a common type (defined in a MAP message).

3.5.4.2.1.12.1 Determine Maximum Number of Signal Groups

Upon request from a management station, the ASC shall return the maximum number of signal groups that can be configured by the ASC.

3.5.4.2.1.12.2 Configure Signal Group Intersection Mapping

Upon request from a management station, the ASC shall store the signal groups and intersection identifiers in use for the intersection. This allows an ASC that controls multiple intersections to map signal groups to intersections for ASC purposes.

3.5.4.2.1.12.3 Configure Signal Group Control Source

Upon request from a management station, the ASC shall store the phase or overlap that controls the signal group.

3.5.4.2.1.12.4 Configure Signal Group Indication Types

Upon request from a management station, the ASC shall store the type of Green (includes WALK), Red (includes DON'T WALK), Flashing Yellow, and Dark movements to be indicated for the signal group in a SPaT message. Valid values are defined by DE_MovementPhaseState in SAE J2735.

3.5.4.2.1.12.5 Configure Signal Group Protected or Permissive State

Upon request from a management station, the ASC shall store the phase or overlap that determines if the signal group indication is a protected or permissive movement. This only applies to signal groups configured to indicate either protected movements or permissive movements are allowed during the GREEN or WALK indications.

3.5.4.2.1.12.6 Configure Signal Group Revocable Lanes

Upon request from a management station, the ASC shall store the set of revocable lanes that activates a signal group when the set of revocable lanes is enabled and deactivates a signal group when revoked. The set of revocable lanes is defined in Section 3.5.4.2.1.4.1, Configure Concurrent Enabled Lanes. If the signal group is enabled, then the movement state for the signal group is included in the SPaT data provided to a CV Application Process. This allows an ASC without MAP data to use correct signal groups in a SPaT Message.

3.5.4.2.1.12.7 Determine Maximum Number of Signal State Entries

Upon request from a management station, the ASC shall return the number of signal state entries supported by the ASC.

3.5.4.2.1.12.8 Configure Customized Signal State Parameters

Upon request from a management station, the ASC shall store customized parameters to determine what signal state as defined by DE_MovementPhaseState in SAE J2725 shall be indicated in a SPaT message.

3.5.4.2.1.13 Retrieve Signal Phase and Timing Time Point

Upon request from a management station, the ASC shall return the time points for the movement start/end times reported in the signal phase and timing data. Time points are in ticks, each tick representing units of tenths of a second, with a value of 0 representing the top of the hour, resulting in a range of 0 to 35999, and a range of 36000 to 36009 during leap seconds. These time points do not need to be synchronized with UTC time or the RSU time.

This requirement allows a TMC operator to view the controller's current time in ticks. Many signal controllers use AC line frequency to determine their internal time—this has the benefit that all signal controllers that use the same line frequency, such as along an arterial, remain synchronized for signal timing coordination. Therefore, the representation of movement start/end time points in the future should not depend on the ASC system time being synchronized with NIST UTC time or being synchronized with the RSU system time.

However, for a connected environment, it is important that all connected devices use time from a known and reliable source. In this context, a reliable source is defined as a time source whose accuracy is known and acceptable. For the purposes of this discussion, the time from a reliable source will be called disciplined time—that is, it "does not accumulate any offset over time." The RSU, which is expected to have access to a time source with disciplined time, can then convert these time points (and the movement start/end times) into time marks in disciplined time.

3.5.4.2.2 Manage Connected Vehicle Detector Requirements

The requirements for an ASC to support vehicle detection in a Connected Vehicle Environment follow.

3.5.4.2.2.1 Enabled Connected Device Connection

Upon request from a management station, the ASC shall store if the detection of connected devices is enabled or disabled. This requirement allows a CV Application Process to enable the detection and processing of connected devices (such as an equipped vehicle) for use by traffic signal controllers, such as for performance measures or actuation purposes.

3.5.4.2.2.2 Configure Vehicle Detector for Connected Vehicle Applications

Upon request from a management station, the ASC shall store if a vehicle detector is configured for connected vehicle detector. This allows CV Application parameters, such as Red Light Violating Warning and Assured Green Period, to apply when an actuation is placed on the vehicle detector.

3.5.4.2.2.3 Configure Connected Vehicle Detector Input Assignment

Upon request from a management station, the ASC shall store which vehicle or pedestrian detectors are actuated when a connected device is detected.

3.5.4.2.2.4 Configure Connected Vehicle Detector Port Assignment

Upon request from a management station, the ASC shall store which communications port on the ASC is associated with this connected vehicle detector. This allows a CV Application Process to place actuations on different vehicle detectors by sending connected vehicle messages to different ports on the ASC.

Add reference to CTI 4001 for RDZ filtering

3.5.4.2.2.5 Configure Assured Green Period Duration

Upon request from a management station, the ASC shall store an AGP for a through vehicle movement at an intersection. The AGP can be calculated according to Section 4.3.2.3 of CTI 4501. A value of 0 indicates that AGP is not in use.

While in Manual Mode, the ASC will not allow a green indication to be terminated before the AGP has been satisfied if the AGP is in use (i.e., if a connected vehicle is within the RLVW detection zone (RDZ)). The ASC will NOT extend a green indication beyond its configured Phase Maximum parameters to accommodate an Assured Green Period (AGP).

The ASC will extend a green indication to accommodate an AGP before accommodating any requests for signal priority. The AGP shall supersede any new or previously granted requests for signal priority at an intersection.

3.5.4.2.2.6 Configure Red Light Violation Warning Application Parameters

Upon request from a management station, the ASC shall store the detector input which reports if a connected vehicle is within the RLVW detection zone (RDZ) of a vehicle through movement at an intersection. This requirement allows the ASC to determine if an AGP is needed for that through movement when calculating the minimum end time for the end of a green indication.

3.5.4.3 Manage ASC – CV Application Process Interface Requirements

The second interface for managing an ASC in a connected vehicle environment is between the ASC Process (ASC) and the CV Application Process. The CV Application Process may be internal to the ASC or external, residing on the RSU or an intermediate device between the RSU and the ASC.

These requirements allow the ASC to deliver signal phase and timing data to the CV Application Process. From an SNMP standpoint, the ASC is the SNMP manager, and the CV Application Process is the agent.

The sub-requirements to manage the data exchanges between the ASC Process and the CV Application Process are:

a) ASC – External CV Application Process Requirements
b) ASC – Internal CV Application Process Requirements
c) Exchange Roadway Geometrics Information Requirements

The requirements to manage the data exchanges between an ASC and a CV Application Process follow.

3.5.4.3.1 ASC - External CV Application Process Requirements

The following requirements are for an ASC Process to exchange information with an external CV Application Process that resides on an RSU or an intermediate device between the RSU and the ASC.

3.5.4.3.1.1 Provide Movement Information Requirements

The SPaT message that is broadcasted by an RSU to connected vehicles includes information about what vehicle (or pedestrian) movements are permitted and when at a signalized intersection. To provide this information the CV Application Process needs movement data from the ASC. The requirements for an ASC to provide current and next movement information to a CV Application Process are as follows.

3.5.4.3.1.1.1 Provide Movement Time Point

Upon request from an ASC, a CV Application Process shall store the time point reference that the ASC will report movement start/end times. Time points are in ticks, each tick representing units of tenths of a second, with a value of 0 representing the top of the hour, resulting in a range of 0 to 35999, and a range of 36000 to 36009 during leap seconds. These time points do not need to be synchronized with UTC time or the RSU time.

This time point is not used in the SAE J2735 SPaT message. The CV Application Process (or RSU) will convert these time points to UTC time to generate the SPaT message.

3.5.4.3.1.1.2 Provide Movement State

Upon request from an ASC, a CV Application Process shall store the overall current state of a movement (for a signal group) at the intersection. The valid states are defined by DE_MovementPhaseState in SAE J2735. If the movement state is not known with a high level of confidence, a value of unavailable is used.

3.5.4.3.1.1.3 Provide Movement Minimum End Time

Upon request from an ASC, a CV Application Process shall store the time point of the earliest possible end of the current movement state (e.g., at the end of a permissive green or at the end of a permissive yellow) of a movement at an intersection, as defined by DE_Timemark in SAE J2735. If the duration of the current state is fixed, this value indicates the end time. This value can be viewed as the earliest possible time point at which the current interval could end, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour.

3.5.4.3.1.1.4 Provide Movement Maximum End Time

Upon request from an ASC, a CV Application Process shall store the time point of the latest possible end of the current movement state (e.g., at the end of a protected green or end of a steady red) of a movement at an intersection, as defined by DE_Timemark in SAE J2735. This value can be viewed as the latest possible time point at which the current interval could end, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour.

If the ASC can calculate the maximum end time, then the ASC is to provide that time point, otherwise, the ASC is to provide a value of unknown.

3.5.4.3.1.1.5 Provide Movement Likely End Time

Upon request from an ASC, a CV Application Process shall store the time point when the current movement state will most likely end (e.g., at the end of a protected green or end of a steady red) at an intersection, as defined by DE_Timemark in SAE J2735. The likely end time point may be predicted based on data available to the ASC. The time point is measured in tenths of a second in the current or next hour.

3.5.4.3.1.1.6 Provide Movement Likely End Time Confidence

Upon request from an ASC, a CV Application Process shall store the statistical confidence that the reported likely end time point of the current movement (e.g., at the end of a protected green or end of a permissive clearance time) at an intersection is accurate. The confidence value is measured as a probability class, as defined by DE_TimeIntervalConfidence in SAE J2735.

3.5.4.3.1.1.7 Provide Next Movement State

Upon request from an ASC, a CV Application Process shall store the movement state immediately after the current movement state (for a signal group) at the intersection. The valid states are defined by DE_MovementPhaseState in SAE J2735. If the next movement state is not known with a high level of confidence, a value of unavailable is used.

3.5.4.3.1.1.8 Provide Next Movement Minimum End Time

Upon request from an ASC, a CV Application Process shall store the time point of the earliest possible end of the movement state immediately after the current movement state at the intersection, as defined by DE_Timemark in SAE J2735. If the end time of the current movement state is known (maximum end time equals minimum end time) and the duration of the next movement state is fixed, this value indicates the end time point of the next movement state. This value can be viewed as the earliest possible time point at which the next interval could end, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour. If the next movement state is unknown, the time point of the earliest possible end time is also unknown.

NOTE: The minimum end time of the next movement will be the minimum end time of the current movement plus the minimum amount of time that the next movement state could time.

3.5.4.3.1.1.9 Provide Next Movement Maximum End Time

Upon request from an ASC, a CV Application Process shall store the time point of the latest possible end of the movement state immediately after the current movement state at the intersection, as defined by DE_Timemark in SAE J2735. This value can be viewed as the latest possible time point at which the next movement state could end, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour. If the next movement state is unknown, the time point of the latest possible end time is also unknown.

When the end time of the current movement state is known (maximum end time equals minimum end time) and the duration of the next movement state is fixed, such as when the ASC is operating in fixed time or the yellow interval duration, the minimum end time of the next movement state will equal the maximum end time of the next movement state.

3.5.4.3.1.1.10 Provide Next Movement Start Time

Upon request from an ASC, a CV Application Process shall store the time point when the next movement state immediately after the current movement state at the intersection will start, as defined by DE_Timemark in SAE J2735. If the next movement state is unknown, the time point of the next movement start time is also unknown. If the start time is unknown, a value of unknown is used. If the next movement state is unknown, the start time will also be unknown. If the next movement state is known, the start time is equal to the movement minimum end time of the current interval.

Note: Start time is always a future time (See CTI 4501, No Current Movement State Start Time). Thus, the start time of the current interval is always unknown.

3.5.4.3.1.1.11 Provide Movement Next Occurrence

Upon request from an ASC, a CV Application Process shall store the estimated time point when the current movement at an intersection is next allowed to proceed (i.e., the movement phase state will be permissive-Movement-Allowed or protected-Movement-Allowed), as defined by DE_Timemark in SAE J2735. The time point is measured in tenths of a second in the current or next hour. This requirement is used to support ECO-driving applications. For fixed time and coordinated signals, next time can be estimated by the ASC, subject to unpredictable events such as signal preemptions, or failures such as a watchdog failure or a conflict monitor. If next time cannot be determined, a value of unknown is used.

3.5.4.3.1.2 Provide Movement Assistance Requirements

The SPaT message in SAE J2735 can also provide potential pedestrian or bicyclist conflicts and queuing information to travelers. The requirements for an ASC to provide this data to an RSU so it can broadcast this information to travelers wishing to traverse through the intersection are defined as follows.

3.5.4.3.1.2.1 Provide Lane Connection Queue Length

Upon request from an ASC, a CV Application Process shall store the distance, in meters, from the stop line of the approach movement to the back edge of the last vehicle in the queue, as measured along the center line of the lane, for a specific movement through the intersection. Valid values are 0 to 10000 meters, where 0 indicates no queue or the queue distance is unknown, and 10000 represents all distance ≥ 10000 meters.

3.5.4.3.1.2.2 Provide Lane Connection Vulnerable Road User Detection

Upon request from an ASC, a CV Application Process shall store if any conflicting pedestrians or bicycles are detected for a specific movement through the intersection. This value is either on or off, with off indicating a high certainty that there is no pedestrian or bicycle present.

3.5.4.3.1.3 Provide Advisory Speed Requirements

The SPaT message in SAE J2735 can also provide speed advisories for specific movements and specific vehicle types. The requirements for an ASC to provide a CV Application Process with advisory speed information for a movement through the intersection are defined as follows.

3.5.4.3.1.3.1 Provide Advisory Speed Type

Upon request from an ASC, a CV Application Process shall store the speed advisory type for a specific movement traversing the intersection. Valid types of speed advisories are defined by DE_AdvisorySpeedType in SAE J2735. Speed advisories may be configured for specific vehicle types as defined in the MAP message for the same intersection.

3.5.4.3.1.3.2 Provide Advisory Speed

Upon request from an ASC, a CV Application Process shall store the advisory speed, in tenths of a meter per second, provided for a specific movement traversing the intersection. Speed advisories may be configured for specific speed advisory types (See 3.5.4.3.1.3.2) or vehicle types as defined in the MAP message for the intersection.

3.5.4.3.1.3.3 Provide Advisory Speed Zone

Upon request from an ASC, a CV Application Process shall store the distance, in meters, upstream from the stop bar that a speed advisory is recommended for a movement traversing the intersection. A value of 10000 indicates that the distance is 10,000 meters or greater.

3.5.4.3.1.3.4 Provide Advisory Speed Vehicle Type

For a specific movement traversing the intersection, upon request from an ASC, a CV Application Process shall store the vehicle type that a speed advisory is intended for. The vehicle type(s) is identified in the MAP message for the intersection. If no vehicle type is identified, then the advisory speed applies to all vehicles.

3.5.4.3.1.4 Provide Road Authority ID

Upon request from an ASC, a CV Application Process shall store the Road Authority ID for an intersection. The Road Authority ID is a unique identifier of the agency that operates and/or maintains the intersection or the ASC. The unique identifier in the US is an object identifier consisting of the Geographic Names Information System (GNIS) codes maintained by the US Geological Survey (USGS). The Road Authority ID is defined by DF_RoadAuthorityID in SAE J2735.

3.5.4.3.1.5 Provide Signal Phase and Timing Intersection Status

Upon request from an ASC, a CV Application Process shall store the status of the ASC as part of the signal phase and timing data. The intersection status values are defined by DE_IntersectionStatusObject in SAE J2735.

3.5.4.3.1.6 Provide Compressed SPaT Information to External CV Application Process

Upon request from an ASC, a CV Application Process shall store the mandatory SPaT information needed to generate a J2735 SPaT message. This data is provided by the ASC in a compressed manner so that the data can be transmitted more efficiently. The connected vehicle environment is expected to have limitations in the data rates and data capacity. This requirement allows the ASC to group sets of data so that the data can be transmitted more efficiently.

3.5.4.3.2 ASC - Internal CV Application Process Requirements

The following requirements are for an CV Application Process within the ASC to exchange information with an RSU process or an external CV Application Process that resides on an RSU or an intermediate device between the RSU and the ASC.

3.5.4.3.2.1 Provide UPER-encoded SPaT Message

A CV Application Process in the ASC shall provide UPER-encoded SPaT Message as defined in SAE J2735 to an RSU or intermediate device between the RSU and the ASC. The SPaT message contains the following information:

- Timestamp
- Intersection Status
- Enabled Lanes
- For each Signal Group (both current followed by the next movement):
    - Movement State
    - Minimum End Time
    - Maximum End Time
    - NextTime
    - StartTime

3.5.4.3.2.2 Retrieve BSMs

A CV Application Process in the ASC shall receive UPER-encoded BSMs as defined in SAE J2735 from an RSU or intermediate device between the RSU and the ASC. The RSU or the intermediate device may filter the BSMs it receives prior to providing the BSMs to the ASC. The ASC processes the BSMs to determine the location of equipped vehicles and their kinematics (velocity, acceleration). The ASC may use this information to support the need to provide an AGP, to manage the signalized intersection safely, and to improve mobility at a signalized intersection.

The data in the BSMs consists of the security certificate, the BSMcoreData, and may include VehicleSafetyExtensions. What data is included in each BSM and the conditions when the data is provided is defined in SAE J3161/1.

3.5.4.3.2.3 Retrieve PSMs

A CV Application Process in the ASC shall receive UPER-encoded PSMs as defined in SAE J2735 from an RSU or intermediate device between the RSU and the ASC. The RSU or the intermediate device may filter the PSMs it receives prior to providing the PSMs to the ASC. The ASC processes the PSMs to determine the location of equipped travelers and their kinematics (velocity). The ASC may use this information to improve the safety of VRUs at a signalized intersection.

3.5.4.3.2.4 Retrieve Actuation Report

Upon request from a CV Application Process in the RSU shall request the actuation report for connected device detectors from an RSU or intermediate device between the RSU and the ASC. An actuation report indicates the connected device detectors that are actuated at the time, based on if any BSMs or PSMs are detected within the detection zone and if the BSM or PSM satisfies the configuration criteria.

3.5.4.3.2.5 Retrieve Detection Report

Upon request from a CV Application Process in the RSU shall request the detection report for connected device detectors from an RSU or intermediate device between the RSU and the ASC. The processed data that can be provided by in a detection report are volume, average speed, average travel time, queue length, average gap, and platoon length.

3.5.4.3.3 Exchange Roadway Geometrics Information Requirements

A broadcasted MAP data message provides roadway geometry information to connected devices in the vicinity of the RSU. A roadway geometry plan defines what roadway geometry data is being broadcasted and contain the roadway configuration for one or more intersections. The requirements for an ASC and a CV Application Process to exchange the roadway geometry plan currently broadcasted by the RSU are defined in the following.

3.5.4.3.3.1 Retrieve MAP Plan in Effect

Upon request from an ASC, a CV Application Process shall provide the MAP plan currently being broadcasted by the RSU. The ASC may use this information to determine if the signal pattern in effect is compatible with the roadway geometry data broadcasted by the RSU.

3.5.4.3.3.2 Confirm MAP Plan Compatibility

An ASC shall confirm that the MAP plan broadcasted by an RSU is compatible with the SPAT data generated. An ASC generates SPAT data for consumption by travelers, however, this SPAT data has limited value unless it is broadcast in conjunction with roadway geometry data, relating the movement information with a lane. Thus, it is important that the SPAT data broadcasted by an RSU is compatible with the MAP plan also broadcasted by the RSU. If the SPAT data is not compatible with the MAP plan currently broadcasted, the SPAT data should not be broadcasted by the RSU.

3.5.4.3.4 Monitor CV Certificate Faults

Upon request from a management station, the ASC shall return if faults pertaining to invalid CV certificates have been detected.

3.5.4.4 Manage ASC - ECLA Interface Requirements

In some scenarios, the signal timing is determined by an external device called the ECLA (See Section 2.3.4.c), which sends commands to the ASC to advance the phase at specific times determined by the ECLA. In this environment, the ASC requires the information below to forward accurate SPaT data to a CV Application Process so a SPaT message can be broadcasted by an RSU in an CV environment. The requirements for an ASC to receive signal timing durations and movement states from an ECLA follow so the data can be shared with a management station follows.

At minimum, the ECLA must send commands to an ASC when the information it is able to share changes. An ECLA may send these commands at a user defined frequency but must send updated commands as soon as it is able to.

Note: the ASC is the final arbiter of the signal timing durations and state for each movement. The ASC is to provide the AGP if necessary – the ECLA is not expected to provide the AGP.

3.5.4.4.1 Receive Current Phase Minimum End Time from an ECLA

Upon request from an ECLA, the ASC shall store the time point of the earliest possible end of the current Active Phases' green indications as defined by DE_Timemark in SAE J2735. If the duration of the current state is fixed, this value indicates the end time. This value can be viewed as the earliest possible time point at which the current interval could end, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour. If the ECLA is unable to provide a time point nor a phase, the ECLA is to provide a value of unknown.

This time is provided by an ECLA for informational purposes and may be ignored by an ASC process.

3.5.4.4.2 Receive Current Phase Maximum End Time from an ECLA

Upon request from an ECLA, the ASC shall store the time point of the latest possible end of the current Active Phases' green indications as defined by DE_Timemark in SAE J2735. This value can be viewed as the latest possible time point at which the current interval could end, except when unpredictable events relating to a preemption or priority call come into play and disrupt a currently active timing plan. The time point is measured in tenths of a second in the current or next hour. If the ECLA is unable to provide a time point nor a phase, the ECLA is to provide a value of unknown.

This time is provided by an ECLA for informational purposes and may be ignored by an ASC process.

3.5.4.4.3 Receive Current Phase Likely End Time from an ECLA

Upon request from an ECLA, the ASC shall store the time point when the current Active Phases' green indications will likely end as defined by DE_Timemark in SAE J2735. The likely end time point may be predicted based on data available to the ASC. The time point is measured in tenths of a second in the current or next hour. If the ECLA is unable to provide a time point nor a phase, the ECLA is to provide a value of unknown.

This time is provided by an ECLA for informational purposes and may be ignored by an ASC process.

3.5.4.4.4 Receive Current Phase Likely End Time Confidence from an ECLA

Upon request from an ECLA, an ASC shall store the statistical confidence that the reported likely end time points of the current Active Phases' green indications are accurate. The confidence value is measured as a probability class, as defined by DE_TimeIntervalConfidence in SAE J2735.

This value is provided by an ECLA for informational purposes and may be ignored by an ASC process.

3.5.4.4.5 Receive Next Phase from an ECLA

Upon request from an ECLA, an ASC shall store the next phase that an ECLA expects to be committed to be active next (after the current Active Phase terminates). If the ECLA is unable to indicate what phases are next, the ECLA is to provide a value of unknown.

This value is provided by an ECLA for informational purposes and may be ignored by an ASC process.

3.5.4.4.6 Receive Compressed ECLA Input Data

Upon request from an ECLA, an ASC shall store the most recent transmission time from an ECLA, the expected end time information for the current Active Phases from an ECLA, and what the expected Next Phases are from an ECLA in a compressed format.

This data is provided by an ECLA for informational purposes and may be ignored by an ASC process.

3.5.5 Backward Compatibility Requirements

Prior versions of NTCIP 1202 use SNMPv1 as its application layer protocol, while NTCIP 1202 v04 uses SNMPv3 as its application layer protocol. A shortcoming of SNMPv1 is its lack of security – thus *NTCIP 9014, Infrastructure Standards Security Assessment (ISSA)*, published in 2021, recommended migrating the NTCIP center-to-field standards from SNMPv1 to SNMPv3. However, an authentication feature in SNMPv3 breaks backward compatibility for every SNMPv1 message exchanged, thus NTCIP 1202 v04 is not backward compatible with previous versions of NTCIP 1202, so this section is not applicable.

3.6 Supplemental Non-communications Requirements

Supplemental requirements for ASC are provided in the following subsections. These requirements do not directly involve communications via the communications interfaces addressed by NTCIP 1202 v04, but, if the supplemental requirement is selected in the PRL, the implementation shall fulfill the stated requirement to claim conformance to NTCIP 1202.

3.6.1 Response Time for Requests

The ASC processes all requests in accordance with all of the rules of the relevant base standards (i.e., NTCIP 1103 v03 and NTCIP 2301), including updating the value in the database and initiating the transmission of the appropriate response (assuming that the ASC has permission to transmit) within the Response Time. If the specification does not indicate the Response Time, the Response Time shall be 25 milliseconds. The Response Time is measured as the time between the receiving of the last byte of the request and the transmission of the first byte of the response.

3.6.2 Condition-based Maximum Transmission Start Time

When a user-specified condition-based exception reporting occurs, the ASC shall initiate the transmission of the appropriate report within the Maximum Transmission Start Time. If the agency specification does not indicate the Condition-based Maximum Transmission Start Time, the Condition-based Maximum Transmission Start Time shall be 10 seconds. The Condition-based Maximum Response Start Time is measured as the time between the time the ASC first detects the occurrence of the event to the time the ASC initiates communications with the management station (e.g., handshake).

3.6.3 Signal Phase and Timing Data Performance Requirements

This section defines the performance requirements for the exchange of signal phase and timing data in a connected vehicle environment. The applicable performance requirements depend on the architecture implemented.

3.6.3.1 SPaT Maximum Transmission Start Time

The ASC shall begin initiating the transmission (SET) of all SPAT data object(s) that changed to the CV Application Process no more than the SPAT Maximum Transmission Start Time after the change to the ground truth of a channel (i.e., when the controller changes the command to the load switch). If the agency specification does not indicate the SPAT Maximum Transmission Start Time, the SPAT Maximum Transmission Start Time shall be 10 milliseconds. The data object may be an individual data object definition or a block object definition.

3.6.3.2 Movement Time Point Minimum Transmission Rate

The ASC shall transmit (SET) the time point reference to the CV Application Process no less than the Movement Time Point Minimum Transmission Rate. If the agency specification does not indicate the Movement Time Point Minimum Transmission Rate, the Movement Time Point Minimum Transmission Rate shall be once per 100 milliseconds.

3.6.3.3 SPaT Maximum Transmission Rate

The ASC shall initiate the transmission (SET) of all SPAT data object(s) to the CV Application Process at least once every 100-millisecond interval, +/- 2 milliseconds. The data object(s) may be individual data object definitions or a block object definition. The 100-millisecond interval is equivalent to the tenth of a second processing cycle for the traffic signal controller. Note that the NEMA TS2 allows a tolerance of 2% in the framing.

3.6.3.4 SPaT Time Accuracy

The point in time reported by the controller in the SPaT data shall be within 100 milliseconds of the time the signal indication changes. When the time of change is known, such as the end of a yellow interval, the minEndTime reported by the controller is equal to the maxEndTime reported. That time should be within 100 milliseconds of the actual time of change (as depicted by the change in ground truth at the load switch). For example, if the time of change occurs 1 millisecond after midnight, the SPaT can report time of change as midnight or 100 milliseconds after midnight.

Section 4
Dialogs [Normative]

Section 4 defines the dialogs (i.e., sequence of data exchanges) that fulfill various Data Exchange requirements defined in Section 3.5. As SNMP communications are largely driven by the management station, most of the requirements define how the device shall respond to the various possible actions a management station might take.

The NTCIP standards effort is based on SNMP. This protocol offers a high degree of flexibility as to how the management station structures its requests. For example, with SNMP, the management station can do any of the following:

a) Send only those requests that are critical at the current time, whereas a standardized dialog typically sends requests relating to all associated data, regardless of whether it is critical for current purposes
b) Combine a number of requests in a single packet, whereas a standardized dialog dictates the exact contents of each packet
c) Separate a group of requests into multiple packets, whereas a standardized dialog dictates the exact contents of each packet
d) Interweave requests from multiple dialogs, whereas a standardized dialog dictates the exact ordering of messages, which are not interrupted with other messages

This flexibility can be a powerful tool allowing a management system to optimize the use of communication facilities, which is the primary reason that SNMP was chosen as the core NTCIP protocol. However, the flexibility also means that there are numerous allowable variations in the management process that a management station may choose to use and that an agent shall support to conform to NTCIP 1202 v04.

Unfortunately, this flexibility presents a challenge to ensuring interoperability. While a conformant ASC is required to support all mandatory operations defined within this standard, ensuring that a given ASC actually supports every possible combination of mandatory and optional requirements would be impractical. Instead, most agencies only require that the device be tested to a standard set of procedures, which would use standardized dialogs (as defined in ISO 26048-1 and Section 4.2). To improve communications efficiency, management stations may use non-standard dialogs (e.g., a combination of GET and/or SET requests that is not defined as a standardized dialog, but which a conformant device is required to support according to the ACCESS rules defined in **Error! Reference source not found.**). Because these more efficient dialogs may not be known until the acquisition of the management station, which may be years after the acquisition of the device, there is a potential for an interoperability problem to arise.

To overcome this complication, Section 4 defines a lowest common denominator approach to communications between a management station and a device. It defines the standardized dialog for each Data Exchange Requirement. Management stations may support other dialogs to fulfill these same requirements, as long as these dialogs are consistent with the rules defined in NTCIP 1202 v04. Such a management station is termed a “consistent management station.” A consistent management station interoperates with any “conformant” device. However, since an agency cannot be certain that a device is 100% conformant to every possible scenario (given practical constraints), interoperability problems could still arise.

A “conformant management station” is required to offer a mode in which it only uses the standardized dialogs as defined in Section 4. With this limited definition, there is relatively little variability in what constitutes a conformant management station. Thus, fully testing a management station for conformance is a relatively straight forward process that can be done within the practical constraints faced by most procuring agencies. Thus, a conformant management station provides an agency with a much greater chance of achieving interoperability with off-the-shelf devices that have been tested against NTCIP 2104 v04, and the designation of such a system is intended to provide a guaranteed base level of interoperability.

The rules for the standardized dialogs follow:

a) The dialogs are defined by a sequence of GET or SET requests. These requests shall equate to the GET and SET operations defined in ISO 26048-1 Section 9 Dialogues.
b) The contents of each request are identified by an object name. Each object name consists of an object type and an instance identifier. Definitions of each object type are provided in Section 4.5 and **Error! Reference source not found.**. The meaning of the instance identifier is provided by these same definitions coupled with standard SNMP rules (see RFC 2578).
c) Each message shall contain all of the objects as shown, unless otherwise indicated.
d) A message shall not contain any other objects.
e) The contents of each message sent by the management station may appear in any order.

Note: Ideally, the order of objects should match the order as shown in NTCIP 1202 to provide the highest probability of interoperability. However, it is recognized that many implementations may use off-the-shelf software, which may prevent the designation of an exact ordering of objects and as a result, this ordering is not a requirement of NTCIP 1202 v04.

a) After sending a message, the management station shall not transmit any other data across the communications channel until the earlier of:
    a. The management station receiving a response from the device; or
    b. The expiration of the maximum response time.
b) If the response indicates an error occurred in the operation, the management station shall exit the process, unless specific error-handling rules are specified by the dialog.
c) Dialogs containing a sequence of only GET requests may request objects in any order.

However, since consistent management stations can alter the order of requests, this standard defines rules for when certain data exchanges are allowed. Unless otherwise indicated, a conformant device shall allow an object to be retrieved (through a GET request) or altered (through a SET request, if the object is writable) at any time.

Finally, Section 4 presents an overview of all of the data defined by this standard, prior to presenting the complete definition for each piece of data in **Error! Reference source not found.**.

4.1 Tutorial [Informative]

The Requirements Traceability Matrix (RTM) in Annex A.3 identifies the standardized dialog that can be used to achieve each of the data exchange requirements defined in Section 3.5. Simple data exchange requirements reference one of the generic SNMP dialogs along with a list of data elements (See ISO 26048-1 Section 9 Dialogues). These equate to a single message being sent (e.g., a GET request) containing the referenced data elements followed by the appropriate response per the generic dialog specification.

Section 4.2 defines the standardized dialogs for the more complicated data exchange requirements. Each dialog in these sections define how the system is designed to work for a given data exchange requirement. It indicates the sequence of actions that a management station has to follow to provide the specific service. Each of these dialogs is defined by a number of steps. Many of the steps reference data elements that are defined in ISO26048-1-Transaction.mib, which is part of the ISO 26048-1 standard. These data elements are also shown in the corresponding row of the RTM along with their precise section number.

The dialogs may also be accompanied by an informative figure that provides a graphical depiction of the normative text. The figures conform to the Unified Modeling Language and depict the management station as an outside actor sending a series of messages to the device and the device returning responses. If there is any conflict between the figure and the text, the text takes precedence.

4.2 Specified Dialogs

This section provides the standardized data exchange sequences that can be used by management stations to ensure interoperable implementations for the various data exchange requirements identified in Section 3.5. Diagrams and graphical representations are included to supplement the text (i.e., not used as a replacement for the text). This section only includes dialogs that have special semantics or impose special restrictions on the operations that are allowed.

4.2.1 Get Block Data

Note: This is a generic dialog that is referenced with specific block names.

The standardized dialog for a management station to retrieve block objects (See Figure 8) shall be as follows:

a) (Precondition) The management station shall be aware of the block data type to be retrieved (0x00 for a Standard Data Block).
b) (Precondition) The management station shall be aware of the block data identifier to be retrieved.
c) (Precondition) The management station shall be aware of the block index value or values (up to 5 different block index values per block object retrieval) to be retrieved (if needed).
d) (Precondition) The management station shall be aware of the block quantity value or values (up to 5 different block quantity values per block object retrieval) to be retrieved (if needed).
e) The management station shall SET the 'ascBlockGetControl' object to ensure that the requested block object and values are valid and supported by the device.
f) If the validity checks fail, the device shall respond with an error status of 'badValue (3)' and the 'ascBlockErrorStatus' object with the value that generated the error, then exit.
g) If the validity check does not fail, the management station shall GET the 'ascBlockData' object utilizing the values in the 'ascBlockGetControl' object.

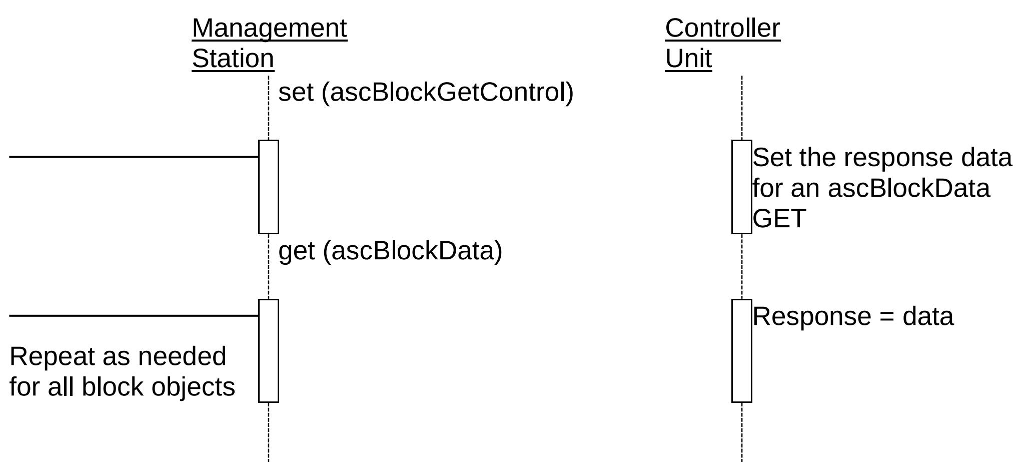

Figure 8 Get Block Data

4.2.2 Set Complex Configuration Parameters (called 'P2' Objects)

Note: This is a generic dialog that is referenced with specific object names.

The standardized dialog for a management station to configure complex configuration parameters ('P2' objects – See Section 4.3.1) (See Figure 9) shall be as follows:

a) (Precondition) The management station shall use the same community name in all data exchanges until the ASC database has been successfully updated.
b) The management station shall GET the fdTransactionMode object until the response value is 'normal' or 'done.' This step indicates that the database is operational and can be modified.
c) The management station shall SET the fdTransactionMode object to a value of 'transaction' until the response value is 'transaction.'
d) The management station shall SET all objects provided to this dialog to their desired values referenced into the device buffer.
e) Once all objects provided to this dialog have been set to the desired values into the device buffer, the management station shall SET the fdTransactionMode object to a value of 'verify” to initiate a consistency check (See Section 4.3.2) to analyze the objects 'in context' treating them as interrelated values.
f) Once the consistency check is complete, the ASC will automatically change the value of fdTransactionMode to 'done.'
g) The management station shall GET the fdTransactionMode object until the response value is 'done.'
h) The management station shall GET the fdTransactionStatus object until the response value is 'doneWithError' or 'doneWithNoError'.
i) If fdTransactionStatus equals 'doneWithNoError', then the ASC shall implement the contents of the device buffer (buffer data). If fdTransactionStatus equals 'doneWithError', the management station shall GET the fdTransactionError object to determine the error with the consistency check.
j) The management station shall SET the fdTransactionMode object to 'normal' or to 'transaction' if another transaction is to be performed. Exit the process.
k) The ASC will discard the contents of the device buffer (buffer data).

See ISO 26048-1 Section 8.21 Transaction Feature and ISO26048-1-Transaction.mib for additional information.

Other values for fdTransactionStatus

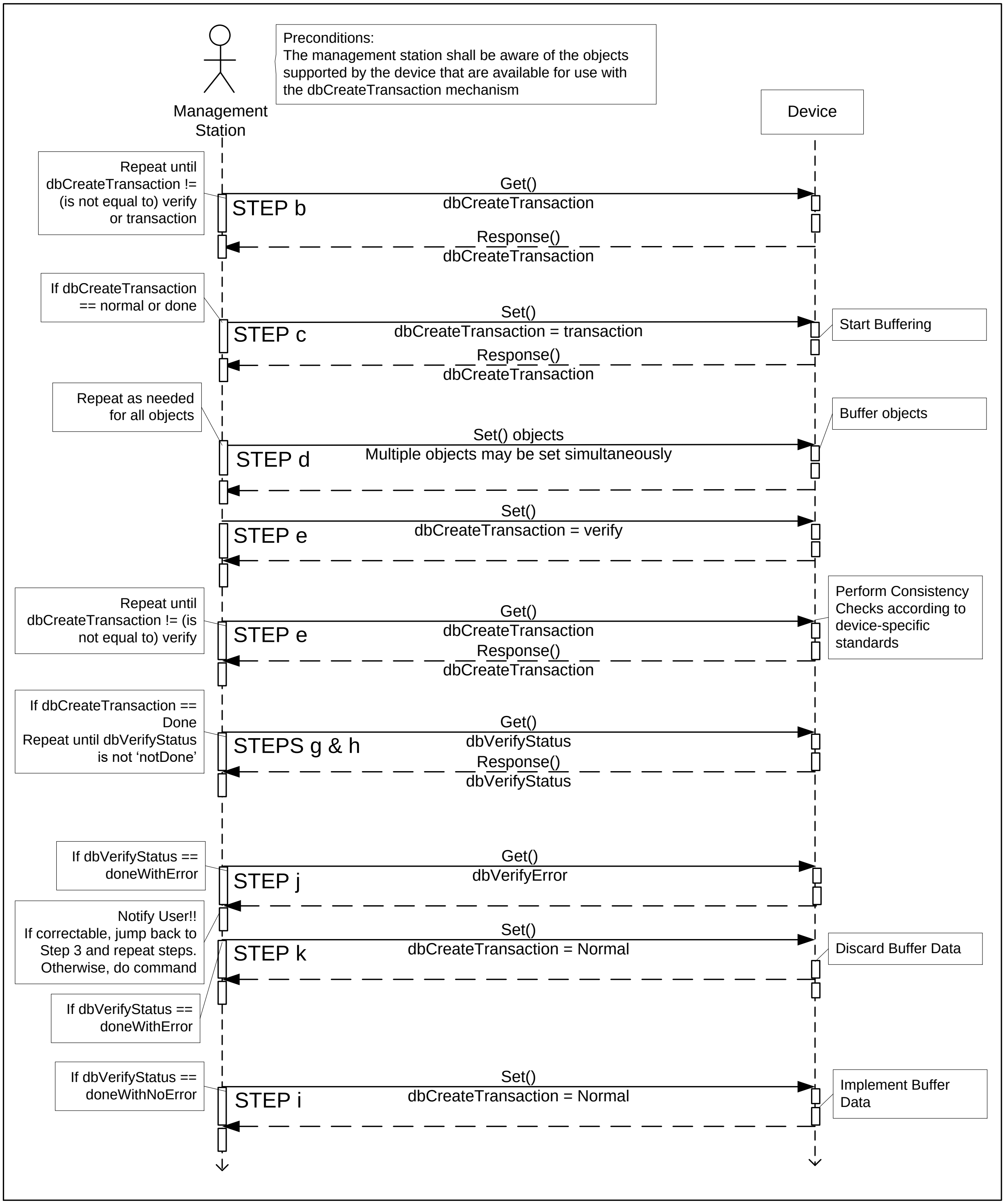

Figure 9 Set Complex Configuration Parameters

4.2.3 Set Block Data

Note: This is a generic dialog that is referenced with specific block names.

The standardized dialog for a management station to configure block objects shall be as follows:

a) (Precondition) The management station shall be aware of the block data type to be configured (0x00 for a Standard Data Block).
b) (Precondition) The management station shall be aware of the block data identifier to be configured.
c) (Precondition) The management station shall be aware of the block index value to be configured (if needed).
d) (Precondition) The management station shall be aware of the block quantity value to be configured (if needed).
e) The management station shall GET the fdTransactionMode object to ensure that the database is operational and can be modified, which would be indicated by a value of 'normal' or 'done.'
f) The management station shall SET the fdTransactionMode object to a value of 'transaction' and ensure that the fdTransactionMode object has a value of 'transaction.'
g) The management station shall SET the ascBlockData object.
h) The management station shall SET all objects (to their desired values) referenced in the buffer.
i) Once all objects have been set to the desired values in the buffer, the management station shall SET the fdTransactionMode object to a value of 'verify" to initiate a consistency check (See Section 4.3.2) to analyze the objects 'in context' treating them as an interrelated value. Once complete, the ASC will automatically change the value of fdTransactionMode to 'done'.
j) The management station shall then retrieve (GET) the fdTransactionStatus (value 'doneWithNoError') and fdTransactionError objects to ensure that the consistency check was successful.

See ISO 26048-1 Section 8.21 and ISO26048-1-Transaction.mib for additional information.

	Management Station		Controller Unit	
		get (fdTransactionMode)		
Repeat until fTransactionMode != verify				Response = fdTransactionMode
		set (fdTransactionMode = transaction)		
If fdTransactionMode == Normal or Done				Start Buffering
		set (ascBlockData)		
Repeat as needed for all block objects				Buffer all block data except dynObj blocks
		set (fdTransactionMode = verify)		
				Perform Consistency Checks
		get (fdTransactioMode)		

Repeat until fdTransactionMode != verify		Response = fdTransactionMode
If fdTransactionMode == Done		Implement Buffer Data
	set (fdTranactionMode = normal)	

Figure 10 Set Block Objects

4.2.4 Setup, Programming, and Processing of I/O Mapping

Because the I/O mapping directly affect the operation of the traffic cabinet, the objects to configure them are considered complex configuration parameters ('P2' objects – See Section **Error! Reference source not found.**). This means that a database transaction is required to change the configuration of the I/O mapping. This dialog is defined in Section 4.2.2.

The I/O mapping requires processing in a consistent order to produce deterministic results. The desired order of operation is:

a) Read external inputs
b) Input mapping – map external inputs to functions
    1. Input functions are OFF/inactive unless an external input mapped to the function is ON/active.
    2. Input functions are OR'd so that if multiple inputs are mapped to the same function, the function will be ON/active if any of the inputs are ON/active, i.e., if multiple inputs are be mapped to the cabinetDoorOpen input, then if any input is active the cabinetDoorOpen input is active.
c) Input logic gate processing
    1. The outputs of input logic gates are OR'd into the input functions with the external inputs, i.e., if the output of an input logic gate is assigned to cabinetDoorOpen, then if the output of the logic gate is ON/active then the cabinetDoorOpen function will be ON/active regardless of the status of any external inputs mapped to cabinetDoorOpen.
d) Traffic processing
    1. Phases and overlaps are mapped to channels during traffic processing
e) Output logic gate processing
f) Output mapping – map functions to external outputs
g) Write external outputs
h) Generate SPAT information from the outputs

4.2.5 Making an I/O Map Active

Making a new I/O map the active map requires a database transaction:

a) Open a database transaction by setting fdTransactionMode to 'transaction' (see the transaction dialog above).
b) Configure the desired I/O map while the fdTransactionMode is open, or have the desired I/O map already configured.
c) Set ascIOactiveMap to the index of the desired I/O map.

d) When all requirements for changing the I/O map set by ascIOactivateRequirement are satisfied, set fdTransactionMode to 'verify.'
e) Get fdTransactionMode, if equal to 'done' then get fdTransactionStatus.
f) If fdtransactionStatus is 'doneWithNoError' then set fdTransactionMode to 'normal'.
g) Else examine fdTransactionError to determine the cause of the error and make corrections as necessary.

Once the transaction is successfully completed the new I/O map will be in effect.

4.2.6 Provide Detection Reports to an ASC

The standardized dialog for a management station to provide detection reports to an ASC shall be as follows:

a) (Precondition) The management station shall be aware of what detection report (columnar) data is to be exchanged.
b) The management station shall GET activeCvDetectors and detectionReportSequence.
c) Make x = the value of detectionReportSequence.
d) For the specified row in detectionReportTable, the management station shall SET AscCvDetectionReportBlockData in accordance with Section 4.2.3. Only the columnar data requested, as configured in detectionReportCollection, shall be sent in AscCvDetectionReportBlockData. If the columnar data is NOT sent, that columnar data shall be SET to the default value.
e) The ASC shall increment detectionReportSequence.

where,
x = the index of the detection report number (detectionReportSequence)

4.2.7 Confirm MAP Compatibility

The standardized dialog for an ASC to confirm the SPAT data generated is compatible with the MAP plan broadcasted by the CV Application Process is as follows:

a) (Precondition) A management station shall SET the spatPortMapActivationCode for each CV Application Process that the ASC is to exchange SPAT data with.
b) The ASC shall GET the mapActivatePlan.0 from the appropriate CV Application Process.
c) The ASC shall compare the mapActivatePlan.0 with spatPortMapActivationCode.x. If the values match, then the ASC may exit the process.
d) If the values do not match, spatPortStatus.x shall be mapError(4), and the ASC shall stop providing SPAT data to this CV Application Process.
e) If the response from Step c indicates an error, the MAP plan was not activated. The management station shall GET mapActivatePlanError.0 to determine the type of error.

where,
x = the index of the RSU port (rsuPortIndex)

4.2.8 Configure Enabled Lanes

The standardized dialog for a management station to command the enabled lanes list on an ASC shall be as follows:

a) (Precondition) The management station shall ensure that the selected enabled lanes are configured as revocable lanes in the current MAP plan.
b) The management station shall SET spatEnabledLanesCommand2.0 to the desired values. This will cause the ASC to perform a consistency check (See Section 4.3.2) on the command.

c) If the response indicates 'noError', the spatEnabledLanesCommand2 has taken effect and the management station shall GET the spatPortStatus.x to ensure that there are no errors preventing the command from taking effect. The management station may then exit the process.
d) If the response from Step b indicates an error, the command did not take effect. The management station shall GET spatPortStatus.x to determine the type of error.

where,
x = the index of the RSU Port (rsuPortIndex)

4.2.9 Configure ISO 26048-1 Objects

The dialogs for a management station to satisfy requirements defined in ISO 26048-1 Section 8 are defined in ISO 26048-1 Section 9.

This section defines additional rules for satisfying requirements defined in ISO 26048-1 Section 8.

4.2.9.1 Static Table Implementation

Most tables defined in ISO 26048-1 are dynamic with an object of RowStatus. Devices that implement ISO 26048-1 dynamic tables as static tables by setting all objects of type RowStatus equal to ‘active(1)’ are conformant to this standard. Implementations of ISO 26048-1 dynamic tables as static tables without the objects of type RowStatus are non-conformant.

4.2.10 Provide Data to an RSU

The dialogs for providing data to an RSU, with the ASC acting as the SNMP Manager, are defined outside of this standard.

The following dialogs are defined in NTCIP 1218 v01A.

4.2.10.1 NTCIP 1218 G.3 SNMP Set Interface

NTCIP 1218 Section G.3 defines the dialog that applies to the following requirements.

a) 3.5.4.3.1.1.1 Provide Movement Time Point
b) 3.5.4.3.1.4 Provide Road Authority ID
c) 3.5.4.3.1.5 Provide Signal Phase and Timing Intersection Status
d) 3.5.4.3.1.6 Provide Compressed SPaT Information to External CV Application Process

4.2.10.2 NTCIP 1218 4.2.8 Generic Table Row Dialog – Dynamic Table

NTCIP 1218 Section 4.2.8 defines that the dialog that applies to the following requirements.

a) 3.5.4.3.1.1.2 Provide Movement State
b) 3.5.4.3.1.1.3 Provide Movement Minimum End Time
c) 3.5.4.3.1.1.4 Provide Movement Maximum End Time
d) 3.5.4.3.1.1.5 Provide Movement Likely End Time
e) 3.5.4.3.1.1.6 Provide Movement Likely End Time Confidence
f) 3.5.4.3.1.1.7 Provide Next Movement State
g) 3.5.4.3.1.1.8 Provide Next Movement Minimum End Time
h) 3.5.4.3.1.1.9 Provide Next Movement Maximum End Time
i) 3.5.4.3.1.1.10 Provide Next Movement Start Time
j) 3.5.4.3.1.2 Provide Movement Assistance Requirements
k) 3.5.4.3.1.3 Provide Advisory Speed Requirements
l) 3.5.4.3.2.1 Provide UPER-encoded SPaT Message

4.3 State-Transition Diagrams

State-Transition diagrams are included for those objects that have states or manage states. The State Transition Diagrams include state-transition tables (listing of the possible state transitions), legitimate transitions, and any illegitimate transitions.

"State-transition diagrams describe all of the states that an object can have, the events under which an object changes state (transitions), the conditions that must be fulfilled before the transition will occur (guards), and the activities undertaken during the life of an object (actions)." (Reference: State-Transition Diagrams: Testing UML Models, Part 4 by Lee Copeland)

The objects for an ASC device do not have states or manage states, but are classified by data parameter type, as defined in Section 4.3.1. For those objects that are defined as a 'critical' data parameter type, the consistency checks defined in Section 4.3.2 are required to be performed. ~~Section 4.3.3 describes the behavior of the device when a non-sequential time change is detected.~~

4.3.1 Object Types

An ASC shall support different types of object types, which are either:

a) pure status/information parameters (called 'S' objects),
b) simple configuration parameters (called 'P' or 'C' objects), or
c) critical configuration parameters that the ASC has to evaluate for consistency before it will use them (called 'P2' objects).

The following symbols are used to indicate type within this standard:

Table 7. Object Types

Symbol	Type
C	Control Object - use of 'fdTransactionMode' in ISO 26048-1 shall NOT delay a SET to this object.
P	Parameter Object - use of 'fdTransactionMode' in ISO 26048-1 to SET this object is optional. The device must support BOTH a SNMP SET without using the 'fdTransactionMode' and a SET using 'fdTransactionMode' per the 4.2.2 dialog.
P2	Transaction Object - use of 'fdTransactionMode' defined in ISO 26048-1 to SET this object is mandatory. An SNMP SET without 'fdTransactionMode' returns an 'inconsistentValue' error.
S	Status / Information Object - this object is read only therefore a SET is not permitted. An SNMP SET without 'fdTransactionMode' returns a 'readOnly' error.

All object types can be retrieved using simple SNMP GET commands.

Pure status/information ('S' objects) parameters are read-only objects, therefore an SNMP SET command is not permitted on these types of parameters. These types of parameters may also be contained in block objects (see 0) or in dynamic objects (see ISO 26048-1 Section 8.7 and ISO26048-1-DynObj.mib).

Simple configuration parameters ('C' or 'P' objects) can be configured using a simple SNMP SET command. However, the use of 'fdTransactionMode' defined in ISO 26048-1 shall NOT delay an SNMP SET to 'C' objects, while the use of 'fdTransactionMode' defined in ISO 26048-1 for 'P' objects is optional. The device also has to support both an SNMP SET without using 'fdTransactionMode' and an SNMP SET using 'fdTransactionMode'. Simple configuration parameters may be contained within block objects (see 0) or in dynamic objects (see ISO 26048-1 Section 8.7 and ISO26048-1-DynObj.mib).

Critical configuration parameters or any settable objects defined within Block Objects shall not be configured using a simple SNMP SET command, but instead shall be configured using a special data exchange mechanism. This data exchange mechanism is defined above in Section 4.2.3 ~~and ISO 26048-1 Section 9 and ISO26048-1-Transaction.mib~~. It is a mechanism to open a buffer in which the new configuration values are to be stored, to download those new configuration values into the buffer, and to instruct the controller to verify the new downloaded values in conjunction with other values stored in the database (this process is referred to as transaction feature and the verification process is referred to as 'consistency checks' – see Section 4.3.2). Only once this process has been completed successfully will the ASC use the downloaded configuration parameter values.

4.3.2 Consistency Checks

Consistency checks assure that certain critical objects are checked “in context” and treated as interrelated values rather than separate non-related data items.

When data is downloaded to a buffer in the ASC operating in the “transaction” mode, as defined by the transaction feature defined in ISO 26048-1, consistency checks shall be performed on downloaded data when the “verify” state of the 'fdTransactionMode’ object is commanded. The consistency checks that shall occur and corresponding error messages are described below. Error messages, if any, may be examined by reading the 'fdTransactionError' object defined in ISO 26048-1 once the ASC has entered the “done” mode of the 'fdTransactionMode' object.

4.3.2.1 Consistency Check Rules

The consistency check rule is stated first, followed by the corresponding error message(s).

- Concurrent Phases, as defined by the phaseConcurrency object, must be in a different ring from phaseNumber (assuming that the phase contained in the phaseNumber object is defined). The error message indicates one or more defined concurrent phases have the same ring assignment as phaseNumber. The value “xx” corresponds to phaseNumber.

  “PHASE xx CONCURRENCY FAULT”

  An example: phaseConcurrency.1 (Phase 1 concurrent phases) includes Phase 2 and phaseRing.1 (Phase 1 Ring) equals phaseRing.2 (Phase 2 Ring). An error message of "PHASE 01 CONCURRENCY FAULT" within the 'fdTransactionError' object is generated.

- Concurrent Phases, as defined by the phaseConcurrency object, must be mutually concurrent with phaseNumber (assuming that the phase contained in the phaseNumber object is defined). The error message indicates one or more defined concurrent phases does not include phaseNumber as a concurrent phase. The value "xx" corresponds to phaseNumber.

  "PHASE xx MUTUAL FAULT"

  An example: phaseConcurrency.1 (Phase 1 concurrent phases) includes phase 5, and phaseConcurrency.5 (Phase 5 concurrent phases) does not include phase 1. An error message of "PHASE 01 MUTUAL FAULT" is provided.

- Phase Sequences, as defined by the sequenceData object, must include phases only once in a given phase sequence. The error message indicates a phase appears more than once in a phase sequence. The value “xx” corresponds to sequenceNumber for sequenceData.

  “SEQ xx SAME PHASE FAULT”

  An example: sequenceData.1.1 (Sequence 01 / Ring 1) is 01-02-03-04-01 (Phase 1 appears twice). An error message of "SEQ 01 SAME PHASE FAULT" is provided.

- Phase Sequences, as defined by the sequenceData object, must include only phases with a ring assignment (phaseRing) equal to sequenceRingNumber. The error message indicates a phase defined by sequenceData does not have a phaseRing equal to sequenceRingNumber. The value “xx” corresponds to sequenceNumber. The value "#" corresponds to sequenceRingNumber.

  “SEQ xx RING # FAULT”

  An example: sequenceData.1.1 (Sequence 01 / Ring 1) is 01-02-03-04-05 and all phaseRing parameters = 1 except phaseRing.5 = 2. An error message of "SEQ 01 RING 1 FAULT" is provided.

- Phase Sequences, as defined by the sequenceData object, must include all phases with a ring assignment (phaseRing) equal to sequenceRingNumber. The error message indicates a phase has been omitted in the sequenceData for sequenceRingNumber. The value “xx” corresponds to sequenceNumber. The value "#" corresponds to sequenceRingNumber.

  “SEQ xx RING # PHS OMITTED”

  A standard dual ring example: sequenceData.1.1 (Sequence 01 / Ring 1) is 01-02-03 (does not include Phase 4). An error message of "SEQ 01 RING 1 PHS OMITTED" is provided.

- Phase Sequences, as defined by the sequenceData object, must be ordered such that all sequenceRingNumber phases within a Concurrency Group can be serviced sequentially without leaving the Concurrency Group of which they are a member. The error message indicates all phases in a Concurrency Group could not be serviced sequentially. The value “xx” corresponds to sequenceNumber.

  “SEQ xx RING SEQ FAULT”

  An standard dual ring example: sequenceData.1.1 (Sequence 01 / Ring 1) is 01-03-02-04 and sequenceData.1.2 (Sequence 01 / Ring 2) is 05-06-07-08. An error message of "SEQ 01 RING SEQ FAULT" is provided.

- Phase Sequences, as defined by the sequenceData object; phases must be arranged, so Concurrency Groups of which phases are a member are sequenced in the same order in all rings for a given sequenceNumber. The error message indicates Concurrency Groups are not in the same order for all. The value “xx” corresponds to sequenceNumber.

  “SEQ xx CG SEQ FAULT”

  An standard dual ring example: sequenceData.1.1 (Sequence 01 / Ring 1) is 01-02-03-04 and sequenceData.1.2 (Sequence 01 / Ring 2) is 07-08-05-06. An error message of "SEQ 01 RING SEQ FAULT" is provided.

- Phase Sequences, as defined by the sequenceData object; phases must be arranged so that it is possible to service all phases (not skip any phase due to compatibility constraints) in all rings in the order defined.

  "SEQ xx SEQUENCING FAULT"

  An example (lead-lag dual ring where phase 1 & 5 cannot operate concurrently): sequenceData.1.1 (Sequence 01 / Ring 1) is 02-01-03-04 and sequenceData.1.2 (Sequence 01 / Ring 2) is 06-05-07-08. An error message of "SEQ 01 SEQUENCING FAULT" is provided.

- Phase Sequences, as defined by the sequenceData object: all sequences must contain entries for all active rings.

  " SEQ xx RING xx EMPTY"
  " SEQ xx ALL RINGS EMPTY"

- The following objects define functionality related to phase assignments. Consistency checks among other things insure that phases specified by these objects may operate concurrently and are defined only once in each string parameter. Note that if the objects are not defined, operation between different CU's may be inconsistent.

  Phase Startup (phaseStartup)
  Automatic Flash Entry Phases (phaseOptions[1])
  Automatic Flash Exit Phases (phaseOptions[2])
  Overlap Included Phases (overlapIncludedPhases)
  Overlap Modifier Phases (overlapModifierPhases)
  Preempt Track Clear Phases (preemptTrackPhase)
  Preempt Dwell Phases (preemptDwellPhase)
  Preempt Dwell Peds (preemptDwellPed)
  Preempt Exit Phases (preemptExitPhase)
  Preempt Cycling Phases (preemptCyclingPhase)
  Preempt Cycling Ped (preemptCyclingPed)

  When the defined phases CAN NOT time concurrently:
  “START PHASE CG FAULT”
  “FLASH ENTRY CG FAULT”
  “FLASH EXIT CG FAULT”
  “PE TRACK PHASE CG FAULT”
  “PE DWELL PHASE CG FAULT”
  “PE EXIT PHASE CG FAULT”

  When the defined phases are in the same ring:
  “START PHASE RING FAULT”
  “FLASH ENTRY RING FAULT”
  “FLASH EXIT RING FAULT”
  “PE TRACK PHASE RING FAULT”
  “PE DWELL PHASE RING FAULT”
  “PE EXIT PHASE RING FAULT”

  When the defined phases are in the string parameter more than once:
  “OVLP INC PHASE MULTI FAULT”
  “OVLP MOD PHASE MULTI FAULT”
  “PE TRACK PHASE MULTI FAULT”
  “PE DWELL PHASE MULTI FAULT”
  “PE DWELL PED MULTI FAULT”
  “PE EXIT PHASE MULTI FAULT”
  “PE CYCLING PHASE MULTI FAULT”
  “PE CYCLING PED MULTI FAULT”
  “PHASE xx CONCURRENCY PHASE MULTI FAULT”

  When a defined phase is disabled.
  “START PHASE DISABLE FAULT”
  “FLASH ENTRY DISABLE FAULT”
  “FLASH EXIT DISABLE FAULT”
  "PE TRACK PHASE DISABLE FAULT"
  “PE DWELL PHASE DISABLE FAULT”

“PE EXIT PHASE DISABLE FAULT”

When a peds parent phase is NOT active.
“PE DWELL PED PARENT FAULT”
“PHASE XX CONCURRENCY PHASE DISABLE FAULT”
“SEQ XX RING X PHASE DISABLE FAULT”

When a peds parent phase is NOT active.
“PE DWELL PED PARENT FAULT”

When the defined phases contain an invalid phase number value:
“PHASE xx CONCURRENCY PHASE NUM FAULT”
“SEQ xx RING xx PHASE NUM FAULT”
“START PHASE xx BAD VALUE FAULT”

- The following objects define functionality related to overlap assignments. Consistency checks insure that overlaps specified by these objects may only be active when an included phase (overlapIncludedPhases) is active.
Preempt Track Clear Overlaps (preemptTrackOverlap)
Preempt Dwell Overlaps (preemptDwellOverlap)
Preempt Cycling Overlap (preemptCyclingOverlap)

When an included phase IS NOT defined to be active:
“PE TRACK OVERLAP FAULT”
“PE DWELL OVERLAP FAULT”

When the defined overlaps are in the string parameter more than once:
“PE TRACK OVLP MULTI FAULT”
“PE DWELL OVLP MULTI FAULT”
“PE CYCLING OVLP MULTI FAULT”

- The following objects define functionality related to coordination patterns. Consistency checks ensure that patterns specified by these objects is active.

Pattern Cycle Length (patternCycleTime)
Pattern Offset Time (patternOffsetTime)
Pattern Split Phase (splitPhase)
Pattern Split Time (splitTime)
Pattern Split Coordinated Phase (splitCoordPhase)

When the sum of phase minimum times exceeds the cycle length:
“PATTERN xx PHASE MINS EXCEED CYCLE”

When the sum of the split times exceeds the cycle length:
“PATTERN xx SPLITS EXCEED CYCLE”

When the pattern offset exceeds the cycle length:
“PATTERN xx OFFSET EXCEEDS CYCLE”

When the splits of a ring have no indicated coordinated phase:
“PATTERN xx RING xx HAS NO SYNC PHASE”

When the minimum time of a phase exceeds its split time:
“PATTERN xx PHASE xx MIN EXCEEDS SPLIT”

- When no consistency faults are detected in the data when leaving “transaction” mode, the following shall be written to the fdTransactionError object:
  “NO VERIFICATION ERROR”

Note that the order of the checks is not defined. Therefore, for a given set of 'bad' data, the Error Message between different ASC's may be inconsistent.

- The following objects define functionality related to I/O mapping. Consistency checks insure that the input and output mappings contain valid functions and indexes:

  I/O Map input / output index (ascIOinputMapIOindex, ascIOoutputMapIOindex)
  I/O Map input / output function (ascIOinputMapFunction, ascIOoutputMapFunction)
  I/O Map input / output function index (ascIOinputMapFuncIndex, ascIOoutputMapFuncIndex)

  “FIO INPUT MAP ROW xx INVALID FCTN”
  “FIO INPUT MAP ROW xx INVALID INDX”
  “FIO OUTPUT MAP ROW xx INVALID FCTN”
  “FIO OUTPUT MAP ROW xx INVALID INDX”
  “TS1 INPUT MAP ROW xx INVALID FCTN”
  “TS1 INPUT MAP ROW xx INVALID INDX”
  “TS1 OUTPUT MAP ROW xx INVALID FCTN”
  “TS1 OUTPUT MAP ROW xx INVALID INDX”
  “BIU xx INPUT MAP ROW xx INVALID FCTN”
  “BIU xx INPUT MAP ROW xx INVALID INDX”
  “BIU xx OUTPUT MAP ROW xx INVALID FCTN”
  “BIU xx OUTPUT MAP ROW xx INVALID INDX”
  “SIU xx INPUT MAP ROW xx INVALID FCTN”
  “SIU xx INPUT MAP ROW xx INVALID INDX”
  “SIU xx OUTPUT MAP ROW xx INVALID FCTN”
  “SIU xx OUTPUT MAP ROW xx INVALID INDX”
  “AUX INPUT MAP INVALID FCTN”
  “AUX INPUT MAP INVALID INDX”

- The enabled lanes included in a spatEnabledLanesCommand2 must match one of the entries in the enabledLaneConcurrency object. The enabledLaneConcurrency object defines which revocable lanes can be enabled concurrently (simultaneously). If ALL the enabled lanes in the spatEnabledLanesCommand2 does not match one of the entries, the command is rejected and an error message is created. The value "xx" defines the lane(s) (in the spatEnabledLanesCommand2) requested to be enabled not defined as a concurrent lane in one of the enabledLaneConcurrency objects.

  "ENABLED LANE xx CONCURRENCY FAULT"

  An example: The only value in the enabledLaneConcurrency entry is 05 06 (indicating lanes 5 and 6 can be concurrent). If the value of spatEnabledLanesCommand2 is SET to 05, an error message of ENABLED LANE 06 CONCURRENCY FAULT is provided.

4.4 Class Diagrams

Not applicable.

Section 5
Management Information Base (MIB) [Normative]

Section 5 defines those objects that are specifically used by Actuated Signal Controllers (ASC). The objects are defined using the OBJECT-TYPE macro as specified in RFCs 2578, 2579, and 2580. The text provided from Section 5 through the end of Section 6 (except the section headings) constitutes the standard NTCIP1202-v04 MIB.

All of the objects defined in this NTCIP 1202 v04 reside under the “asc” node of the global naming tree. To aid in object management, the “asc” node has been subdivided into logical categories, each defined by a node under the “asc” node. The individual objects are then located under the appropriate node.

Conformance requirements for any object is determined by the use of the Requirements Traceability Matrix (RTM) in Annex A. To support any defined Requirement, an implementation shall support all objects to which the Requirement traces in the RTM. The value of the STATUS field for every object in the MIB is "mandatory," and indicates that it is mandatory if any associated Requirement is selected.

For all bitmapped objects, if a bit is zero (0), then the referenced function is disabled or not supported, and if a bit is one (1), then the referenced function is enabled or supported.

A computer readable format of this information, called a Management Information Base, is available from NEMA (ntcip@nema.org). The MIB has been verified using SMI SNMP Validator based on libsmi (smilint 0.4.7) (https://www.simpleweb.org/ietf/mibs/validate/).

5.1 MIB Header

```
NTCIP1202-v04 DEFINITIONS ::= BEGIN

-- the following OBJECT IDENTIFIERS are used in the ASC MIB:
IMPORTS
    DisplayString, ifIndex                    FROM RFC1213-MIB-Modified
    SnmpAdminString
                                              FROM SNMP-FRAMEWORK-MIB
                                              -- RFC 3411
    zeroDotZero                               FROM SNMPv2-SMI
    ITSOerString                              FROM ISO26048-1-FieldDevice-TC
    devices                                   FROM NTCIP8004-Transportation

    MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY,
    Integer32, Unsigned32, IpAddress, Counter32        FROM SNMPv2-SMI
                                              -- RFC 2578
    RowStatus, StorageType, VariablePointer
                                              FROM SNMPv2-TC
                                              -- RFC 2579
    MODULE-COMPLIANCE, OBJECT-GROUP
                                              FROM SNMPv2-CONF
                                              -- RFC 2580
;
```

```
-- ***********************************************************************
-- Administrative Objects
-- ***********************************************************************
asc MODULE-IDENTITY
  LAST-UPDATED "202501282300Z"

  ORGANIZATION "NTCIP ASC WG"
  CONTACT-INFO
   "name: NTCIP Coordinator
    email: ntcip@nema.org
    postal:  National Electrical Manufacturers Association
             1300 North 17th Street, Suite 1752
             Rosslyn, VA 22209-3801
             USA"
  DESCRIPTION
  "<Definition> This MIB defines the structure of management information
    under the asc node, which is used to manage Actuated Signal
    Controllers (ASC) Objects.
  <Object Identifier> 1.3.6.1.4.1.1206.4.2.1"
  REVISION "202501282300Z"
  DESCRIPTION "NTCIP 1202 v04 - Change to SMIv2"
  REVISION "202309190500Z"
  DESCRIPTION "NTCIP 1202 v0335 - Last update"
  ::= { devices 1 }

--5.2 Phase Parameters
phase OBJECT-IDENTITY
    STATUS current
    DESCRIPTION  "<Definition> Phase parameters.
        for ASC traffic controllers.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1"

    ::= { asc 1 }

-- This node shall contain objects that configure, monitor or
-- control phase functions for this device.
```

5.2 Phase Parameters

```
phase OBJECT-IDENTITY
    STATUS current
    DESCRIPTION  "<Definition> Phase parameters.
        for ASC traffic controllers.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1"
    ::= { asc 1 }
```

```
-- This node shall contain objects that configure, monitor or
-- control phase functions for this device.
```

5.2.1 Maximum Phases

```
maxPhases OBJECT-TYPE
    SYNTAX Integer32 (2..255)
    UNITS "phase"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION  "<Definition> The Maximum Number of Phases this Controller
        Unit supports. This object indicates the maximum rows which shall
        appear in the phaseTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.1
    <Object Type> S"
    ::= { phase 1 }
```

5.2.2 Phase Table

```
phaseTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhaseEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit phase
parameters. The number of rows in this table is equal to the maxPhases object.

        The objects in this table correspond to the matching objects in the
        first set (phaseSetIndex = 1) of the phaseSetTable.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2"
    ::= { phase 2 }

phaseEntry OBJECT-TYPE
    SYNTAX PhaseEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Controller Unit phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1"
    INDEX { phaseNumber }
    ::= { phaseTable 1 }

PhaseEntry ::= SEQUENCE {
    phaseNumber                     Integer32,
    phaseWalk                       Integer32,
    phasePedestrianClear            Integer32,
    phaseMinimumGreen               Integer32,
```



```
    phasePassage                    Integer32,
    phaseMaximum1                   Integer32,
    phaseMaximum2                   Integer32,
    phaseYellowChange               Integer32,
    phaseRedClear                   Integer32,
    phaseRedRevert                  Integer32,
    phaseAddedInitial               Integer32,
    phaseMaximumInitial             Integer32,
    phaseTimeBeforeReduction        Integer32,
    phaseCarsBeforeReduction        Integer32,
    phaseTimeToReduce               Integer32,
    phaseReduceBy                   Integer32,
    phaseMinimumGap                 Integer32,
    phaseDynamicMaxLimit            Integer32,
    phaseDynamicMaxStep             Integer32,
    phaseStartup                    INTEGER,
    phaseOptions                    Integer32,
    phaseRing                       Integer32,
    phaseConcurrency                OCTET STRING,
    phaseMaximum3                   Integer32,
    phasePedClearDuringVehicleClear Integer32,
    phasePedServiceLimit            Integer32,
    phaseDontWalkRevert             Integer32,
    phasePedAlternateClearance      Integer32,
    phasePedAlternateWalk           Integer32,
    phasePedAdvanceWalkTime         Integer32,
    phasePedDelayTime               Integer32,
    phaseAdvWarnGrnStartTime        Integer32,
    phaseAdvWarnRedStartTime        Integer32,
    phaseAltMinTimeTransition       Integer32,
    phaseWalkDuringTransition       Integer32,
    phasePedClearDuringTransition   Integer32    }
```

5.2.2.1 Phase Number

```
phaseNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "phase"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The phase number for objects in this row. This
        value shall not exceed the maxPhases object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.1"
    ::= { phaseEntry 1 }
```

5.2.2.2 Phase Walk Parameter

```
phaseWalk OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Walk Parameter in seconds. This shall
        control the amount of time the WALK indication shall be displayed.
        This parameter shall be used regardless whether the pedestrian
        indication associated with this phase is for a ped-only phase or for
        a pedestrian indication that runs parallel to a vehicle phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.2
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.2.a"
    ::= { phaseEntry 2 }
```

5.2.2.3 Phase Pedestrian Clear Parameter

```
phasePedestrianClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Pedestrian Clear Parameter in seconds.
        This shall control the duration of the Pedestrian Clearance output
        (if present) and the flashing period of the Don't Walk output.
        This parameter shall be used regardless whether the pedestrian
        indication associated with this phase is for a ped-only phase or for
        a pedestrian indication that runs parallel to a vehicle phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.3
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.2.b"
    ::= { phaseEntry 3 }
```

5.2.2.4 Phase Minimum Green Parameter

```
phaseMinimumGreen OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Minimum Green Parameter in seconds (NEMA
        TS 2 range: 1-255 sec). The first timed portion of the Green interval
        which may be set in consideration of the storage of vehicles between
        the zone of detection for the approach vehicle detector(s) and the
```

```
        stop line.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.4
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.a.(1)"
    ::= { phaseEntry 4 }
```

5.2.2.5 Phase Passage Parameter

```
phasePassage OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Passage Parameter in tenth seconds (0-25.5
        sec). Passage Time, Vehicle Interval, Preset Gap, Vehicle Extension:
        the extensible portion of the Green shall be a function of vehicle
        actuations that occur during the Green interval. The phase shall
        remain in the extensible portion of the Green interval as long as the
        passage timer is not timed out. The timing of this portion of the
        green interval shall be reset with each subsequent vehicle actuation
        and shall not commence to time again until the vehicle actuation is
        removed or the maximum green timer has expired.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.5
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.a.(2)"
    ::= { phaseEntry 5 }
```

5.2.2.6 Phase Maximum Green 1 Parameter

```
phaseMaximum1 OBJECT-TYPE
    SYNTAX Integer32 (0..999)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Maximum 1 Parameter in seconds (NEMA TS 2
        range: 1-255 sec). This time setting shall determine the maximum
        length of time this phase may be held in Green in the presence of a
        serviceable conflicting call. In the absence of a serviceable
        conflicting call the Maximum Green timer shall be held reset unless
        Max Vehicle Recall is enabled for this phase. This is the default
        maximum value to use. It may be overridden via an external input,
        coordMaximumMode, or other method.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.6
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1, 3.5.3.2.1.a.(3) and 3.5.3.5"
    ::= { phaseEntry 6 }
```

5.2.2.7 Phase Maximum Green 2 Parameter

```
phaseMaximum2 OBJECT-TYPE
    SYNTAX Integer32 (0..999)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Maximum 2 Parameter in seconds (NEMA TS 2
        range: 1-255 sec). This time setting shall determine the maximum
        length of time this phase may be held in Green in the presence of a
        serviceable conflicting call. In the absence of a serviceable
        conflicting call the Maximum Green timer shall be held reset unless
        Max Vehicle Recall is enabled for this phase. This may be implemented
        as the max green timer via an external input, coordMaximumMode or
        other method.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.7
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1, 3.5.3.2.1.a.(3), 3.5.3.5
     and 3.5.4.1 (7)"
    ::= { phaseEntry 7 }
```

5.2.2.8 Phase Yellow Change Parameter

```
phaseYellowChange OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Yellow Change Parameter in tenth seconds
        (NEMA TS 2 range: 3-25.5 sec). Following the Green interval of each
        phase the CU shall provide a Yellow Change interval which is timed
        according to the Yellow Change parameter for that phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.8
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.5.a"
    ::= { phaseEntry 8 }
```

5.2.2.9 Phase Red Clearance Parameter

```
phaseRedClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Red Clearance Parameter in tenth seconds
```



```
        (0-25.5 sec). Following the Yellow Change interval for each phase, the
        CU shall provide a Red Clearance interval which is timed according to
        the Red Clearance parameter for that phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.9
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.5.b"
    ::= { phaseEntry 9 }
```

5.2.2.10 Phase Red Revert Parameter

```
phaseRedRevert OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Red revert time parameter in tenths of seconds
        (0-25.5 sec). A minimum RED indication to be timed following the
        Yellow Change interval and prior to the next display of Green on the
        same signal output driver group.

        The unitRedRevert parameter shall act as a minimum red revert time for
        all signal displays. The phaseRedRevert parameter may increase the red
        revert time for a specific phase. If the phaseRedRevert parameter is
        less than the unitRedRevert the unitRedRevert time shall be used.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.10
    <Object Type> P"
    ::= { phaseEntry 10 }
```

5.2.2.11 Phase Added Initial Parameter

```
phaseAddedInitial OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Added Initial Parameter in tenths of
        seconds (0-25.5 sec). Added Initial parameter (Seconds / Actuation)
        shall determine the time by which the variable initial time period
        will be increased from zero with each vehicle actuation received
        during the associated phase Yellow and Red intervals.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.11
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(1).(b)"
    ::= { phaseEntry 11 }
```

5.2.2.12 Phase Maximum Initial Parameter

```
phaseMaximumInitial OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Maximum Initial Parameter in seconds
        (0-255 sec). The maximum value of the variable initial timing period.
        Variable Initial timing shall equal the lesser of [added initial
        (seconds / actuation) * number of actuations] or [ Max Initial ].
        The variable initial time shall not be less than Minimum Green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.12
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.2.1.b.(1).(c)"
    ::= { phaseEntry 12 }
```

5.2.2.13 Phase Time Before Reduction Parameter

```
phaseTimeBeforeReduction OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Time Before Reduction (TBR) Parameter in
        seconds (0-255 sec). The Time Before Reduction period shall begin
        when the phase is Green and there is a serviceable conflicting call.
        If the serviceable conflicting call is removed before completion of
        this time (or time to reduce), the timer shall reset. Upon completion
        of the TBR period or the Cars Before Reduction (CBR) parameter is
        satisfied, whichever occurs first, the linear reduction of the
        allowable gap from the Passage Time shall begin.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.13
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"
    ::= { phaseEntry 13 }
```

5.2.2.14 Phase Cars Before Reduction Parameter

```
phaseCarsBeforeReduction OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "vehicle"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Cars Before Reduction (CBR) Parameter
        (0-255 vehicles). When the phase is Green and the sum of the cars
        waiting (vehicle actuations during Yellow & Red intervals) on
```

```
        serviceable conflicting phases equals or exceeds the CBR parameter or
        the Time Before Reduction (TBR) parameter is satisfied, whichever
        occurs first, the linear reduction of the allowable gap from the
        Passage Time shall begin.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.14
    <Object Type> P"
    ::= { phaseEntry 14 }
```

5.2.2.15 Phase Time To Reduce Parameter

```
phaseTimeToReduce OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Time To Reduce Parameter in seconds
        (0-255 sec). This parameter shall control the rate of reduction of the
        allowable gap between the Passage Time and Minimum Gap setting.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.15
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"
    ::= { phaseEntry 15 }
```

5.2.2.16 Phase Reduce By Parameter

```
phaseReduceBy OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object may be used for volume density gap
        reduction as an alternate to the linear reduction defined by NEMA TS 1
        and TS 2. It contains the tenths of seconds to reduce the gap by
        (0.0 - 25.5 seconds). The frequency of reduction shall produce the
        Minimum Gap after a time equal to the 'phaseTimeToReduce' object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.16
    <Object Type> P"
    ::= { phaseEntry 16 }
```

5.2.2.17 Phase Minimum Gap Parameter

```
phaseMinimumGap OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Minimum Gap Parameter in tenth seconds
```

```
        (0-25.5 sec). The reduction of the allowable gap shall continue until
        the gap reaches a value equal to or less than the minimum gap as set
        on the Minimum Gap control after which the allowable gap shall remain
        fixed at the values set on the Minimum Gap control.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.17
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"
    ::= { phaseEntry 17 }
```

5.2.2.18 Phase Dynamic Max Limit Parameter

```
phaseDynamicMaxLimit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object shall determine either the upper or
        lower limit of the running max in seconds (0-255 sec) during dynamic
        max operation. The normal maximum (i.e. Max1, Max2, etc.) shall
        determine the other limit as follows:
        When this value is larger than the normal maximum, it is the upper limit.
        When this value is smaller than the normal maximum, it is the lower
limit.
        Setting dynamicMaxLimit greater than zero enables dynamic max
            operation with the normal maximum used as the initial maximum
            setting. See dynamicMaxStep for details on dynamic max operation.

        Max Vehicle Recall or a failed detector that is assigned to the
associated phase shall disable dynamic max operation for the phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.18
    <Object Type> P"
    ::= { phaseEntry 18 }
```

5.2.2.19 Phase Dynamic Max Step Parameter

```
phaseDynamicMaxStep OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object shall determine the automatic
adjustment
        to the running max in tenths of seconds (0-25.5). When an ASC decides
        to adjust the running max for the phase, the value of this object shall
        be added or subtracted to the running max per cycle.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.19
```

```
    <Object Type> P"
    ::= { phaseEntry 19 }
```

5.2.2.20 Phase Startup Parameter

```
phaseStartup OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     phaseNotOn (2),
                     greenWalk (3),
                     greenNoWalk (4),
                     yellowChange (5),
                     redClear (6)}
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The Phase Startup parameter is an enumerated
        integer which selects the startup state for each phase after
        restoration of a defined power interruption or activation of the
        external start input. The following entries are defined:
            other: this phase is not enabled (phaseOptions bit 0=0 or
                phaseRing=0) or initializes in a state not defined by
                this standard.
            phaseNotOn: this phase initializes in a Red state (the phase is
                not active and no intervals are timing).
            greenWalk: this phase initializes at the beginning of the minimum
                green and walk timing intervals.
            greenNoWalk: this phase initializes at the beginning of the
                minimum green timing interval.
            yellowChange: this phase initializes at the beginning of the
                Yellow Change interval.
            redClear: this phase initializes at the beginning of the Red
                Clearance interval.

        The value for this phase is the same for all phase sets. There is no
object in the phaseSetTable corresponding to phaseStartup.
        <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.20
        <Object Type> P2"
    REFERENCE "NEMA TS 2 Clause 3.5.5.1 and 3.5.5.12"
    ::= { phaseEntry 20 }
```

5.2.2.21 Phase Options

```
phaseOptions OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition>Optional phase functions
```

(0 = False/Disabled, 1 = True/Enabled)

Bit 15: AddedInitialCalculation - If set (1) the CU shall compare counts from all associated AddedInitial detectors and use the largest count value for the calculations. If clear (0) the CU shall sum all associated AddedInitial detector counts and use this sum for the calculations. Support is optional.

Bit 14: Conditional Service Enable - in multi-ring configurations when set to 1 causes a gapped/maxed phase to conditionally service a preceding actuated vehicle phase when sufficient time remains before max time out of the phase(s) not prepared to terminate. Support is optional.
REFERENCE NEMA TS 2 Clause 3.5.3.9

Bit 13: Actuated Rest-in-Walk - when set to 1 causes an actuated phase to rest in Walk when there is no serviceable conflicting call at the end of Walk Timing or if Max Vehicle Recall is enabled.

Bit 12: Guaranteed Passage - when set to 1 enables an actuated phase operating in volume density mode (using gap reduction) to retain the right of way for the unexpired portion of the Passage time following the decision to terminate the green due to a reduced gap. Support is optional. The value for this phase is the same for all phase sets.

Bit 11: Simultaneous Gap Disable - in multi-ring configurations when set to 1 disables a gapped out phase from reverting to the extensible portion. Support is optional.
REFERENCE NEMA TS 2 Clause 3.5.5.3

Bit 10: Dual Entry Phase - in multi-ring configurations when set to 1 causes the phase to become active upon entry into a concurrency group (crossing a barrier) when no calls exist in its ring within its concurrency group.
REFERENCE NEMA TS 2 Clause 3.5.5.3

Bit 9: Soft Vehicle Recall - when set to 1 causes a call on a phase when all conflicting phases are in green dwell or red dwell and there are no serviceable conflicting calls. Support is optional.

Bit 8: Ped. Recall - when set to 1 causes a recurring pedestrian demand which shall function in the same manner as an external pedestrian call except that it shall not recycle the pedestrian service until a conflicting phase is serviced.
REFERENCE NEMA TS 2 Clause 3.5.3.7

Bit 7: Max Vehicle Recall - when set to 1 causes a call on a phase such that the timing of the Green interval for that phase shall be extended to Maximum Green time.
REFERENCE NEMA TS 2 Clause 3.5.3.5

Bit 6: Min. Vehicle Recall - when set to 1 causes recurring demand for vehicle service on the phase when that phase is not in its Green interval.
REFERENCE NEMA TS 2 Clause 3.5.3.6

Bit 5: Non Lock Detector Memory - when set to 0 will cause the call to be

```
        locked at the beginning of the yellow interval. When set to 1 call
        locking will depend on the detectorOptions object.
        REFERENCE NEMA TS 2 Clause 3.5.3.4
    Bit 4: Non-Actuated 2 - when set to 1 causes a phase to respond to the
        Call To Non-Actuated 2 input (if present) or other method. Support
        is optional.
        REFERENCE NEMA TS 2 Clause 3.5.5.5.8
    Bit 3: Non-Actuated 1 - when set to 1 causes a phase to respond to the
        Call To Non-Actuated 1 input (if present) or other method. Support is
        optional.
        REFERENCE NEMA TS 2 Clause 3.5.5.5.8
    Bit 2: Automatic Flash Exit Phase - The CU shall move immediately to the
        beginning of the phase(s) programmed as Exit Phase(s) when Automatic
        Flash terminates. The value for this phase is the same for all phase
sets. Support is optional.
        REFERENCE NEMA TS 2 Clause 3.9.1.2.1
    Bit 1: Automatic Flash Entry Phase - When Automatic Flash is called, the
        CU shall service the Entry Phase(s), clear to an All Red, then
        initiate flashing operation. The value for this phase is the same for all
phase sets. Support is optional.
        REFERENCE NEMA TS 2 Clause 3.9.1.2.1
    Bit 0: Enabled Phase - provide a means to define whether this phase is
        used in the current configuration. A disabled phase shall not provide
        any outputs nor respond to any phase inputs. The object phaseRing = 0
        has the same effect. The value for this phase is the same for all phase
sets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.22
    <Object Type> P2"
    ::= { phaseEntry 21 }
```

5.2.2.22 Phase Ring

```
phaseRing OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "ring"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase ring number (1..maxRings) that identified
        the ring which contains the associated phase. This value must not
        exceed the maxRings object value. If the ring number is zero, the
        phase is disabled (phaseOptions Bit 0 = 0 has the same effect).

    The value for this phase is the same for all phase sets. There is no object
in the phaseSetTable corresponding to phaseRing.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.22
    <Object Type> P2"
```

```
    ::= { phaseEntry 22 }
```

5.2.2.23 Phase Concurrency

```
phaseConcurrency OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition>Each octet contains a phase number (binary value)
        that may run concurrently with the associated phase. Phases that are
        contained in the same ring may NOT run concurrently.

        The value for this phase is the same for all phase sets. There is no
object in the phaseSetTable corresponding to phaseConcurrency.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.23
    <Object Type> P2"
    ::= { phaseEntry 23 }
```

5.2.2.24 Phase Maximum Green 3 Parameter

```
phaseMaximum3 OBJECT-TYPE
    SYNTAX Integer32 (0..999)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Maximum 3 Parameter in seconds. This time
        setting shall determine the maximum length of time this phase may be
        held in Green in the presence of a serviceable conflicting call. In
        the absence of a serviceable conflicting call, the Maximum Green timer
        shall be held reset unless Max Vehicle Recall is enabled for this
        phase. This may be implemented as the max green timer via an external
        input, coordMaximumMode or other method.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.24
    <Object Type> P"
    ::= { phaseEntry 24 }
```

5.2.2.25 Phase Ped Clearance During Vehicle Clearance Parameter

```
phasePedClearDuringVehicleClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time that the pedestrian clearance
        may extend into the vehicle clearance time (yellow and red) for a
        phase. This parameter is expressed in 0.1 second increments ranging
        from 0.0 to 25.5 seconds.
```

```
        This object was named phaseYellowandRedChangeTimeBeforeEndPedClear in
1202 v03.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.25
    <Object Type> P"
    ::= { phaseEntry 25 }
```

5.2.2.26 Phase Pedestrian Service Limit Parameter

```
phasePedServiceLimit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This parameter indicates whether and how many
        times this phase is allowed to recycle the pedestrian movement during
        a cycle. This parameter is typically used for ped-only, signalized
intersections
        (mostly mid-block) that are within a coordinated roadway. If set to
        '1', no recycle is allowed and the pedestrian movement can be shown
        only up to once. If set to '2', the pedestrian movement can be shown
        up to twice during a cycle, etc. A value of zero means there is no limit.

        This object was named phasePedWalkService in 1202 v03.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.26
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseEntry 26 }
```

5.2.2.27 Phase Don't Walk Revert Parameter

```
phaseDontWalkRevert OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Don't Walk revert time parameter in tenth
        seconds. A minimum DON'T WALK indication to be timed following the
        pedestrian clearance interval prior to the next WALK indication on
        the same signal output driver group.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.27
    <Object Type> P"
    ::= { phaseEntry 27 }
```

5.2.2.28 Phase Alternate Pedestrian Clearance Time Parameter

```
phasePedAlternateClearance OBJECT-TYPE
    SYNTAX Integer32 (0..255)
```

```
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An alternate (replacement) time, in seconds,
        for the duration of the pedestrian clearance output (if present) and
        the flashing DON'T WALK indication.

        This parameter may be used for a parallel pedestrian indication in
        conjunction with a vehicle phase or with a ped-only phase to support
extend pedestrian clearance times.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.28
    <Object Type> P"
    ::= { phaseEntry 28 }
```

5.2.2.29 Phase Alternate Pedestrian Walk Time Parameter

```
phasePedAlternateWalk OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An alternate (replacement) time, in seconds, for
        a pedestrian Walk. This shall control the amount of time the WALK
        indication shall be displayed.

        This parameter may be used for a parallel pedestrian indication in
        conjunction with a vehicle phase or with a ped-only phase to support
extended walk times.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.29
    <Object Type> P"
    ::= { phaseEntry 29 }
```

5.2.2.30 Phase Pedestrian Advance Walk Time Parameter

```
phasePedAdvanceWalkTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time, in tenths of a second from
        0 to 25.5 seconds, that the phase's pedestrian WALK
        indication starts before the start of the phase's vehicle GREEN
indication.

        This parameter is used in response to a pedestrian detector input as
configured in pedestrianDetectorOptions or if pedestrian recall is enabled.
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.30
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseEntry 30 }
```

5.2.2.31 Phase Pedestrian Delay Walk Time Parameter

```
phasePedDelayTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time, in tenths of a second from
        0 to 25.5 seconds, that the phase's pedestrian WALK
        indication starts after the start of the phase's vehicle GREEN
indication.

    This parameter is used in response to a pedestrian detector input.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.31
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseEntry 31 }
```

5.2.2.32 Phase Advance Green Indication Start Time Parameter

```
phaseAdvWarnGrnStartTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time, in tenths of a second for a
        period of 0.0 to 25.5 seconds, that an advance warning signal
        indication is displayed before the start of phase Green. The warning
        signal is placed upstream of the phase's approach and indicates that
        the phase's GREEN indication is about to start or has started.

        The duration of the advance green indication should not exceed the total
amount of vehicle clearance time of the phase(s) that is being terminated prior
to the start of this phase.

        Note: The Advance Warning Green terminates at the end of the green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.32
    <Object Type> P"
    ::= { phaseEntry 32 }
```

5.2.2.33 Phase Advance Red Indication Start Time Parameter

```
phaseAdvWarnRedStartTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time, in tenths of a second for a
        range of 0.0 to 25.5 seconds, prior to the start of the phase's RED
        indication that an advance warning signal, placed upstream of the
        phase's approach, turns on.
        Note: The Advance Warning Red terminates at the end of Red.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.33
    <Object Type> P"
    ::= { phaseEntry 33 }
```

5.2.2.34 Phase Alternate Minimum Green Time During Transitions

```
phaseAltMinTimeTransition OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the alternate minimum
        green time that is used during transitions, in seconds from 1 to 255
        seconds. This object can be applied during transitions or signal
        priority. A value of zero indicates that this object is not used during
        transitions. This value shall not exceed phaseMinimumGreen for this
phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.34
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseEntry 34 }
```

5.2.2.35 Phase Alternate Walk Time During Transitions

```
phaseWalkDuringTransition OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the alternate walk time that
is used during transitions, in seconds from 1 to 255
        seconds. This object can be applied during transitions or signal
priority. A value of zero indicates that this object is not used during
        transitions. This value shall not exceed phaseWalk for this phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.35
```

```
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseEntry 35 }
```

5.2.2.36 Phase Alternate Pedestrian Clearance Time During Transitions

```
phasePedClearDuringTransition OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the alternate minimum
pedestrian clearance time walk time that is used during transitions, in seconds
from 1 to 255 seconds. This object can be applied during transitions or signal
priority. A value of zero indicates that this object is not used during
transitions. This value shall not exceed phasePedestrianClear for this phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.2.1.36
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseEntry 36 }
```

5.2.3 Maximum Phase Groups

```
    maxPhaseGroups OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS   "groups"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Phase Groups (8 Phases per
group) this Controller Unit supports. This value is equal to TRUNCATE [(maxPhases
+ 7) / 8].
    This object indicates the maximum rows which shall appear in the
phaseStatusGroupTable and phaseControlGroupTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.3
    <Object Type> S"
    ::= { phase 3 }
```

5.2.4 Phase Status Group Table

```
phaseStatusGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhaseStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit Phase Output
        (Red, Yellow, & Green) and Call (vehicle & pedestrian) status in
        groups of eight Phases. The number of rows in this table is equal to
        the maxPhaseGroups object.
```

```
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4"
    ::= { phase 4 }

phaseStatusGroupEntry OBJECT-TYPE
    SYNTAX PhaseStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Red, Yellow, & Green Output Status and Vehicle
        and Pedestrian Call for eight Controller Unit Phases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1"
    INDEX   { phaseStatusGroupNumber }
    ::= { phaseStatusGroupTable 1 }


PhaseStatusGroupEntry ::= SEQUENCE {
    phaseStatusGroupNumber      Integer32,
    phaseStatusGroupReds        Integer32,
    phaseStatusGroupYellows     Integer32,
    phaseStatusGroupGreens      Integer32,
    phaseStatusGroupDontWalks   Integer32,
    phaseStatusGroupPedClears   Integer32,
    phaseStatusGroupWalks       Integer32,
    phaseStatusGroupVehCalls    Integer32,
    phaseStatusGroupPedCalls    Integer32,
    phaseStatusGroupPhaseOns    Integer32,
    phaseStatusGroupPhaseNexts  Integer32 }
```

5.2.4.1 Phase Status Group Number

```
phaseStatusGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS   "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The Phase Status Group number for objects in
        this row. This value shall not exceed the maxPhaseGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.1"
    ::= { phaseStatusGroupEntry 1 }
```

5.2.4.2 Phase Status Group Reds

```
phaseStatusGroupReds OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION "<Definition> Phase Red Output Status Mask, when a bit = 1,
        the Phase Red is currently active. When a bit = 0, the Phase Red is
        NOT currently active. This does not include phases that are disabled.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.2
    <Object Type> S"
    ::= { phaseStatusGroupEntry 2 }
```

5.2.4.3 Phase Status Group Yellows

```
phaseStatusGroupYellows OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Phase Yellow Output Status Mask, when a bit = 1,
        the Phase Yellow is currently active. When a bit = 0, the Phase
        Yellow is NOT currently active.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.3
    <Object Type> S"
    ::= { phaseStatusGroupEntry 3 }
```

5.2.4.4 Phase Status Group Greens

```
phaseStatusGroupGreens OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Phase Green Output Status Mask, when a bit = 1,
        the Phase Green is currently active. When a bit = 0, the Phase Green
        is NOT currently active.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
```

```
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.4
    <Object Type> S"
    ::= { phaseStatusGroupEntry 4 }
```

5.2.4.5 Phase Status Group Don't Walks

```
phaseStatusGroupDontWalks OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Phase Dont Walk Output Status Mask, when a
        bit = 1, the Phase Dont Walk is currently active. When a bit = 0,
        the Phase Dont Walk is NOT currently active.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.5
    <Object Type> S"
    ::= { phaseStatusGroupEntry 5 }
```

5.2.4.6 Phase Status Group Pedestrian Clears

```
phaseStatusGroupPedClears OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Phase Ped Clear Output Status Mask, when a
        bit = 1, the Phase Ped Clear is currently active. When a bit = 0,
        the Phase Ped Clear is NOT currently active.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
```

```
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.6
    <Object Type> S"
    ::= { phaseStatusGroupEntry 6 }
```

5.2.4.7 Phase Status Group Walks

```
phaseStatusGroupWalks OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Walk Output Status Mask, when a bit = 1,
        the Phase Walk is currently active. When a bit = 0, the Phase Walk
        is NOT currently active.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.7
    <Object Type> S"
    ::= { phaseStatusGroupEntry 7 }
```

5.2.4.8 Phase Status Group Vehicle Calls

```
phaseStatusGroupVehCalls OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Phase Vehicle Call Status Mask, when a bit = 1,
        the Phase vehicle currently has a call for service, regardless of where
the call was placed from.
        When a bit = 0, the Phase vehicle currently does NOT have a call for
service.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
```

```
        Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.8
    <Object Type> S"
    ::= { phaseStatusGroupEntry 8 }
```

5.2.4.9 Phase Status Group Pedestrian Calls

```
phaseStatusGroupPedCalls OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Phase Pedestrian Call Status Mask, when a
        bit = 1, the Phase pedestrian currently has a call for service,
regardless of where the call was placed from.
        When a bit = 0, the Phase pedestrian currently does NOT have a
        call for service.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.9
    <Object Type> S"
    ::= { phaseStatusGroupEntry 9 }
```

5.2.4.10 Phase Status Group Phase Ons

```
phaseStatusGroupPhaseOns OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Phase On Status Mask, when a bit = 1, the Phase
        is currently active. When a bit = 0, the Phase currently is NOT
        active. The phase is ON during the Green, Yellow, & Red Clearance
        intervals of that phase. It shall be permissible for this
        STATUS to be True (bit=1) during the Red Dwell state.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
```



```
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.10
    <Object Type> S"
    ::= { phaseStatusGroupEntry 10 }
```

5.2.4.11 Phase Status Group Phase Nexts

```
phaseStatusGroupPhaseNexts OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Phase Next Status Mask, when a bit = 1, the
        Phase currently is committed to be NEXT in sequence & remains present
        until the phase becomes active (On/Timing). When a bit = 0, the Phase
        currently is NOT committed to be NEXT in sequence. The phase next to
        be serviced shall be determined at the end of the green interval of
        the terminating phase; except that if the decision cannot be made at
        the end of the Green interval, it shall not be made until after the
        end of all Vehicle Change & Clearance intervals.
            Bit 7: Phase # = (phaseStatusGroupNumber * 8)
            Bit 6: Phase # = (phaseStatusGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseStatusGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseStatusGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseStatusGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseStatusGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseStatusGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.4.1.11
    <Object Type> S"
    ::= { phaseStatusGroupEntry 11 }
```

5.2.5 Phase Control Table

```
phaseControlGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhaseControlGroupEntry
    UNITS "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit Phase Control
        in groups of eight phases. The number of rows in this table is equal
        to the maxPhaseGroups object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5"
    ::= { phase 5 }

phaseControlGroupEntry OBJECT-TYPE
```

```
    SYNTAX PhaseControlGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Phase Control for eight Controller Unit phases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5.1"
    INDEX   { phaseControlGroupNumber }
    ::= { phaseControlGroupTable 1 }

PhaseControlGroupEntry ::= SEQUENCE {
    phaseControlGroupNumber     Integer32,
    phaseControlGroupPhaseOmit  Integer32,
    phaseControlGroupPedOmit    Integer32,
    phaseControlGroupHold       Integer32,
    phaseControlGroupForceOff   Integer32,
    phaseControlGroupVehCall    Integer32,
    phaseControlGroupPedCall    Integer32 }
```

5.2.5.1 Phase Control Group Number

```
phaseControlGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS   "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The Phase Control Group number for objects in
        this row. This value shall not exceed the maxPhaseGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5.1.1"
    ::= { phaseControlGroupEntry 1 }
```

5.2.5.2 Phase Omit Control

```
phaseControlGroupPhaseOmit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
        omit phases from being serviced in the device. When a bit = 1, the
        device shall activate the System Phase Omit control for that phase.
        When a bit = 0, the device shall not activate the System Phase Omit
        control for that phase.
            Bit 7: Phase # = (phaseControlGroupNumber * 8)
            Bit 6: Phase # = (phaseControlGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseControlGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseControlGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseControlGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseControlGroupNumber * 8) - 5
```

```
            Bit 1: Phase # = (phaseControlGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseControlGroupNumber * 8) - 7
        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO
        (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5.1.2
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.3.11.2"
    ::= { phaseControlGroupEntry 2 }
```

5.2.5.3 Pedestrian Omit Control

```
phaseControlGroupPedOmit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
        omit peds from being serviced in the device. When a bit = 1, the
        device shall activate the System Ped Omit control for that phase. When
        a bit = 0, the device shall not activate the System Ped Omit control
        for that phase.
            Bit 7: Phase # = (phaseControlGroupNumber * 8)
            Bit 6: Phase # = (phaseControlGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseControlGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseControlGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseControlGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseControlGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseControlGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseControlGroupNumber * 8) - 7
        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO
        (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5.1.3
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.3.11.3"
    ::= { phaseControlGroupEntry 3 }
```

5.2.5.4 Phase Hold Control

```
phaseControlGroupHold OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
        hold phases in the device. When a bit = 1, the device shall activate
        the System Phase Hold control for that phase. When a bit = 0, the
```

```
        device shall not activate the System Phase Hold control for that phase.
            Bit 7: Phase # = (phaseControlGroupNumber * 8)
            Bit 6: Phase # = (phaseControlGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseControlGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseControlGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseControlGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseControlGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseControlGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseControlGroupNumber * 8) - 7
        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO
        (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5.1.4
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.3.11.1"
    ::= { phaseControlGroupEntry 4 }
```

5.2.5.5 Phase Force Off Control

```
phaseControlGroupForceOff OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to apply force offs on a per
        phase basis. When a bit = 1, the device shall activate the System
        Phase Force Off control for that phase. When a bit = 0, the device
        shall not activate the System Phase Force Off control for that phase.
        When the phase green terminates, the associated bit shall be reset to
zero.
            Bit 7: Phase # = (phaseControlGroupNumber * 8)
            Bit 6: Phase # = (phaseControlGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseControlGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseControlGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseControlGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseControlGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseControlGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseControlGroupNumber * 8) - 7
        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO
        (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5.1.5
    <Object Type> C"
    ::= { phaseControlGroupEntry 5 }
```

5.2.5.6 Phase Vehicle Call Control

```
phaseControlGroupVehCall OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
        place calls for vehicle service in the device. When a bit = 1, the
        device shall place a call for vehicle service on that phase. When a
        bit = 0, the device shall not place a call for vehicle service on
        that phase.
            Bit 7: Phase # = (phaseControlGroupNumber * 8)
            Bit 6: Phase # = (phaseControlGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseControlGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseControlGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseControlGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseControlGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseControlGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseControlGroupNumber * 8) - 7
        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO
        (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5.1.6
    <Object Type> C"
    ::= { phaseControlGroupEntry 6 }
```

5.2.5.7 Phase Pedestrian Call Control

```
phaseControlGroupPedCall OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
        place calls for ped service in the device. When a bit = 1, the device
        shall place a call for ped service on that phase. When a bit = 0, the
        device shall not place a call for ped service on that phase.
            Bit 7: Phase # = (phaseControlGroupNumber * 8)
            Bit 6: Phase # = (phaseControlGroupNumber * 8) - 1
            Bit 5: Phase # = (phaseControlGroupNumber * 8) - 2
            Bit 4: Phase # = (phaseControlGroupNumber * 8) - 3
            Bit 3: Phase # = (phaseControlGroupNumber * 8) - 4
            Bit 2: Phase # = (phaseControlGroupNumber * 8) - 5
            Bit 1: Phase # = (phaseControlGroupNumber * 8) - 6
            Bit 0: Phase # = (phaseControlGroupNumber * 8) - 7
        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO
```

```
        (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.5.1.7
    <Object Type> C"
    ::= { phaseControlGroupEntry 7 }
```

5.2.6 Maximum Phase Sets

```
maxPhaseSets OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION  "<Definition> The Maximum Number of Phase Sets this Controller
        Unit supports. This object indicates the maximum value for the
        phaseSetTable index phaseSetNumber.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.6
    <Object Type> S"
    ::= { phase 6 }
```

5.2.7 Phase Set Table

```
phaseSetTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhaseSetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit phase
        parameters. The number of rows in this table is equal to the maxPhases
        object times the maxPhaseSets object.

        The first set in this table (phaseSetNumber = 1) shall match is
corresponding objects in the phaseTable, unless explicitly defined. For example:
            phaseSetWalk.1.n = phaseWalk.n
            phaseSetPedestrianClear.1.n = phasePedestrianClear.n:
            phaseSetMinimumGreen.1.n = phaseMinimumGreen.n
             etc…
            phaseSetPedClearDuringTransition.1.n =
phasePedClearDuringTransition.n
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7"
    ::= { phase 7 }

phaseSetEntry OBJECT-TYPE
    SYNTAX PhaseSetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a set of Controller Unit phases.
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1"
    INDEX { phaseSetNumber, phaseNumber }
    ::= { phaseSetTable 1 }

PhaseSetEntry ::= SEQUENCE {
    phaseSetNumber                   Integer32,
    phaseSetWalk                     Integer32,
    phaseSetPedestrianClear          Integer32,
    phaseSetMinimumGreen             Integer32,
    phaseSetPassage                  Integer32,
    phaseSetMaximum1                 Integer32,
    phaseSetMaximum2                 Integer32,
    phaseSetYellowChange             Integer32,
    phaseSetRedClear                 Integer32,
    phaseSetRedRevert                Integer32,
    phaseSetAddedInitial             Integer32,
    phaseSetMaximumInitial           Integer32,
    phaseSetTimeBeforeReduction      Integer32,
    phaseSetCarsBeforeReduction      Integer32,
    phaseSetTimeToReduce             Integer32,
    phaseSetReduceBy                 Integer32,
    phaseSetMinimumGap               Integer32,
    phaseSetDynamicMaxLimit          Integer32,
    phaseSetDynamicMaxStep           Integer32,
    phaseSetOptions                  Integer32,
    phaseSetMaximum3                 Integer32,
    phaseSetPedClearDuringVehicleClear Integer32,
    phaseSetPedServiceLimit          Integer32,
    phaseSetDontWalkRevert           Integer32,
    phaseSetPedAlternateClearance    Integer32,
    phaseSetPedAlternateWalk         Integer32,
    phaseSetPedAdvanceWalkTime       Integer32,
    phaseSetPedDelayTime             Integer32,
    phaseSetAdvWarnGrnStartTime      Integer32,
    phaseSetAdvWarnRedStartTime      Integer32,
    phaseSetAltMinTimeTransition     Integer32,
    phaseSetWalkDuringTransition     Integer32,
    phaseSetPedClearDuringTransition Integer32    }
```

5.2.7.1 Phase Set Number

```
phaseSetNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS not-accessible
    STATUS current
```

```
    DESCRIPTION "<Definition> The set number for objects in this row. This
        value shall not exceed the maxPhaseSets object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.1"
    ::= { phaseSetEntry 1 }
```

5.2.7.2 Phase Set Walk Parameter

```
phaseSetWalk OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Walk Parameter in seconds. This shall
        control the amount of time the Walk indication shall be displayed.
        This parameter shall be used regardless whether the pedestrian
        indication associated with this phase is for a ped-only phase or for
        a pedestrian indication that runs parallel to a vehicle phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.2
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.2.a"
    ::= { phaseSetEntry 2 }
```

5.2.7.3 Phase Set Pedestrian Clear Parameter

```
phaseSetPedestrianClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Pedestrian Clear Parameter in seconds.
        This shall control the duration of the Pedestrian Clearance output
        (if present) and the flashing period of the Don't Walk output.

        This parameter shall be used regardless whether the pedestrian
        indication associated with this phase is for a ped-only phase or for
        a pedestrian indication that runs parallel to a vehicle phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.3
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.2.b"
    ::= { phaseSetEntry 3 }
```

5.2.7.4 Phase Set Minimum Green Parameter

```
phaseSetMinimumGreen OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION "<Definition> Phase Minimum Green Parameter in seconds (NEMA
        TS 2 range: 1-255 sec). The first timed portion of the Green interval
        which may be set in consideration of the storage of vehicles between
        the zone of detection for the approach vehicle detector(s) and the
        stop line.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.4
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.a.(1)"
    ::= { phaseSetEntry 4 }
```

5.2.7.5 Phase Set Passage Parameter

```
phaseSetPassage OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Passage Parameter in tenth seconds
        (0-25.5 sec). Passage Time, Vehicle Interval, Preset Gap, Vehicle
        Extension: the extensible portion of the Green shall be a function of
        vehicle actuations that occur during the Green interval. The phase
        shall remain in the extensible portion of the Green interval as long
        as the passage timer is not timed out. The timing of this portion of
        the green interval shall be reset with each subsequent vehicle
        actuation and shall not commence to time again until the vehicle
        actuation is removed or the maximum green timer has expired.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.5
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.a.(2)"
    ::= { phaseSetEntry 5 }
```

5.2.7.6 Phase Set Maximum Green 1 Parameter

```
phaseSetMaximum1 OBJECT-TYPE
    SYNTAX Integer32 (0..999)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Maximum 1 Parameter in seconds (NEMA TS 2
        range: 1-255 sec). This time setting shall determine the maximum
        length of time this phase may be held in Green in the presence of a
        serviceable conflicting call. In the absence of a serviceable
        conflicting call, the Maximum Green timer shall be held reset unless
        Max Vehicle Recall is enabled for this phase. This is the default
        maximum value to use. It may be overridden via an external input,
```

```
        coordMaximumMode, or other method.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.6
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1, 3.5.3.2.1.a.(3) and 3.5.3.5"
    ::= { phaseSetEntry 6 }
```

5.2.7.7 Phase Set Maximum Green 2 Parameter

```
phaseSetMaximum2 OBJECT-TYPE
    SYNTAX Integer32 (0..999)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Maximum 2 Parameter in seconds (NEMA TS 2
        range: 1-255 sec). This time setting shall determine the maximum
        length of time this phase may be held in Green in the presence of a
        serviceable conflicting call. In the absence of a serviceable
        conflicting call the Maximum Green timer shall be held reset unless
        Max Vehicle Recall is enabled for this phase. This may be implemented
        as the max green timer via an external input, coordMaximumMode or
        other method.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.7
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1, 3.5.3.2.1.a.(3), 3.5.3.5
     and 3.5.4.1 (7)"
    ::= { phaseSetEntry 7 }
```

5.2.7.8 Phase Set Yellow Change Parameter

```
phaseSetYellowChange OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Yellow Change Parameter in tenth seconds
        (NEMA TS 2 range: 3-25.5 sec). Following the Green interval of each
        phase the CU shall provide a Yellow Change interval which is timed
        according to the Yellow Change parameter for that phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.8
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.5.a"
    ::= { phaseSetEntry 8 }
```

5.2.7.9 Phase Set Red Clearance Parameter

```
phaseSetRedClear OBJECT-TYPE
```

```
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Red Clearance Parameter in tenth seconds
        (0-25.5 sec). Following the Yellow Change interval for each phase, the
        CU shall provide a Red Clearance interval which is timed according to
        the Red Clearance parameter for that phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.9
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.5.b"
    ::= { phaseSetEntry 9 }
```

5.2.7.10 Phase Set Red Clearance Parameter

```
phaseSetRedRevert OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Red revert time parameter in tenths of seconds
        (0-25.5 sec). A minimum RED indication to be timed following the
        Yellow Change interval and prior to the next display of Green on the
        same signal output driver group.

        The unitRedRevert parameter shall act as a minimum red revert time for
        all signal displays. The phaseSetRedRevert parameter may increase the red
        revert time for a specific phase. If the phaseSetRedRevert parameter is
        less than the unitRedRevert the unitRedRevert time shall be used.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.10
    <Object Type> P"
    ::= { phaseSetEntry 10 }
```

5.2.7.11 Phase Set Added Initial Parameter

```
phaseSetAddedInitial OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Added Initial Parameter in tenths of
        seconds (0-25.5 sec). Added Initial parameter (Seconds / Actuation)
        shall determine the time by which the variable initial time period
        will be increased from zero with each vehicle actuation received
        during the associated phase Yellow and Red intervals.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.11
```

```
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(1).(b)"
    ::= { phaseSetEntry 11 }
```

5.2.7.12 Phase Set Maximum Initial Parameter

```
phaseSetMaximumInitial OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Maximum Initial Parameter in seconds
        (0-255 sec). The maximum value of the variable initial timing period.
        Variable Initial timing shall equal the lesser of [added initial
        (seconds / actuation) * number of actuations] or [ Max Initial ].
        The variable initial time shall not be less than Minimum Green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.12
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.2.1.b.(1).(c)"
    ::= { phaseSetEntry 12 }
```

5.2.7.13 Phase Set Time Before Reduction Parameter

```
phaseSetTimeBeforeReduction OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Time Before Reduction (TBR) Parameter in
        seconds (0-255 sec). The Time Before Reduction period shall begin
        when the phase is Green and there is a serviceable conflicting call.
        If the serviceable conflicting call is removed before completion of
        this time (or time to reduce), the timer shall reset. Upon completion
        of the TBR period or the CarsBeforeReduction (CBR) parameter is
        satisfied, whichever occurs first, the linear reduction of the
        allowable gap from the Passage Time shall begin.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.13
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"
    ::= { phaseSetEntry 13 }
```

5.2.7.14 Phase Set Cars Before Reduction Parameter

```
phaseSetCarsBeforeReduction OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "vehicle"
```

```
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Cars Before Reduction (CBR) Parameter
        (0-255 vehicles). When the phase is Green and the sum of the cars
        waiting (vehicle actuations during Yellow & Red intervals) on
        serviceable conflicting phases equals or exceeds the CBR parameter or
        the Time Before Reduction (TBR) parameter is satisfied, whichever
        occurs first, the linear reduction of the allowable gap from the
        Passage Time shall begin.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.14
    <Object Type> P"
    ::= { phaseSetEntry 14 }
```

5.2.7.15 Phase Set Time to Reduce Parameter

```
phaseSetTimeToReduce OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Time To Reduce Parameter in seconds
        (0-255 sec). This parameter shall control the rate of reduction of the
        allowable gap between the Passage Time and Minimum Gap setting.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.15
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"
    ::= { phaseSetEntry 15 }
```

5.2.7.16 Phase Set Reduce By Parameter

```
phaseSetReduceBy OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object may be used for volume density gap
        reduction as an alternate to the linear reduction defined by NEMA TS 1
        and TS 2. It contains the tenths of seconds to reduce the gap by
        (0.0 - 25.5 seconds). The frequency of reduction shall produce the
        Minimum Gap after a time equal to the 'phaseSetTimeToReduce' object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.16
    <Object Type> P"
    ::= { phaseSetEntry 16 }
```


5.2.7.17 Phase Set Minimum Gap Parameter

```
phaseSetMinimumGap OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Minimum Gap Parameter in tenth seconds
        (0-25.5 sec). The reduction of the allowable gap shall continue until
        the gap reaches a value equal to or less than the minimum gap as set
        on the Minimum Gap control after which the allowable gap shall remain
        fixed at the values set on the Minimum Gap control.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.17
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.b.(2)"
    ::= { phaseSetEntry 17 }
```

5.2.7.18 Phase Set Dynamic Max Limit

```
phaseSetDynamicMaxLimit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object shall determine either the upper or
        lower limit of the running max in seconds (0-255 sec) during dynamic
        max operation. The normal maximum (i.e. phaseSetMaximum1,
phaseSetMaximum2, etc.) shall
        determine the other limit as follows:
        When this value is larger than the normal maximum, it is the upper limit.
        When this value is smaller than the normal maximum, it is the lower
limit.
        Setting phaseSetDynamicMaxLimit greater than zero enables dynamic max
            operation with the normal maximum used as the initial maximum
            setting. See phaseSetDynamicMaxStep for details on dynamic max
operation.

        Max Vehicle Recall or a failed detector that is assigned to the
associated
            phase shall disable dynamic max operation for the phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.18
    <Object Type> P"
    ::= { phaseSetEntry 18 }
```

5.2.7.19 Phase Set Dynamic Max Step

```
phaseSetDynamicMaxStep OBJECT-TYPE
```

```
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object shall determine the automatic 
adjustment
        to the running max in tenths of seconds (0-25.5). When an ASC decides
        to adjust the running max for the phase, the value of this object shall
        be added or subtracted to the running max per cycle.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.19
    <Object Type> P"
    ::= { phaseSetEntry 19 }
```

5.2.7.20 Phase Set Options

```
phaseSetOptions OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition>Optional phase functions
        (0 = False/Disabled, 1 = True/Enabled)
    Bit 15: Not used (AddedInitialCalculation only in the object phaseOptions)
    Bit 14: Conditional Service Enable - in multi-ring configurations when set
        to 1 causes a gapped/maxed phase to conditionally service a preceding
        actuated vehicle phase when sufficient time remains before max time out
        of the phase(s) not prepared to terminate. Support is optional.
        REFERENCE NEMA TS 2 Clause 3.5.3.9
    Bit 13: Actuated Rest-in-Walk - when set to 1 causes an actuated phase to
        Rest-in-Walk when there is no serviceable conflicting call at the end
        of Walk Timing or if Max Vehicle Recall is enabled.
    Bit 12: Reserved -- Not Used (Guaranteed Passage only in the object 
phaseOptions)
    Bit 11: Simultaneous Gap Disable - in multi-ring configurations when set
        to 1 disables a gapped out phase from reverting to the extensible
        portion. Support is optional.
        REFERENCE NEMA TS 2 Clause 3.5.5.3
    Bit 10: Dual Entry Phase - in multi-ring configurations when set to 1
        causes the phase to become active upon entry into a concurrency group
        (crossing a barrier) when no calls exist in its ring within its
        concurrency group.
        REFERENCE NEMA TS 2 Clause 3.5.5.3
    Bit 9: Soft Vehicle Recall - when set to 1 causes a call on a phase when
        all conflicting phases are in green dwell or red dwell and there are
        no serviceable conflicting calls. Support is optional.
    Bit 8: Ped. Recall - when set to 1 causes a recurring pedestrian demand
        which shall function in the same manner as an external pedestrian call
```

```
        except that it shall not recycle the pedestrian service until a
        conflicting phase is serviced.
        REFERENCE NEMA TS 2 Clause 3.5.3.7
    Bit 7: Max Vehicle Recall - when set to 1 causes a call on a phase such
        that the timing of the Green interval for that phase shall be extended
        to Maximum Green time.
        REFERENCE NEMA TS 2 Clause 3.5.3.5
    Bit 6: Min. Vehicle Recall - when set to 1 causes recurring demand for
        vehicle service on the phase when that phase is not in its Green
        interval.
        REFERENCE NEMA TS 2 Clause 3.5.3.6
    Bit 5: Non Lock Detector Memory - when set to 0 will cause the call to be
        locked at the beginning of the yellow interval. When set to 1 call
        locking will depend on the detectorOptions object.
        REFERENCE NEMA TS 2 Clause 3.5.3.4
    Bit 4: Non-Actuated 2 - when set to 1 causes a phase to respond to the
        Call To Non-Actuated 2 input (if present) or other method. Support
        is optional.
        REFERENCE NEMA TS 2 Clause 3.5.5.5.8
    Bit 3: Non-Actuated 1 - when set to 1 causes a phase to respond to the
        Call To Non-Actuated 1 input (if present) or other method. Support is
        optional.
        REFERENCE NEMA TS 2 Clause 3.5.5.5.8
    Bit 2: Reserved --Not used (Automatic Flash Exit Phase only in object
phaseOptions)
    Bit 1: Reserved --Not used (Automatic Flash Entry only in the object
phaseOptions)
    Bit 0: Reserved --Not Used (Enabled Phase only in the object phaseOptions)
    A SET of a 'reserved' bit to a value other than zero (0) shall return a
wrongValue(10) error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.20
    <Object Type> P"
    ::= { phaseSetEntry 20 }
```

5.2.7.21 Phase Set Maximum 3 Parameter

```
phaseSetMaximum3 OBJECT-TYPE
    SYNTAX Integer32 (0..999)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Phase Maximum 3 Parameter in seconds. This time
        setting shall determine the maximum length of time this phase may be
        held in Green in the presence of a serviceable conflicting call. In
        the absence of a serviceable conflicting call, the Maximum Green timer
        shall be held reset unless Max Vehicle Recall is enabled for this
```

```
        phase. This may be implemented as the max green timer via an external
        input, coordMaximumMode, or other method.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.21
    <Object Type> P"
    ::= { phaseSetEntry 21 }
```

5.2.7.22 Phase Set Maximum 3 Parameter

```
phaseSetPedClearDuringVehicleClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time that the pedestrian clearance
        may extend into the vehicle clearance time (yellow and red) for a
        phase. This parameter is expressed in 0.1 second increments ranging
        from 0.0 to 25.5 seconds.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.22
    <Object Type> P"
    ::= { phaseSetEntry 22 }
```

5.2.7.23 Phase Set Pedestrian Service Limit

```
phaseSetPedServiceLimit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This parameter indicates whether and how many
        times this phase is allowed to recycle the pedestrian movement during
        a cycle. This parameter is typically used for ped-only, signalized
intersections
        (mostly mid-block) that are within a coordinated roadway. If set to
        '1', no recycle is allowed and the pedestrian movement can be shown
        only up to once. If set to '2', the pedestrian movement can be shown
        up to twice during a cycle, etc. A value of zero means there is no limit.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.23
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseSetEntry 23 }
```

5.2.7.24 Phase Set Pedestrian Don’t Walk Revert Parameter

```
phaseSetDontWalkRevert OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
```



```
    STATUS current
    DESCRIPTION "<Definition> Don't Walk revert time parameter in tenth
        seconds. A minimum DON'T WALK indication to be timed following the
        pedestrian clearance interval prior to the next WALK indication on
        the same signal output driver group.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.24
    <Object Type> P"
    ::= { phaseSetEntry 24 }
```

5.2.7.25 Phase Set Alternate Pedestrian Clearance Time Parameter

```
phaseSetPedAlternateClearance OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An alternate (replacement) time, in seconds, for
        the duration of the pedestrian clearance output (if present) and the
        flashing period of the DON'T WALK output.

        This parameter may be used for a parallel pedestrian indication in
        conjunction with a vehicle phase or with a ped-only phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.25
    <Object Type> P"
    ::= { phaseSetEntry 25 }
```

5.2.7.26 Phase Set Alternate Pedestrian Walk Time Parameter

```
phaseSetPedAlternateWalk OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An alternate (replacement) time, in seconds, for
        a pedestrian walk. This shall control the amount of time the Walk
        indication shall be displayed.

        This parameter may be used for a parallel pedestrian indication in
        conjunction with a vehicle phase or with a ped-only phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.26
    <Object Type> P"
    ::= { phaseSetEntry 26 }
```

5.2.7.27 Phase Set Pedestrian Advance Walk Time Parameter

```
phaseSetPedAdvanceWalkTime OBJECT-TYPE
```



```
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time, in tenths of a second from
        0 to 25.5 seconds, that the phase's pedestrian Walk
        indication starts before the start of the phase's vehicle GREEN
indication.

        This parameter is used as the result of a pedestrian detector input as
configured in pedestrianSetDetectorOptions or if pedestrian recall is enabled.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.27
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseSetEntry 27 }
```

5.2.7.28 Phase Set Pedestrian Delay Walk Time Parameter

```
phaseSetPedDelayTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time, in tenths of a second from
        0 to 25.5 seconds, that the phase's pedestrian Walk
        indication starts after the start of the phase's vehicle GREEN
indication.

        This parameter is used as result of a pedestrian detector input.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.28
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseSetEntry 28 }
```

5.2.7.29 Phase Set Advance Green Indication Start Time Parameter

```
phaseSetAdvWarnGrnStartTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time, in tenths of a second for a
        period of 0.0 to 25.5 seconds, that an advance warning signal
        indication is displayed before the start of phase Green. The warning
        signal is placed upstream of the phase's approach and indicates that
        the phase's GREEN indication is about to start or has started.
```

```
        The value of this object should not exceed the total amount of
        clearance time of the phase(s) that is being terminated prior to the
        start of this phase.

        Note: The Advance Warning Green terminates at the end of the green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.29
    <Object Type> P"
    ::= { phaseSetEntry 29 }
```

5.2.7.30 Phase Set Advance Red Indication Start Time Parameter

```
phaseSetAdvWarnRedStartTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The amount of time, in tenths of a second for a
        range of 0.0 to 25.5 seconds, prior to the start of the phase's RED
        indication that an advance warning signal, placed upstream of the
        phase's approach, turns on.

        Note: The Advance Warning Red terminates at the end of Red.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.30
    <Object Type> P"
    ::= { phaseSetEntry 30 }
```

5.2.7.31 Phase Set Alternate Minimum Green Time During Transitions

```
phaseSetAltMinTimeTransition OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the alternate minimum green
        time that is used during transitions, in seconds from 1 to 255 seconds.
        This object can be applied during transitions or signal priority. A
        value of 0 indicates that this object is not used during transitions.
        This value shall not exceed phaseSetMinimumGreen for this phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.31
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseSetEntry 31 }
```

5.2.7.32 Phase Set Alternate Walk Time During Transitions

```
phaseSetWalkDuringTransition OBJECT-TYPE
```

```
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the alternate walk time that
is used during transitions, in seconds from 1 to 255
        seconds. This object can be applied during transitions or signal
priority. A value of zero indicates that this object is not used during
        transitions. This value shall not exceed phaseSetWalk.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.32
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseSetEntry 32 }
```

5.2.7.33 Phase Set Alternate Pedestrian Clearance Time During Transitions

```
phaseSetPedClearDuringTransition OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the alternate minimum
pedestrian clearance time walk time that is used during transitions, in seconds
from 1 to 255 seconds. This object can be applied during transitions or signal
priority. A value of zero indicates that this object is not used during
transitions. This value shall not exceed phaseSetPedestrianClear.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.7.1.33
    <Object Type> P"
    DEFVAL { 0 }
    ::= { phaseSetEntry 33 }
```

5.3 Detector Parameters

```
detector OBJECT IDENTIFIER
    ::= { asc 2 }

-- This defines a node for supporting detector objects.
```

5.3.1 Maximum Vehicle Detectors

```
maxVehicleDetectors OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "detector"
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION "<Definition> The Maximum Number of Vehicle Detectors this
        Controller Unit supports. This object indicates the maximum rows
        which shall appear in the vehicleDetectorTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.1
    <Object Type> S"
    ::= { detector 1 }
```

5.3.2 Vehicle Detector Parameter Table

```
vehicleDetectorTable OBJECT-TYPE
    SYNTAX SEQUENCE OF VehicleDetectorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit vehicle
        detector parameters. The number of rows in this table is equal to the
        maxVehicleDetectors object.

        The objects in this table correspond to the matching objects in the
        first set (vehicleDetectorSetNumber = 1) of the vehicleDetectorSetTable.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2"
    ::= { detector 2 }

vehicleDetectorEntry OBJECT-TYPE
    SYNTAX VehicleDetectorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Controller Unit
        detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1"
    INDEX   { vehicleDetectorNumber }
    ::= { vehicleDetectorTable 1 }

VehicleDetectorEntry ::= SEQUENCE {
    vehicleDetectorNumber               Integer32,
    vehicleDetectorOptions              Integer32,
    vehicleDetectorDescription          SnmpAdminString,
    vehicleDetectorCallPhase            Integer32,
    vehicleDetectorSwitchPhase          Integer32,
    vehicleDetectorDelay                Integer32,
    vehicleDetectorExtend               Integer32,
    vehicleDetectorQueueLimit           Integer32,
    vehicleDetectorNoActivity           Integer32,
    vehicleDetectorMaxPresence          Integer32,
    vehicleDetectorErraticCounts        Integer32,
    vehicleDetectorFailTime             Integer32,
```



```
    vehicleDetectorAlarms                  Integer32,
    vehicleDetectorReportedAlarms          Integer32,
    vehicleDetectorReset                   Integer32,
    vehicleDetectorOptions2                Integer32,
    vehicleDetectorPairedDetector          Integer32,
    vehicleDetectorPairedDetectorSpacing   Integer32,
    vehicleDetectorAvgVehicleLength        Integer32,
    vehicleDetectorLength                  Integer32,
    vehicleDetectorTravelMode              INTEGER }
```

5.3.2.1 Vehicle Detector Number

```
vehicleDetectorNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "detector"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The vehicle detector number for objects in this
        row. The value shall not exceed the maxVehicleDetectors object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.1"
    ::= { vehicleDetectorEntry 1 }
```

5.3.2.2 Vehicle Detector Options Parameter

```
vehicleDetectorOptions OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Vehicle Detector Options Parameter as follows
        (0=Disabled, 1=Enabled):
    Bit 7: Call - if Enabled, the CU shall place a demand for vehicular
        service on the assigned phase when the phase is not timing the green
        interval and an actuation is present.
    Bit 6: Queue - if Enabled, the CU shall extend the green interval of the
        assigned phase until a gap occurs (no actuation) or until the green
        has been active longer than the vehicleDetectorQueueLimit time.
        This is optional.
    Bit 5: AddedInitial - if Enabled, the CU shall accumulate detector
        actuation counts for use in the added initial calculations. Counts
        shall be accumulated from the beginning of the yellow interval to the
        beginning of the green interval.
    Bit 4: Passage - if Enabled, the CU shall maintain a reset to the
        associated phase passage timer for the duration of the detector
        actuation when the phase is green.
    Bit 3: Red Lock Call - if Enabled, the detector will lock a call to the
```



```
        assigned phase if an actuation occurs while the phase is not timing
        Green or Yellow. This mode is optional.
    Bit 2: Yellow Lock Call - if Enabled, the detector will lock a call to
        the assigned phase if an actuation occurs while the phase is not
        timing Green.
    Bit 1: Reserved. This bit was previously used to enable occupancy detection.
    Bit 0: Reserved. This bit was previously used to enable volume detection.

    A SET of both bits 2 & 3 = 1 shall result in bit 2=1 and bit 3=0.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.2
    <Object Type> P"
    ::= { vehicleDetectorEntry 2}
```

5.3.2.3 Vehicle Detector Description

```
vehicleDetectorDescription OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A textual string indicate a user description for
the vehicle detector (e.g. location). This value is the same for all vehicle
detector sets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.3
    <Object Type> P"
    ::= { vehicleDetectorEntry 3 }
```

5.3.2.4 Vehicle Detector Call Phase Parameter

```
vehicleDetectorCallPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "phase"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object contains assigned phase number for the
detector input associated with this row. A value of zero indicates this detector
does not call a phase. The value shall not exceed the value of maxPhases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.4
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.5.5.4 and 3.5.5.5.5"
    ::= { vehicleDetectorEntry 4 }
```

5.3.2.5 Vehicle Detector Switch Phase Parameter

```
vehicleDetectorSwitchPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "phase"
```

```
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Switch Phase Parameter (i.e., Phase
        Number). The phase to which a vehicle detector actuation shall be
        switched when the assigned phase is Yellow or Red and the Switch Phase
        is Green. The value shall not exceed the value of maxPhases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.5
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.5.5.4.c"
    ::= { vehicleDetectorEntry 5 }
```

5.3.2.6 Vehicle Detector Delay Parameter

```
vehicleDetectorDelay OBJECT-TYPE
    SYNTAX Integer32 (0..2550)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Delay Parameter in tenth seconds
        (0-255.0 sec). The period a detector actuation (input recognition)
        shall be delayed when the phase is not Green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.6
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.5.5.4.a"
    ::= { vehicleDetectorEntry 6 }
```

5.3.2.7 Vehicle Detector Delay Parameter

```
vehicleDetectorExtend OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Extend Parameter in tenth seconds
        (0-25.5 sec). The period a vehicle detector actuation (input duration)
        shall be extended from the point of termination, when the phase is
        Green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.7
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.5.5.4.b"
    ::= { vehicleDetectorEntry 7 }
```

5.3.2.8 Vehicle Detector Queue Limit

```
vehicleDetectorQueueLimit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
```



```
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Queue Limit parameter in seconds
        (0-255 sec). The length of time that an actuation from a queue
        detector may continue into the phase green. This time begins when the
        phase becomes green and when it expires any associated detector inputs
        shall be ignored. This time may be shorter due to other overriding
        device parameters (i.e. Maximum time, Force Offs, ...).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.8
    <Object Type> P"
    ::= { vehicleDetectorEntry 8 }
```

5.3.2.9 Vehicle Detector No Activity Parameter

```
vehicleDetectorNoActivity OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector No Activity diagnostic Parameter in
        minutes (1-65535 min.). If an active detector does not exhibit an
        actuation in the specified period, it is considered a fault by the
        diagnostics and the detector is classified as Failed. A value of zero for
        this object shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.9
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.1"
    ::= { vehicleDetectorEntry 9 }
```

5.3.2.10 Vehicle Detector Maximum Presence Parameter

```
vehicleDetectorMaxPresence OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Maximum Presence diagnostic Parameter
        in minutes (0-255 min.). If an active detector exhibits continuous
        detection for too long a period, it is considered a fault by the
        diagnostics and the detector is classified as Failed. A value of zero for
        this object shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.10
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.2"
    ::= { vehicleDetectorEntry 10 }
```

5.3.2.11 Vehicle Detector Erratic Counts Parameter

```
vehicleDetectorErraticCounts OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "count"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Erratic Counts diagnostic Parameter in
        counts/minute (0-255 cpm). If an active detector exhibits excessive
        actuations, it is considered a fault by the diagnostics and the
        detector is classified as Failed. A value of zero for this object shall
        disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.11
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.3"
    ::= { vehicleDetectorEntry 11 }
```

5.3.2.12 Vehicle Detector Fail Time Parameter

```
vehicleDetectorFailTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Fail Time in seconds (0..255 sec). If a
        detector diagnostic indicates that the associated detector input is
        failed, then a call shall be placed on the associated phase during all
        non-green intervals. When each green interval begins the call shall be
        maintained for the length of time specified by this object and then
        removed. If the value of this object equals the maximum value (255)
        then a constant call shall be placed on the associated phase (max
        recall). If the value of this object equals zero, then no call shall be
        placed on the associated phase for any interval (no recall). Compliant
        devices may support a limited capability for this object (i.e. only
        max recall). At a minimum, the max recall setting must be supported.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.12
    <Object Type> P"
    ::= { vehicleDetectorEntry 12 }
```

5.3.2.13 Vehicle Detector Alarms

```
vehicleDetectorAlarms OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION "<Definition> This object shall return indications of detector
        alarms as identified by the CU. Detector Alarms are indicated as follows:

        Bit 7: Other Fault - The detector has failed due to some other cause.
        Bit 6: Reserved.
        Bit 5: Reserved.
        Bit 4: Configuration Fault - Detector is assigned but is not supported.
        Bit 3: Communications Fault - Communications to the device (if
                present) have failed.
        Bit 2: Erratic Output Fault - This detector has been flagged as
                non-operational due to erratic outputs (excessive counts) by
                the CU detector diagnostic.
        Bit 1: Max Presence Fault - This detector has been flagged as
                non-operational due to a presence indicator that exceeded the
                maximum expected time by the CU detector diagnostic.
        Bit 0: No Activity Fault - This detector has been flagged as
                non-operational due to lower than expected activity by the CU
                detector diagnostic.

        Once set a bit shall maintain its state as long as the condition
        exists. The bit shall clear when the condition no longer exists.

        The value of this object is the same for all vehicle detector sets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.13
    <Object Type> S"
    ::= { vehicleDetectorEntry  13 }
```

5.3.2.14 Vehicle Detector Reported Alarms

```
vehicleDetectorReportedAlarms OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object shall return detector device
        reported alarms communicated to the CU through an external device.

        Detector Alarms are indicated as follows:
        Bit 7: Reserved.
        Bit 6: Reserved.
        Bit 5: Reserved.
        Bit 4: Excessive Change Fault - This detector has been flagged as
                non-operational due to an inductance change that exceeded
                expected values.
        Bit 3: Shorted Loop Fault - This detector has been flagged as
```

```
            non-operational due to a shorted loop wire.
        Bit 2: Open Loop Fault - This detector has been flagged as
            non-operational due to an open loop (broken wire).
        Bit 1: Watchdog Fault - This detector has been flagged as
            non-operational due to a watchdog error.
        Bit 0: Other - This detector has been flagged as non-operational due
            to some other error.

        Once set a bit shall maintain its state as long as the condition
        exists. The bit shall clear when the condition no longer exists.

        The value of this object is the same for all vehicle detector sets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.14
    <Object Type> S"
    ::= { vehicleDetectorEntry  14 }
```

5.3.2.15 Vehicle Detector Reset

```
vehicleDetectorReset OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object when set to TRUE (one) shall cause
        the CU to command the associated detector to reset. This object shall
        automatically return to FALSE (zero) after the CU has issued the reset
        command.

        Note: this may affect other detector (detector channels) that are
        physically attached to a common reset line.

        There is no object in the vehicleDetectorSetTable corresponding to
        vehicleDetectorReset.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.15
    <Object Type> C"
    ::= { vehicleDetectorEntry 15 }
```

5.3.2.16 Vehicle Detector Options 2

```
vehicleDetectorOptions2 OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A bit-mapped value as defined below for
        configuring detector options.
    <Format>
        Bit 7: Reserved
```

```
        Bit 6: Reserved
        Bit 5: Reserved
        Bit 4: Reserved
        Bit 3    0=DISABLED, 1=ENABLED  Connected Vehicle Detector: When enabled,
                    this detector is assigned to a Connected Vehicle Detector as
                    configured in the cvDetectorTable. Any CV Application
                    Parameters, such as Red Light Violation Warning and Assured
                    Green Period, apply if this detector is actuated.
        Bit 2    0=CUSTOM, 1=NTCIP  Default Detector Speed Mode Option. For a
                    vehicle detector operating in pairs, this option is used
                    when there is an error on one of the paired detectors. It
                    identifies how the controller should calculate speed
                    without the other detector. CUSTOM indicates a
                    manufacturer specific calculation. NTCIP indicates the use
                    of the calculation Speed = (Average Vehicle Length +
                    Detector Length) / Detect Time.
        Bit 1    0=TRAIL, 1=LEAD  Detector Placement Option. For a vehicle
                    detector operating in pairs, this option indicates the
                    leading and trailing detectors. LEAD indicates that the
                    detector is the leading detector of the pair. TRAIL
                    indicates that the detector is a trailing detector in
                    the pair.
        Bit 0    0=DISABLED, 1=ENABLED  Enable Speed Data. If enabled, the
                    detector is used to collect speed data (See
                    volumeOccupancyTable and detectorAvgSpeed). This
                    capability may not be supported on all detector inputs to
                    a device.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.16
    <Object Type> P"
    ::= { vehicleDetectorEntry 16 }
```

5.3.2.17 Vehicle Detector Paired Detector

```
vehicleDetectorPairedDetector OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This is a detector identifier
        (vehicleDetectorNumber) that is used to determine speed. A value of zero
        indicates there is no paired detector. Setting this value will
        automatically add this detector as the given detector's
        vehicleDetectorPairedDetector.  This value shall not exceed
maxVehicleDetectors.

        The value of this object is the same for all vehicle detector sets.
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.17
    <Object Type> P"
    DEFVAL { 0 }
    ::= { vehicleDetectorEntry 17 }
```

5.3.2.18 Vehicle Detector Paired Detector Spacing

```
vehicleDetectorPairedDetectorSpacing OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "centimeter"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This parameter allows the user to set the
        spacing, in centimeters between paired detectors for use in
        calculating vehicle speeds. This parameter is measured from the
        leading edge of one detector to the leading edge of the paired
        detector. A value of zero indicates there is no paired detector.

        The value of this object is the same for all vehicle detector sets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.18
    <Object Type> P"
    DEFVAL { 0 }
    ::= { vehicleDetectorEntry 18 }
```

5.3.2.19 Vehicle Detector Average Vehicle Length

```
vehicleDetectorAvgVehicleLength OBJECT-TYPE
    SYNTAX Integer32 (1..4000)
    UNITS "centimeter"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This parameter allows the user to set the
        average vehicle length for use in determining speed and
        classification. This allows for a range of lengths between 0.01 meters
        to 40 meters in length.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.19
    <Object Type> P"
    ::= { vehicleDetectorEntry 19 }
```

5.3.2.20 Vehicle Detector Average Vehicle Length

```
vehicleDetectorLength OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    UNITS "centimeter"
    MAX-ACCESS read-write
    STATUS current
```



```
    DESCRIPTION "<Definition> This parameter allows the user to set the length
        of the detection zone. In the case of a loop detector, this is the
        length of the loop.

        Values 01 to 4000 are used to represent the length.
        This allows for a range of lengths between 0.01 meters to 40 meters in
        length. The value of 65535 shall be returned to represent no length
        set. Values 4001 to 65534 are not used.

        The value of this object is the same for all vehicle detector sets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.20
    <Object Type> P"
    DEFVAL { 65535 }
    ::= { vehicleDetectorEntry 20 }
```

5.3.2.21 Vehicle Detector Travel Mode

```
vehicleDetectorTravelMode OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     vehicle (2),
                     transit (3),
                     bicycle (4) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This parameter allows the user to identify
        detectors for specific types of travel modes.
        other:   refers to a detector for a travel type not defined in this
standard
        vehicle: refers to a detector identified for vehicles.
        transit: refers to a detector identified for transit vehicles.
        bicycle: refers to a detector identified for bicycles.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.21
    <Object Type> P"
    DEFVAL { 2 }
    ::= { vehicleDetectorEntry 21 }
```

5.3.3 Maximum Vehicle Detector Groups

```
maxVehicleDetectorGroups OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The maximum number of detector groups
        (8 detectors per group) this device supports. This value is equal to
        TRUNCATE [(maxVehicleDetectors + 7 ) / 8]. This object indicates the
```

```
        maximum number of rows which shall appear in the
        vehicleDetectorStatusGroupTable and the vehicleDetectorControlGroupTable.

        This object was named maxVehicleDetectorStatusGroups in 1202 v03.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.3
    <Object Type> S"
    ::= { detector 3 }
```

5.3.4 Vehicle Detector Status Group Table

```
vehicleDetectorStatusGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF VehicleDetectorStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing detector status in groups of
        eight detectors. The number of rows in this table is equal to the
        maxVehicleDetectorGroups object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.4"
    ::= { detector 4 }

vehicleDetectorStatusGroupEntry OBJECT-TYPE
    SYNTAX VehicleDetectorStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A group (row) of detector status.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.4.1"
    INDEX  { vehicleDetectorStatusGroupNumber }
    ::= { vehicleDetectorStatusGroupTable  1 }

VehicleDetectorStatusGroupEntry ::= SEQUENCE {
    vehicleDetectorStatusGroupNumber    Integer32,
    vehicleDetectorStatusGroupActive    Integer32,
    vehicleDetectorStatusGroupAlarms    Integer32 }
```

5.3.4.1 Vehicle Detector Status Group Number

```
vehicleDetectorStatusGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The detector status group number for objects in
        this row. This value shall not exceed the
        maxVehicleDetectorGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.4.1.1"
```

```
    ::= { vehicleDetectorStatusGroupEntry 1 }
```

5.3.4.2 Vehicle Detector Status Group Active

```
vehicleDetectorStatusGroupActive OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object shall return the detection STATUS of
        each detector associated with the group. Each detector shall be
        represented as ON (detect) or OFF (no-detect) by individual bits in
        this object. If a detector is ON then the associated bit shall be set
        (1). If a detector is OFF then the associated bit shall be clear (0).

        Bit 7: Det # = ( vehicleDetectorStatusGroupNumber * 8)
        Bit 6: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 1
        Bit 5: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 2
        Bit 4: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 3
        Bit 3: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 4
        Bit 2: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 5
        Bit 1: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 6
        Bit 0: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.4.1.2
    <Object Type> S"
    ::= { vehicleDetectorStatusGroupEntry 2 }
```

5.3.4.3 Vehicle Detector Alarm Group Status

```
vehicleDetectorStatusGroupAlarms OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object shall return the alarm status of the
        detectors associated with the group. Each detector alarm status shall
        be represented as ON or OFF by individual bits in this object. If any
        detector alarm (defined in the vehicleDetectorAlarm object) is active
        the associated bit shall be set (1). If a detector alarm is not active
        the associated bit shall be clear (0).

        Bit 7: Det # = ( vehicleDetectorStatusGroupNumber * 8)
        Bit 6: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 1
        Bit 5: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 2
        Bit 4: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 3
        Bit 3: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 4
        Bit 2: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 5
        Bit 1: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 6
```

```
        Bit 0: Det # = ( vehicleDetectorStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.4.1.3
    <Object Type> S"
    ::= { vehicleDetectorStatusGroupEntry 3 }
```

5.3.5 Vehicle Detector Data

```
volumeOccupancyReport   OBJECT IDENTIFIER
    ::= { detector 5 }
-- This node contains the objects necessary to support volume / occupancy
reporting.
```

5.3.5.1 Volume / Occupancy Sequence

```
volumeOccupancySequence OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "sequence"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object defines a sequence number for
        Volume/Occupancy data collection. This object is used to detect
        duplicate or missing reports. The value cycles within the limits of 0
        to 255. This object is incremented by one at the expiration of the
        volumeOccupancyPeriod time.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.1
    <Object Type> S"
    ::= { volumeOccupancyReport 1 }
```

5.3.5.2 Volume / Occupancy Period

```
volumeOccupancyPeriod OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "This object indicates the number of seconds
        (0-3600 seconds) that comprise the Volume/Occupancy/Speed collection
        period. When the collection period expires the device shall increment
        the volumeOccupancySequence, update the volumeOccupancyTable entries
        and reset the volume occupancy timer.

        A value of zero indicates no sampling is performed.

        A value of 65535 indicates that the sample period is equal to the current
        cycle length recorded at local zero. If the sample period is
        configured to use the cycle length but the ASC is running in Free
```

```
        mode, then no data collection is performed.

        Value       Indication
        0           No sampling is performed
        1-3600      Volume/Occupancy/Speed period in seconds
        3601-65534  Reserved
        65535       Sample period is same as cycle period recorded at local zero.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.2
    <Object Type> P"
    ::= { volumeOccupancyReport 2 }
```

5.3.5.3 Volume / Occupancy Table

```
volumeOccupancyTable OBJECT-TYPE
    SYNTAX SEQUENCE OF VolumeOccupancyEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Detector Volume, Occupancy
        and Speed data collected from vehicle detectors. The number of rows
in          this table is equal to the maxVehicleDetectors object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.4"
    ::= { volumeOccupancyReport 4 }

volumeOccupancyEntry OBJECT-TYPE
    SYNTAX VolumeOccupancyEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The Volume, Occupancy and Speed data collected
        for one of the detectors in the device.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.4.1"
    INDEX   { vehicleDetectorNumber }
    ::= { volumeOccupancyTable 1 }

VolumeOccupancyEntry ::= SEQUENCE {
    detectorVolume      Integer32,
    detectorOccupancy   Integer32,
    detectorAvgSpeed    Integer32 }
```

5.3.5.3.1 Volume Data

```
detectorVolume OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "vehicle"
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION "<Definition> Detector Volume data collected over the
        volumeOccupancyPeriod. This value shall range from 0 to 65534 indicating
the volume of traffic crossing the associated vehicleDetectorNumber during the
collection period. The value 65535 shall indicate volume overflow.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.4.1.1
    <Object Type> S"
    ::= { volumeOccupancyEntry 1 }
```

5.3.5.3.2 Occupancy Data

```
detectorOccupancy OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "percent"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Detector Occupancy as a percentage of the
        volumeOccupancyPeriod over which the data was collected or Detector
        Unit Diagnostic Information. The value of the object shall indicate
        occupancy or detector diagnostic information as follows:

        Range   Meaning
        0-200   Detector Occupancy in 0.5% Increments
        201-209 Reserved
        210     Max Presence Fault
        211     No Activity Fault
        212     Open loop Fault
        213     Shorted loop Fault
        214     Excessive Change Fault
        215     Reserved
        216     Watchdog Fault
        217     Erratic Output Fault
        218-255 Reserved

        Faults shall be indicated for all collection periods during which a
        fault is detected if either occupancy data or volume data is being
        collected. The highest numbered fault shall be presented if more than
        one fault is active (i.e. indicate OpenLoop rather than NoActivity).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.4.1.2
    <Object Type> S"
    ::= { volumeOccupancyEntry 2 }
```

5.3.5.3.3 Speed Data

```
detectorAvgSpeed OBJECT-TYPE
    SYNTAX Integer32 (0..511)
    UNITS "0.5 kilometers/hour"
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION  "<Definition>Average vehicle speed during the
        volumeOccupancyPeriod over which the data was collected. The value of
        the object shall indicate average vehicle speed as follows:

        Range Meaning
        0-508 Average vehicle speed in 0.5 kilometers per hour
        509   Reserved
        510   Average vehicle speed is 255 kilometers per hour or higher
        511   Invalid or missing value
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.4.1.3
    <Object Type> S"
    DEFVAL { 511 }
    ::= { volumeOccupancyEntry 3 }
```

5.3.5.4 Volume / Occupancy Sample Time

```
detectorSampleTime OBJECT-TYPE
    SYNTAX Counter32
    UNITS "second"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "The local time, expressed in seconds since 00:00:00
        (midnight) January 1, 1970 of the same time offset, representing the
        end time of the last completed vehicle detector data collection
        period. This value changes by 3600 seconds in response to a DST event.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.6
    <Object Type> S"
    ::= { volumeOccupancyReport 6 }
```

5.3.5.5 Volume / Occupancy Sample Duration

```
detectorSampleDuration OBJECT-TYPE
    SYNTAX Integer32 (0..3600)
    UNITS "second"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the number of seconds
        (1-3600 seconds) that comprise the duration of the pedestrian detector
        data collection period. A value of zero indicates that duration is
        invalid.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.5.7
    <Object Type> S"
    ::= { volumeOccupancyReport 7 }
```

5.3.6 Maximum Pedestrian Detectors

```
maxPedestrianDetectors OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "detector"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Pedestrian Detectors this
        Controller Unit supports. This object indicates the maximum rows which
        shall appear in the pedestrianDetectorTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.6
    <Object Type> S"
    ::= { detector 6 }
```

5.3.7 Pedestrian Detector Parameter Table

```
pedestrianDetectorTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PedestrianDetectorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit pedestrian
        detector parameters. The number of rows in this table is equal to the
        maxPedestrianDetectors object.

        The objects in this table correspond to the matching objects in the
        first set (pedestrianDetectorSetNumber = 1) of the
pedestrianDetectorSetTable.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7"
    ::= { detector 7 }

pedestrianDetectorEntry OBJECT-TYPE
    SYNTAX PedestrianDetectorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Controller Unit
        pedestrian detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1"
    INDEX   { pedestrianDetectorNumber }
    ::= { pedestrianDetectorTable 1 }

PedestrianDetectorEntry ::= SEQUENCE {
    pedestrianDetectorNumber        Integer32,
    pedestrianDetectorCallPhase     Integer32,
    pedestrianDetectorNoActivity    Integer32,
    pedestrianDetectorMaxPresence   Integer32,
```

```
    pedestrianDetectorErraticCounts Integer32,
    pedestrianDetectorAlarms        Integer32,
    pedestrianDetectorReset         Integer32,
    pedestrianButtonPushTime        Integer32,
    pedestrianDetectorOptions       Integer32,
    pedestrianDetectorDescription   SnmpAdminString }
```

5.3.7.1 Pedestrian Detector Number

```
pedestrianDetectorNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "detector"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The pedestrianDetector number for objects in
        this row. The value shall not exceed the maxPedestrianDetectors object
        value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.1"
    ::= { pedestrianDetectorEntry 1 }
```

5.3.7.2 Pedestrian Detector Call Phase Parameter

```
pedestrianDetectorCallPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "phase"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object contains assigned phase number for
        the pedestrian detector input associated with this row. The associated
        detector call capability is enabled when this object is set to a
        non-zero value. The value shall not exceed the value of maxPhases. A
value of zero indicates no call phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.2
    <Object Type> P"
    ::= { pedestrianDetectorEntry 2 }
```

5.3.7.3 Pedestrian Detector No Activity Parameter

```
pedestrianDetectorNoActivity OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Pedestrian Detector No Activity diagnostic
        Parameter in minutes (1-65535 min.). If an active detector does not
        exhibit an actuation in the specified period, it is considered a fault
```

```
        by the diagnostics and the detector is classified as Failed. A value
        of 0 for this object shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.3
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.1"
    ::= { pedestrianDetectorEntry 3 }
```

5.3.7.4 Pedestrian Detector Maximum Presence Parameter

```
pedestrianDetectorMaxPresence OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Pedestrian Detector Maximum Presence diagnostic
        Parameter in minutes (0-255 min.). If an active detector exhibits
        continuous detection for too long a period, it is considered a fault
        by the diagnostics and the detector is classified as Failed. A value
        of 0 for this object shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.4
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.2"
    ::= { pedestrianDetectorEntry 4 }
```

5.3.7.5 Pedestrian Detector Erratic Counts Parameter

```
pedestrianDetectorErraticCounts OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "count"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Pedestrian Detector Erratic Counts diagnostic
        Parameter in counts/minute (0-255 cpm). If an active detector exhibits
        excessive actuations, it is considered a fault by the diagnostics and
        the detector is classified as Failed. A value of zero for this object
        shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.5
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.3"
    ::= { pedestrianDetectorEntry 5 }
```

5.3.7.6 Pedestrian Detector Alarms

```
pedestrianDetectorAlarms OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION "<Definition> This object shall return indications of pedestrian
detector
        alarms identified by the CU. Detector Alarms are indicated as follows (0
= False, 1 = True):

        Bit 7: Other Fault - The detector has failed due to some other cause.
        Bit 6: Reserved.
        Bit 5: Reserved.
        Bit 4: Configuration Fault - Detector is assigned but is not supported.
        Bit 3: Communications Fault - Communications to the device (if present)
have failed.
        Bit 2: Erratic Output Fault - This detector has been flagged as
                non-operational due to erratic outputs (excessive counts) by
                the CU detector diagnostic.
        Bit 1: Max Presence Fault - This detector has been flagged as
                non-operational due to a presence indicator that exceeded the
                maximum expected time by the CU detector diagnostic.
        Bit 0: No Activity Fault - This detector has been flagged as
                non-operational due to lower than expected activity by the CU
                detector diagnostic

        Once set a bit shall maintain its state as long as the condition
        exists. The bit shall clear when the condition no longer exists.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.6
    <Object Type> S"
    ::= { pedestrianDetectorEntry  6 }
```

5.3.7.7 Pedestrian Detector Reset

```
pedestrianDetectorReset OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object when set to TRUE (one) shall cause
        the CU to command the associated detector to reset. This object shall
        automatically return to FALSE (zero) after the CU has issued the reset
        command.

        Note: this may affect other detector (detector channels) that are
        physically attached to a common reset line.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.7
    <Object Type> C"
    DEFVAL { 0 }
    ::= { pedestrianDetectorEntry 7 }
```

5.3.7.8 Pedestrian Push Button Duration Parameter

```
pedestrianButtonPushTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The minimum amount of time, in tenths of a
        second, a pedestrian call button is pressed to actuate additional
        accessible features such as increased pedestrian crossing times
        (phasePedAlternateWalk) or pedestrian clearance times
        (phasePedAlternateClearance). A value of zero indicates that all
        accessible pedestrian signal (APS) features are disabled for the
        associated detector.
        This value is the same for all pedestrian detector sets.
        NOTE the pedestrian detector must be configured to allow Alternate
timing. pedestrianDetectorOptions or pedestrianSetDetectorOptions.

        This value is the same for all pedestrian detector sets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.8
    <Object Type> P"
    DEFVAL { 0 }
    ::= { pedestrianDetectorEntry 8 }
```

5.3.7.9 Pedestrian Detector Parameter

```
pedestrianDetectorOptions OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION " <Definition> Pedestrian Detector Options Parameter as
        follows (0=Disabled, 1=Enabled):

        Bit 7: Reserved.
        Bit 6: Reserved.
        Bit 5: Reserved.
        Bit 4: Advanced Walk: If enabled, the detector will place calls for the
phase's pedestrian WALK indication to start before the phase's vehicle GREEN
indication.
        Bit 3: Delayed Walk: If enabled, the detector will place calls for the
phase's pedestrian WALK indication to start after the phase's vehicle GREEN
indication.
        Bit 2: Non-locking: If enabled, detector will place non-locked calls
instead of locked calls.
```

```
        Bit 1: Alternate timing: If enabled, the detector will place calls for
alternate ped timing instead of normal ped timing.
        Bit 0: Presence: If enabled, the detector indicates presence of
                pedestrians in the crosswalk instead of placing calls for
                service.

        A SET of both bits 3 & 4 shall result in bit 3=0 and bit 4=1. A SET of a
'reserved' bit to a value other than zero (0) shall return a wrongValue(10)
error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.9
    <Object Type> P"
    DEFVAL { 0 }
    ::= { pedestrianDetectorEntry 9 }
```

5.3.7.10 Pedestrian Detector Description

```
pedestrianDetectorDescription OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A textual string indicate a user description for
the vehicle detector (e.g. location). This value is the same for all pedestrian
detector sets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.7.1.10
    <Object Type> P"
    ::= { pedestrianDetectorEntry 10 }
```

5.3.8 Maximum Pedestrian Detector Groups

```
maxPedestrianDetectorGroups OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition>The maximum number of pedestrian detector status
        groups (8 detectors per group) this device supports. This value is
        equal to TRUNCATE [(maxPedestrianDetectors + 7 ) / 8]. This object
        indicates the maximum number of rows which shall appear in the
        pedestrianDetectorStatusGroupTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.8
    <Object Type> S"
    ::= { detector 8 }
```

5.3.9 Pedestrian Detector Status Group Table

```
pedestrianDetectorStatusGroupTable OBJECT-TYPE
```

```
    SYNTAX SEQUENCE OF PedestrianDetectorStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing pedestrian detector status in
        groups of eight detectors. The number of rows in this table is equal
        to the maxPedestrianDetectorGroups object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.9"
    ::= { detector 9 }

pedestrianDetectorStatusGroupEntry OBJECT-TYPE
    SYNTAX PedestrianDetectorStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A group (row) of pedestrian detector status.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.9.1"
    INDEX { pedestrianDetectorStatusGroupNumber }
    ::= { pedestrianDetectorStatusGroupTable 1 }

PedestrianDetectorStatusGroupEntry ::= SEQUENCE {
    pedestrianDetectorStatusGroupNumber Integer32,
    pedestrianDetectorStatusGroupActive Integer32,
    pedestrianDetectorStatusGroupAlarms Integer32 }
```

5.3.9.1 Pedestrian Detector Status Group Number

```
pedestrianDetectorStatusGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The pedestrian detector status group number for
        objects in this row. This value shall not exceed the
        maxPedestrianDetectorGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.9.1.1"
    ::= { pedestrianDetectorStatusGroupEntry 1 }
```

5.3.9.2 Pedestrian Detector Status Group Active

```
pedestrianDetectorStatusGroupActive OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition>This object shall return the detection status of
        each pedestrian detector associated with the group. Each detector
        shall be represented as ON (detect) or OFF (no-detect) by individual
```

```
        bits in this object. If a detector is ON then the associated bit shall
        be set (1). If a detector is OFF then the associated bit shall be
        clear (0).

        Bit 7: Det # = ( pedestrianDetectorStatusGroupNumber * 8)
        Bit 6: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 1
        Bit 5: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 2
        Bit 4: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 3
        Bit 3: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 4
        Bit 2: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 5
        Bit 1: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 6
        Bit 0: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.9.1.2
    <Object Type> S"
    ::= { pedestrianDetectorStatusGroupEntry 2 }
```

5.3.9.3 Pedestrian Detector Status Group Active

```
pedestrianDetectorStatusGroupAlarms OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object shall return the alarm status of the
        pedestrian detectors associated with the group. Each pedestrian
        detector alarm status shall be represented as ON or OFF by individual
        bits in this object. If any pedestrian detector alarm (defined in the
        pedestrianDetectorAlarms object) is active the associated bit shall be
        set (1). If a pedestrian detector alarm is not active the associated
        bit shall be clear (0).

        Bit 7: Det # = ( pedestrianDetectorStatusGroupNumber * 8)
        Bit 6: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 1
        Bit 5: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 2
        Bit 4: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 3
        Bit 3: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 4
        Bit 2: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 5
        Bit 1: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 6
        Bit 0: Det # = ( pedestrianDetectorStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.9.1.3
    <Object Type> S"
    ::= { pedestrianDetectorStatusGroupEntry 3 }
```

5.3.10 Pedestrian Detector Report

```
pedestrianDetectorReport OBJECT IDENTIFIER
    ::= { detector 10 }
```

```
-- This node contains the objects necessary to support pedestrian detector
reporting.
```

5.3.10.1 Pedestrian Detector Report

```
pedestrianDetectorSequence OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "sequence"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object defines a Sequence Number for the
        pedestrian detector data collection. This object is used to detect
        duplicate or missing reports. The value cycles within the limits of
        0 to 255. This object is incremented by one at the expiration of the
        pedestrianDetectorPeriod time.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.1
    <Object Type> S"
    ::= { pedestrianDetectorReport 1 }
```

5.3.10.2 Pedestrian Detector Report

```
pedestrianDetectorPeriod OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the number of seconds
        (1-3600 seconds) that comprise the pedestrian detector data
        collection period. When the collection period expires the device shall
        increment the pedestrianDetectorSequence, update the
        pedestrianSampleTable entries, and reset the pedestrian volume timer. A
        value of 0 indicates that no sampling is to be performed.

        A value of 65534 indicates that the pedestrian detector data
        collection period is equal to vehicle sample period in effect.

        A value of 65535 indicates that the sample period equal to current
        cycle length recorded at local zero. If the sample period is
        configured to use the cycle length but the ASC is running in Free
        mode, then no data collection is performed.

        Value       Indication
        0           No pedestrian data collection is performed
        1-3600      Pedestrian data collection period in seconds
        3601-65533  Reserved
```

```
        65534       Pedestrian data collection period is equal to the vehicle
                        sample period in effect
        65535       Pedestrian data collection period is same as cycle period
                        recorded at local zero
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.2
    <Object Type> P"
    ::= { pedestrianDetectorReport 2 }
```

5.3.10.3 Pedestrian Sample Table

```
pedestrianSampleTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PedestrianSampleEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing pedestrian data collected.
        The number of rows in this table is equal to the
        maxPedestrianDetectors object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.4"
    ::= { pedestrianDetectorReport 4 }

pedestrianSampleEntry OBJECT-TYPE
    SYNTAX PedestrianSampleEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The data collected for one of the detectors in
        the device as part of a pedestrian detector data collection.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.4.1"
    INDEX { pedestrianDetectorNumber }
    ::= { pedestrianSampleTable 1 }

PedestrianSampleEntry ::= SEQUENCE {
    pedestrianDetectorVolume     Integer32,
    pedestrianDetectorActuations Integer32,
    pedestrianDetectorServices   Integer32 }
```

5.3.10.3.1 Pedestrian Sample Volume

```
pedestrianDetectorVolume OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "pedestrian"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition>Pedestrian detector data collected over the data
        collection period. This value shall range from 0 to 254 indicating the
        volume of pedestrians crossing the associated pedestrian detector zone
```



```
        during the data collection period. The value 255 shall indicate volume
        overflow.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.4.1.1
    <Object Type> S"
    ::= { pedestrianSampleEntry 1 }
```

5.3.10.3.2 Pedestrian Sample Actuations

```
pedestrianDetectorActuations OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "actuations"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition>Pedestrian actuations collected over the data
        collection period. The value of the object shall indicate pedestrian
        actuations or pedestrian detector diagnostic information as follows:

        Value   Indication
        0-200   Number of actuations
        201     Number of actuations exceeds 200.
        202-208 Reserved
        209     Other Fault
        210     Max Presence Fault
        211     No Activity Fault
        212     Reserved
        213     Reserved
        214     Reserved
        215     Configuration Fault
        216     Communications Fault
        217     Erratic Output Fault
        218-255 Reserved

        Faults shall be indicated for all collection periods during which a
        fault is detected if either pedestrian volume or pedestrian actuations
        is being collected. The highest numbered fault shall be presented if
        more than one fault is active (i.e. indicate OpenLoop rather than
        NoActivity).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.4.1.2
    <Object Type> S"
    ::= { pedestrianSampleEntry 2 }
```

5.3.10.3.3 Pedestrian Sample Services

```
pedestrianDetectorServices OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "service"
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The number of pedestrian services (number of
        times the ped transitioned from don't walk to walk) collected over the
        data collection period. This value shall range from 0-254. A value of
        255 indicates an overflow condition.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.4.1.3
    <Object Type> S"
    ::= { pedestrianSampleEntry 3 }
```

5.3.10.4 Pedestrian Volume / Actuation Sample Time

```
pedestrianDetectorSampleTime OBJECT-TYPE
    SYNTAX Counter32
    UNITS "second"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "The local time expressed in seconds since 00:00:00 (midnight)
        January 1, 1970 of the same time offset, representing the end time of
        the last completed pedestrian detector data collection period. This
        value changes by 3600 seconds in response to a DST event.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.5
    <Object Type> S"
    ::= { pedestrianDetectorReport 5 }
```

5.3.10.5 Pedestrian Volume / Actuation Sample Duration

```
pedestrianDetectorSampleDuration OBJECT-TYPE
    SYNTAX Integer32 (0..3600)
    UNITS "second"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the number of seconds
        (1-3600 seconds) that comprise the sampling of the pedestrian detector
        data collection period. A value of zero indicates that duration is
        invalid.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.6
    <Object Type> S"
    ::= { pedestrianDetectorReport 6 }
```

5.3.10.6 Pedestrian Volume / Actuation Sample Duration

```
pedestrianDetectorSampleDuration OBJECT-TYPE
    SYNTAX Integer32 (0..3600)
    UNITS "second"
    MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION "<Definition> This object indicates the number of seconds
        (1-3600 seconds) that comprise the sampling of the pedestrian detector
        data collection period. A value of zero indicates that duration is
        invalid.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.10.6
    <Object Type> S"
    ::= { pedestrianDetectorReport 6 }
```

5.3.11 Vehicle Detector Control Group Table

```
vehicleDetectorControlGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF VehicleDetectorControlGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing vehicle detector control in
        groups of eight detectors. The number of rows in this table is equal
        to the maxVehicleDetectorGroups object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.12"
    ::= { detector 12 }

vehicleDetectorControlGroupEntry OBJECT-TYPE
    SYNTAX VehicleDetectorControlGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A group (row) of vehicle detector controls.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.12.1"
    INDEX  { vehicleDetectorControlGroupNumber }
    ::= { vehicleDetectorControlGroupTable  1 }

VehicleDetectorControlGroupEntry ::= SEQUENCE {
    vehicleDetectorControlGroupNumber    Integer32,
    vehicleDetectorControlGroupActuation Integer32 }
```

5.3.11.1 Vehicle Detector Control Group Number

```
vehicleDetectorControlGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The vehicle detector control group number for
        objects in this row. This value shall not exceed the
        maxVehicleDetectorGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.12.1.1"
```

```
    ::= { vehicleDetectorControlGroupEntry 1 }
```

5.3.11.2 Vehicle Detector Control Group Actuation

```
vehicleDetectorControlGroupActuation OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
        place an actuation on a vehicle detector. When a bit = 1, an actuation
        is placed on the vehicle detector. When a bit = 0, no actuation is
        placed on the vehicle detector. An NTCIP actuation is placed using
        this object and is treated the same as an external actuation, so all
        detector functions are still applicable, including delay, extension,
        diagnostics, and report objects, such as
        vehicleDetectorStatusGroupActive and volumeOccupancyReport.

        Bit 7: Det # = ( vehicleDetectorControlGroupNumber * 8)
        Bit 6: Det # = ( vehicleDetectorControlGroupNumber * 8) - 1
        Bit 5: Det # = ( vehicleDetectorControlGroupNumber * 8) - 2
        Bit 4: Det # = ( vehicleDetectorControlGroupNumber * 8) - 3
        Bit 3: Det # = ( vehicleDetectorControlGroupNumber * 8) - 4
        Bit 2: Det # = ( vehicleDetectorControlGroupNumber * 8) - 5
        Bit 1: Det # = ( vehicleDetectorControlGroupNumber * 8) - 6
        Bit 0: Det # = ( vehicleDetectorControlGroupNumber * 8) - 7

        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO
        (see unitBackupTime).
    <Object Type> C
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.12.1.2"
    ::= { vehicleDetectorControlGroupEntry 2 }
```

5.3.12 Pedestrian Detector Control Group Table

```
pedestrianDetectorControlGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PedestrianDetectorControlGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing pedestrian detector control
        in groups of eight detectors. The number of rows in this table is
        equal to the maxPedestrianDetectorGroups object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.13"
    ::= { detector 13 }
```

```
pedestrianDetectorControlGroupEntry OBJECT-TYPE
    SYNTAX PedestrianDetectorControlGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A group (row) of pedestrian detector controls.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.13.1"
    INDEX  { pedestrianDetectorControlGroupNumber }
    ::= { pedestrianDetectorControlGroupTable  1 }

PedestrianDetectorControlGroupEntry ::= SEQUENCE {
    pedestrianDetectorControlGroupNumber    Integer32,
    pedestrianDetectorControlGroupActuation Integer32 }
```

5.3.12.1 Pedestrian Detector Control Group Number

```
pedestrianDetectorControlGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The pedestrian detector control group number for
        objects in this row. This value shall not exceed the
        maxPedestrianDetectorGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.13.1.1"
    ::= { pedestrianDetectorControlGroupEntry 1 }
```

5.3.12.2 Pedestrian Detector Control Group Actuation

```
pedestrianDetectorControlGroupActuation OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
        place an actuation on a pedestrian detector. When a bit = 1, an
        actuation is placed on the pedestrian detector. When a bit = 0, no
        actuation is placed on the pedestrian detector. An NTCIP actuation is
        placed using this object and is treated the same as an external
        actuation, so all detector functions are still applicable, including
        delay, extension, diagnostics, and report objects, such as
        pedestrianDetectorStatusGroupActive and pedestrianDetectorReport.

        Bit 7: Det # = ( pedestrianDetectorControlGroupNumber * 8)
        Bit 6: Det # = ( pedestrianDetectorControlGroupNumber * 8) - 1
        Bit 5: Det # = ( pedestrianDetectorControlGroupNumber * 8) - 2
        Bit 4: Det # = ( pedestrianDetectorControlGroupNumber * 8) - 3
        Bit 3: Det # = ( pedestrianDetectorControlGroupNumber * 8) - 4
```

```
        Bit 2: Det # = ( pedestrianDetectorControlGroupNumber * 8) - 5
        Bit 1: Det # = ( pedestrianDetectorControlGroupNumber * 8) - 6
        Bit 0: Det # = ( pedestrianDetectorControlGroupNumber * 8) - 7

        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO
        (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.13.1.2
    <Object Type> C"
    ::= { pedestrianDetectorControlGroupEntry 2 }
```

5.3.13 Maximum Vehicle Detector Sets

```
maxVehicleDetectorSets OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Vehicle Detector Sets this
        Controller Unit supports. This object indicates the maximum value for
        the vehicleDetectorSetTable index vehicleDetectorSetNumber.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.14
    <Object Type> S"
    ::= { detector 14 }
```

5.3.14 Vehicle Detector Set Table

```
vehicleDetectorSetTable OBJECT-TYPE
    SYNTAX SEQUENCE OF VehicleDetectorSetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit vehicle
        detector parameters. The number of rows in this table is equal to the
        maxVehicleDetectors object times the maxVehicleDetectorSets object.

        The first set in this table (vehicleDetectorSetNumber = 1) shall match is
corresponding objects in the vehicleDetectorTable, unless explicitly defined.
        For example:
            vehicleDetectorSetOptions.1.n = vehicleDetectorOptions.n
            vehicleDetectorSetCallPhase.1.n = vehicleDetectorCallPhase.n:
            vehicleDetectorSetSwitchPhase.1.n = vehicleDetectorSwitchPhase.n
            etc…
            vehicleDetectorSetTravelMode.1.n = vehicleDetectorTravelMode.n
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15"
    ::= { detector 15 }
```

```
vehicleDetectorSetEntry OBJECT-TYPE
    SYNTAX VehicleDetectorSetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a one of a set of Controller Unit
        detectors.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1"
    INDEX   { vehicleDetectorSetNumber, vehicleDetectorNumber }
    ::= { vehicleDetectorSetTable 1 }

VehicleDetectorSetEntry ::= SEQUENCE {
    vehicleDetectorSetNumber                Integer32,
    vehicleDetectorSetOptions               Integer32,
    vehicleDetectorSetCallPhase             Integer32,
    vehicleDetectorSetSwitchPhase           Integer32,
    vehicleDetectorSetDelay                 Integer32,
    vehicleDetectorSetExtend                Integer32,
    vehicleDetectorSetQueueLimit            Integer32,
    vehicleDetectorSetNoActivity            Integer32,
    vehicleDetectorSetMaxPresence           Integer32,
    vehicleDetectorSetErraticCounts         Integer32,
    vehicleDetectorSetFailTime              Integer32,
    vehicleDetectorSetOptions2              Integer32,
    vehicleDetectorSetAvgVehicleLength      Integer32,
    vehicleDetectorSetTravelMode            INTEGER }
```

5.3.14.1 Vehicle Detector Set Number

```
vehicleDetectorSetNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The vehicle detector set index for objects in
        this row. The value shall not exceed maxVehicleDetectorSets.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.1"
    ::= { vehicleDetectorSetEntry 1 }
```

5.3.14.2 Vehicle Detector Set Options Parameter

```
vehicleDetectorSetOptions OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Vehicle Detector Set Options Parameter as
```

```
        follows (0=Disabled, 1=Enabled):
    Bit 7: Call - if Enabled, the CU shall place a demand for vehicular
        service on the assigned phase when the phase is not timing the green
        interval and an actuation is present.
    Bit 6: Queue - if Enabled, the CU shall extend the green interval of the
        assigned phase until a gap occurs (no actuation) or until the green
        has been active longer than the vehicleDetectorQueueLimit time.
        This is optional.
    Bit 5: AddedInitial - if Enabled, the CU shall accumulate detector
        actuation counts for use in the added initial calculations. Counts
        shall be accumulated from the beginning of the yellow interval to the
        beginning of the green interval.
    Bit 4: Passage - if Enabled, the CU shall maintain a reset to the
        associated phase passage timer for the duration of the detector
        actuation when the phase is green.
    Bit 3: Red Lock Call - if Enabled, the detector will lock a call to the
        assigned phase if an actuation occurs while the phase is not timing
        Green or Yellow. This mode is optional.
    Bit 2: Yellow Lock Call - if Enabled, the detector will lock a call to
        the assigned phase if an actuation occurs while the phase is not
        timing Green.
    Bit 1: Reserved
    Bit 0: Reserved

    A SET of both bits 2 & 3 = 1 shall result in bit 2=1 and bit 3=0.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.2
    <Object Type> P"
    ::= { vehicleDetectorSetEntry 2}
```

5.3.14.3 Vehicle Detector Set Call Phase Parameter

```
vehicleDetectorSetCallPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "phase"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object contains assigned phase number for
        the detector input associated with this row. The associated detector
        call capability is enabled when this object is set to a non-zero
        value. The value shall not exceed the value of maxPhases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.3
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.5.5.4 and 3.5.5.5.5"
    ::= { vehicleDetectorSetEntry 3 }
```

5.3.14.4 Vehicle Detector Set Switch Phase Parameter

```
vehicleDetectorSetSwitchPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "phase"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Switch Phase Parameter (i.e., Phase
        Number). The phase to which a vehicle detector actuation shall be
        switched when the assigned phase is Yellow or Red and the Switch Phase
        is Green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.4
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.5.5.4.c"
    ::= { vehicleDetectorSetEntry 4 }
```

5.3.14.5 Vehicle Detector Set Delay Parameter

```
vehicleDetectorSetDelay OBJECT-TYPE
    SYNTAX Integer32 (0..2550)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Delay Parameter in tenth seconds
        (0-255.0 sec). The period a detector actuation (input recognition)
        shall be delayed when the phase is not Green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.5
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.5.5.4.a"
    ::= { vehicleDetectorSetEntry 5 }
```

5.3.14.6 Vehicle Detector Set Extend Parameter

```
vehicleDetectorSetExtend OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Extend Parameter in tenth seconds
        (0-25.5 sec). The period a vehicle detector actuation (input duration)
        shall be extended from the point of termination, when the phase is
        Green.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.6
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.5.5.4.b"
    ::= { vehicleDetectorSetEntry 6 }
```

5.3.14.7 Vehicle Detector Set Extend Parameter

```
vehicleDetectorSetQueueLimit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Queue Limit parameter in seconds
        (0-255 sec). The length of time that an actuation from a queue
        detector may continue into the phase green. This time begins when the
        phase becomes green and when it expires any associated detector inputs
        shall be ignored. This time may be shorter due to other overriding
        device parameters (i.e. Maximum time, Force Offs, ...).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.7
    <Object Type> P"
    ::= { vehicleDetectorSetEntry 7 }
```

5.3.14.8 Vehicle Detector Set No Activity Parameter

```
vehicleDetectorSetNoActivity OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector No Activity diagnostic Parameter in
        minutes (1-65535 min.). If an active detector does not exhibit an
        actuation in the specified period, it is considered a fault by the
        diagnostics and the detector is classified as Failed. A value of zero for
        this object shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.8
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.1"
    ::= { vehicleDetectorSetEntry 8 }
```

5.3.14.9 Vehicle Detector Set Maximum Presence Parameter

```
vehicleDetectorSetMaxPresence OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Maximum Presence diagnostic Parameter
        in minutes (0-255 min.). If an active detector exhibits continuous
        detection for too long a period, it is considered a fault by the
        diagnostics and the detector is classified as Failed. A value of zero for
```



```
        this object shall disable this diagnostic for this detector.

        This object corresponds to the object vehicleDetectorSwitchPhase
        in the vehicleDetectorTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.9
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.2"
    ::= { vehicleDetectorSetEntry 9 }
```

5.3.14.10 Vehicle Detector Set Erratic Counts Parameter

```
vehicleDetectorSetErraticCounts OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "count"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Erratic Counts diagnostic Parameter in
        counts/minute (0-255 cpm). If an active detector exhibits excessive
        actuations, it is considered a fault by the diagnostics and the
        detector is classified as Failed. A value of zero for this object shall
        disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.10
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.3"
    ::= { vehicleDetectorSetEntry 10 }
```

5.3.14.11 Vehicle Detector Set Fail Time Parameter

```
vehicleDetectorSetFailTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Detector Fail Time in seconds (0..255 sec). If a
        detector diagnostic indicates that the associated detector input is
        failed, then a call shall be placed on the associated phase during all
        non-green intervals. When each green interval begins the call shall be
        maintained for the length of time specified by this object and then
        removed. If the value of this object equals the maximum value (255)
        then a constant call shall be placed on the associated phase (max
        recall). If the value of this object equals zero then no call shall be
        placed on the associated phase for any interval (no recall). Compliant
        devices may support a limited capability for this object (i.e. only
        max recall). At a minimum, the max recall setting must be supported.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.11
    <Object Type> P"
```

```
    ::= { vehicleDetectorSetEntry 11 }
```

5.3.14.12 Vehicle Detector Set Options 2

```
vehicleDetectorSetOptions2 OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A bit-mapped value as defined below for
        configuring detector options.
    <Format>
        Bit 7: Reserved
        Bit 6: Reserved
        Bit 5: Reserved
        Bit 4: Reserved
        Bit 3: Reserved
        bit 2    0=CUSTOM, 1=NTCIP  Default Detector Speed Mode Option. For a
                    vehicle detector operating in pairs, this option is used
                    when there is an error on one of the paired detectors. It
                    identifies how the controller should calculate speed
                    without the other detector. CUSTOM indicates a
                    manufacturer specific calculation. NTCIP indicates the use
                    of the calculation Speed = (Average Vehicle Length +
                    Detector Length) / Detect Time.
        bit 1    0=TRAIL, 1=LEAD  Detector Placement Option. For a vehicle
                    detector operating in pairs, this option indicates the
                    leading and trailing detectors. LEAD indicates that the
                    detector is the leading detector of the pair. TRAIL
                    indicates that the detector is a trailing detector in
                    the pair.
        bit 0    0=DISABLED, 1=ENABLED  Enable Speed Data. If enabled, the
                    detector is used to collect speed data (See
                    volumeOccupancyTable and detectorAvgSpeed). This
                    capability may not be supported on all detector inputs to
                    a device.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.12
    <Object Type> P"
    ::= { vehicleDetectorSetEntry 12 }
```

5.3.14.13 Vehicle Detector Set Average Vehicle Length

```
vehicleDetectorSetAvgVehicleLength OBJECT-TYPE
    SYNTAX Integer32 (1..4000)
    UNITS "centimeter"
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION "<Definition> This parameter allows the user to set the
        average vehicle length for use in determining speed and
        classification. This allows for a range of lengths between 0.01 meters
        to 40 meters in length.

        Different sets may be used to support different vehicle mixes based on
time of day, such as cars during rush hour and trucks at night.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.13
    <Object Type> P"
    ::= { vehicleDetectorSetEntry 13 }
```

5.3.14.14 Vehicle Detector Set Travel Mode

```
vehicleDetectorSetTravelMode OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     vehicle (2),
                     transit (3),
                     bicycle (4) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This parameter allows the user to identify
        detectors for specific types of travel modes.

        other:   refers to a detector for a travel type not defined in this
                    standard
        vehicle: refers to a detector identified for vehicles.
        transit: refers to a detector identified for transit vehicles.
        bicycle: refers to a detector identified for bicycles.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.15.1.14
    <Object Type> P"
    DEFVAL { 2 }
    ::= { vehicleDetectorSetEntry 14 }
```

5.3.15 Maximum Pedestrian Detector Sets

```
maxPedestrianDetectorSets OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Pedestrian Detector Sets this
        Controller Unit supports. This object indicates the maximum value for
        the PedestrianDetectorSetTable index pedestrianDetectorSetNumber.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.16
    <Object Type> S"
    ::= { detector 16 }
```

5.3.16 Pedestrian Detector Set Table

```
pedestrianDetectorSetTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PedestrianDetectorSetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit pedestrian
        detector parameters. The number of rows in this table is equal to the
        maxPedestrianDetectorSets object.

           The first set in this table (pedestrianDetectorSetNumber = 1) shall
match is corresponding objects in the pedestrianDetectorTable, unless explicitly
defined.
        For example:
            pedestrianDetectorSetCallPhase.1.n = pedestrianDetectorCallPhase.n:
            pedestrianDetectorSetSwitchPhase.1.n =
pedestrianDetectorSwitchPhase.n
            etc…
            pedestrianDetectorSetOptions.1.n = pedestrianDetectorOptions.n
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.17"
    ::= { detector 17 }

pedestrianDetectorSetEntry OBJECT-TYPE
    SYNTAX PedestrianDetectorSetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Controller Unit
        pedestrian detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.17.1"
    INDEX   { pedestrianDetectorSetNumber, pedestrianDetectorNumber }
    ::= { pedestrianDetectorSetTable 1 }

PedestrianDetectorSetEntry ::= SEQUENCE {
    pedestrianDetectorSetNumber         Integer32,
    pedestrianDetectorSetCallPhase      Integer32,
    pedestrianDetectorSetNoActivity     Integer32,
    pedestrianDetectorSetMaxPresence    Integer32,
    pedestrianDetectorSetErraticCounts  Integer32,
    pedestrianDetectorSetOptions        Integer32 }
```

5.3.16.1 Pedestrian Detector Set Number

```
pedestrianDetectorSetNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
```

```
    UNITS "set"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The pedestrianDetector set index for objects in
        this row. The value shall not exceed the maxPedestrianDetectorSets object
        value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.17.1.1"
    ::= { pedestrianDetectorSetEntry 1 }
```

5.3.16.2 Pedestrian Detector Set Call Phase Parameter

```
pedestrianDetectorSetCallPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "phase"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object contains assigned phase number for
        the pedestrian detector input associated with this row. The associated
        detector call capability is enabled when this object is set to a
        non-zero value. The value shall not exceed the value of maxPhases.
        A value of zero indicates no call phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.17.1.2
    <Object Type> P"
    ::= { pedestrianDetectorSetEntry 2 }
```

5.3.16.3 Pedestrian Detector Set No Activity Parameter

```
pedestrianDetectorSetNoActivity OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Pedestrian Detector Set No Activity diagnostic
        Parameter in minutes (1-65535 min.). If an active detector does not
        exhibit an actuation in the specified period, it is considered a fault
        by the diagnostics and the detector is classified as Failed. A value
        of 0 for this object shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.17.1.3
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.1"
    ::= { pedestrianDetectorSetEntry 3 }
```

5.3.16.4 Pedestrian Detector Set No Maximum Presence Parameter

```
pedestrianDetectorSetMaxPresence OBJECT-TYPE
    SYNTAX Integer32 (0..255)
```

```
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Pedestrian Detector Set Maximum Presence diagnostic
        Parameter in minutes (0-255 min.). If an active detector exhibits
        continuous detection for too long a period, it is considered a fault
        by the diagnostics and the detector is classified as Failed. A value
        of 0 for this object shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.17.1.4
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.2"
    ::= { pedestrianDetectorSetEntry 4 }
```

5.3.16.5 Pedestrian Detector Set No Maximum Presence Parameter

```
pedestrianDetectorSetErraticCounts OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "count"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Pedestrian Detector Set Erratic Counts diagnostic
        Parameter in counts/minute (0-255 cpm). If an active detector exhibits
        excessive actuations, it is considered a fault by the diagnostics and
        the detector is classified as Failed. A value of zero for this object
        shall disable this diagnostic for this detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.17.1.5
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.3.1.4.3"
    ::= { pedestrianDetectorSetEntry 5 }
```

5.3.16.6 Pedestrian Detector Set Options

```
pedestrianDetectorSetOptions OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION " <Definition> Pedestrian Detector Set Options Parameter as
        follows (0=Disabled, 1=Enabled):

        Bit 7: Reserved.
        Bit 6: Reserved.
        Bit 5: Reserved.
        Bit 4: Advanced Walk: If enabled, the detector will place calls for the
                phase's pedestrian WALK indication to start before the phase's
                vehicle GREEN indication.
        Bit 3: Delayed Walk: If enabled, the detector will place calls for the
```

```
            phase's pedestrian WALK indication to start after the phase's
            vehicle GREEN indication.
        Bit 2: Non-locking: If enabled, detector will place non-locked calls
            instead of locked calls.
        Bit 1: Alternate timing: If enabled, the detector will place calls for
            alternate ped timing instead of normal ped timing.
        Bit 0: Presence: If enabled, the detector indicates presence of
            pedestrians in the crosswalk instead of placing calls for
            service.

        A SET of both bits 3 & 4 shall result in bit 3=0 and bit 4=1. A SET of
        a 'reserved' bit to a value other than zero (0) shall return a
        wrongValue(10) error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.17.1.6
    <Object Type> P"
    DEFVAL { 0 }
    ::= { pedestrianDetectorSetEntry 6 }
```

5.3.17 Detector Diagnostics Reset

```
unitDetectorDiagnosticReset OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object when set to TRUE (one) shall cause
        the CU to clear detector faults and reset diagnostic counters for ALL
        vehicle and pedestrian detectors.
        All diagnostic counters with thresholds configured in the following
        objects are reset to zero:
            vehicleDetectorNoActivity or vehicleDetectorSetNoActivity
            vehicleDetectorMaxPresence or vehicleDetectorSetMaxPresence
            vehicleDetectorErraticCounts or vehicleDetectorSetErraticCounts
            pedestrianDetectorNoActivity or pedestrianDetectorSetNoActivity
            pedestrianDetectorMaxPresence or pedestrianDetectorSetMaxPresence
            pedestrianDetectorErraticCounts or pedestrianDetectorSetErraticCounts

        The value of these objects do not change when the diagnostic counters
        are reset.

        This object shall automatically return to FALSE (zero) after the CU has
        issued the reset command.

        This object does not command the CU to reset detectors (see
        vehicleDetectorReset and pedestrianDetectorReset)
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.2.2.1.18
```

```
    <Object Type> C"
    ::= { detector 18 }
```

5.4 Unit Parameters

```
unit OBJECT IDENTIFIER
    ::= { asc 3 }

--This defines a node for supporting unit objects.
```

5.4.1 Startup Flash Parameter

```
unitStartUpFlash OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Unit Start up Flash time parameter in seconds
        (0 to 255 sec). The period/state (Start-Up Flash) occurs when power is
        restored following a device defined power interruption. During the
        Start-Up Flash state, the Fault Monitor and Voltage Monitor outputs
        shall be inactive (if present) and the Channel Flash settings shall be
        overridden.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.1
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.9.1.1"
    ::= { unit 1 }
```

5.4.2 Automatic Ped Clear Parameter

```
unitAutoPedestrianClear OBJECT-TYPE
    SYNTAX INTEGER { disable(1),
                     enable (2) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Unit Automatic Ped Clear parameter (1 =
        False/Disable 2=True/Enable). When enabled, the CU shall time the
        Pedestrian Clearance interval when Manual Control Enable is active and
        prevent the Pedestrian Clearance interval from being terminated by the
        Interval Advance input.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.2
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.5.3.10"
    ::= { unit 2 }
```

5.4.3 Backup Time Parameter

```
unitBackupTime OBJECT-TYPE
```

```
    SYNTAX Integer32 (0..16777216)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The Backup Time in seconds (0-16777216 sec). When
        any of the defined system control parameters is SET, the backup timer
        is reset. After reset, the CU times the unitBackupTime interval. If the
        unitBackupTime interval expires without a SET operation to any of the
        system control parameters, then the CU shall revert to Backup Mode.

        A value of zero for this object shall disable this feature.
        The BackUp timer reverts to an expired state when unitManualBackup is
enabled.
        A SET to this object does NOT reset the backup timer.

        The system control parameters are:
            phaseControlGroupPhaseOmit,
            phaseControlGroupPedOmit,
            phaseControlGroupHold,
            phaseControlGroupForceOff,
            phaseControlGroupVehCall,
            phaseControlGroupPedCall,
            systemPatternControl,
            systemSyncControl,
            preemptControlState,
            ringControlGroupStopTime,
            ringControlGroupForceOff,
            ringControlGroupMax2,
            ringControlGroupMaxInhibit,
            ringControlGroupPedRecycle,
            ringControlGroupRedRest,
            ringControlGroupOmitRedClear,
            unitControlV4,
            specialFunctionOutputControl,
            ringControlGroupMax3,
            vehicleDetectorControlGroupActuation,
            pedestrianDetectorControlGroupActuation,
            actionPlanControl,
            eclaDataTimestamp,
            eclaRingCurrentPhase,
            eclaRingGreenMinEndTime,
            eclaRingGreenMaxEndTime,
            eclaRingGreenLikelyEndTime,
            eclaRingEndTimeConfidence, and
            eclaRingNextPhase
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.3
    <Object Type> P"
    ::= { unit 3 }
```

5.4.4 Unit Red Revert Parameter

```
unitRedRevert OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The red revert in tenth seconds (0.0 - 25.5
        sec). This value shall provide the minimum red revert time for all
        phases (i.e. if it is greater than a phaseRedRevert object value, then
        this value shall be used as the red revert time for the affected
        phase). This object provides a minimum Red indication following the
        Yellow Change interval and prior to the next display of Green on the
        same signal output driver group.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.4
    <Object Type> P"
    ::= { unit 4 }
```

5.4.5 Unit Control Status

```
unitControlStatus OBJECT-TYPE
    SYNTAX INTEGER { other (1),
    systemControl (2),
    systemStandby (3),
    backupMode(4),
    manual (5),
    localTimebase (6),
    interconnect (7),
    interconnectBackup (8),
    remoteManualControl (9),
    localManualControl (10) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Control Mode for Pattern, Flash, or Free at
        the device:
        other: control by a source other than those listed here.
        systemControl: control by central commands.
        systemStandby: control by local based on central command to use local
control.
        backupMode: Backup Mode (see unitBackupTime).
        manual: when coordOperationalMode is not equal to 0.
        localTimebase: control by the local Time Base.
```

```
        interconnect: control by the local Interconnect inputs.
        interconnectBackup: control by local timebase control due to invalid
Interconnect inputs or loss of sync.
        remoteManualControl: control by central command via remote MCE commands
(See unitMCEIntAdv and unitMCETimeout).
        localManualControl: control via MCE and Interval Advance inputs (e.g.,
police panel)
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.5
    <Object Type> S"
    ::= { unit 5 }
```

5.4.6 Unit Flash Status

```
unitFlashStatus OBJECT-TYPE
    SYNTAX INTEGER { other(1),
    notFlash(2),
    automatic(3),
    localManual(4),
    controllerFaultFlash(5),
    smu(6),
    startup(7),
    preempt (8),
    remoteManual(9) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Flash modes:

        other: the CU is in flash for some other reason.
        notFlash: the CU is not in Flash
        automatic: the CU is currently in an Automatic Flash state due to the CU
programmed settings.
        localManual: the Controller Unit Local Flash input is active, MMU
                Flash input is not active, and Flash is not commanded by
                the central system.
        controllerFaultFlash: the CU initiated the Flash State due to a detected
fault.
        smu: the Controller Unit Signal Monitoring Unit Flash input is active and
the CU is not in Start-Up Flash. This was referred to as an MMU in 1202 v03.
        start-up: the CU is currently timing the Start-Up Flash period.
        preempt: the CU is currently timing the preempt Flash.
        remoteManual: coordOperationalMode or systemPatternControl is equal to
255.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.6
    <Object Type> S"
    ::= { unit 6 }
```

5.4.7 Maximum Alarm Groups

```
maxAlarmGroups OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of alarm
        groups (8 alarm inputs per group) this device supports. This object
        indicates the maximum rows which shall appear in the alarmGroupTable
        object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.11
    <Object Type> S"
    ::= { unit 11 }
```

5.4.8 Alarm Group Table

```
alarmGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AlarmGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table contains external alarm input status in
groups of eight inputs. The number of rows in this table is equal to the
        maxAlarmGroups object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.12"
    ::= { unit 12 }

alarmGroupEntry OBJECT-TYPE
    SYNTAX AlarmGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Status for eight alarm inputs.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.12.1"
    INDEX { alarmGroupNumber }
    ::= { alarmGroupTable 1 }

AlarmGroupEntry ::= SEQUENCE {
    alarmGroupNumber    Integer32,
    alarmGroupState     Integer32 }
```

5.4.8.1 Alarm Group Number

```
alarmGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION "<Definition> The alarm group number for objects in this row.
        This value shall not exceed the maxAlarmGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.12.1.1"
```

5.4.8.2 Alarm Group State

```
alarmGroupState OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Alarm input state bit field. When a bit = 1, the
        associated physical alarm input is active. When a bit = 0, the
        associated alarm input is NOT active.

            Bit 7: Alarm Input # = ( alarmGroupNumber * 8)
            Bit 6: Alarm Input # = ( alarmGroupNumber * 8) -1
            Bit 5: Alarm Input # = ( alarmGroupNumber * 8) -2
            Bit 4: Alarm Input # = ( alarmGroupNumber * 8) -3
            Bit 3: Alarm Input # = ( alarmGroupNumber * 8) -4
            Bit 2: Alarm Input # = ( alarmGroupNumber * 8) -5
            Bit 1: Alarm Input # = ( alarmGroupNumber * 8) -6
            Bit 0: Alarm Input # = ( alarmGroupNumber * 8) -7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.12.1.2
    <Object Type> S"
    ::= {alarmGroupEntry 2 }
```

5.4.9 Maximum Special Function Outputs

```
maxSpecialFunctionOutputs OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "output"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Special Functions this
Controller Unit supports.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.13
    <Object Type> S"
    ::= { unit 13 }
```

5.4.10 Maximum Special Function Outputs

```
specialFunctionOutputTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SpecialFunctionOutputEntry
    MAX-ACCESS not-accessible
    STATUS current
```

```
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit
        special function output objects. The number of rows in this table is
        equal to the maxSpecialFunctionOutputs object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.14"
    ::= { unit 14 }

specialFunctionOutputEntry OBJECT-TYPE
    SYNTAX SpecialFunctionOutputEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Control for Actuated Controller Unit system
        special functions.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.14.1"
    INDEX   { specialFunctionOutputNumber }
    ::= { specialFunctionOutputTable 1 }

SpecialFunctionOutputEntry ::= SEQUENCE {
    specialFunctionOutputNumber          Integer32,
    specialFunctionOutputControl         Integer32,
    specialFunctionOutputStatus          Integer32,
    specialFunctionOutputControlSource   BITS    }
```

5.4.10.1 Special Function Output Number

```
specialFunctionOutputNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "output"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The special function output number associated
        with object in this row. This value shall not exceed the
        maxSpecialFunctionOutputs object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.14.1.1
    <Object Type> S"
    ::= { specialFunctionOutputEntry 1 }
```

5.4.10.2 Special Function Output Control

```
specialFunctionOutputControl OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The special function output (logical or
        physical) in the device may be controlled by this object. 0 = OFF &
        1 = ON. The device shall reset this object to ZERO when in BACKUP
```

```
        Mode.  A write to this object shall reset the BACKUP timer (see
        unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.14.1.3
    <Object Type> C"
    ::= { specialFunctionOutputEntry  3 }
```

5.4.10.3 Special Function Output Status

```
specialFunctionOutputStatus OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The current status (ON-OFF) of the special
        function output (logical or physical) in the device. 0 = OFF & 1 = ON.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.14.1.4
    <Object Type> S"
    ::= { specialFunctionOutputEntry  4 }
```

5.4.10.4 Special Function Output Control Source

```
specialFunctionOutputControlSource OBJECT-TYPE
    SYNTAX BITS {   other(0),
                    remote(1),
                    timebase(2),
                    frontPanel(3) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates why the special function is
active:
    Valid values are:
        Bit 0: Other - the special function was activated from a source not
defined in this standard.
        Bit 1: Remote- the special function was activated remotely through a
management station locally or from the central system.
        Bit 2: Timebase - the special function was activated by the scheduler.
        Bit 3: Front Panel: the special function was commanded from the front
panel.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.14.1.5
    <Object Type> S"
    ::= { specialFunctionOutputEntry  5 }
```

5.4.11 Remote Manual Control Timer

```
unitMCETimeout OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "second"
    MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION "<Definition> This object performs two functions. First, it
        enables manual operation the same as the MCE controller input.

        Second, it serves as a timeout value for a failsafe. If the value is
        non-zero, remote manual control is enabled. When a SET to this object
        is performed, the controller shall load the value into a timer that
        decrements once per second. If the timer reaches zero or if the object
        is SET to zero by the management station, the controller shall
        automatically disable remote manual control. This forces the
        management station to continually refresh this timer to maintain
        remote manual operation. When in remote manual mode, the controller
        should respond exactly as if the MCE input was active.

        This value is the current value of the timer.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.15
    <Object Type> C"
    ::= { unit 15 }
```

5.4.12 Remote Manual Control Advance Command

```
unitMCEIntAdv OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object acts as a remote interval advance
        function. When SET to 1, the controller will behave as if the Interval
        Advance input (a.k.a. police 'pickle') was toggled. When remote manual
        control mode is enabled (see unitMCETimeout) the controller shall
        advance to the next interval and reset the value of this object to 0.
        If this object is SET to 1 when the controller is not in remote manual
        control mode, the controller shall return an inconsistentValue(12) error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.16
    <Object Type> C"
    ::= { unit 16 }
```

5.4.13 ASC Elevation – Antenna Offset

```
ascElevationOffset OBJECT-TYPE
    SYNTAX Integer32 (0..31)
    UNITS "meter"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition>The offset in height, in meters, from the antenna
        of a GNSS or similar geopositioning device to the base of CU cabinet.
        It is assumed that the antenna is at a height higher than the base of
```

```
        the CU structure.

        Values of 0 to 30 provides a range from 0 meters to 30 meters. The
        value of 31 represents unknown.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.17
    <Object Type> P"
    DEFVAL { 31 }
    ::= { unit 17 }
```

5.4.14 Startup Flash Mode

```
unitStartUpFlashMode OBJECT-TYPE
    SYNTAX INTEGER { other(1),
                     automaticFlash (2),
                     cabinetFlash (3),
                     allRedControllerFlash (4) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Defines the startup flash state for the unit. The
        possible states are:

    other: The startup flash state is a state not defined by this standard.
    Automatic Flash: The startup flash state for each signal indication is
        defined by the channelFlash object for each channel.
    Cabinet Flash: Upon startup, the intersection shall flash based on the
        cabinet's internal flash transfer relay configuration.
    All Red Controller Flash: Upon startup, the intersection shall flash
        all-red regardless of the channelFlash object settings.
    <Object Type> P
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.18"
    DEFVAL { 2 }
    ::= { unit 18 }
```

5.4.15 Communications

```
commPorts OBJECT IDENTIFIER
    ::= { unit 23 }

-- This node shall contain objects that configure, monitor or control
communications ports functions for this device.
```

5.4.15.1 Maximum Communications Ports

```
maxCommPorts OBJECT-TYPE
    SYNTAX Integer32 (1..16)
    UNITS "port"
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of communications Ports this
device supports and is the maximum value of ifNumber allowed.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.23.1
    <Object Type> S"
    ::= { commPorts 1 }
```

5.4.15.2 Communications Ports Table

```
commPortsTable OBJECT-TYPE
    SYNTAX SEQUENCE OF CommPortsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table used to describe and configure the
communication ports on the CU. The number of rows in this table is equal to the
maxCommPorts object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.23.2"
    ::= { commPorts 2 }

commPortsEntry OBJECT-TYPE
    SYNTAX CommPortsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a row in the Communication
Ports Table which is used to extend the ifTable of RFC 1213. It uses ifIndex from
RFC 1213 as the index, so all communications ports in the controller has to be
included in the ifTable including async RS232 ports which are also in the
rs232PortTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.23.2.1"
    INDEX { ifIndex }
    ::= { commPortsTable 1 }

CommPortsEntry ::= SEQUENCE {
    commPortType                INTEGER,
    commPortTypeIndex           Integer32,
    commPortEnable              INTEGER }
```

5.4.15.2.1 Communications Port Type

```
commPortType OBJECT-TYPE
    SYNTAX INTEGER { other (1),
    ethernet (2),
    rs232like (3) }
    MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION "<Definition> This object identifies the port type. Although RFC
1213 contains ifType, this object provides the user/device further guidance to
additional objects that define the port and its configuration. The valid port
types are defined as follows:
other: the port type is not defined by this standard.
Ethernet: the port type is Ethernet based (i.e. IP) and configured per NTCIP 2104
and 2202
rs232like: the port type is rs-232-like (per RFC 1317) and may be either an
asynchronous or synchronous serial. Port configuration is contained in RFC 1317
rs232PortTable
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.23.2.1.1
    <Object Type> S"
    ::= { commPortsEntry 1 }
```

5.4.15.2.2 Communications Port Type Number Parameter

```
commPortTypeIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object contains the value to be used as the
index into the appropriate communication port configuration table. If
commPortType = rs232like, then this value is used for RFC 1317 rs232PortIndex
into the rs232PortTable. If commPortType is ethernet then this object is
redundant and should have the same value as ifIndex.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.23.2.1.2
    <Object Type> S"
    ::= { commPortsEntry 2 }
```

5.4.15.2.3 Communications Port Enable

```
commPortEnable OBJECT-TYPE
    SYNTAX INTEGER { enabled (1),
                     disabled (2) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to enable/disable a
communications port on the device unless prohibited by the hardware specification
or hardware standard. Unused communications port numbers shall be set to disabled
(2), and by default, all available communications ports shall be disabled. A GET
of this object returns the current state of the port.

Note that the device may include other means to disable a communications port
(e.g. Port 1 Disable input). Attempting to enable a port that is disabled by
```

```
these other means will fail and a subsequent GET of this object will indicate
that the port is still disabled.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.23.2.1.3
    <Object Type> C"
    ::= { commPortsEntry 3 }
```

5.4.16 Unit Alarm Status

```
unitAlarmStatus OBJECT-TYPE
    SYNTAX BITS {  cycleFault(0),
                   coordFault(1),
                   coordFail(2),
                   cycleFail(3),
                   localFree(4),
                   coordActive(5),
                   ioLinkError(6),
                   smuLinkError(7),
                   rsuLinkError(8),
                   cvCertificateError(9),
                   preemmptMaxPresence(10),
                   eclaActive(11)    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Device Alarm Mask ( 0 = False, 1 = True) as
        follows:
        Bit 0: Cycle Fault - When the CU is operating in the coordinated mode and
cycling diagnostics indicate that a serviceable call exists that has not been
serviced for two cycles.
        Bit 1: Coord Fault - When a Cycle Fault is in effect and the serviceable
call has been serviced within two cycles after the Cycle Fault.
        Bit 2: Coord Fail - When a Coord Fault is in effect and a Cycle Fault
occurs again within two cycles of the coordination retry.
        Bit 3: Cycle Fail - When a local CU is operating in the non-coordinated
mode, whether the result of a Cycle Fault or Free being the current normal mode,
and cycling diagnostics indicate that a serviceable call exists that has not been
serviced for two cycles.
        Bit 4: Local Free - When any of the CU inputs and/or programming cause it
not to run coordination. The localFreeStatus object indicates why the CU is not
running coordination.
        Bit 5: Coordination Active - When coordination is active and not
preempted or overridden.
        Bit 6: IO Link Error - When the CU has a communication error with an IO
device in the cabinet.
        Bit 7: SMU Link Error - Whenever the CU is unable to communicate to the
signal monitoring unit because the the communication link is failed (e.g.,
timeouts, errors) or because the signal monitoring unit has been removed.
```

```
        Bit 8: RSU Link Error - Whenever the CU is configured to communicate with
a RSU but the communications link is failed (e.g., timeouts, errors).
        Bit 9: CV Certificate Error - Whenever the CU detects an error related to
invalid connected vehicle certificates.
        Bit 10: Preempt Maximum Presence - Preempt maximum presence time
exceeded.
        Bit 11: ECLA Active- The CU is currently being controlled by an External
Local Control Application.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.28
    <Object Type> S"
    ::= { unit 28 }
```

5.4.17 Short Alarm Status

```
shortAlarmStatusV4 OBJECT-TYPE
    SYNTAX BITS { preemptActive(0),
                flashActive(1),
                transitioning(2),
                localOveride(3),
                coordinationAlarm(4),
                detectorFault(5),
                externalAlarm(6),
                stopTime(7),
                priorityCall(8) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Short Alarm Mask ( 0 = False, 1 = True) as
        follows:

        Bit 0: Preempt - When any of the CU Preempt inputs become active.
        Bit 1: Flash Active - When the CU is in Flash. The unitFlashStatus object
indicates the reason the CU is in flash.
        Bit 2: Transitioning - Whenever the CU is performing an offset
                transition (correction in process).
        Bit 3: Local Override - When any of the CU inputs and/or programming
                cause it not to run coordination.
        Bit 4: Coordination Alarm - When the CU is not running the called
                pattern without offset correction within three cycles of the
                command. An offset correction requiring less than three cycles
                due to cycle overrun caused by servicing a pedestrian call
                shall not cause a Coordination Alarm.
        Bit 5: Detector Fault - When any vehicle or pedestrian detector fault
occurs.
        Bit 6: External Alarm - When a user configured alarm input is active. The
alarmGroupTable indicates which alarm input is active.
        Bit 7: Stop Time - When the Stop Time input is active.
```

```
        Bit 8: Priority Call - The CU is currently responding to a Priority Call
using NTCIP 1211 or another method.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.29
    <Object Type> S"
    ::= { unit 29 }
```

5.4.18 Unit Control

```
unitControlV4 OBJECT-TYPE
    SYNTAX BITS {  globalMinRecall(0),
                nonactuated1(1),
                nonactuated2(2),
                walkRestModifier(3),
                interconnect(4) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
        activate unit functions in the device (0 = False / Disabled, 1 = True
        / Enabled) as follows:

        Bit 0: Global Minimum Recall - when set to 1, causes a recurring
                demand on all vehicle phases for a minimum vehicle service.
                REFERENCE NEMA TS 2 Clause 3.5.5.5.9
        Bit 1: Call to Non-Actuated 1 - when set to 1, causes any phase(s)
                appropriately programmed in the phaseOptions object to operate
                in the Non-Actuated Mode.
                REFERENCE NEMA TS 2 Clause 3.5.5.5.8
        Bit 2: Call to Non-Actuated 2 - when set to 1, causes any phase(s)
                appropriately programmed in the phaseOptions object to operate
                in the Non-Actuated Mode.
                REFERENCE NEMA TS 2 Clause 3.5.5.5.8
        Bit 3: Walk Rest Modifier - when set to 1, causes non-actuated phases
                to remain in the timed-out Walk state (rest in Walk) in the
                absence of a serviceable conflicting call.
                REFERENCE NEMA TS 2 Clause 3.5.5.5.13
        Bit 4: Interconnect - when set to 1, shall cause the interconnect
                inputs to operate at a higher priority than the timebase
                control (TBC On Line).
                REFERENCE NEMA TS 2 Clause 3.6.2.3 and 3.8.3

    The device shall reset this object to ZERO when in BACKUP Mode. A write to
this object shall reset the BACKUP timer (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.30
    <Object Type> C"
    ::= { unit 30 }
```

5.4.19 Unit Manual Backup

```
unitManualBackup OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to enable BackUp Mode,
regardless of the current BackUp timer.
    When enabled, the BackUp timer reverts to an expired state and remains
expired until this setting is disabled.
    When enabled, all objects defined in unitBackUpTime (C objects) are reset to
ZERO and any SET attempt to these objects shall result in an error of
inconsistentValue(12).
     (0 = False / Disabled, 1 = True/ Enabled)
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.3.31
    <Object Type> P"
    ::= { unit 31 }
```

5.5 Coordination Parameters

```
coord OBJECT IDENTIFIER
    ::= { asc 4 }

-- The coord node contains objects that support coordination configuration,
status and control functions for the device.
```

5.5.1 Coordination Operational Mode Parameter

```
coordOperationalMode OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the operational mode for
        coordination. The possible modes are:

        Value   DESCRIPTION
        0       Automatic - this mode provides for coord operation, free,
                    and flash to be determined automatically by the possible
                    sources (i.e. Interconnect, Time Base, or System Commands).
        1-253   Manual Pattern - these modes provide for Coord operation
                    running this pattern. This selection of pattern overrides
                    all other pattern commands.
        254     Manual Free - this mode provides for Free operation without
                    coordination or Automatic Flash from any source.
        255     Manual Flash - this mode provides for Automatic Flash without
                    coordination or Free from any source.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.1
```

```
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.6.2.4"
    ::= { coord 1 }
```

5.5.2 Coordination Correction Mode Parameter

```
coordCorrectionMode OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     dwell (2),
                     shortway (3),
                     addOnly (4),
                     subtractOnly (5) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the Coordination Correction
Mode.
        The possible modes are:
        other: the coordinator establishes a new offset by a mechanism not
            defined in this standard.
        dwell: when changing offset, the coordinator shall establish a new
            offset by dwelling in the coord phase(s) until the desired offset
            is reached.
        shortway (Smooth): when changing offset, the coordinator shall
            establish a new offset by adding or subtracting to/from the
            timings in a manner that limits the cycle change. This operation
            is performed in a device specific manner.
        addOnly: when changing offset, the coordinator shall establish a new
            offset by adding to the timings in a manner that limits the cycle
            change. This operation is performed in a device specific manner.
        subtractOnly: when changing offset, the coordinator shall establish a
            new offset by subtracting from the timings in a manner that limits
            the cycle change. This operation is performed in a device specific
            manner.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.2
    <Object Type> P"
    DEFVAL { 3 }
    ::= { coord 2 }
```

5.5.3 Coordination Correction Mode Parameter

```
coordMaximumMode OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     maximum1 (2),
                     maximum2 (3),
                     maxInhibit (4),
                     maximum3 (5) }
```

```
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "This object defines the Coordination Maximum Mode. The possible
        modes are:
        other: the maximum mode is determined by some other mechanism not
            defined in this standard.
        maximum1: the internal Maximum 1 Timing shall be effective while
            coordination is running a pattern.
        maximum2: the internal Maximum 2 Timing shall be effective while
            coordination is running a pattern.
        maxInhibit: the internal Maximum Timing shall be inhibited while
            coordination is running a pattern.
        maximum3: the internal Maximum 3 Timing shall be effective while
            coordination is running a pattern.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.3
    <Object Type> P"
    DEFVAL { 4 }
    ::= { coord 3 }
```

5.5.4 Coordination Force Mode Parameter

```
unitCoordForceMode OBJECT-TYPE
    SYNTAX INTEGER { other(1),
                     floating (2),
                     fixed (3) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the Unit Force Mode. The
        possible modes are:
        other: the CU implements a mechanism not defined in this standard.
        floating: each non-coord phase will be forced to limit its time to the
            split time value. This allows unused split time to revert to the
            coord phase.
        fixed: each non-coord phase will be forced at a fixed position in the
            cycle. This allows unused split time to revert to the following
             phase.
    This does not apply to phases operating with Max Vehicle Recall, a non-
actuated split mode, or a Call to Non-Actuated input.
    This object was named coordForceMode in 1202 v03.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.4
    <Object Type> P"
    DEFVAL { 3 }
    ::= { coord 4 }
```

5.5.5 Maximum Patterns

```
maxPatterns OBJECT-TYPE
    SYNTAX Integer32 (1..253)
    UNITS   "patterns"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The maximum number of Patterns this Controller Unit
supports. This object indicates how many rows are in the patternTable object (254
and 255 are defined as non-pattern Status for Free and Flash).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.5
    <Object Type> S"
    ::= { coord 5 }
```

5.5.6 Pattern Table

```
patternTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PatternEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit
coordination Pattern parameters. The number of rows in this table is equal to the
maxPatterns object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7"
    ::= { coord 7 }

patternEntry OBJECT-TYPE
    SYNTAX PatternEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Actuated Controller Unit
pattern.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1"
    INDEX   { patternNumber }
    ::= { patternTable 1 }

PatternEntry ::= SEQUENCE {
    patternNumber                   Integer32,
    patternCycleTime                Integer32,
    patternOffsetTime               Integer32,
    patternSplitNumber              Integer32,
    patternSequenceNumber           Integer32,
    patternMaximumMode              INTEGER,
    patternSpatEnabledLanes         Integer32,
    patternReferencePhase           Integer32,
```



```
    patternReferencePoint           INTEGER,
    patternPhaseSet                 Integer32,
    patternOverlapSet               Integer32,
    patternVehicleDetectorSet       Integer32,
    patternPedestrianDetectorSet    Integer32,
    patternSpecialFunction          Unsigned32 }
```

5.5.6.1 Pattern Number

```
patternNumber OBJECT-TYPE
    SYNTAX Integer32 (1..253)
    UNITS   "pattern"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The pattern number for objects in this row. This
        value shall not exceed the maxPatterns object value. This value may be
referenced by ISO 26048-1.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.1"
    ::= { patternEntry 1 }
```

5.5.6.2 Pattern Cycle Time Parameter

```
patternCycleTime OBJECT-TYPE
    SYNTAX Integer32 (0..999)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The patternCycleTime object specifies the length
        of the pattern cycle in seconds (NEMA TS 2 range: 30-255). A value of
zero
        is used for a pattern where the ASC runs in Free mode. Otherwise, the
        pattern is configured for coordination.

        A pattern cycle time less than adequate to service the minimum
        requirements of all phases shall result in Free mode. While this
        condition exists, the Local Free bit of unitAlarmStatus shall be 1, the
        Local Override bit of shortAlarmStatusV4 shall be 1, the value
        localFreeStatus shall be badPattern(7), and the Bad Cycle Time bit of
        coordPatternFaultStatus shall be 1.

        The minimum time to serve a phase must satisfy vehicle minimum green,
        walk (if configured), pedestrian clear (if configured), yellow change,
        and red clearance time.

        If the pattern cycle time is zero and the associated split table (if
        any) contains values greater than zero, then the ASC shall utilize the
```

```
        split time values as maximum values for each phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.2
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.6.2.1.1"
    DEFVAL { 0 }
    ::= { patternEntry 2 }
```

5.5.6.3 Pattern Offset Time Parameter

```
patternOffsetTime OBJECT-TYPE
    SYNTAX Integer32 (0..998)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The patternOffsetTime defines by how many seconds
        (NEMA TS 2 range: 0-254) the local time zero shall lag the system time
        zero (synchronization pulse) for this pattern.

        An offset value equal to or greater than the patternCycleTime other
        than 0 shall result in the ASC running in Free mode. While this
        condition exists, the Local Free bit of unitAlarmStatus shall be 1, the
        Local Override bit of shortAlarmStatusV4 shall be 1, the value of
        localFreeStatus shall be badPattern(7), and the Invalid Offset bit of
        coordPatternFaultStatus shall be 1.

        The value of this object is ignored when patternCycleTime is 0.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.3
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.6.2.2"
    DEFVAL { 0 }
    ::= { patternEntry 3 }
```

5.5.6.4 Pattern Split Number Parameter

```
patternSplitNumber OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS   "split"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to locate information in the
splitTable to use for this pattern. This value shall not exceed the maxSplits
object value. A value of zero is used for a pattern where the ASC runs in Free
mode.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.4
    <Object Type> P"
    DEFVAL { 0 }
```

```
    ::= { patternEntry 4 }
```

5.5.6.5 Pattern Sequence Number Parameter

```
patternSequenceNumber OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "sequence"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to locate information in the
sequenceTable to use with this pattern. This value shall not exceed the
maxSequences object value. A value of zero indicates no sequence is specified and
Automatic Flash without coordination is to run when this pattern is called.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.5
    <Object Type> P"
    DEFVAL { 1 }
    ::= { patternEntry 5 }
```

5.5.6.6 Pattern Maximum Mode Parameter

```
patternMaximumMode OBJECT-TYPE
    SYNTAX INTEGER { other (1),
        coordMaximumMode (2),
        maxInhibit (3),
        maximum1 (4),
        maximum2 (5),
        maximum3 (6) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object represents the maximum mode to be used
for the pattern. The valid maximum modes are:
other: the maximum mode is determined by some other mechanism not defined in this
standard.
coordMaximumMode: use the maximum mode defined by the coordMaximumMode object.
maxInhibit: the internal maximum timing shall be inhibited while coordination is
running this pattern.
maximum1: the internal Maximum 1 Timing shall be effective while coordination is
running this pattern.
maximum2: the internal Maximum 2 Timing shall be effective while coordination is
running this pattern.
maximum3: the internal Maximum 3 Timing shall be effective while coordination is
running this pattern.

This object was named patternOptions in 1202 v03.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.7
    <Object Type> P"
```

```
    DEFVAL { 2 }
    ::= { patternEntry 7 }
```

5.5.6.7 Pattern Enabled Lanes

```
patternSpatEnabledLanes OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This value is equal to the enabledLaneIndex
        (row) in the spatEnabledLanesConcurrencyTable that contains the
        enabledLaneConcurrency lanes to be active when this pattern is active.
        A value of zero indicates that no revocable lanes are to be enabled as
part of this pattern. Any lane that is not revocable remains active. This value
shall not exceed maxPatterns.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.8
    <Object Type> P"
    DEFVAL { 0 }
    ::= { patternEntry 8 }
```

5.5.6.8 Pattern Reference Phase Parameter

```
patternReferencePhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "phase"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to identify the phase that
contains the reference point for when the pattern cycle starts and ends. This
value shall not exceed maxPhases. This phase is must be coordinated phase if the
pattern is used for coordination. The reference point is identified by
patternReferencePoint. A value of zero is used for a pattern where the ASC runs
in Free mode.

    If the selected phase is not in the sequence identified in
patternSequenceNumber or if this phase is not a coordinated phase as specified in
associated split table (if any), the device shall operate in Free Mode. While
this condition exists, the Local Free bit of unitAlarmStatus shall be 1, the
Local Override bit of shortAlarmStatusV4 shall be 1, the value of localFreeStatus
is badPattern(7), and the Invalid Reference Point bit of coordPatternFaultStatus
is 1.

    The value of this object is ignored when patternCycleTime is 0.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.9
    <Object Type> P"
    DEFVAL { 0 }
```

```
    ::= { patternEntry 9 }
```

5.5.6.9 Pattern Reference Point Parameter

```
patternReferencePoint OBJECT-TYPE
    SYNTAX INTEGER { other(1),
                    greenBegin (2),
                    yellowBegin (3),
                    redBegin (4),
                    redEnd (5) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to indicate the location of the
reference point for when the pattern cycle starts and ends. This object
identifies an interval within the phase identified by the patternReferencePhase.
The valid values are:

    other: The reference point is a position not defined by this object.
    greenBegin: The start of the vehicle GREEN interval. If
phasePedAdvanceWalkTime is used, then this is the start of the walk interval.
    yellowBegin: The start of the yellow change interval.
    redBegin: The start of the red clearance interval.
    redEnd: The end of the red clearance interval.

    The value of this object is ignored when patternCycleTime is 0.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.10
    <Object Type> P"
    ::= { patternEntry 10 }
```

5.5.6.10 Pattern Phase Set Number Parameter

```
patternPhaseSet OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to identify the phase set to be
used with this pattern. This value shall not exceed the maxPhaseSets object
value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.11
    <Object Type> P"
    DEFVAL { 1 }
    ::= { patternEntry 11 }
```

5.5.6.11 Pattern Overlap Set Number Parameter

```
patternOverlapSet OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to identify the overlap set to
be used with this pattern. This value shall not exceed the maxOverlapSets object
value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.12
    <Object Type> P"
    DEFVAL { 1 }
    ::= { patternEntry 12 }
```

5.5.6.12 Pattern Vehicle Detector Set Number Parameter

```
patternVehicleDetectorSet OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to identify the vehicle
detector set to be used with this pattern. This value shall not exceed the
maxVehicleDetectorSets object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.13
    <Object Type> P"
    DEFVAL { 1 }
    ::= { patternEntry 13 }
```

5.5.6.13 Pattern Pedestrian Detector Set Number Parameter

```
patternPedestrianDetectorSet OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to identify the vehicle
detector set to be used with this pattern. This value shall not exceed the
maxVehicleDetectorSets object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.14
    <Object Type> P"
    DEFVAL { 1 }
    ::= { patternEntry 14 }
```

5.5.6.14 Pattern Special Functions Parameter

```
patternSpecialFunction OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The Special Functions that shall be active when
this pattern is active. Each bit represents a special function # from 1 to 32.
        Bit 31: Special Function 32
        Bit 30: Special Function 31
        Bit 29: Special Function 30
        .... .... .... ....
        Bit 3: Special Function 4
        Bit 2: Special Function 3
        Bit 1: Special Function 2
        Bit 0: Special Function 1

        Bit = 0 - False/Disabled, Bit = 1 - True/Enabled

        If the ASC supports less than 32 special functions, a SET of a bit for an
unsupported special function to a value other than zero (0) shall return a
wrongValue(10) error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.7.1.15
    <Object Type> P"
     DEFVAL { 0 }
    ::= { patternEntry 15 }
```

5.5.7 Maximum Splits

```
maxSplits OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS   "split"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The maximum number of Split Plans this Actuated
Controller Unit supports. This object indicates how many Split plans are in the
splitTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.8
    <Object Type> S"
    ::= { coord 8 }
```

5.5.8 Split Table

```
splitTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SplitEntry
    MAX-ACCESS not-accessible
    STATUS current
```

```
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit
coordination split parameters. The number of rows in this table is equal to
maxSplits.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9"
    ::= { coord 9 }

splitEntry OBJECT-TYPE
    SYNTAX SplitEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Split type Parameters for a specific  Actuated
Controller Unit phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9.1"
    INDEX   { splitNumber, splitPhase }
    ::= { splitTable 1 }

SplitEntry ::= SEQUENCE {
    splitNumber         Integer32,
    splitPhase          Integer32,
    splitTime           Integer32,
    splitMode           INTEGER,
    splitCoordPhase     INTEGER,
    splitOptions        Integer32,
    splitCoordForceMode INTEGER}
```

5.5.8.1 Split Numbers

```
splitNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS   "split"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The object defines which rows of the split table
comprise a split group. All rows that have the same splitNumber are in the same
split group. The value of this object shall not exceed the maxSplits object
value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9.1.1"
    ::= { splitEntry 1 }
```

5.5.8.2 Split Phase Number

```
splitPhase OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "phase"
    MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION "<Definition> The phase number for objects in this row. The value
of this object shall not exceed the maxPhases object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9.1.2"
    ::= { splitEntry 2 }
```

5.5.8.3 Split Time Parameter

```
splitTime OBJECT-TYPE
    SYNTAX Integer32 (0..999)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The time in seconds the splitPhase is allowed to
receive (i.e. before a Force Off is applied) when constant demands exist on all
phases. In floating force mode for coordination this is always the maximum time a
non-coordinated phase is allowed to receive. In fixed force mode for
coordination, the actual allowed time may be longer if a previous phase gapped
out.

    The splitTime includes all phase clearance times for the associated phase.
The split time shall at minimum the sum of the phase minimum service requirements
for the phase. When the time is NOT adequate to service the minimum service
requirements of the phase, the ASC shall operate in Free Mode.

    The minimum time to serve a phase must satisfy vehicle minimum green, walk
(if configured), pedestrian clear (if configured), yellow change, and red
clearance time.

    If the patternCycleTime entry of the associated patternTable entry is zero
(i.e. the device is in Free Mode), then the value of this object shall be
applied, if non-zero, as a maximum time for the associated phase.

    If the sum of all splitTime values for this split group is less than the
patternCycleTime entry of the associated patternTable entry, all extra time is
allotted to the coordinated phase in each ring.

    If the sum of all splitTime values for this split group is greater than the
patternCycleTime entry of the associated patternTable entry (and the
patternCycleTime is not zero) the device shall operate in the Free Mode. While
this condition exists, the splitOverrun bit of coordPatternFaultStatus is 1.

    While the Free Mode condition exists, the Local Free bit of unitAlarmStatus
shall be 1, the Local Override bit of shortAlarmStatusV4 shall be 1, and the
value of localFreeStatus is badPattern(7).
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9.1.3
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.6.2.1.2"
    ::= { splitEntry 3 }
```

5.5.8.4 Split Mode Parameter

```
splitMode OBJECT-TYPE
    SYNTAX INTEGER { other(1),
        none (2),
        minimumVehicleRecall (3),
        maximumVehicleRecall (4),
        pedestrianRecall (5),
        maximumVehicleAndPedestrianRecall (6),
        phaseOmitted (7),
        nonActuated (8) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines operational characteristics of
the phase. The following options are available:
other: the operation is not specified in this standard
none: no split mode control.
minimumVehicleRecall: this phase operates with a minimum vehicle recall.
maximumVehicleRecall: this phase operates with a maximum vehicle recall. This
value shall also be used for bicycle phase recalls and transit phase recalls.
pedestrianRecall: this phase operates with a pedestrian recall. The settings for
minimumVehicleRecall also apply.
maximumVehicleAndPedestrianRecall: this phase operates with a maximum vehicle &
pedestrian recall.
phaseOmitted: this phase is omitted.
nonActuated: this phase operates with a fixed split time.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9.1.4
    <Object Type> P"
     DEFVAL { 2 }
    ::= { splitEntry 4 }
```

5.5.8.5 Split Coordinated Phase

```
splitCoordPhase OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> To select the associated phase as a coordinated
phase this object shall be set to TRUE (non zero).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9.1.5"
    ::= { splitEntry 5 }
```

5.5.8.6 Split Options

```
splitOptions OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Optional split functions (0 = False/Disabled, 1 =
True/Enabled).
Bit 7: Reserved
Bit 6: Reserved
Bit 5: Reserved
Bit 4: Reserved
Bit 3: Reserved
Bit 2: Reserved
Bit 1: Reserved
Bit 0: Transition Phase Omit - To allow the associated phase to be omitted during
coord Correction Mode (transitions), this object shall be set to TRUE (1). If the
associated phase is not allowed to be omitted, this object shall be set to FALSE
(0).

A SET of a 'reserved' bit to a value other than zero (0) shall return a
wrongValue(10) error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9.1.6
    <Object Type> P"
    ::= { splitEntry 6 }
```

5.5.8.7 Split Phase Coord Force Mode Parameter

```
splitCoordForceMode OBJECT-TYPE
    SYNTAX INTEGER { other(1),
                     unitCoordForceMode(2),
                     floating (3),
                     fixed (4) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the Force Mode for the phase
when the split is in use. The possible modes are:
        other: the CU implements a mechanism not defined in this standard or this
object is not applicable.
        unitCoordForceMode: The CU uses the current setting of the
unitCoordForceMode object for this phase.
        floating: The phase will be forced to limit its time to the splitTime
value.
        fixed: The phase will be forced off at a fixed position in the cycle,
allowing its on-time to exceed the splitTime value
```

```
    The value of this object for this phase is not used when splitCoordPhase is
equal to '1' or if the splitMode is nonActuated.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.9.1.7
    <Object Type> P"
     DEFVAL { 2 }
    ::= { splitEntry 7}
```

5.5.9 Coordination Pattern Status

```
coordPatternStatus OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object defines the running coordination
pattern/mode in the device. The possible values are:
    Value   Description
    1-253   Pattern - indicates the currently running pattern
    254 Manual Free - indicates Manual Free operation without coordination.
    255 Manual Flash - indicates Manual Flash without coordination.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.10
    <Object Type> S"
    ::= { coord 10 }
```

5.5.10 Local Free Status

```
localFreeStatus OBJECT-TYPE
    SYNTAX INTEGER { other(1),
    notFree(2),
    commandFree(3),
    transitionFree(4),
    inputFree(5),
    coordFree(6),
    badPattern(7),
    badCycleTime(8),--deprecated
    splitOverrun (9),--deprecated
    invalidOffset (10),--deprecated
    failed(11) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Free modes:
    other: some other condition has caused the device to run in free mode.
    notFree: the unit is not running in free mode.
    commandFree: the CU is running Free because there is no pattern command.
    transitionFree: the CU has a pattern command but is cycling to a point to
begin coordination.
```

```
    inputFree: one of the CU inputs cause it to not respond to coordination.
    coordFree: the CU is programmed to run Free.
    badPattern: The CU is running Free because the called pattern is invalid. The
coordPatternFaultStatus object indicates why the called pattern is invalid.
    failed: cycling diagnostics have called for Free.

An ASC may provide diagnostics beyond those stated herein. Therefore, for a set
of given bad data, the free status between devices may be inconsistent.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.11
    <Object Type> S"
    ::= { coord 11 }
```

5.5.11 Coordination Cycle Status

```
coordCycleStatus OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "second"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Coord Cycle Status represents the current
position
    in the local coord cycle of the running pattern (0 to 1898 sec). This value
normally
    counts down from patternCycleTime to zero. This value may exceed the
patternCycleTime
    during a coord cycle with offset correction (patternCycleTime + correction).

    Values 1899-65534 are not used. A value of 65535 indicates that the
controller is not in coordination because it is running Free or in preemption.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.12
    <Object Type> S"
    ::= { coord 12 }
```

5.5.12 Coordination Cycle Status

```
coordCycleStatus OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "second"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Coord Cycle Status represents the current
position
    in the local coord cycle of the running pattern (0 to 1898 sec). This value
normally
    counts down from patternCycleTime to zero. This value may exceed the
patternCycleTime
```

```
    during a coord cycle with offset correction (patternCycleTime + correction).

    Values 1899-65534 are not used. A value of 65535 indicates that the
controller is not in coordination because it is running Free or in preemption.

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.12
    <Object Type> S"
    ::= { coord 12 }
```

5.5.13 System Pattern Control

```
systemPatternControl OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to establish the Called
        System Pattern/Mode for the device. The possible values are:

        Value   DESCRIPTION
        0       Standby - the system relinquishes control of the device.
        1-253   Pattern - these values indicate the system commanded pattern
        254     Free - this value indicates a call for Manual Free
        255     Flash - this value indicates a call for Manual Flash

        If an unsupported / invalid pattern is called, Free shall be the
        operational mode. The device shall reset this object to ZERO when in
        BACKUP Mode. A write to this object shall reset the Backup timer to
        ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.14
    <Object Type> C"
    DEFVAL { 0 }
    ::= { coord 14 }
```

5.5.14 System Sync Control

```
systemSyncControl OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to establish the system
        reference point for the Called System Pattern by providing the current
        position in the system pattern cycle (0-998 sec). The device shall
        recognize a write to this object as a command to establish the time
        until the next system reference point. Thereafter, the system
        reference point shall be assumed to occur at a frequency equal to the
```

```
        patternCycleTime.

        Values 999-65534 are not used. When the value in the object is 65535, the
system REFERENCE point shall
        be referenced to the local Time Base in accordance with its
        programming.

        This CU must maintain an accuracy of 0.1 seconds based on the receipt
        of the SET packet. The device shall reset this object to ZERO when in
        BACKUP Mode. A write to this object shall reset the Backup timer to
        ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.15
    <Object Type> C"
    ::= { coord  15 }
```

5.5.15 System Sync Control

```
coordCurrentOffset OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "second"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The value of the offset currently in effect from 0
to 998 seconds. This value value may change between or during cycles as a result
of the CU coordination process transitioning to or maintaining the programmed
pattern offset. This value may also change during the cycle as the result of a
priority call.

    Values 999-65534 are not used. A value of 65535 indicates that the controller
is not in coordination because it is running Free or in preemption.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.17
    <Object Type> S"
    REFERENCE "NEMA TS 2 Clause 3.6.2.2"
    ::= { coord 17 }
```

5.5.16 Coordination Pattern Source

```
coordPatternSource OBJECT-TYPE
    SYNTAX BITS     {   other(0),
                        remote(1),
                        timebased(2),
                        frontPanel(3),
                        backup(4) }
    MAX-ACCESS read-only
    STATUS current
```



```
    DESCRIPTION "<Definition> This object indicates why the current pattern or
mode is currently in effect.
    Valid values are:
        other: the current pattern or mode was activated from a source not
defined in this standard.
        remote: the current pattern or mode was commanded from a management
locally or through a central system.
        timebased: the current pattern or mode is running as scheduled and not as
a result of the CU going into backup mode.
        frontPanel: the current pattern or mode was commanded from the front
panel.
        backup: the current pattern or mode is running because the CU went into
backup mode. The CU may go into backup mode due to a failure.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.18
    <Object Type> S"
    ::= { coord 18 }
```

5.5.17 Coordination Pattern Fault Source

```
coordPatternFaultStatus OBJECT-TYPE
    SYNTAX BITS { other(0),
                badCycleTime(1),
                splitOverrun(2),
                invalidOffset(3),
                invalidReferencePoint(4) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates any fault with the called
pattern if the called pattern is invalid.
    Vailid values are:
        Bit 0: Other - The pattern cannot run due to a reason not defined in this
standard.
        Bit 1: Bad Cycle Time - The pattern cycle time is not adequate to service
the minimum requirements of all phases, except when patternCycleTime is 0. The
minimum time to serve a phase must satisfy vehicle minimum green, walk (if
configured), pedestrian clear (if configured), yellow change, and red clearance
time.
        Bit 2: Split Overrun - The sum of all split times exceeds the programmed
patternCycleTime value.
        Bit 3: Invalid Offset - The programmed pattern offset time value is not
less than the programmed cycle time value, except when patternCycleTime is 0.
        Bit 4: Invalid Reference Point - If the pattern is configured for
coordination (patternCycleTime is greater than 0), the pattern reference point is
invalid if the patternReferencePhase is not a coordinated phase as defined by
```

```
splitCoordPhase in the associated split group or not in the pattern sequence as
defined in the sequenceData of the associated sequence.

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.4.19
    <Object Type> S"
    ::= { coord 19 }
```

5.6 Time Base Parameters

```
timebaseAsc OBJECT IDENTIFIER
    ::= { asc 5 }

-- This object is an identifier used to group all objects for support of timebase
functions. If a device implements timebase functions then these objects shall be
supported.
```

5.6.1 Time Base Pattern Sync Parameter

```
timebaseAscPatternSync OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "minute"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Pattern Sync Reference in minutes past midnight.
This is the time that the master cycle begins and local cycle offset is relative.
When the value is 65535, the controller unit shall use the Action time as the
Sync Reference for that pattern. This value is defined in ISO 26048-1.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.1
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.8.2"
    ::= { timebaseAsc 1 }
```

5.6.2 Maximum Time Base Actions

```
maxTimebaseAscActions OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "action"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Actions this device
        supports. This object indicates the maximum rows which shall appear
        in the timebaseAscActionTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.2
    <Object Type> S"
    ::= { timebaseAsc 2 }
```

5.6.3 Time Base Asc Action Table

```
timebaseAscActionTable OBJECT-TYPE
    SYNTAX SEQUENCE OF TimebaseAscActionEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit Time
        Base action parameters. The number of rows in this table is equal to
        the maxTimebaseAscActions object.

        All the functionality of the timebaseAscActionTable is supported by
patternTable. Previous versions of NTCIP 1202 only supported use of the
timebaseAscActionTable to implement scheduled special functions. NTCIP 1202 v04
recommends using the patternTable to implement scheduled special functions as
part of timing patterns but implementations that use the timebaseAscActionTable
are still conformant.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.3"
    ::= { timebaseAsc 3 }

timebaseAscActionEntry OBJECT-TYPE
    SYNTAX TimebaseAscActionEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Action Parameters for a Actuated Controller Unit
        Time Base Program.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.3.1"
    INDEX   { timebaseAscActionNumber }
    ::= { timebaseAscActionTable 1 }

TimebaseAscActionEntry ::= SEQUENCE {
    timebaseAscActionNumber      Integer32,
    timebaseAscPattern           Integer32,
    timebaseAscSpecialFunction   Unsigned32,
    timebaseAscEnabledLane       Integer32 }
```

5.6.3.1 Time Base Action Number

```
timebaseAscActionNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "action"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The time base Action number for objects in this
        row. This value shall not exceed the maxTimebaseAscActions object
        value. This value may be referenced by ISO 26048-1.
```



```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.3.1.1"
    ::= { timebaseAscActionEntry 1 }
```

5.6.3.2 Time Base Action Pattern Parameter

```
timebaseAscPattern OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "pattern"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The Pattern that shall be active when this
        Action is active. The value shall not exceed the value of maxPatterns,
        except for 254 (Free) and 255 (Flash). A pattern of zero indicates that
no pattern
        is being selected. A pattern = 0 relinquishes control to entity of a
        lower priority than timebase and allows that entity to control (i.e.,
        interconnect if available).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.3.1.2
    <Object Type> P"
    ::= { timebaseAscActionEntry 2 }
```

5.6.3.3 Time Base Special Function Parameter

```
timebaseAscSpecialFunction OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The Special Functions that shall be active when
this Action is active. Each bit represents a special function from 1 to 32.
        Bit 31: Special Function 32
        Bit 30: Special Function 31
        Bit 29: Special Function 30
        .... .... .... .... .....
        Bit 3: Special Function 4
        Bit 2: Special Function 3
        Bit 1: Special Function 2
        Bit 0: Special Function 1

        Bit = 0 - False/Disabled, Bit = 1 - True/Enabled

        If the ASC supports less than 32 special functions, a SET of a bit for an
unsupported special function to a value other than zero (0) shall return a
wrongValue(10) error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.3.1.4"
    ::= { timebaseAscActionEntry 4 }
}
```


5.6.3.4 Time Base Action Enabled Lane

```
timebaseAscEnabledLane OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This value is equal to the enabledLaneIndex
        (row) in the spatEnabledLanesConcurrencyTable that contains the
        enabledLaneConcurrency lanes to be active when this Action is active.
        A value of zero indicates that no lanes are to be be enabled as part
        of this action.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.3.1.4
    <Object Type> P"
    ::= { timebaseAscActionEntry 5 }
```

5.6.4 Time Base Asc Action Status

```
timebaseAscActionStatus OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the current time base
        Action Table row that will be used when the CU is in Time Base
        operation. A value of zero indicates that no time base Action is
        selected.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.4
    <Object Type> S"
    ::= { timebaseAsc 4 }
```

5.6.5 Action Plan Command

```
actionPlanControl OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to activate a configured
        action plan by referencing the Action number (timebaseAscActionNumber).
        When this action plan is in effect, the CU shall operate as if the
        action plan has been activated by the time base scheduler. A value of
        0 shall deactivate the action plan and returns to what would normally
        have been in operation if the action plan was not in effect.

        If an unsupported / invalid Action number is called, Free shall be the
        operational mode.

        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO (see
```

```
        unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.5.5
    <Object Type> C"
    DEFVAL { 0 }
    ::= { timebaseAsc 5 }
```

5.7 Preempt Parameters

```
preempt OBJECT IDENTIFIER
    ::= { asc 6 }

-- The preempt node contains objects that support preempt input functions for the
device.
```

5.7.1 Maximum Preempts

```
maxPreempts OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "preempt"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Preempts this Actuated
Controller Unit supports. This object indicates the maximum rows which shall
appear in the preemptTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.1
    <Object Type> S"
    REFERENCE "NEMA TS 2 Clause 3.7"
    ::= { preempt 1 }
```

5.7.2 Preempt Table

```
preemptTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PreemptEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit
preemption parameters. The number of rows in this table is equal to the
maxPreempts object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2"
    ::= { preempt 2 }

preemptEntry OBJECT-TYPE
    SYNTAX PreemptEntry
    MAX-ACCESS not-accessible
    STATUS current
```

```
    DESCRIPTION "<Definition> Parameters for a specific Actuated Controller Unit
preemptor.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1"
    INDEX    { preemptNumber }
    ::= { preemptTable 1}

PreemptEntry ::= SEQUENCE {
    preemptNumber              Integer32,
    preemptControl             Integer32,
    preemptLink                Integer32,
    preemptDelay               Integer32,
    preemptMinimumDuration     Integer32,
    preemptMinimumGreen        Integer32,
    preemptMinimumWalk         Integer32,
    preemptEnterPedClear       Integer32,
    preemptTrackGreen          Integer32,
    preemptDwellGreen          Integer32,
    preemptMaximumPresence     Integer32,
    preemptTrackPhase          OCTET STRING,
    preemptDwellPhase          OCTET STRING,
    preemptDwellPed            OCTET STRING,
    preemptExitPhase           OCTET STRING,
    preemptState               INTEGER,
    preemptTrackOverlap        OCTET STRING,
    preemptDwellOverlap        OCTET STRING,
    preemptCyclingPhase        OCTET STRING,
    preemptCyclingPed          OCTET STRING,
    preemptCyclingOverlap      OCTET STRING,
    preemptEnterYellowChange   Integer32,
    preemptEnterRedClear       Integer32,
    preemptTrackYellowChange   Integer32,
    preemptTrackRedClear       Integer32,
    preemptSequenceNumber      Integer32,
    preemptExitType            INTEGER }
```

5.7.2.1 Preempt Number

```
preemptNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "preempt"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The preempt number for objects in this row. The
value shall not exceed the maxPreempts object value. When all preemptControl
objects have a value where bit 2 = 0, each preemptNumber routine shall be a
```

```
higher priority and override all preemptNumber routines that have a larger
preemptNumber.

When a preemptControl object has a value where bit 2 = 1, the next higher
preemptNumber becomes of equal priority with the preemptNumber but may still be a
higher priority than larger preemptNumbers depending on bit 2 of the relavent
preemptControl objects.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.1"
    ::= { preemptEntry 1 }
```

5.7.2.2 Preempt Control Parameter

```
preemptControl OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Preempt Miscellaneous Control Parameter Mask
(Bit=0: False/Disabled, Bit=1: True/Enabled) as follows:
Bit 7: Reserved
Bit 6: All Red Entry - the CU shall enter to all red at the start of the preempt.
Bit 5: All Red Flash Exit - the CU shall enter to all red flash instead of normal
operations when the preemptMaximumPresence is exceeded. When this bit is not set
(0), the CU shall exit the preempt as specified in preemptExitType.
Bit 4: Preempt Enable - enables or disables this preemption input. Disabling
preempts should be done with extreme caution.
Bit 3: Flash Dwell - the CU shall cause the phases listed in the
preemptDwellPhase object to flash Yellow during the Dwell interval. All active
phases not listed in preemptDwellPhase shall flash Red.
The CU shall cause the overlaps listed in the preemptDwellOverlap object to flash
Yellow during the Dwell state. All active overlaps not listed in
preemptDwellOverlap shall flash Red. Preempt cycling phase programming is ignored
if this bit is set. This control is optional.
Bit 2: Preempt Override Preempt Lock- provide a means to define whether this
preempt shall NOT override the next higher numbered Preempt (preemptNumber + 1).
When this bit is set (1), this preempt shall not override the next higher
numbered preempt. Lowered numbered preempts override higher numbered preempts.
For example, 1 overrides 3, and the only way to get 3 equal to 1, is to set both
1 and 2 to NOT override the next higher numbered preempt. This parameter shall be
ignored when preemptNumber equals maxPreempts.
Bit 1: Preempt Override Flash Lock - provide a means to define whether this
preempt shall NOT override Automatic Flash. When set (1) this preempt shall NOT
override Automatic Flash.
Bit 0: Non-Locking Memory - provide a means to enable an operation which does not
require detector memory. When set (1) a preempt sequence shall not occur if the
preempt input terminates prior to expiration of the preemptDelay time.
```

```
    A SET of a 'reserved' bit to a value other than zero (0) shall return a
wrongValue(10) error.

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.2
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.7.2.1 and 3.7.2.2"
    DEFVAL { 0 }
    ::= { preemptEntry 2 }
```

5.7.2.3 Preempt Link Parameter

```
preemptLink OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "preempt"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object provides a means to define a higher
priority preempt to be combined (linked) with this preempt. At the end of
preemptDwellGreen, the linked preempt shall receive an automatic call that shall
be maintained as long as the demand for this preempt is active. Any value that is
not a higher priority preempt or a valid preempt shall be ignored. The value
shall not exceed the maxPreempts object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.3
    <Object Type> P"
    DEFVAL { 0 }
    ::= { preemptEntry 3 }
```

5.7.2.4 Preempt Delay Parameter

```
preemptDelay OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Preempt Delay Time in seconds (0-999 sec). This
value determines the time the preempt input shall be active prior to initiating
any preempt sequence. A non-locking preempt input which is removed prior to the
completion of this time shall not cause a preempt sequence to occur.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.4
    <Object Type> P"
    DEFVAL { 0 }
    ::= { preemptEntry 4 }
```

5.7.2.5 Preempt Duration Parameter

```
preemptMinimumDuration OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
```

```
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Preempt Minimum Duration Time in seconds. This
value determines the minimum time during which the preempt is active. Duration
begins timing at the end of Preempt Delay (if non zero) and will prevent an exit
from the Dwell interval until this time has elapsed.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.5
    <Object Type> P"
    DEFVAL { 0 }
    ::= { preemptEntry 5 }
```

5.7.2.6 Preempt Minimum Green Parameter

```
preemptMinimumGreen OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Preempt Minimum Green Time in seconds (0-255 sec).
A preempt initiated transition shall not cause the termination of an existing
Green indication for lesser of the phase's Minimum Green time or this period.
CAUTION - if this value is zero, the currently active phases' GREEN indications
are terminated immediately.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.6
    <Object Type> P"
    DEFVAL { 255 }
    ::= { preemptEntry 6 }
```

5.7.2.7 Preempt Minimum Walk Parameter

```
preemptMinimumWalk OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Preempt Minimum Walk Time in seconds (0-255 sec). A
preempt initiated transition shall not cause the termination of an existing WALK
prior to its display for the lesser of the phase's Walk time or this period.
CAUTION - if this value is zero, currently active phases' WALK indications are
terminated immediately.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.7
    <Object Type> P"
    DEFVAL { 255 }
    ::= { preemptEntry 7 }
```

5.7.2.8 Preempt Enter Pedestrian Clear Parameter

```
preemptEnterPedClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Enter Ped Clearance Time in seconds (0-255 sec).
This parameter controls the ped clear timing for a normal Walk signal terminated
by a preempt initiated transition. A preempt initiated transition shall not cause
the termination of a Pedestrian Clearance (Flashing DON'T WALK) prior to its
display for the lesser of the phase's Pedestrian Clearance time or this period.
CAUTION - currently active phases' Pedestrian Clearance indications are
terminated immediately or skipped.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.8
    <Object Type> P"
    DEFVAL { 255 }
    ::= { preemptEntry 8 }
```

5.7.2.9 Preempt Track Green Parameter

```
preemptTrackGreen OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Track Clear Green Time in seconds (0-255 sec). This
parameter controls the green timing for the track clearance movement. Track Clear
phase(s) are enabled in the preemptTrackPhase object. If this value is zero, the
track clearance movement is omitted, regardless of preemptTrackPhase programming.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.9
    <Object Type> P"
    DEFVAL { 0 }
    ::= { preemptEntry 9 }
```

5.7.2.10 Preempt Minimum Dwell Parameter

```
preemptDwellGreen OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Minimum Dwell interval in seconds (1-255 sec). This
parameter controls the minimum timing for the dwell interval. Phase(s) active
during the Dwell interval are enabled in preemptDwellPhase and
preemptCyclingPhase objects. Dwell phases may dwell for this amount of time
before any phases cycle during the preempt. The Dwell interval shall not
```

```
terminate prior to the completion of preemptMinimumDuration, preemptDwellGreen
(this object), and the call is no longer present.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.10
    <Object Type> P"
    DEFVAL { 0 }
    ::= { preemptEntry 10 }
```

5.7.2.11 Preempt Maximum Presence Parameter

```
preemptMaximumPresence OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Preempt Maximum Presence time in seconds (0-65535
sec). This value determines the maximum time which a preempt call may remain
active and be considered valid. When the preempt call has been active for this
time period, the CU shall return to normal operation. This preempt call shall be
considered invalid until such time as a change in state occurs (no longer
active). When set to zero the preempt maximum presence time is disabled.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.11
    <Object Type> P"
    DEFVAL { 0 }
    ::= { preemptEntry 11 }
```

5.7.2.12 Preempt Track Phase Parameter

```
preemptTrackPhase OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains a
phaseNumber (binary value) that shall be active during the Preempt Track
Clearance intervals. The values of phaseNumber used here shall not exceed
maxPhases or violate the Consistency Checks defined in Section 4.3.2.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.12
    <Object Type> P2"
    DEFVAL { "" }
    ::= { preemptEntry 12 }
```

5.7.2.13 Preempt Dwell Phase Parameter

```
preemptDwellPhase OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains a
phaseNumber (binary value) that specifies the phase(s) to be served in the
Preempt Dwell interval. The phase(s) defined in preemptCyclingPhase shall occur
after those defined herein. The values of phaseNumber used here shall not exceed
maxPhases or violate the Consistency Checks defined in Section 4.3.2.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.13
    <Object Type> P2"
    DEFVAL { "" }
    ::= { preemptEntry 13 }
```

5.7.2.14 Preempt Dwell Ped Parameter

```
preemptDwellPed OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains a
phaseNumber (binary value) that specifies the pedestrian movement(s) to be served
in the Preempt Dwell interval. The peds defined in preemptCyclingPed shall occur
after those defined herein. The values of phaseNumber used here shall not exceed
maxPhases or violate the Consistency Checks defined in Section 4.3.2.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.14
    <Object Type> P2"
    DEFVAL { "" }
    ::= { preemptEntry 14 }
```

5.7.2.15 Preempt Exit Phase Parameter

```
preemptExitPhase OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains a
phaseNumber (binary value) that shall be active following Preempt. The values of
phaseNumber used here shall not exceed maxPhases or violate the Consistency
Checks defined in Section 4.3.2.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.15
    <Object Type> P2"
    DEFVAL { "" }
    ::= { preemptEntry 15 }
```

5.7.2.16 Preempt State

```
preemptState OBJECT-TYPE
    SYNTAX INTEGER { other (1),
```

```
    notActive (2),
    notActiveWithCall (3),
    entryStarted (4),
    trackService (5),
    dwell (6),
    linkActive (7),
    exitStarted (8),
    maxPresence (9),
    advancedPreempt (10) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Preempt State provides status on which state the
associated preempt is in. The states are as follows:
other: preempt service is not specified in this standard.
notActive: preempt input is not active, this preempt is not active.
notActiveWithCall: preempt input is active, preempt service has not started.
entryStarted: preempt service is timing the entry intervals.
trackService: preempt service is timing the track intervals.
dwell: preempt service is timing the dwell intervals.
linkActive: preempt service is performing linked operation.
exitStarted: preempt service is timing the exit intervals.
maxPresence: preempt input has exceeded maxPresence time
advancedPreempt: preempt service is timing the advanced preemption time.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.16
    <Object Type> S"
    ::= { preemptEntry 16}
```

5.7.2.17 Preempt Track Overlap Parameter

```
preemptTrackOverlap OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains an
overlapNumber (binary value) that shall be active during the Preempt Track Clear
intervals. The values of overlapNumber used here shall not exceed maxOverlaps or
violate the consistency checks defined in Section 4.3.2. Any overlap not
specified will not be active during the preempt even if its parent phases are
active.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.17
    <Object Type> P2"
    DEFVAL { "" }
    ::= { preemptEntry 17 }
```

5.7.2.18 Preempt Dwell Overlap Parameter

```
preemptDwellOverlap OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains an
overlapNumber (binary value) that is allowed during the Preempt Dwell interval.
The values of overlapNumber used here shall not exceed maxOverlaps or violate the
consistency checks defined in Section 4.3.2. Any overlap not specified will not
be active during the preempt even if its parent phases are active.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.18
    <Object Type> P2"
    DEFVAL { "" }
    ::= { preemptEntry 18 }
```

5.7.2.19 Preempt Cycling Phase Parameter

```
preemptCyclingPhase OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains a
phaseNumber (binary value) that is allowed to cycle during the Preempt Dwell
interval after preemptDwellTime has lapsed. The values of phaseNumber used here
shall not exceed maxPhases or violate the Consistency Checks defined in Section
4.3.2.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.19
    <Object Type> P2"
    DEFVAL { "" }
    ::= { preemptEntry 19 }
```

5.7.2.20 Preempt Cycling Ped Parameter

```
preemptCyclingPed OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains a
phaseNumber (binary value) indicating a pedestrian movement that is allowed to
cycle during the Preempt Dwell interval after preemptDwellTime has lapsed. The
values of phaseNumber used here shall not exceed maxPhases or violate the
consistency checks defined in Section 4.3.2.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.20
    <Object Type> P2"
    DEFVAL { "" }
```

```
    ::= { preemptEntry 20 }
```

5.7.2.21 Preempt Cycling Overlap Parameter

```
preemptCyclingOverlap OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains a
overlapNumber (binary value) that is allowed to cycle during the Preempt Dwell
interval. The values of overlapNumber used here shall not exceed maxOverlaps or
violate the consistency checks defined in Section 4.3.2. Any overlap not
specified will not be active during the preempt even if its parent phases are
active.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.21
    <Object Type> P2"
    DEFVAL { "" }
    ::= { preemptEntry 21 }
```

5.7.2.22 Preempt Enter Yellow Change Parameter

```
preemptEnterYellowChange OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Enter Yellow Change in tenth seconds (0-25.5  sec).
This parameter controls the yellow change timing for a normal Yellow Change
signal terminated by a preempt initiated transition. A preempt initiated
transition shall not cause the termination of a Yellow Change prior to its
display for the lesser of the phase's Yellow Change time or this period. CAUTION
- if this value is zero, phase Yellow Change is terminated immediately.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.22
    <Object Type> P"
    DEFVAL { 255 }
    ::= { preemptEntry 22 }
```

5.7.2.23 Preempt Enter Red Clearance Parameter

```
preemptEnterRedClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Enter Red Clear in tenth seconds (0-25.5 sec). This
parameter controls the red clearance timing for a normal Red Clear signal
```

```
terminated by a preempt initiated transition. A preempt initiated transition
shall not cause the termination of a Red Clear prior to its display for the
lesser of the phase's Red Clear time or this period. CAUTION - if this value is
zero, phase Red Clear is terminated immediately.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.23
    <Object Type> P"
    DEFVAL { 255 }
    ::= { preemptEntry 23 }
```

5.7.2.24 Preempt Track Yellow Change Parameter

```
preemptTrackYellowChange OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Track Clear Yellow Change time in tenth seconds (0-
25.5 sec). The lesser of the phase's Yellow Change time or this parameter
controls the yellow timing for the track clearance movement. Track clear phases
are specified in the preemptTrackPhase object. The track clearance movement is
enabled in the preemptTrackGreen object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.24
    <Object Type> P"
    DEFVAL { 255 }
    ::= { preemptEntry 24 }
```

5.7.2.25 Preempt Track Red Clearance Parameter

```
preemptTrackRedClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Track Clear Red Clearance time in tenth seconds (0-
25.5 sec). The lesser of the phase's Red Clearance time or this parameter
controls the Red Clearance timing for the track clearance movement. Track clear
phases are specified in the preemptTrackPhase object. The track clearance
movement is enabled in the preemptTrackGreen object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.25
    <Object Type> P"
    DEFVAL { 255 }
    ::= { preemptEntry 25 }
```

5.7.2.26 Preempt Sequence Number

```
preemptSequenceNumber OBJECT-TYPE
```

```
    SYNTAX Integer32 (0..255)
    UNITS "sequence"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to configure the sequence to
run during the preempt's dwell duration. The selected sequence defines the order
in which the phases defined the preemptCyclingPhase and preemptCyclingPed objects
are allowed to cycle. Phases in the selected sequence that are not defined in
preemptCyclingPhase nor preemptCyclingPed do not cycle.

This value is a sequenceNumber used to select a sequence from the sequenceTable
and shall not exceed the maxSequences object value. A value of zero indicates
that the sequence to run is the one that was in effect when the preempt was
activated.

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.26
    <Object Type> P"
    DEFVAL { 0 }
    ::= { preemptEntry 26 }
```

5.7.2.27 Preempt Exit Type

```
preemptExitType OBJECT-TYPE
    SYNTAX INTEGER { exitPhases (1),
    queueDelayRecovery (2),
    shortService (3),
    exitCoord (4)}
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the exit strategy (type) to be
used following the end of the preempt. The exit types are as follows:
exitPhases: the CU immediately enters the exit phases to be active as configured
queueDelayRecovery: the CU immediately enters the phase with the highest demand
or longest wait time
shortService: the CU immediately enters the first short service phase. The first
short service phase is a phase where only the preempt minimum green time was
serviced during the advanced preemption time or the right-of-way transfer time
exitCoord: the CU immediately returns to the place in the coordinated cycle where
the ASC would have been if there was no preempt
    <Object Type> P
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.2.1.27"
    ::= { preemptEntry 27 }
```

5.7.3 Preempt Control Table

```
preemptControlTable OBJECT-TYPE
```

```
    SYNTAX SEQUENCE OF PreemptControlEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table contains the control objects that allow
the preempts to be activated remotely. There shall be one control object for each
preempt input supported by the device. The number of rows in this table shall be
equal to maxPreempts.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.3"
    ::= { preempt 3 }

preemptControlEntry OBJECT-TYPE
    SYNTAX PreemptControlEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Control objects for each preempt input. These
objects allow the system to activate preempt functions remotely.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.3.1"
    INDEX   { preemptControlNumber }
    ::= { preemptControlTable 1 }

PreemptControlEntry ::= SEQUENCE {
    preemptControlNumber Integer32,
    preemptControlState  Integer32 }
```

5.7.3.1 Preempt Control Number

```
preemptControlNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "preempt"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object shall indicate the preempt input number
controlled by the associated preemptControlState object in this row.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.3.1.1"
    ::= { preemptControlEntry  1 }
```

5.7.3.2 Preempt Control State

```
preemptControlState OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object when set to ON (one) shall cause the
associated preempt actions to occur unless the actions have already been started
by the physical preempt input. The preempt shall remain active as long as this
```

```
object is ON or the physical preempt input is ON. This object when set to OFF
(zero) shall cause the physical preempt input to control the associated preempt
actions. The device shall reset this object to ZERO when in BACKUP Mode. A write
to this object shall reset the Backup timer to ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.3.1.2
    <Object Type> C"
    ::= { preemptControlEntry  2 }
```

5.7.4 Preempt Status

```
preemptStatus OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object defines the preempt number that is
currently being serviced in the device. A value of zero indicate no preempt is
being served.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.4"
    ::= { preempt 4 }
```

5.7.5 Maximum Preempt Groups

```
maxPreemptGroups OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Preempt Groups (8 Preempt per
group) this CU supports. This value is equal to TRUNCATE [(maxPreempts + 7) / 8].
This object indicates the maximum rows which shall appear in the
preemptStatusGroupTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.5
    <Object Type> S"
    ::= { preempt 5 }
```

5.7.6 Preempt Status Table

```
preemptStatusGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PreemptStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing the CU preempt input signal
status in groups of eight Preempts. The number of rows in this table is equal to
the maxPreemptGroups object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.6"
```

```
    ::= { preempt 6 }

preemptStatusGroupEntry OBJECT-TYPE
    SYNTAX PreemptStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Preempt input signal status for preempt inputs in
groups of eight.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.6.1"
    INDEX { preemptStatusGroupNumber }
    ::= { preemptStatusGroupTable 1 }

PreemptStatusGroupEntry ::= SEQUENCE {
    preemptStatusGroupNumber Integer32,
    preemptStatusGroup       Integer32  }
```

5.7.6.1 Preempt Status Group Number

```
preemptStatusGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The Preempt StatusGroup number for objects in this
row. This value shall not exceed the maxPreemptGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.6.1.1
    <Object Type> S"
    ::= { preemptStatusGroupEntry 1 }
```

5.7.6.2 Preempt Status Group

```
preemptStatusGroup OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Preempt Input Signal Status Mask. When a bit = 1, a
preempt input signal is detected or the preemptControlState is ON for that input,
and when a bit = 0, no preempt input signal is detected and the
preemptControlState is OFF for that input.
Bit 7: Preempt # = (preemptStatusGroupNumber * 8)
Bit 6: Preempt # = (preemptStatusGroupNumber * 8) - 1
Bit 5: Preempt # = (preemptStatusGroupNumber * 8) - 2
Bit 4: Preempt # = (preemptStatusGroupNumber * 8) - 3
Bit 3: Preempt # = (preemptStatusGroupNumber * 8) - 4
Bit 2: Preempt # = (preemptStatusGroupNumber * 8) - 5
Bit 1: Preempt # = (preemptStatusGroupNumber * 8) - 6
```

```
Bit 0: Preempt # = (preemptStatusGroupNumber * 8) - 7

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.6.1.2
    <Object Type> S"
    ::= { preemptStatusGroupEntry 2 }
```

5.7.7 Preempt Queue Delay Table

```
preemptQueueDelayTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PreemptQueueDelayEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing CU detector parameters for the
queue delay recovery exit strategy. The number of rows in this table will not
exceed the maxVehicleDetectors object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.7"
    ::= { preempt 7 }

preemptQueueDelayEntry OBJECT-TYPE
    SYNTAX PreemptQueueDelayEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific CU preempt input if the
queue delay recovery exit strategy is used.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.7.1"
    INDEX { preemptNumber, vehicleDetectorNumber }
    ::= { preemptQueueDelayTable 1}

PreemptQueueDelayEntry ::= SEQUENCE {
    preemptDetectorWeight Integer32 }
```

5.7.7.1 Preempt Detector Weight

```
preemptDetectorWeight OBJECT-TYPE
    SYNTAX Integer32 (0..1000)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the relative weight for the
associated detector when using the detector data to determine the queue delay
recovery exit strategy from a preempt input. The association between the
vehicleDetectorNumber and a phase is identified by the vehicleDetectorCallPhase.
A higher number for this object indicates a larger weight for the demand and wait
time for that phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.7.1.1
    <Object Type> P"
```

```
    ::= { preemptQueueDelayEntry 1 }
```

5.7.8 Maximum Preemption Gates

```
maxPreemptGates OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "gate"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The maximum number of preempt gates this CU
supports. This object indicates the maximum rows which shall appear in the
preemptGateTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.8
    <Object Type> S"
    ::= { preempt 8 }
```

5.7.9 Preempt Gate Table

```
preemptGateTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PreemptGateEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing the status of the gates that may
be lowered during a preempt sequence.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.9"
    ::= { preempt 9 }

preemptGateEntry OBJECT-TYPE
    SYNTAX PreemptGateEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Gate status for preempt sequences.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.9.1"
    INDEX { preemptGateNumber }
    ::= { preemptGateTable 1 }

PreemptGateEntry ::= SEQUENCE {
    preemptGateNumber      INTEGER,
    preemptGateStatus      INTEGER,
    preemptGateDescription SnmpAdminString }
```

5.7.9.1 Preempt Gate Number

```
preemptGateNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
```

```
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The Preempt Gate number for objects in this row.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.9.1.1"
    ::= { preemptGateEntry 1 }
```

5.7.9.2 Preempt Gate Status

```
preemptGateStatus OBJECT-TYPE
    SYNTAX INTEGER { other(1),
                     unknown(2),
                     up(3),
                     down(4) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The status of a gate that may be lowered during
        a preempt sequence.
        unknown: The status of unknown or no gate is present
        other: The gate is neither in the locked up or locked down position
        up: The gate is in an up position
        down: The gate is in a down position
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.9.1.2
    <Object Type> S"
    DEFVAL { 2 }
    ::= { preemptGateEntry 2 }
```

5.7.9.1 Preempt Gate Description

```
preemptGateDescription OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A textual string indicate a user description for
the gate (e.g. location).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.6.9.1.3
    <Object Type> P"
    ::= { preemptGateEntry 3 }
```

5.8 Ring Parameters

```
ring  OBJECT IDENTIFIER
    ::= { asc  7 }

-- The ring node contains objects that support ring configuration, status and
control functions in the device.
```

5.8.1 Maximum Rings

```
maxRings OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "ring"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The value of this object shall specify the maximum
number of rings this device supports.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.1
    <Object Type> S"
    ::= { ring 1 }
```

5.8.2 Maximum Sequences

```
maxSequences OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "sequence"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The value of this object shall specify the maximum
number of sequence plans this device supports.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.2
    <Object Type> S"
    ::= { ring 2 }
```

5.8.3 Sequence Table

```
sequenceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SequenceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table contains all the sequence plans for the
controller. A sequence plan shall consist of one row for each ring that the CU
supports. Each row defines the phase service order for that ring.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.3"
    ::= { ring 3 }

sequenceEntry OBJECT-TYPE
    SYNTAX SequenceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Phase Sequence Parameters for an Actuated
Controller Unit.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.3.1"
    INDEX   { sequenceNumber, sequenceRingNumber }
```

```
    ::= { sequenceTable 1 }

SequenceEntry ::= SEQUENCE {
    sequenceNumber     Integer32,
    sequenceRingNumber Integer32,
    sequenceData       OCTET STRING }
```

5.8.3.1 Sequence Number

```
sequenceNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "sequence"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This number identifies a sequence plan. Each row of
the table contains the phase sequence for a ring. A sequence plan shall consist
of one row for each ring that defines the phase sequences for that ring.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.3.1.1"
    ::= { sequenceEntry 1 }
```

5.8.3.2 Sequence Ring Number

```
sequenceRingNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "ring"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This number identifies the ring number this phase
sequence applies to.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.3.1.2"
    ::= { sequenceEntry 2 }
```

5.8.3.3 Sequence Data

```
sequenceData OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet is a Phase Number (binary value) within
the associated ring number. The phase number value shall not exceed the maxPhases
object value. The order of phase numbers determines the phase sequence for the
ring. The phase numbers shall not be ordered in a manner that would violate the
Consistency Checks defined in Section 4.3.2. Additionally, this object does not
specify where barriers are present in the ring - the presences of barriers is
determined by phaseConcurrency.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.3.1.3
    <Object Type> P2"
```

```
    ::= { sequenceEntry 3 }
```

5.8.4 Maximum Ring Control Groups

```
maxRingControlGroups OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The maximum number of Ring Control Groups (8 rings
per group) this Actuated Controller Unit supports. This value is equal to
TRUNCATE[(maxRings + 7) / 8]. This object indicates the maximum rows which shall
appear in the ringControlGroupTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.4
    <Object Type> S"
    ::= { ring 4 }
```

5.8.5 Ring Control Group Table

```
ringControlGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF RingControlGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit Ring
Control in groups of eight rings. The number of rows in this table is equal to
the maxRingControlGroups object.
    <TableType> static
    <<Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5"
    ::= { ring 5 }

ringControlGroupEntry OBJECT-TYPE
    SYNTAX RingControlGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Ring Control for eight Actuated Controller Unit
rings.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1"
    INDEX   { ringControlGroupNumber }
    ::= { ringControlGroupTable 1 }

RingControlGroupEntry ::= SEQUENCE {
    ringControlGroupNumber      Integer32,
    ringControlGroupStopTime    Integer32,
    ringControlGroupForceOff    Integer32,
    ringControlGroupMax2        Integer32,
    ringControlGroupMaxInhibit  Integer32,
```

```
    ringControlGroupPedRecycle   Integer32,
    ringControlGroupRedRest      Integer32,
    ringControlGroupOmitRedClear Integer32,
    ringControlGroupMax3         Integer32 }
```

5.8.5.1 Ring Control Group Number

```
ringControlGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The Ring Control Group number for objects in this
row. This value shall not exceed the maxRingControlGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.1"
    ::= { ringControlGroupEntry 1 }
```

5.8.5.2 Ring Stop Time Control

```
ringControlGroupStopTime OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
stop timing in the device. The device shall activate/deactivate the System Stop
Time control for a ring according to the respective bit value as follows:
bit = 0 - deactivate the ring control
bit = 1 - activate the ring control
Bit 7: Ring # = (ringControlGroupNumber * 8)
Bit 6: Ring # = (ringControlGroupNumber * 8) - 1
Bit 5: Ring # = (ringControlGroupNumber * 8) - 2
Bit 4: Ring # = (ringControlGroupNumber * 8) - 3
Bit 3: Ring # = (ringControlGroupNumber * 8) - 4
Bit 2: Ring # = (ringControlGroupNumber * 8) - 5
Bit 1: Ring # = (ringControlGroupNumber * 8) - 6
Bit 0: Ring # = (ringControlGroupNumber * 8) - 7
The device shall reset this object to ZERO when in BACKUP Mode. A write to this
object shall reset the Backup timer to ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.2
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.4.1.6"
    ::= { ringControlGroupEntry 2 }
```

5.8.5.3 Ring Force Off Control

```
ringControlGroupForceOff OBJECT-TYPE
```

```
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
terminate phases via a force off command in the device. The device shall
activate/deactivate the System Force Off control for a ring according to the
respective bit value as follows:
bit = 0 - deactivate the ring control
bit = 1 - activate the ring control
Bit 7: Ring # = (ringControlGroupNumber * 8)
Bit 6: Ring # = (ringControlGroupNumber * 8) - 1
Bit 5: Ring # = (ringControlGroupNumber * 8) - 2
Bit 4: Ring # = (ringControlGroupNumber * 8) - 3
Bit 3: Ring # = (ringControlGroupNumber * 8) - 4
Bit 2: Ring # = (ringControlGroupNumber * 8) - 5
Bit 1: Ring # = (ringControlGroupNumber * 8) - 6
Bit 0: Ring # = (ringControlGroupNumber * 8) - 7
The device shall reset this object to ZERO when in BACKUP Mode. A write to this
object shall reset the Backup timer to ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.3
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.4.1.1"
    ::= { ringControlGroupEntry 3 }
```

5.8.5.4 Ring Max 2 Control

```
ringControlGroupMax2 OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
request Maximum 2 timings in the device. The device shall activate/deactivate the
System Maximum 2 control for a ring according to the respective bit value as
follows:
bit = 0 - deactivate the ring control
bit = 1 - activate the ring control
Bit 7: Ring # = (ringControlGroupNumber * 8)
Bit 6: Ring # = (ringControlGroupNumber * 8) - 1
Bit 5: Ring # = (ringControlGroupNumber * 8) - 2
Bit 4: Ring # = (ringControlGroupNumber * 8) - 3
Bit 3: Ring # = (ringControlGroupNumber * 8) - 4
Bit 2: Ring # = (ringControlGroupNumber * 8) - 5
Bit 1: Ring # = (ringControlGroupNumber * 8) - 6
Bit 0: Ring # = (ringControlGroupNumber * 8) - 7
```

```
The device shall reset this object to ZERO when in BACKUP Mode. A write to this
object shall reset the Backup timer to ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.4
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.4.1.7"
    ::= { ringControlGroupEntry 4 }
```

5.8.5.5 Ring Max Inhibit Control

```
ringControlGroupMaxInhibit OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "<Definition> This object is used to allow a remote entity to request
        internal maximum timings be inhibited in the device. The device shall
        activate/deactivate the System Max Inhibit control for a ring
        according to the respective bit value as follows:
            bit = 0 - deactivate the ring control
            bit = 1 - activate the ring control

        Bit 7: Ring # = (ringControlGroupNumber * 8)
        Bit 6: Ring # = (ringControlGroupNumber * 8) - 1
        Bit 5: Ring # = (ringControlGroupNumber * 8) - 2
        Bit 4: Ring # = (ringControlGroupNumber * 8) - 3
        Bit 3: Ring # = (ringControlGroupNumber * 8) - 4
        Bit 2: Ring # = (ringControlGroupNumber * 8) - 5
        Bit 1: Ring # = (ringControlGroupNumber * 8) - 6
        Bit 0: Ring # = (ringControlGroupNumber * 8) - 7

        The device shall reset this object to ZERO when in BACKUP Mode. A
        write to this object shall reset the Backup timer to ZERO (see
        unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.5
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.4.1.3"
    ::= { ringControlGroupEntry 5 }
```

5.8.5.6 Ring Ped Recycle Control

```
ringControlGroupPedRecycle OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
request a pedestrian recycle in the device. The device shall activate/deactivate
```

```
the System Ped Recycle control for a ring according to the respective bit value
as follows:
bit = 0 - deactivate the ring control
bit = 1 - activate the ring control
Bit 7: Ring # = (ringControlGroupNumber * 8)
Bit 6: Ring # = (ringControlGroupNumber * 8) - 1
Bit 5: Ring # = (ringControlGroupNumber * 8) - 2
Bit 4: Ring # = (ringControlGroupNumber * 8) - 3
Bit 3: Ring # = (ringControlGroupNumber * 8) - 4
Bit 2: Ring # = (ringControlGroupNumber * 8) - 5
Bit 1: Ring # = (ringControlGroupNumber * 8) - 6
Bit 0: Ring # = (ringControlGroupNumber * 8) - 7
The device shall reset this object to ZERO when in BACKUP Mode. A write to this
object shall reset the Backup timer to ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.6
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.4.1.5"
    ::= { ringControlGroupEntry 6 }
```

5.8.5.7 Ring Red Rest Control

```
ringControlGroupRedRest OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
request red rest in the device. The device shall activate/deactivate the System
Red Rest control for a ring according to the respective bit value as follows:
bit = 0 - deactivate the ring control
bit = 1 - activate the ring control
Bit 7: Ring # = (ringControlGroupNumber * 8)
Bit 6: Ring # = (ringControlGroupNumber * 8) - 1
Bit 5: Ring # = (ringControlGroupNumber * 8) - 2
Bit 4: Ring # = (ringControlGroupNumber * 8) - 3
Bit 3: Ring # = (ringControlGroupNumber * 8) - 4
Bit 2: Ring # = (ringControlGroupNumber * 8) - 5
Bit 1: Ring # = (ringControlGroupNumber * 8) - 6
Bit 0: Ring # = (ringControlGroupNumber * 8) - 7
The device shall reset this object to ZERO when in BACKUP Mode. A write to this
object shall reset the Backup timer to ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.7
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.4.1.2"
    ::= { ringControlGroupEntry 7 }
```


5.8.5.8 Ring Omit Red Control

```
ringControlGroupOmitRedClear OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
omit red clearances in the device. The device shall activate/deactivate the
System Omit Red Clear control for a ring according to the respective bit value as
follows:
bit = 0 - deactivate the ring control
bit = 1 - activate the ring control
Bit 7: Ring # = (ringControlGroupNumber * 8)
Bit 6: Ring # = (ringControlGroupNumber * 8) - 1
Bit 5: Ring # = (ringControlGroupNumber * 8) - 2
Bit 4: Ring # = (ringControlGroupNumber * 8) - 3
Bit 3: Ring # = (ringControlGroupNumber * 8) - 4
Bit 2: Ring # = (ringControlGroupNumber * 8) - 5
Bit 1: Ring # = (ringControlGroupNumber * 8) - 6
Bit 0: Ring # = (ringControlGroupNumber * 8) - 7
The device shall reset this object to ZERO when in BACKUP Mode. A write to this
object shall reset the Backup timer to ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.8
    <Object Type> C"
    REFERENCE "NEMA TS 2 Clause 3.5.4.1.4"
    ::= { ringControlGroupEntry 8 }
```

5.8.5.9 Ring Max 3 Control

```
ringControlGroupMax3 OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to allow a remote entity to
request Maximum 3 timings in the device. The device shall activate/deactivate the
System Maximum 3 control for a ring according to the respective bit value as
follows:
bit = 0 - deactivate the ring control
bit = 1 - activate the ring control
Bit 7: Ring # = (ringControlGroupNumber * 8)
Bit 6: Ring # = (ringControlGroupNumber * 8) - 1
Bit 5: Ring # = (ringControlGroupNumber * 8) - 2
Bit 4: Ring # = (ringControlGroupNumber * 8) - 3
Bit 3: Ring # = (ringControlGroupNumber * 8) - 4
Bit 2: Ring # = (ringControlGroupNumber * 8) - 5
Bit 1: Ring # = (ringControlGroupNumber * 8) - 6
```

```
Bit 0: Ring # = (ringControlGroupNumber * 8) - 7
The device shall reset this object to ZERO when in BACKUP Mode.
A write to this object shall reset the Backup timer to ZERO (see unitBackupTime).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.5.1.9
    <Object Type> C"
    ::= { ringControlGroupEntry 9 }
```

5.8.6 Ring Status Table

```
ringStatusTable OBJECT-TYPE
    SYNTAX SEQUENCE OF RingStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit Ring
Status. The number of rows in this table is equal to the maxRings object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.6"
    ::= { ring 6 }

ringStatusEntry OBJECT-TYPE
    SYNTAX RingStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Ring Status for an Actuated Controller Unit ring.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.6.1"
    INDEX   { sequenceRingNumber }
    ::= { ringStatusTable 1 }

RingStatusEntry ::= SEQUENCE {
    ringStatus          INTEGER,
    ringOnPhase         Integer32,
    ringOnPhaseDuration Unsigned32 }
```

5.8.6.1 Ring Status

```
ringStatus OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The phase status for the phase indicated in the
ringActivePhase object.
Bit 15: Reserved
Bit 14: Reserved
Bit 13: Reserved
Bit 12: Reserved
```

```
Bit 11: Don't Walk - When bit = 1, the active phase in the ring is currently
displaying a steady DON'T WALK indication.
Bit 10: Flashing Don't Walk - When bit = 1, the active phase in the ring is
timing a pedestrian clearance interval and displaying a Flashing DON'T WALK
indication.
Bit 9: Rest-in-Walk - When bit = 1, the active phase in the ring is currently
resting in Walk when there is no serviceable conflicting call or because Max
Vehicle Recall is enabled.
Bit 8: Delayed Pedestrian Waiting - When bit = 1 the active phase in the ring has
a delayed pedestrian interval and is currently displaying a vehicle GREEN
indication before displaying WALK indication.
Bit 7: Walk - When bit = 1, the active phase in the ring is displaying a WALK
indication.
Bit 6: Advanced Walk - When bit = 1, the active phase in the ring is timing a
Leading Pedestrian Interval.
Bit 5: Force Off - When bit = 1, the active phase in the ring was terminated by
Force Off
Bit 4: Max Out - When bit = 1, the active phase in the ring was terminated by Max
Out
Bit 3: Gap Out - When bit = 1, the active phase in the ring was terminated by Gap
Out
Bit 2: Coded Status Bit C
Bit 1: Coded Status Bit B
Bit 0: Coded Status Bit A
+======+=====+=====+=====+===============+
| Code |    Bit States   | State         |
|  ##  |  A  |  B  |  C  | Names         |
+======+=====+=====+=====+===============+
|   0  |  0  |  0  |  0  | Min Green     |
|   1  |  1  |  0  |  0  | Extension     |
|   2  |  0  |  1  |  0  | Maximum |
|   3  |  1  |  1  |  0  | Green Rest    |
|   4  |  0  |  0  |  1  | Yellow Change |
|   5  |  1  |  0  |  1  | Red Clearance |
|   6  |  0  |  1  |  1  | Red Rest      |
|   7  |  1  |  1  |  1  | Undefined     |
+======+=====+=====+=====+===============+
NEMA TS 2 Clause 3.5.4.2 provides further definition of Coded Status Bits.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.6.1.1
    <Object Type> S"
    ::= { ringStatusEntry 1 }
```

5.8.6.2 Ring On Phase

```
ringOnPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
```

```
    UNITS "phase"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The phaseNumber of the phase in this ring that is
currently on in a green, yellow, or red clearance interval. A value of zero
indicates no phase is on for this ring. This value shall not exceed the value of
maxPhases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.6.1.2
    <Object Type> S"
    ::= { ringStatusEntry 2 }
```

5.8.6.3 Ring On Phase Duration

```
ringOnPhaseDuration OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    UNITS "decisecond"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This value is the time, in tenth seconds, that the
phase in ringOnPhase has been on since it began its vehicle GREEN (or WALK for
leading pedestrian intervals) indication. This value shall reset to zero when the
phase in ringOnPhase completes its red clearance interval.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.7.6.1.3
    <Object Type> S"
    ::= { ringStatusEntry 3 }
```

5.9 Channel Parameters

```
channel OBJECT IDENTIFIER
    ::= { asc 8 }

--This defines a node for supporting channel objects.
```

5.9.1 Maximum Channels

```
maxChannels OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS   "channel"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Channels this Actuated
Controller Unit supports. This object indicates the maximum rows which shall
appear in the channel table.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.1
    <Object Type> S"
    ::= { channel 1 }
```

5.9.2 Channel Table

```
channelTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ChannelEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit channel
parameters. The number of rows in this table is equal to the maxChannels object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.2"
    ::= { channel 2 }

channelEntry OBJECT-TYPE
    SYNTAX ChannelEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Actuated Controller Unit
channel.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.2.1"
    INDEX   { channelNumber }
    ::= { channelTable 1 }

ChannelEntry ::= SEQUENCE {
    channelNumber          Integer32,
    channelControlSource   Integer32,
    channelControlType     INTEGER,
    channelFlash           Integer32    }
```

5.9.2.1 Channel Number

```
channelNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS   "channel"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The channel number for objects in this row. This
value shall not exceed the maxChannels object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.2.1.1"
    ::= { channelEntry 1 }
```

5.9.2.2 Channel Control Source Parameter

```
channelControlSource OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
```



```
    DESCRIPTION "<Definition> This object defines the channel control source
(which Phase or Overlap). The value shall not exceed maxPhases or maxOverlaps as
determined by channelControlType object:
Value 00 = No Control (Not in Use)
Value 01 = Phase 01 or Overlap A
Value 02 = Phase 02 or Overlap B
||
Value 15 = Phase 15 or Overlap O
Value 16 = Phase 16 or Overlap P
etc.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.2.1.2
    <Object Type> P2"
    ::= { channelEntry 2 }
```

5.9.2.3 Channel Control Type Parameter

```
channelControlType OBJECT-TYPE
    SYNTAX INTEGER { other (1),
        phaseVehicle (2),
        phasePedestrian (3),
        overlap (4),
        pedOverlap (5),
        queueJump (6) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the channel control type
(Vehicle Phase, Pedestrian Phase, or Overlap):
other: The channel controls an other type of display.
phaseVehicle: The channel controls a vehicle phase display. Also valid for
bicycle phases and transit phases.
phasePedestrian: The channel controls a pedestrian phase display.
overlap: The channel controls an overlap display, which might include flashing
yellow arrows, flashing red arrows, vehicle overlaps, bicycle overlaps and
transit overlaps.
pedOverlap: The channel controls an overlap for pedestrian display.
queueJump: The channel controls a queue jump display typically used for transit
priority.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.2.1.3
    <Object Type> P2"
    ::= { channelEntry 3 }
```

5.9.2.4 Channel Flash Parameter

```
channelFlash OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION "<Definition> This object defines the channel state during
Automatic Flash.
Bit 7: Reserved
Bit 6: Reserved
Bit 5: Reserved
Bit 4: Flash Alternate Second
Bit=0: Flash Alternate First (wig) & Bit=1: Flash Alternate Second (wag)
Bit 3: Flash Alternate Half Hertz
Bit=0: Off/Disabled & Bit=1: On/Enabled
Bit 2: Flash Red
Bit=0: Off/Red Dark & Bit=1: On/Flash Red
Bit 1: Flash Yellow
Bit=0: Off/Yellow Dark & Bit=1: On/Flash Yellow
Bit 0: Reserved
A SET of both bits 1 & 2 shall result in bit 1=0 and bit 2=1.
A SET of a 'reserved' bit to a value other than zero shall return a
wrongValue(10) error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.2.1.4
    <Object Type> P"
    ::= { channelEntry 4 }
```

5.9.3 Maximum Channel Status Groups

```
maxChannelStatusGroups OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The maximum number of Channel Status Groups (8
channels per group) this Actuated Controller Unit supports. This value is equal
to TRUNCATE [(maxChannels + 7) / 8]. This object indicates the maximum rows which
shall appear in the channelStatusGroupTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.3
    <Object Type> S"
    ::= { channel 3 }
```

5.9.4 Channel Status Group Table

```
channelStatusGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ChannelStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit channel
output (Red, Yellow, & Green) status in groups of eight channels. The number of
rows in this table is equal to the maxChannelStatusGroups object.
```

```
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.4"
    ::= { channel 4 }

channelStatusGroupEntry OBJECT-TYPE
    SYNTAX ChannelStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Red, Yellow, & Green Output Status for eight
Actuated Controller Unit channels.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.4.1"
    INDEX   { channelStatusGroupNumber }
    ::= { channelStatusGroupTable 1 }

ChannelStatusGroupEntry ::= SEQUENCE {
    channelStatusGroupNumber  Integer32,
    channelStatusGroupReds    Integer32,
    channelStatusGroupYellows Integer32,
    channelStatusGroupGreens  Integer32 }
```

5.9.4.1 Channel Status Group Number

```
channelStatusGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The channelStatusGroup number for objects in this
row. This value shall not exceed the maxChannelStatusGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.4.1.1"
    ::= { channelStatusGroupEntry 1 }
```

5.9.4.2 Channel Status Group Reds

```
channelStatusGroupReds OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Channel Red Output Status Mask, when a bit=1, the
Channel Red is currently active. When a bit=0, the Channel Red is NOT currently
active.
Bit 7: Channel # = (channelStatusGroupNumber * 8)
Bit 6: Channel # = (channelStatusGroupNumber * 8) - 1
Bit 5: Channel # = (channelStatusGroupNumber * 8) - 2
Bit 4: Channel # = (channelStatusGroupNumber * 8) - 3
```

```
Bit 3: Channel # = (channelStatusGroupNumber * 8) - 4
Bit 2: Channel # = (channelStatusGroupNumber * 8) - 5
Bit 1: Channel # = (channelStatusGroupNumber * 8) - 6
Bit 0: Channel # = (channelStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.4.1.2
    <Object Type> S"
    ::= { channelStatusGroupEntry 2 }
```

5.9.4.3 Channel Status Group Yellows

```
channelStatusGroupYellows OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Channel Yellow Output Status Mask, when a bit=1,
the Channel Yellow is currently active. When a bit=0, the Channel Yellow is NOT
currently active.
Bit 7: Channel # = (channelStatusGroupNumber * 8)
Bit 6: Channel # = (channelStatusGroupNumber * 8) - 1
Bit 5: Channel # = (channelStatusGroupNumber * 8) - 2
Bit 4: Channel # = (channelStatusGroupNumber * 8) - 3
Bit 3: Channel # = (channelStatusGroupNumber * 8) - 4
Bit 2: Channel # = (channelStatusGroupNumber * 8) - 5
Bit 1: Channel # = (channelStatusGroupNumber * 8) - 6
Bit 0: Channel # = (channelStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.4.1.3
    <Object Type> S"
    ::= { channelStatusGroupEntry 3 }
```

5.9.4.4 Channel Status Group Greens

```
channelStatusGroupGreens OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Channel Green Output Status Mask, when a bit=1, the
Channel Green is currently active. When a bit=0, the Channel Green is NOT
currently active.
Bit 7: Channel # = (channelStatusGroupNumber * 8)
Bit 6: Channel # = (channelStatusGroupNumber * 8) - 1
Bit 5: Channel # = (channelStatusGroupNumber * 8) - 2
Bit 4: Channel # = (channelStatusGroupNumber * 8) - 3
Bit 3: Channel # = (channelStatusGroupNumber * 8) - 4
Bit 2: Channel # = (channelStatusGroupNumber * 8) - 5
Bit 1: Channel # = (channelStatusGroupNumber * 8) - 6
Bit 0: Channel # = (channelStatusGroupNumber * 8) - 7
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.8.4.1.4
    <Object Type> S"
    ::= { channelStatusGroupEntry 4 }
```

5.10 Overlap Parameters

```
overlap   OBJECT IDENTIFIER
    ::= { asc 9 }

-- This node contains objects that configure, monitor and control overlap
functions.
```

5.10.1 Maximum Overlaps

```
maxOverlaps OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "overlap"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Overlaps this Actuated
        Controller Unit supports. This object indicates the maximum number of
        rows which shall appear in the overlapTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.1
    <Object Type> S"
    ::= { overlap 1 }
```

5.10.2 Overlap Table

```
overlapTable OBJECT-TYPE
    SYNTAX SEQUENCE OF OverlapEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit
        overlap parameters. The number of rows in this table is equal to the
        maxOverlaps object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2"
    ::= { overlap 2 }

overlapEntry OBJECT-TYPE
    SYNTAX OverlapEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Actuated Controller
        Unit overlap.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1"
```

```
    INDEX { overlapNumber }
    ::= { overlapTable 1 }

OverlapEntry ::= SEQUENCE {
    overlapNumber                Integer32,
    overlapType                  INTEGER,
    overlapIncludedPhases        OCTET STRING,
    overlapModifierPhases        OCTET STRING,
    overlapTrailGreen            Integer32,
    overlapTrailYellow           Integer32,
    overlapTrailRed              Integer32,
    overlapWalk                  Integer32,
    overlapPedClearance          Integer32,
    overlapConflictingPedPhases OCTET STRING }
```

5.10.2.1 Overlap Number

```
overlapNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "overlap"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The overlap number for objects in this row. The
        value shall not exceed the maxOverlaps object. The value maps to the
        Overlap as follows:
            1 = Overlap A, 2 = Overlap B etc.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.1
    <Object Type> S"
    ::= { overlapEntry 1 }
```

5.10.2.2 Overlap Type

```
overlapType OBJECT-TYPE
    SYNTAX INTEGER { other(1),
                     normal (2),
                     minusGreenYellow (3),
                     pedestrianNormal (4),
                     fYAThreeSection (5),
                     fYAFourSection (6),
                     fRAThreeSection (7),
                     fRAFourSection (8),
                     transit-2 (9),
                     minusGreenYellowAlternate (10)}
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION "<Definition> The type of overlap operation for this row. This
value is the same for all overlap sets. The types are as follows:

        other: The overlap operates in another mode than those described herein.

        normal: The overlap output shall be controlled by the
            overlapIncludedPhases when this type is indicated. The overlap
            output shall be green in the following situations:
                (1) when an overlap included phase is green.
                (2) when an overlap included phase is yellow (or red clearance)
                    and an overlap included phase is next.
            The overlap output shall be yellow when an included phase is
            yellow and an overlap included phase is not next. The overlap
            output shall be red whenever the overlap green and yellow are
            not ON.

        minusGreenYellow: The overlap output shall be controlled by the
            overlapIncludedPhases and the overlapModifierPhases if this type
            is indicated. The overlap output shall be green in the following
            situations:
            (1) when an overlap included phase is green and an overlap
                modifier phase is NOT green.
            (2) when an overlap included phase is yellow (or red clearance)
                and an overlap included phase is next and an overlap modifier
                phase is NOT green.
            The overlap output shall be yellow when an overlap included phase
            is yellow and an overlap modifier phase is NOT yellow and an
            overlap included phase is not next. The overlap output shall be
            red whenever the overlap green and yellow are not ON.

        pedestrianNormal: The overlap output shall be controlled by the
            overlapIncludedPhases when this type is indicated. The overlap
            output shall be Walk in the following situations:
            (1) when an overlap included phase is in a walk interval.
            (2) when an overlap included phase is in a pedestrian clearance
                    interval and an overlap included phase is next.
            Upon completion of the Walk interval, the overlap enters the
            pedestrian clearance interval if another included overlap phase is
not next in the associated phase sequence.
            The overlap output shall exit the pedestrian clearance interval to
            steady Dont Walk when the programmed pedestrian clearance time
            expires. The overlap output shall be steady Dont Walk whenever the
            overlap Walk and pedestrian clearance are not ON.

        fYAThreeSection: The overlap output shall be controlled by the
```


overlapIncludedPhases and the overlapModifierPhases if this type is indicated. It shall be used with a 3-section signal head where the overlap output drives the green arrow, combined yellow/flashing yellow arrow, and red arrow. The permissive through phase opposing the left-turn signal is the overlapIncludedPhases and the associated left-turn protected phase is the overlapModifierPhases.
The overlap output shall be FYA in the following situations:
(1) when an overlap included phase is green and an overlap modifier phase is NOT green.
(2) when an overlap included phase is yellow (or red clearance), an overlap included phase is next or an overlap modifier phase is next, and a modifier phase is NOT green.

The overlap output shall be yellow:
(1) when an overlap included phase is yellow, an overlap included phase is not next, and an overlap modifier phase is NOT green.
(2) when an overlap modifier phase is yellow.

The overlap output shall be red:
(1) when an overlap included phase is red, an overlap modifier phase is NOT green, and an overlap modifier phase is NOT yellow.
(2) when an overlap modifier phase is timing a red-clearance interval.

The overlap output shall be green:
(1) when an overlap modifier phase is green.

fYAFourSection: The overlap output shall be controlled by the overlapIncludedPhases and the overlapModifierPhases if this type is indicated. It shall be used with a 4-section signal head where the overlap output drives the flashing yellow arrow, yellow and red. The permissive through phase opposing the left-turn signal is the overlapIncludedPhases and the associated left-turn protected phase is the overlapModifierPhases.
The overlap output shall be FYA in the following situations:
(1) when an overlap included phase is green and an overlap modifier phase is NOT green.
(2) when an overlap included phase is yellow (or red clearance), an overlap included phase or an overlap modifier phase is next and an overlap modifier phase is NOT green.

The overlap output shall be yellow:
(1) when an overlap included phase is yellow, an overlap

```
            included phase is not next, and an overlap modifier phase
            is NOT green.
        (2) when an overlap modifier phase is yellow.

    The overlap output shall be red:
        (1) when an overlap included phase is red, an overlap modifier
            phase is NOT green, and an overlap modifier phase is NOT
            yellow.
        (2) when an overlap modifier phase is timing a red-clearance
            interval.

    The overlap output shall be blank/dark:
        (1) when an overlap modifier phase is green

fRAThreeSection: The overlap output shall be controlled by the
    overlapIncludedPhases and the overlapModifierPhases if this type
    is indicated. The overlap output drives the green arrow, yellow
    arrow, and combined red/flashing red arrow. The
    overlapIncludedPhases is an opposing through phase and the
    overlapModifierPhases is a protected left turn phase.

    The overlap output shall be green when an overlap modifier phase
    is green.

    The overlap output shall be yellow:
        (1) when an overlap modifier phase is yellow.
        (2) when an overlap modifier phase is red and an overlap
            included phase is yellow.

    The overlap output shall be red when the overlap modifier and
        included phases are red.
    The overlap output shall be flashing red when an overlap included
        phase is green and an overlap modifier phase is red.

fRAFourSection: The overlap output shall be controlled by the
    overlapIncludedPhases and the overlapModifierPhases if this type
    is indicated. The overlap output drives the yellow arrow, red
    arrow, and flashing red arrow. The overlapIncludedPhases is an
    opposing through phase and the overlapModifierPhases is a
    protected left turn phase.

    The overlap outputs shall be blank when the overlapModifierPhase is
green.

    The overlap output shall be yellow:
```

```
                (1) when an overlap modifier phase is yellow.
                (2) when an overlap modifier phase is red and an overlap included
phase is yellow.

            The overlap output shall be red when an overlap modifier phase and
            an overlap included phase are red.
            The overlap output shall be flashing red when an overlap included
            phase is green and an overlap modifier phase is red.

        transit-2: The overlap output shall be controlled by the
            overlapIncludedPhases when this type is indicated. The overlap
            output drives a 2-section bar signal for transit vehicles using
            overlap green (vertical bar) and red (horizontal bar) outputs.

            The overlap output shall be green in the following situations:
                (1) when an overlap included phase is green.

            The overlap output shall be flashing green when an overlap
            included phase is yellow and an overlap included phase is not next.

            The overlap output shall be red whenever an overlap included phase
            is red.

        minusGreenYellowAlternate: The overlap output shall be controlled by
            the overlapIncludedPhases and the overlapModifierPhases if this
            type is indicated. The overlap output shall be green in the
            following situations:
                (1) when an overlap included phase is green and an overlap
                    modifier phase is NOT green.
                (2) when an overlap included phase is yellow (or red
                    clearance) and an overlap included phase is next and an
                    overlap modifier phase is NOT green and an overlap
                    modifier phase is not next.

            The overlap output shall be yellow when an overlap included phase
            is yellow and an overlap modifier phase is NOT yellow and an
            overlap included phase is not next. The overlap output shall be
            red whenever the overlap green and yellow are not ON.

        Note: Each enumeration requires the user to understand and avoid
        violation of MUTCD operational guidelines.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.2
    <Object Type> P2"
    DEFVAL {2}
    ::= { overlapEntry 2 }
```

5.10.2.3 Overlap Included Phase Parameter

```
overlapIncludedPhases OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet is a Phase (number) that shall be an
        included phase for the overlap. The phase number value shall not
        exceed the maxPhases object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.3
    <Object Type> P2"
    ::= { overlapEntry 3 }
```

5.10.2.4 Overlap Modifier Phase Parameter

```
overlapModifierPhases OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet is a Phase (number) that shall be a
        modifier phase for the overlap. The phase number value shall not
        exceed the maxPhases object value. The use of this object is defined
        by the overlapType.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.4
    <Object Type> P2"
    ::= { overlapEntry 4 }
```

5.10.2.5 Overlap Trailing Green Parameter

```
overlapTrailGreen OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Trailing Green Parameter in seconds
        (0-255 sec). When this value is greater than zero and the overlap
        green (or walk) would normally terminate, the overlap green (or walk)
        shall be extended by this additional time. This is applicable to
        vehicle phases, bicycle phases, and transit phases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.5
    <Object Type> P"
    ::= { overlapEntry 5 }
```

5.10.2.6 Overlap Trailing Yellow Change Parameter

```
overlapTrailYellow OBJECT-TYPE
```

```
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Trailing Yellow Change Parameter in
        tenth seconds (NEMA range: 3.0-25.5 sec). When the overlap green has
        been extended (Trailing Green), this value shall determine the current
        length of the Yellow Change interval for the overlap. This is
        applicable to vehicle phases, bicycle phases, and transit phases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.6
    <Object Type> P"
    ::= { overlapEntry 6 }
```

5.10.2.7 Overlap Trailing Red Clearance Parameter

```
overlapTrailRed OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Trailing Red Clearance Parameter in tenth
        seconds (0-25.5 sec). When the overlap green has been extended
        (Trailing Green), this value shall determine the current length of the
        Red Clearance interval for the overlap. This is applicable to vehicle
        phases, bicycle phases, and transit phases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.7
    <Object Type> P"
    ::= { overlapEntry 7 }
```

5.10.2.8 Overlap Walk Parameter

```
overlapWalk OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Walk Parameter in seconds (1-255 sec).
        This value is the length of the walk interval for a pedestrian
        overlap. Upon completion of the Walk interval, the overlap enters the
        pedestrian clearance interval. A value of zero indicates that the
phaseWalk parameter of the currently running parent phase is used for this
overlap.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.8
    <Object Type> P"
    ::= { overlapEntry 8 }
```


5.10.2.9 Overla Pedestrian Clearance Parameter

```
overlapPedClearance OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Pedestrian Clearance Parameter in
        seconds (1-255 sec). This value is the length of the pedestrian
        clearance interval. A value of zero indicates that the
phasePedestrianClear parameter of the currently running parent phase is used for
this overlap
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.9
    <Object Type> P"
```

5.10.2.10 Overlap Conflicting Pedestrian Phase Parameter

```
overlapConflictingPedPhases OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet is a Phase (number) that shall be a
        pedestrian modifier phase for the overlap. The phase number value
        shall not exceed the maxPhases object value.

        If the overlap type is 'normal', a non-null value would suppress the
        overlap when the pedestrian phase is active (in the walk or clearance
        interval). Upon completion of the active pedestrian phase and upon
        completion of a clearance interval (MUTCD requires 3 seconds), the
        overlap is allowed to proceed to the green state.

        If the overlap type is fYAThreeSection or fYAFourSection, a non-null
        value would maintain the overlap red state when the pedestrian phase
        is active (in the walk or clearance interval). Upon completion of the
        active pedestrian phase and upon completion of a clearance interval
        (MUTCD requires 3 seconds), the overlap is allowed to proceed to the
        flashing yellow state.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.2.1.10
    <Object Type> P2"
    ::= { overlapEntry 10 }
```

5.10.3 Maximum Overlap Status Groups

```
maxOverlapStatusGroups OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS read-only
```



```
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Overlap Status Groups (8
        overlaps per group) this Actuated Controller Unit supports. This value
        is equal to TRUNCATE [(maxOverlaps + 7) / 8]. This object indicates
        the maximum rows which shall appear in the overlapStatusGroupTable
        object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.3
    <Object Type> S"
    ::= { overlap 3 }
```

5.10.4 Overlap Status Group Table

```
overlapStatusGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF OverlapStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit
        overlap output (Red, Yellow, & Green) status in groups of eight
        overlaps. The number of rows in this table is equal to the
        maxOverlapStatusGroups object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.4"
    ::= { overlap 4 }

overlapStatusGroupEntry OBJECT-TYPE
    SYNTAX OverlapStatusGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Red, Yellow, & Green Output Status for eight
        Actuated Controller Unit overlaps.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.4.1"
    INDEX   { overlapStatusGroupNumber }
    ::= { overlapStatusGroupTable 1 }

OverlapStatusGroupEntry ::= SEQUENCE {
    overlapStatusGroupNumber  Integer32,
    overlapStatusGroupReds    Integer32,
    overlapStatusGroupYellows Integer32,
    overlapStatusGroupGreens  Integer32 }
```

5.10.4.1 Overlap Status Group Number

```
overlapStatusGroupNumber OBJECT-TYPE
    SYNTAX Integer32 (1..32)
    UNITS "group"
    MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION "<Definition> The Overlap Status Group number for objects in
        this row. This value shall not exceed the maxOverlapStatusGroups
        object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.4.1.1
    <Object Type> S"
    ::= { overlapStatusGroupEntry 1 }
```

5.10.4.2 Overlap Status Group Reds

```
overlapStatusGroupReds OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Overlap Red Output Status Mask, when a bit=1,
        the Overlap Red is currently active. When a bit=0, the Overlap Red is
        NOT currently active.

        Bit 7: Overlap # = (overlapStatusGroupNumber * 8)
        Bit 6: Overlap # = (overlapStatusGroupNumber * 8) - 1
        Bit 5: Overlap # = (overlapStatusGroupNumber * 8) - 2
        Bit 4: Overlap # = (overlapStatusGroupNumber * 8) - 3
        Bit 3: Overlap # = (overlapStatusGroupNumber * 8) - 4
        Bit 2: Overlap # = (overlapStatusGroupNumber * 8) - 5
        Bit 1: Overlap # = (overlapStatusGroupNumber * 8) - 6
        Bit 0: Overlap # = (overlapStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.4.1.2
    <Object Type> S"
    ::= { overlapStatusGroupEntry 2 }
```

5.10.4.3 Overlap Status Group Yellows

```
overlapStatusGroupYellows OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Overlap Yellow Output Status Mask, when a bit=1,
        the Overlap Yellow is currently active. When a bit=0, the Overlap
        Yellow is NOT currently active.

        Bit 7: Overlap # = (overlapStatusGroupNumber * 8)
        Bit 6: Overlap # = (overlapStatusGroupNumber * 8) - 1
        Bit 5: Overlap # = (overlapStatusGroupNumber * 8) - 2
        Bit 4: Overlap # = (overlapStatusGroupNumber * 8) - 3
        Bit 3: Overlap # = (overlapStatusGroupNumber * 8) - 4
        Bit 2: Overlap # = (overlapStatusGroupNumber * 8) - 5
```

```
        Bit 1: Overlap # = (overlapStatusGroupNumber * 8) - 6
        Bit 0: Overlap # = (overlapStatusGroupNumber * 8) - 7

        For pedestrianNormal overlap type, this object is used to represent
        the pedestrian clearance interval.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.4.1.3
    <Object Type> S"
    ::= { overlapStatusGroupEntry 3 }
```

5.10.4.4 Overlap Status Group Greens

```
overlapStatusGroupGreens OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Overlap Green Output Status Mask, when a bit=1,
        the Overlap Green is currently active. When a bit=0, the Overlap Green
        is NOT currently active.

        Bit 7: Overlap # = (overlapStatusGroupNumber * 8)
        Bit 6: Overlap # = (overlapStatusGroupNumber * 8) - 1
        Bit 5: Overlap # = (overlapStatusGroupNumber * 8) - 2
        Bit 4: Overlap # = (overlapStatusGroupNumber * 8) - 3
        Bit 3: Overlap # = (overlapStatusGroupNumber * 8) - 4
        Bit 2: Overlap # = (overlapStatusGroupNumber * 8) - 5
        Bit 1: Overlap # = (overlapStatusGroupNumber * 8) - 6
        Bit 0: Overlap # = (overlapStatusGroupNumber * 8) - 7
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.4.1.4
    <Object Type> S"
    ::= { overlapStatusGroupEntry 4 }
```

5.10.5 Maximum Overlap Sets

```
maxOverlapSets OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "set"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Overlap Sets this Actuated
        Controller Unit supports. This object indicates the maximum number of
        rows which shall appear in the overlapSetTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.5
    <Object Type> S"
    ::= { overlap 5 }
```


5.10.6 Overlap Set Table

```
overlapSetTable OBJECT-TYPE
    SYNTAX SEQUENCE OF OverlapSetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit
        overlap parameters. The number of rows in this table is equal to the
        maxOverlaps object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6"
    ::= { overlap 6 }

overlapSetEntry OBJECT-TYPE
    SYNTAX OverlapSetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Actuated Controller
        Unit overlap.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1"
    INDEX   { overlapSetNumber, overlapNumber }
    ::= { overlapSetTable 1 }

OverlapSetEntry ::= SEQUENCE {
    overlapSetNumber                Integer32,
    overlapSetIncludedPhases       OCTET STRING,
    overlapSetModifierPhases       OCTET STRING,
    overlapSetTrailGreen           Integer32,
    overlapSetTrailYellow          Integer32,
    overlapSetTrailRed             Integer32,
    overlapSetWalk                 Integer32,
    overlapSetPedClearance         Integer32,
    overlapSetConflictingPedPhases OCTET STRING }
```

5.10.6.1 Overlap Set Number

```
overlapSetNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "overlap set"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The overlap number for objects in this row. The
        value shall not exceed the maxOverlapSets object. The value maps to the
        Overlap as follows:
            1 = Overlap A, 2 = Overlap B etc.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.1
```

```
    <Object Type> S"
    ::= { overlapSetEntry 1 }
```

5.10.6.2 Overlap Set Included Phase Parameter

```
overlapSetIncludedPhases OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet is a Phase (number) that shall be an
        included phase for the overlap. The phase number value shall not
        exceed the maxPhases object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.2
    <Object Type> P2"
    ::= { overlapSetEntry 2 }
```

5.10.6.3 Overlap Set Modifier Phase Parameter

```
overlapSetModifierPhases OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet is a Phase (number) that shall be a
        modifier phase for the overlap. The phase number value shall not
        exceed the maxPhases object value. The use of this object is defined
        by the overlapType.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.3
    <Object Type> P2"
    ::= { overlapSetEntry 3 }
```

5.10.6.4 Overlap Set Trailing Green Parameter

```
overlapSetTrailGreen OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Trailing Green Parameter in seconds
        (0-255 sec). When this value is greater than zero and the overlap
        green (or walk) would normally terminate, the overlap green (or walk)
        shall be extended by this additional time. This is applicable to
        vehicle phases, bicycle phases, and transit phases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.4
    <Object Type> P"
    ::= { overlapSetEntry 4 }
```


5.10.6.5 Overlap Set Trailing Yellow Change Parameter

```
overlapSetTrailYellow OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Trailing Yellow Change Parameter in
        tenth seconds (NEMA range: 3.0-25.5 sec). When the overlap green has
        been extended (Trailing Green), this value shall determine the current
        length of the Yellow Change interval for the overlap. This is
        applicable to vehicle phases, bicycle phases, and transit phases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.5
    <Object Type> P"
    ::= { overlapSetEntry 5 }
```

5.10.6.6 Overlap Set Trailing Red Clearance Parameter

```
overlapSetTrailRed OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Trailing Red Clearance Parameter in tenth
        seconds (0-25.5 sec). When the overlap green has been extended
        (Trailing Green), this value shall determine the current length of the
        Red Clearance interval for the overlap. This is applicable to vehicle
        phases, bicycle phases, and transit phases.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.6
    <Object Type> P"
    ::= { overlapSetEntry 6 }
```

5.10.6.7 Overlap Set Walk Parameter

```
overlapSetWalk OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Walk Parameter in seconds (0-255 sec).
        This value is the length of the walk interval for a pedestrian
        overlap. Upon completion of the Walk interval, the overlap enters the
        pedestrian clearance interval.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.7
    <Object Type> P"
    ::= { overlapSetEntry 7 }
```

5.10.6.8 Overlap Set Pedestrian Clearance Parameter

```
overlapSetPedClearance OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Overlap Pedestrian Clearance Parameter in
        seconds (0-255 sec). This value is the length of the pedestrian
        clearance interval.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.8
    <Object Type> P"
    ::= { overlapSetEntry 8 }
```

5.10.6.9 Overlap Set Conflicting Pedestrian Phase Parameter

```
overlapSetConflictingPedPhases OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet is a Phase (number) that shall be a
        pedestrian modifier phase for the overlap. The phase number value
        shall not exceed the maxPhases object value.

        If the overlap type is 'normal', a non-null value would suppress the
        overlap when the pedestrian phase is active (in the walk or clearance
        interval). Upon completion of the active pedestrian phase and upon
        completion of a clearance interval (MUTCD requires 3 seconds), the
        overlap is allowed to proceed to the green state.

        If the overlap type is fYAThreeSection or fYAFourSection, a non-null
        value would maintain the overlap red state when the pedestrian phase
        is active (in the walk or clearance interval). Upon completion of the
        active pedestrian phase and upon completion of a clearance interval
        (MUTCD requires 3 seconds), the overlap is allowed to proceed to the
        flashing yellow state.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.9.6.1.9
    <Object Type> P2"
    ::= { overlapSetEntry 9 }
```

5.11 TS2 Port 1 Parameters

```
ts2port1  OBJECT IDENTIFIER
    ::= { asc 10 }

-- This object is an identifier used to group all objects for support of NEMA TS
2 (Clause 3.3.1) Port 1 activities.
```

5.11.1 Maximum Port 1 Addresses

```
maxPort1Addresses OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "address"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of Port 1 addresses this
        Actuated Controller Unit supports. This object indicates the maximum
        rows which shall appear in the port1Table object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.10.1
    <Object Type> S"
    ::= { ts2port1 1 }
```

5.11.2 Port 1 Table

```
port1Table OBJECT-TYPE
    SYNTAX SEQUENCE OF Port1Entry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit
        port 1 parameters. The number of rows in this table is equal to
        maxPort1Addresses object. Address 255 is reserved for the all stations
        (link devices) address.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.10.2"
    ::= { ts2port1 2 }

port1Entry OBJECT-TYPE
    SYNTAX Port1Entry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a conceptual row in the
        port 1 Table.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.10.2.1"
    INDEX { port1Number }
    ::= { port1Table 1 }

Port1Entry ::= SEQUENCE {
    port1Number        Integer32,
    port1DevicePresent Integer32,
    port1Frame40Enable Integer32,
    port1Status        INTEGER,
    port1FaultFrame    Integer32 }
```


5.11.2.1 Port 1 Number

```
port1Number OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "address"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The (Port 1 address plus one) for objects in
        this row. This value shall not exceed the maxPort1Addresses object
        value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.10.2.1.1"
    ::= { port1Entry 1 }
```

5.11.2.2 Port 1 Device Present

```
port1DevicePresent OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to program the CU as to the
        presence or absence of a device for this Port 1 address. The CU shall
        transmit Command Frames only to those devices that are present as
        determined by this programming.
            True (one) - the device is present.
            False (zero) - the device is not present.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.10.2.1.2
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.3.1.4"
    ::= { port1Entry 2 }
```

5.11.2.3 Port 1 Frame 40 Enable

```
port1Frame40Enable OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> To enable or disable Frame 40 messages to the
        device at this Port 1 address. Frame 40 is used to poll the secondary
        stations for a secondary to secondary message exchange. Command 40
        series frames shall be transmitted only to those devices that are
        enabled, as determined by this programming.
            TRUE (one) - Enable frame 40 messages for this device.
            FALSE (zero) - Disable frame 40 messages for this device.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.10.2.1.3
    <Object Type> P"
    REFERENCE "NEMA TS 2 Clause 3.3.1.4.1"
    ::= { port1Entry 3 }
```

5.11.2.4 Port 1 Status

```
port1Status OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     online (2),
                     responseFault (3) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the communications status
        with the associated device:

        other: This indicates that some other communications fault has
            been detected.
        online: This indicates that at least five of the most recent 10
            response transfers were received correctly.
        responseFault: This indicates that more than 5 of the most recent 10
            response transfers were received incorrectly.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.10.2.1.4
    <Object Type> S"
    ::= { port1Entry 4 }
```

5.11.2.5 Port 1 Fault Frame

```
port1FaultFrame OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the frame number that
        caused the most recent fault.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.10.2.1.5
    <Object Type> S"
    ::= { port1Entry 5 }
```

5.12 ASC Block Objects

```
ascBlock  OBJECT IDENTIFIER
    ::= { asc 11 }

-- This object is an identifier used to group all objects for support of ASC
Block Upload and Download activities.
```

5.12.1 ASC Block Get Control

```
ascBlockGetControl OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4..12))
    MAX-ACCESS read-write
    STATUS current
```


DESCRIPTION "<Definition> An OER encoded string of reference parameters for ASC Block Uploads. The parameter values in this string are:

```
ascBlockDataType    INTEGER (0..255)
ascBlockDataID      INTEGER (0..255)
ascBlockIndex1      INTEGER (0..255) if needed
ascBlockQuantity1   INTEGER (0..255) if needed
ascBlockIndex2      INTEGER (0..255) if needed
ascBlockQuantity2   INTEGER (0..255) if needed
ascBlockIndex3      INTEGER (0..255) if needed
ascBlockQuantity3   INTEGER (0..255) if needed
ascBlockIndex4      INTEGER (0..255) if needed
ascBlockQuantity4   INTEGER (0..255) if needed
ascBlockIndex5      INTEGER (0..255) if needed
ascBlockQuantity5   INTEGER (0..255) if needed
```

A GET of ascBlockData shall utilize values currently in this object to define the data to be returned.

A SET of this object shall be evaluated for validity and Error Status of badValue(3) be returned for the following conditions:
1) ascBlockDataType is not supported
2) ascBlockDataID is not supported
3) ascBlockIndex1 is zero or not supported
4) ascBlockQuantity1 is zero or ascBlockIndex1 + ascBlockQuantity1 - 1 is not supported
5) ascBlockIndex2 is zero or not supported
6) ascBlockQuantity2 is zero or ascBlockIndex2 + ascBlockQuantity2) - 1 is not supported
7) ascBlockIndex3 is zero or not supported
8) ascBlockQuantity3 is zero or ascBlockIndex3 + scBlockQuantity3) - 1 is not supported
9) ascBlockIndex4 is zero or not supported

10) ascBlockQuantity4 is zero or ascBlockIndex4 + ascBlockQuantity4) - 1 is not supported
11) ascBlockIndex5 is zero or not supported
12) ascBlockQuantity5 is zero or ascBlockIndex5 + scBlockQuantity5) - 1 is not supported
13) if the SET length is zero or incorrect for ascBlockDataType & ascBlockDataID
14) if the GetResponse length for a GET on ascBlockData using maximum data field sizes would exceed a local limitation

When this validity check fails, ascBlockErrorStatus shall be set equal to the Bullet Value above that generated the error.

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.11.1
    <Object Type> C"
    ::= { ascBlock 1 }
```

5.12.2 ASC Block Data

```
ascBlockGetControl OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4..12))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An OER encoded string of reference parameters for
ASC Block Uploads. The parameter values in this string are:

ascBlockDataType    INTEGER (0..255)
ascBlockDataID      INTEGER (0..255)
ascBlockIndex1      INTEGER (0..255) if needed
ascBlockQuantity1   INTEGER (0..255) if needed
ascBlockIndex2      INTEGER (0..255) if needed
ascBlockQuantity2   INTEGER (0..255) if needed
ascBlockIndex3      INTEGER (0..255) if needed
ascBlockQuantity3   INTEGER (0..255) if needed
ascBlockIndex4      INTEGER (0..255) if needed
ascBlockQuantity4   INTEGER (0..255) if needed
ascBlockIndex5      INTEGER (0..255) if needed
ascBlockQuantity5   INTEGER (0..255) if needed

A GET of ascBlockData shall utilize values currently in this object to define the
data to be returned.

A SET of this object shall be evaluated for validity and Error Status of
badValue(3) be returned for the following conditions:
1) ascBlockDataType is not supported
2) ascBlockDataID is not supported
3) ascBlockIndex1 is zero or not supported
4) ascBlockQuantity1 is zero or ascBlockIndex1 + ascBlockQuantity1 - 1 is not
supported
5) ascBlockIndex2 is zero or not supported
6) ascBlockQuantity2 is zero or ascBlockIndex2 + ascBlockQuantity2) - 1 is not
supported
7) ascBlockIndex3 is zero or not supported
8) ascBlockQuantity3 is zero or ascBlockIndex3 + scBlockQuantity3) - 1 is not
supported
9) ascBlockIndex4 is zero or not supported

10) ascBlockQuantity4 is zero or ascBlockIndex4 + ascBlockQuantity4) - 1 is not
supported
```

```
11) ascBlockIndex5 is zero or not supported
12) ascBlockQuantity5 is zero or ascBlockIndex5 + scBlockQuantity5) - 1 is not
supported
13) if the SET length is zero or incorrect for ascBlockDataType & ascBlockDataID
14) if the GetResponse length for a GET on ascBlockData using maximum data field
sizes would exceed a local limitation

When this validity check fails, ascBlockErrorStatus shall be set equal to the
Bullet Value above that generated the error.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.11.1
    <Object Type> C"
    ::= { ascBlock 1 }

--5.12.2    ASC Block Data
ascBlockData OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(6..65535))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An OER encoded string used for uploading and
downloading ASC parameters. See SECTION 6 for encoding and decoding the block. A
SET on this object shall require the use of the Transaction feature defined in
ISO 26048-1.

A SET of this object shall be evaluated for validity and Error Status of
badValue(3) be returned for the following conditions:
1) ascBlockDataType is not supported
2) ascBlockDataID is not supported
3) ascBlockIndex1 is zero or not supported
4) ascBlockQuantity1 is zero or ascBlockIndex1 + ascBlockQuantity1 - 1 is not
supported
5) ascBlockIndex2 is zero or not supported
6) ascBlockQuantity2 is zero or ascBlockIndex2 + ascBlockQuantity2) - 1 is not
supported
7) ascBlockIndex3 is zero or not supported
8) ascBlockQuantity3 is zero or ascBlockIndex3 + ascBlockQuantity3) - 1 is not
supported
9) ascBlockIndex4 is zero or not supported
10) ascBlockQuantity4 is zero or ascBlockIndex4 + ascBlockQuantity4) - 1 is not
supported
11) ascBlockIndex5 is zero or not supported
12) ascBlockQuantity5 is zero or ascBlockIndex5 + ascBlockQuantity5) - 1 is not
supported
13) if the SET length is zero or incorrect for ascBlockDataType & ascBlockDataID
14) if the SET (SEQUENCE OF) value is incorrect.
```

```
When this validity check fails, ascBlockErrorStatus shall be set equal to the
Bullet Value above that generated the error.

A SET that includes an unsupported value for a supported data element shall
return an Error Status of badValue(3) and ascBlockErrorStatus shall be set equal
to: (data Sequence # * 100) + data Element #

A SET that includes a non-zero or non-null value in the position of an
unsupported data element shall return an Error Status of badValue(3) and
ascBlockErrorStatus shall be set equal to: (data Sequence # * 100) + data Element
#

A GET on this object shall utilize values currently in ascBlockGetControl to
define the data to be returned. When ascBlockGetControl has invalid data, an
Error STATUS of badValue(3) shall be returned.

A GET shall return a zero or null value in the position of an unsupported object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.11.2
    <Object Type> C"
    ::= { ascBlock 2 }
```

5.12.3 ASC Block Error Status

```
ascBlockErrorStatus OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object defines the data element within
ascBlockGetControl or ascBlockData that caused a badValue(3) ErrorStatus. This
object should equal zero after any successful SET to ascBlockGetControl or
ascBlockData.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.11.3
    <Object Type> S"
    ::= { ascBlock 3 }
```

5.13 I/O Mapping

```
ascIOmapping OBJECT IDENTIFIER
    ::= { asc 13 }

-- This node contains objects that configure, monitor or control input and output
mapping in the ASC
```

5.13.1 I/O Mapping Control

```
ascIOmapControl OBJECT IDENTIFIER
    ::= { ascIOmapping 1 }
```

```
-- This node contains objects that control the current I/O map
```

5.13.1.1 Maximum Number of I/O Maps

```
ascIOmaxMaps OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of I/O maps
this ASC supports. This object indicates the number of rows in the
ascIOmapsTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.1.1
    <Object Type> S"
    ::= { ascIOmapControl 1 }
```

5.13.1.2 Active I/O Map

```
ascIOactiveMap OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object selects the active I/O map. This object
has to be changed using the Transaction feature defined in ISO 26048-1. A
Database Transaction that changes this object or edits the currently active I/O
map has to satisfy the activation requirements in ascIOactivateRequirement at the
time fdTransactionMode is set to verify (3) for the transaction to successfully
verify.

      The value of this object cannot exceed the value of ascIOmaxMaps.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.1.2
    <Object Type> S"
    DEFVAL { 1 }
    ::= { ascIOmapControl 2 }
```

5.13.1.3 Conditions for Activating New I/O Map

```
ascIOactivateRequirement OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object returns the conditions under which an
ASC will confirm a DB transaction that activates a new I/O map or edits the
currently active I/O map. The requirements are selected in a bitmap format:
        Bit 0 - cabinetDoorOpen -- I/O input cabinetDoorOpen active
        Bit 1 - inFlash         -- in any flash state
        Bit 2 - allRedFlash     -- in programmed all red flash
```

```
        Bit 3 - cabinetFlash    -- in CVM flash, input localFlashSense active
        Bit 4 - restart         -- changes take effect only after a restart
        Bit 5 - reserved
        Bit 6 - reserved
        Bit 7 - reserved
    Note: if all I/O mapping values being set are the same as the   current
values, the DB transaction shall succeed without requiring any conditions set by
this object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.1.3
    <Object Type> S"
    ::= { ascIOmapControl 3 }
```

5.13.2 I/O Maps Maximum Inputs

```
ascIOmapMaxInputs OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of I/O
mapping input functions this ASC supports. This object indicates the number of
rows in the ascIOinputMapTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.2
    <Object Type> S"
    ::= { ascIOmapping 2 }
```

5.13.3 I/O Maps Maximum Outputs

```
ascIOmapMaxOutputs OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of I/O
mapping output functions this ASC supports. This object indicates the number of
rows in the ascIOoutputMapTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.3
    <Object Type> S"
    ::= { ascIOmapping 3 }
```

5.13.4 I/O Input Map Table

```
ascIOinputMapTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AscIOinputMapTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing mapping for an ASC Controller's
inputs to functions. The number of I/O input map tables is equal to the value of
ascIOmaxMaps.
```

```
      The total number of rows in the table is ascIOmapMaxInputs. Only one I/O
input map may be active at any one time, and is selected by ascIOactiveMap.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4"
    ::= { ascIOmapping 4 }

ascIOinputMapTableEntry OBJECT-TYPE
    SYNTAX AscIOinputMapTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a conceptual row in the
ascIOinputMapTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1"
    INDEX { ascIOmapNumber,  ascIOinputMapIOindex }
    ::= { ascIOinputMapTable 1 }

AscIOinputMapTableEntry ::= SEQUENCE {
   ascIOmapNumber            Integer32,    -- up to ascIOmaxMaps
   ascIOinputMapIOindex      Integer32,      -- up to ascIOmapMaxInputs
   ascIOinputMapDeviceType   INTEGER,    -- enum custom, FIO,TS1,BIU,SIU,AUX,
reserved
   ascIOinputMapDevicePNN    Integer32,    -- NEMA PNN if DeviceType is custom
   ascIOinputMapDevicePtype Integer32,    -- Custom device type
   ascIOinputMapDeviceAddr   Integer32,    -- only used if needed (BIU, SIU)
   ascIOinputMapDevicePin    INTEGER,      -- device I/O pin index
   ascIOinputMapFuncType     Integer32,    -- 0=STD, else nemaPrivate vendor code
   ascIOinputMapFuncPtype    Integer32,    -- Custom function type set
   ascIOinputMapFunction     Integer32,    -- function
   ascIOinputMapFuncIndex    Integer32 }  -- index if function support more than
one input or output
```

5.13.4.1 I/O Map Number

```
ascIOmapNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The value of this object shall not exceed the
ascIOmaxMaps value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.1
    <Object Type> S"
    ::= { ascIOinputMapTableEntry 1 }
```

5.13.4.2 I/O Input Map Index

```
--5.13.4.2  I/O Input Map Index
```



```
ascIOinputMapIOindex OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The I/O index for this row of the table. The range
will not exceed ascIOmapMaxInputs.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.2
    <Object Type> S "
    ::= { ascIOinputMapTableEntry 2 }
```

5.13.4.3 I/O Input Map Device Type

```
ascIOinputMapDeviceType OBJECT-TYPE
    SYNTAX INTEGER { unused (1),
                  custom (2),
                  fio (3),
                  ts1 (4),
                  biu (5),
                  siu (6),
                  aux (7) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is an selects the device type for this
row of the table.
      A value of unused (1) means that this row of the table is unused (all
values of ascIOinputMapIOindex up to ascIOmapMaxInputs may not be needed by every
mapping).

A custom type is a manufacturer defined device which also requires a
ascIOinputMapDevicePNN and a ascIOinputMapDevicePtype to fully specify the
device.

Values > 7 are reserved for future device types.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.3
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 3 }
```

5.13.4.4 I/O Input Map Custom Device Manufacturer ID

```
ascIOinputMapDevicePNN OBJECT-TYPE
 SYNTAX Integer32 (0..65535)
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION "<Definition> The object is used to further define a device type
when ascIOinputMapDeviceType is custom (2). The value of this object will be the
```

```
manufacturer's Private Node Number (PNN) as assigned by NEMA
(1.3.6.1.4.1.1206.3.PNN). This is the same identifier used for ASC custom blocks.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.4
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 4 }
```

5.13.4.5 I/O Input Map Custom Device Type

```
ascIOinputMapDevicePtype OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The object is used to further define a device type
when ascIOinputMapDeviceType is custom (2). The value of this object will
identify a custom device type unique to the manufacturer specified by
ascIOinputMapDevicePNN.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.5
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 5 }
```

5.13.4.6 I/O Input Map Device Address

```
ascIOinputMapDeviceAddr OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An address for the device at this table row. The
address is used for devices like BIUs and SIUs that require an address. The value
should be zero if the address is not needed for this row.
      If the ascIOinputMapDeviceType is biu (4), the address values are:
                1    - Traffic Facilities BIU #1
                2    - Traffic Facilities BIU #2
                3    - Traffic Facilities BIU #3
                4    - Traffic Facilities BIU #4
                5-8 - reserved
                9    - Detector BIU #1
                10   - Detector BIU #2
                11   - Detector BIU #3
                12   - Detector BIU #4
                13-16 - reserved
      Note that these values are the BIU SDLC address + 1.

      If the ascIOinputMapDeviceType is siu (5), the address values are:
                1  - reserved
                2  - 14-pack output SIU position 1
                3  - reserved
```

```
                4  - 14-pack output SIU position 3
                5  -  6-pack output SIU position 4
                6  -  6-pack output SIU position 1
                7  -  6-pack output SIU position 2
                8  -  6-pack output SIU position 3
                9  - reserved
                10 - input SIU #1
                11 - input SIU #2
                12 - input SIU #3
                13 - input SIU #4
                14 - input SIU #5
                15 - reserved
        Note that these values are the SIU SDLC address + 1.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.6
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 6 }
```

5.13.4.7 I/O Input Map Device Pin

```
ascIOinputMapDevicePin OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An index for the I/O pin at this table row. The
range for this value will depend upon the value of other objects in the row.
      The range for ascIOinputMapDeviceType:
            fio (3) pin range is AscIOmapFIOinputs
            ts1 (4) pin range is AscIOmapTS1inputs
            biu (5) pin range is AscIOmapBIUinputs
            siu (6) pin range is AscIOmapSIUinputs
            aux (7) pin range is AscIOmapAUXinputs
      The range for a custom device type will be determined by the manufacturer
defining it.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.7
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 7 }
```

5.13.4.8 I/O Input Map Function Type

```
ascIOinputMapFuncType OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object selects the function type for this row
of the table. A value of zero references the standard input functions defined by
AscIOinputType and the ascIOinputMapFunction.
```

```
      Any other value is a manufacturer's Private Node Number (PNN) as assigned
by NEMA (1.3.6.1.4.1.1206.3.PNN). In this case the  function is defined by the
PNN (the manufacturer), ascIOinputMapFuncPtype (which of the manufacturer's
multiple function sets), and the ascIOinputMapFunction.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.8
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 8 }
```

5.13.4.9 I/O Input Map Custom Function Type

```
ascIOinputMapFuncPtype OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The object is used to further define a function
type when ascIOinputMapFuncType is not zero. The value of this object will
identify a custom function type unique to the manufacturer specified by
ascIOinputMapDevicePNN.

This allows a manufacturer to have multiple sets of functions for their one
Private Node Number.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.9
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 9 }
```

5.13.4.10 I/O Input Map Function

```
ascIOinputMapFunction OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object specifies the function that is mapped
to the I/O pin specified by this row.
      For example, if ascIOinputMapFuncType is zero (standard) then the function
could be any AscIOinputType value such as vehicleDetector (51).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.10
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 10 }
```

5.13.4.11 I/O Input Map Function Index

```
ascIOinputMapFuncIndex OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
```



```
    DESCRIPTION "<Definition> An index for the function at this table row. For
example, if ascIOinputMapFuncType is zero (standard) and the
ascIOinputMapFunction is vehicleDetector (51) then this index will determine
which detector input it is (1 to maxVehicleDetectors).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.4.1.11
    <Object Type> P"
    ::= { ascIOinputMapTableEntry 11 }
```

5.13.5 I/O Input Map Status Table

```
ascIOinputMapStatusTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AscIOinputMapStatusTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing status for the current mapping
for an ASC controller's inputs.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.5"
    ::= { ascIOmapping 5 }

ascIOinputMapStatusTableEntry OBJECT-TYPE
    SYNTAX AscIOinputMapStatusTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a conceptual row in the
ascIOinputMapStatusTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.5.1"
    INDEX { ascIOmapNumber, ascIOinputMapIOindex }
    ::= { ascIOinputMapStatusTable   1 }

AscIOinputMapStatusTableEntry ::= SEQUENCE {
   ascIOinputMapDevPinDescr   SnmpAdminString, -- description of input pin
   ascIOinputMapDevPinStatus  Integer32 }     -- 0 or 1 for active
```

5.13.5.1 I/O Input Map Device Pin Description

```
ascIOinputMapDevPinDescr OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object returns the name of the input pin, such
as 'C1-39 Detector 2'.
      Since the physical pins are determined by the controller hardware, the
value is read-only.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.5.1.1
    <Object Type> S"
```

```
    ::= { ascIOinputMapStatusTableEntry 1 }
```

5.13.5.2 I/O Input Map Device Pin Status

```
ascIOinputMapDevPinStatus OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object returns the current status of an input
pin: inactive/OFF (0) or active/ON (1).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.5.1.2
    <Object Type> S"
    ::= { ascIOinputMapStatusTableEntry 2 }
```

5.13.6 I/O Output Map Table

```
ascIOoutputMapTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AscIOoutputMapTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing mapping for an ASC Controller's
inputs and outputs to functions.
      The number of I/O map tables is equal to the value of ascIOmaxMaps.
      The total number of rows in the table is ascIOmapMaxOutputs.
      Only one I/O map may be active at any one time, and is selected by
ascIOactiveMap.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6"
    ::= { ascIOmapping 6 }

ascIOoutputMapTableEntry OBJECT-TYPE
    SYNTAX AscIOoutputMapTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a conceptual row in the
ascIOoutputMapTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1"
    INDEX { ascIOmapNumber, ascIOoutputMapIOindex }
    ::= { ascIOoutputMapTable 1 }

AscIOoutputMapTableEntry ::= SEQUENCE {
   ascIOoutputMapIOindex       Integer32,    -- up to ascIOmapMaxOutputs
   ascIOoutputMapDeviceType    INTEGER,    -- enum custom, FIO,TS1,BIU,SIU,AUX,
reserved
   ascIOoutputMapDevicePNN     Integer32,    -- NEMA PNN if DeviceType is custom
   ascIOoutputMapDevicePtype   Integer32,    -- Custom device type
```

```
   ascIOoutputMapDeviceAddr    Integer32,   -- only used if needed (BIU, SIU)
   ascIOoutputMapDevicePin     Integer32,   -- device I/O pin index
   ascIOoutputMapFuncType      Integer32,   -- 0=STD, else nemaPrivate vendor
code
   ascIOoutputMapFuncPtype     Integer32,   -- Custom function type
   ascIOoutputMapFunction      Integer32,   -- function
   ascIOoutputMapFuncIndex     Integer32 }  -- index if function support more
than one input or output
```

5.13.6.1 I/O Output Map Index

```
ascIOoutputMapIOindex OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The I/O index for this row of the table. The range
is 1 to ascIOmapMaxOutputs.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.1
    <Object Type> S"
    ::= { ascIOoutputMapTableEntry 1 }
```

5.13.6.2 I/O Output Map Device Type

```
ascIOoutputMapDeviceType OBJECT-TYPE
    SYNTAX INTEGER { unused (1),
                  custom (2),
                  fio (3),
                  ts1 (4),
                  biu (5),
                  siu (6),
                  aux (7) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is an selects the device type
for         this row of the table.
      A value of unused (1) means that this row of the table is unused (all
values of ascIOoutputMapIOindex up to or ascIOmapMaxOutputs may not be needed by
every mapping).
      A custom type is a manufacturer defined device which also requires a
ascIOoutputMapDevicePNN and a ascIOoutputMapDevicePtype to fully specify the
device.
      Values > 7 are reserved for future device types.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.2
    <Object Type> P"
    ::= { ascIOoutputMapTableEntry 2 }
```

5.13.6.3 I/O Output Map Custom Device Manufacturer

```
ascIOoutputMapDevicePNN OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The object is used to further define a device type
when ascIOoutputMapDeviceType is custom (2). The value of this object will be the
manufacturer's Private Node Number (PNN) as assigned by NEMA
(1.3.6.1.4.1.1206.3.PNN). This is the same identifier used for ASC custom blocks.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.3
    <Object Type> P"
```

5.13.6.4 I/O Output Map Custom Device Type

```
ascIOoutputMapDevicePtype OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The object is used to further define a device type
when ascIOoutputMapDeviceType is custom (2). The value of this object will
identify a custom device type unique to the manufacturer specified by
ascIOoutputMapDevicePNN.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.4
    <Object Type> P"
    ::= { ascIOoutputMapTableEntry 4 }
```

5.13.6.5 I/O Output Map Device Address

```
ascIOoutputMapDeviceAddr OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An address for the device at this table row. The
address is used for devices like BIUs and SIUs that require an address. The value
should be zero if the address is not needed for this row.
      If the ascIOoutputMapDeviceType is biu (4), the address values are:
                1   - Traffic Facilities BIU #1
                2   - Traffic Facilities BIU #2
                3   - Traffic Facilities BIU #3
                4   - Traffic Facilities BIU #4
                5-8 - reserved
                9   - Detector BIU #1
                10  - Detector BIU #2
                11  - Detector BIU #3
                12  - Detector BIU #4
                13-16 - reserved
```

```
      Note that these values are the BIU SDLC address + 1.

      If the ascIOoutputMapDeviceType is siu (5), the address values  are:
                1  - reserved
                2  - 14-pack output SIU position 1
                3  - reserved
                4  - 14-pack output SIU position 3
                5  -  6-pack output SIU position 4
                6  -  6-pack output SIU position 1
                7  -  6-pack output SIU position 2
                8  -  6-pack output SIU position 3
                9  - reserved
                10 - input SIU #1
                11 - input SIU #2
                12 - input SIU #3
                13 - input SIU #4
                14 - input SIU #5
                15 - reserved
      Note that these values are the SIU SDLC address + 1.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.5
    <Object Type> P"
    ::= { ascIOoutputMapTableEntry 5 }
```

5.13.6.6 I/O Output Map Device Pin

```
ascIOoutputMapDevicePin OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> An index for the I/O pin at this table row. The
range for this value will depend upon the value of other objects in the row.
      The range for ascIOmapDeviceType:
            fio (3) pin range is AscIOmapFIOoutputs
            ts1 (4) pin range is AscIOmapTS1outputs
            biu (5) pin range is AscIOmapBIUoutputs
            siu (6) pin range is AscIOmapSIUoutputs
            aux (7) pin range is AscIOmapAUXoutputs
      The range for a custom device type will be determined by the manufacturer
defining it.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.6
    <Object Type> P"
    ::= { ascIOoutputMapTableEntry 6 }
```

5.13.6.7 I/O Output Map Function Type

```
ascIOoutputMapFuncType OBJECT-TYPE
```

```
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object selects the function type for this row
of the table. A value of zero references the standard output functions defined by
AscIOoutputType and the ascIOoutputMapFunction.
      Any other value is a manufacturer's Private Node Number (PNN) as assigned
by NEMA (1.3.6.1.4.1.1206.3.PNN). In this case the function is defined by the PNN
(the manufacturer), ascIOoutputMapFuncPtype (which of the manufacturer's multiple
function sets), and the ascIOoutputMapFunction.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.7
    <Object Type> P"
    ::= { ascIOoutputMapTableEntry 7 }
```

5.13.6.8 I/O Output Map Custom Function Type

```
ascIOoutputMapFuncPtype OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The object is used to further define a function
type when ascIOoutputMapFuncType is not zero. The value of this object will
identify a custom function type unique to the manufacturer specified by
ascIOoutputMapDevicePNN.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.8
    <Object Type> P"
    ::= { ascIOoutputMapTableEntry 8 }
```

5.13.6.9 I/O Output Map Function

```
ascIOoutputMapFunction OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object specifies the function that is mapped
to the I/O pin specified by this row. For example, if ascIOoutputMapFuncType is
zero (standard) then the function could be any AscIOoutputType value such as
channelGreen (6).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.9
    <Object Type> P"
    ::= { ascIOoutputMapTableEntry 9 }
```

5.13.6.10 I/O Output Map Function Index

```
ascIOoutputMapFuncIndex OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION "<Definition> An index for the function at this table row. For
example, if ascIOoutputMapFuncType is zero (standard) and the
ascIOoutputMapFunction is channelGreen (6), then this index will determine which
channelGreen output it is (1 to maxChannels).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.6.1.11
    <Object Type> P"
    ::= { ascIOoutputMapTableEntry 11 }
```

5.13.7 I/O Output Map Status Table

```
ascIOoutputMapStatusTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AscIOoutputMapStatusTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing status for the current mapping
for an ASC controller's outputs.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.7"
    ::= { ascIOmapping 7 }

ascIOoutputMapStatusTableEntry OBJECT-TYPE
    SYNTAX AscIOoutputMapStatusTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a conceptual row in the
ascIOoutputMapStatusTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.7.1"
    INDEX { ascIOmapNumber, ascIOoutputMapIOindex }
    ::= { ascIOoutputMapStatusTable   1 }

AscIOoutputMapStatusTableEntry ::= SEQUENCE {
   ascIOoutputMapDevPinDescr    SnmpAdminString, -- description of output pin
   ascIOoutputMapDevPinStatus   Integer32 }      -- 0 or 1 for active
```

5.13.7.1 I/O Output Map Device Description

```
ascIOoutputMapDevPinDescr OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object returns the name of the output pin,
such as 'C1-39 Detector 2'. Since the physical pins are determined by the
controller hardware, the value is read-only.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.7.1.1
    <Object Type> S"
```

```
    ::= { ascIOoutputMapStatusTableEntry 1 }
```

5.13.7.2 I/O Output Map Device Pin Status

```
ascIOoutputMapDevPinStatus OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object returns the current status of an output
pin: inactive/OFF (0) or active/ON (1).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.7.1.2"
    ::= { ascIOoutputMapStatusTableEntry 2 }
```

5.13.8 I/O Map Description Table

```
ascIOmapDescriptionTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AscIOmapDescriptionTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing mapping for an ASC Controller's
inputs and outputs to functions. The number of I/O map tables is equal to the
value of ascIOmaxMaps.
      Each table contains a row for each input the supported by the ASC. The
total number of rows in the table is ascIOmapMaxInputs.
      Only one I/O input map may be active at any one time, and is selected by
ascIOactiveMap.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.8"
    ::= { ascIOmapping 8 }

ascIOmapDescriptionTableEntry OBJECT-TYPE
    SYNTAX AscIOmapDescriptionTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a conceptual set of rows in the
ascIOinputMapTable and ascIOoutputMapTable tables corresponding to an
ascIOmapNumber.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.8.1"
    INDEX { ascIOmapNumber }
    ::= { ascIOmapDescriptionTable   1 }

AscIOmapDescriptionTableEntry ::= SEQUENCE {
   ascIOmapDescription   SnmpAdminString }
```

5.13.8.1 I/O Map Description

```
ascIOmapDescription OBJECT-TYPE
```

```
    SYNTAX SnmpAdminString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object returns the name of the I/O map.  There
is only one name for the I/O map for each value of ascIOmapNumber.
      This map name corresponds to ascIOinputMapTable and ascIOoutputMapTable
rows with the same ascIOmapNumber index.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.8.1.1
    <Object Type> P"
    ::= { ascIOmapDescriptionTableEntry 1 }
```

5.13.9 I/O Map Input Functions

```
ascIOmapInputFunctions OBJECT IDENTIFIER ::= { ascIOmapping 9 }

-- Enumeration for I/O mapping input functions
AscIOinputType ::= INTEGER {
    unusedInput             (1),    -- an input pin that is unused, not mapped
    ioUsedAsOutput          (2),    -- an I/O pin that is being used as an output
(SIU or BIU)
    logicInput              (3),    -- an input that is used only by the
programmable logic
    addressBit              (4),
    alarmInput              (5),
    alternateSequence       (6),
    autoFlashRequest        (7),
    cabinetDoorOpen         (8),
    callToNonActuated       (9),
    clockUpdate             (10),
    conflictMonitorStatus   (11),
    cycleAdvance            (12),
    dimmingEnable           (13),
    externalStart           (14),
    forceOffRing            (15),
    freeRequest             (16),
    hardwareControl         (17),
    indicatorLampControl    (18),
    inhibitMaxRing          (19),
    intervalAdvance         (20),
    localFlashSense         (21),
    manualControlEnable     (22),
    max2Ring                (23),
    max3AllRings            (24),
    max4AllRings            (25),
    maxRecall               (26),
    maxWalk                 (27),
```

```
    minRecall               (28),
    mmuCmuFlashSense        (29),
    modeSelectBit           (30),
    offsetInput             (31),
    omitRedClearRing        (32),
    patternSelect           (33),
    pedestrianDetector      (34),
    pedestrianOmit          (35),
    pedestrianRecycleRing   (36),
    phaseHold               (37),
    phaseOmit               (38),
    preemptGateDown         (39),
    preemptGateUp           (40),
    preemptHealthy          (41),
    preemptInput            (42),
    preemptInputAdvanced    (43),
    priorityCheckout        (44),
    priorityRequest         (45),
    redRestRing             (46),
    specialFunctionInput    (47),
    stopTimeAllRings        (48),
    stopTimeRing            (49),
    tbcOnline               (50),
    testInput               (51),
    timingPlanInput         (52),
    vehicleDetector         (53),
    vehicleDetectorFault    (54),
    walkRestModifier        (55) }
```

5.13.9.1 I/O Map Maximum Input Functions

```
ascIOmapMaxInputFunctions OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of I/O
mapping input functions this ASC supports. This object indicates the number of
rows in the ascIOmapInputFuncTable.
      The value of this object is equal to the number of AscIOinputType
enumerations.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.9.1
    <Object Type> S"
    ::= { ascIOmapInputFunctions 1 }
```


5.13.9.2 I/O Map Input Functions Table

```
ascIOmapInputFuncTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AscIOmapInputFuncEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table referencing the ASC I/O mapping input
functions. These are functions that may be assigned to the ASC inputs.
      The number of rows in this table is equal to ascIOmapMaxInputFunctions. The
entries in this table correspond to the values of the AscIOinputType enumeration.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.9.2"
    ::= { ascIOmapInputFunctions 2 }

ascIOmapInputFuncEntry OBJECT-TYPE
    SYNTAX AscIOmapInputFuncEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a conceptual row in the
ascIOmapInputFuncTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.9.2.1"
    INDEX { ascIOinputIndex }
    ::= { ascIOmapInputFuncTable 1 }

AscIOmapInputFuncEntry ::= SEQUENCE {
    ascIOinputIndex          Integer32,
    ascIOinputMaxFuncIndex   Integer32,
    ascIOinputFunctionName   SnmpAdminString }
```

5.13.9.2.1 I/O Map Input Functions Table Index

```
ascIOinputIndex OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Indicates the row number of the entry in the
ascIOmapInputFuncTable. The value of this object shall not exceed the
ascIOmapMaxInputFunctions value.
      These indexes correspond to the values of the AscIOinputType enumeration.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.9.2.1.1
    <Object Type> S"
    ::= { ascIOmapInputFuncEntry 1 }
```

5.13.9.2.2 I/O Map Input Maximum Index

```
ascIOinputMaxFuncIndex OBJECT-TYPE
```

```
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Some input functions support an array of inputs.
This object indicates the maximum array index for this input function.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.9.2.1.2
    <Object Type> S"
    ::= { ascIOmapInputFuncEntry 2 }
```

5.13.9.2.3 I/O Map Input Function Name

```
ascIOinputFunctionName OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains an ASCII string describing the
input function.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.9.2.1.3
    <Object Type> S"
    ::= { ascIOmapInputFuncEntry 3 }
```

5.13.10 I/O Map Output Functions

```
ascIOmapOutputFunctions OBJECT IDENTIFIER
    ::= { ascIOmapping 10 }

-- Enumeration for I/O mapping output functions
AscIOoutputType ::= INTEGER {
    unusedOutput            (1),    -- an output pin that is unused, not mapped
    ioUsedAsInput           (2),    -- an I/O pin that is being used as an input
(SIU or BIU)
    logicOutput             (3),    -- an output function from the programmable
logic
    advWarnGrn              (4),
    advWarnRed              (5),
    alarmOutput             (6),
    automaticFlashStatus    (7),
    channelGreen            (8),
    channelRed              (9),
    channelYellow           (10),
    codedStatusBitA         (11),
    codedStatusBitB         (12),
    codedStatusBitC         (13),
    detectorResetSlots      (14),
    detectorReset           (15),
    faultMonitor            (16),
```

```
    flashingLogic           (17),
    freeStatus              (18),
    offsetOutput            (19),
    phaseCheck              (20),
    phaseNext               (21),
    phaseOn                 (22),
    preemptActive           (23),
    preemptActiveAdvanced   (24),
    specialFunctionOutput   (25),
    tbcAuxOutput            (26),
    timingPlanOutput        (27),
    voltageMonitor          (28),
    watchdog                (29) }
```

5.13.10.1 I/O Map Maximum Output Functions

```
ascIOmapMaxOutputFunctions OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of I/O
mapping output functions this ASC supports. This object indicates the number of
rows in the ascIOmapOutputFuncTable.
      The value of this object is equal to the number of AscIOoutputType
enumerations.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.10.1
    <Object Type> S"
    ::= { ascIOmapOutputFunctions 1 }
```

5.13.10.2 I/O Map Output Functions Table

```
ascIOmapOutputFuncTable OBJECT-TYPE
 SYNTAX SEQUENCE OF AscIOmapOutputFuncEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "<Definition> A table referencing the ASC I/O mapping output
functions. These are functions that may be assigned         to the ASC outputs.
      The number of rows in this table is equal to ascIOmapMaxOutputFunctions.
      The entries in this table correspond to the values of the AscIOoutputType
enumeration.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.10.2"
    ::= { ascIOmapOutputFunctions 2 }

ascIOmapOutputFuncEntry OBJECT-TYPE
    SYNTAX AscIOmapOutputFuncEntry
```

```
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object defines a conceptual row in the
ascIOmapOutputFuncTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.10.2.1"
    INDEX { ascIOoutputIndex }
    ::= { ascIOmapOutputFuncTable 1 }

AscIOmapOutputFuncEntry ::= SEQUENCE {
    ascIOoutputIndex          Integer32,
    ascIOoutputMaxFuncIndex   Integer32,
    ascIOoutputFunctionName   SnmpAdminString }
```

5.13.10.2.1 I/O Map Output Functions Table Index

```
ascIOoutputIndex OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Indicates the row number of the entry in the
ascIOmapOutputFuncTable. The value of this object shall not exceed the
ascIOmapMaxOutputFunctions value. These indexes correspond to the values of the
AscIOoutputType enumeration.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.10.2.1.1
    <Object Type> S"
    ::= { ascIOmapOutputFuncEntry 1 }
```

5.13.10.2.2 I/O Map Output Function Maximum Index

```
ascIOoutputMaxFuncIndex OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Some output functions support an array of outputs.
This object indicates the maximum array index for this output function.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.10.2.1.2
    <Object Type> S"
    ::= { ascIOmapOutputFuncEntry 2 }
```

5.13.10.2.3 I/O Map Output Function Name

```
ascIOoutputFunctionName OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION "<Definition> This object contains an ASCII string describing the
output function.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.13.10.2.1.3
    <Object Type> S"
    ::= { ascIOmapOutputFuncEntry 3 }
```

5.13.11 I/O Map FIO Pins

```
ascIOmapFIO OBJECT IDENTIFIER ::= { ascIOmapping 11 }
```

5.13.11.1 I/O Map FIO Input Pins

```
AscIOmapFIOinputs ::= INTEGER { -- Enumeration for 2070-2A FIO (170) inputs
    pinC1-39 (1),   -- Detector 2
    pinC1-40 (2),   -- Detector 16
    pinC1-41 (3),   -- Detector 8
    pinC1-42 (4),   -- Detector 22
    pinC1-43 (5),   -- Detector 3
    pinC1-44 (6),   -- Detector 17
    pinC1-45 (7),   -- Detector 9
    pinC1-46 (8),   -- Detector 23
    pinC1-47 (9),   -- Detector 6
    pinC1-48 (10),  -- Detector 20
    pinC1-49 (11),  -- Detector 12
    pinC1-50 (12),  -- Detector 26
    pinC1-51 (13),  -- RR1 Preempt
    pinC1-52 (14),  -- RR2 Preempt
    pinC1-53 (15),  -- Manual Control
    pinC1-54 (16),  -- Spare
    pinC1-55 (17),  -- Detector 15
    pinC1-56 (18),  -- Detector 1
    pinC1-57 (19),  -- Detector 21
    pinC1-58 (20),  -- Detector 7
    pinC1-59 (21),  -- Detector 27
    pinC1-60 (22),  -- Detector 13
    pinC1-61 (23),  -- Detector 28
    pinC1-62 (24),  -- Detector 14
    pinC1-63 (25),  -- Detector 4
    pinC1-64 (26),  -- Detector 18
    pinC1-65 (27),  -- Detector 10
    pinC1-66 (28),  -- Detector 24
    pinC1-67 (29),  -- Ped Detector 1
    pinC1-68 (30),  -- Ped Detector 3
    pinC1-69 (31),  -- Ped Detector 2
    pinC1-70 (32),  -- Ped Detector 4
    pinC1-71 (33),  -- EVA Preempt
```

```
    pinC1-72 (34),  -- EVB Preempt
    pinC1-73 (35),  -- EVC Preempt
    pinC1-74 (36),  -- EVD Preempt
    pinC1-75 (37),  -- Spare
    pinC1-76 (38),  -- Detector 5
    pinC1-77 (39),  -- Detector 19
    pinC1-78 (40),  -- Detector 11
    pinC1-79 (41),  -- Detector 25
    pinC1-80 (42),  -- Interval Advance
    pinC1-81 (43),  -- Flash Sense
    pinC1-82 (44),  -- Stop Time
    pinC11-10 (45), -- C11 inputs all spare
    pinC11-11 (46),
    pinC11-12 (47),
    pinC11-13 (48),
    pinC11-15 (49),
    pinC11-16 (50),
    pinC11-17 (51),
    pinC11-18 (52),
    pinC11-19 (53),
    pinC11-20 (54),
    pinC11-21 (55),
    pinC11-22 (56),
    pinC11-23 (57),
    pinC11-24 (58),
    pinC11-25 (59),
    pinC11-26 (60),
    pinC11-27 (61),
    pinC11-28 (62),
    pinC11-29 (63),
    pinC11-30 (64) }
```

5.13.11.2 I/O Map FIO Output Pins

```
AscIOmapFIOoutputs ::= INTEGER {    -- Enumeration for 2070-2A FIO (170) outputs
    pinC1-2   (1),    -- Ped 4 red
    pinC1-3   (2),    -- Ped 4 green
    pinC1-4   (3),    -- Phase 4 red
    pinC1-5   (4),    -- Phase 4 yellow
    pinC1-6   (5),    -- Phase 4 green
    pinC1-7   (6),    -- Phase 3 red
    pinC1-8   (7),    -- Phase 3 yellow
    pinC1-9   (8),    -- Phase 3 green
    pinC1-10  (9),    -- Ped 2 red
    pinC1-11  (10),   -- Ped 2 green
    pinC1-12  (11),   -- Phase 2 red
```

```
pinC1-13  (12),  -- Phase 2 yellow
pinC1-15  (13),  -- Phase 2 green
pinC1-16  (14),  -- Phase 1 red
pinC1-17  (15),  -- Phase 1 yellow
pinC1-18  (16),  -- Phase 1 green
pinC1-19  (17),  -- Ped 8 red
pinC1-20  (18),  -- Ped 8 green
pinC1-21  (19),  -- Phase 8 red
pinC1-22  (20),  -- Phase 8 yellow
pinC1-23  (21),  -- Phase 8 green
pinC1-24  (22),  -- Phase 7 red
pinC1-25  (23),  -- Phase 7 yellow
pinC1-26  (24),  -- Phase 7 green
pinC1-27  (25),  -- Ped 6 red
pinC1-28  (26),  -- Ped 6 green
pinC1-29  (27),  -- Phase 6 red
pinC1-30  (28),  -- Phase 6 yellow
pinC1-31  (29),  -- Phase 6 green
pinC1-32  (30),  -- Phase 5 red
pinC1-33  (31),  -- Phase 5 yellow
pinC1-34  (32),  -- Phase 5 green
pinC1-35  (33),  -- Ped 2 yellow
pinC1-36  (34),  -- Ped 6 yellow
pinC1-37  (35),  -- Ped 4 yellow
pinC1-38  (36),  -- Ped 8 yellow
pinC1-83  (37),  -- Spare
pinC1-84  (38),  -- Spare
pinC1-85  (39),  -- Overlap D red
pinC1-86  (40),  -- Overlap D yellow
pinC1-87  (41),  -- Overlap D green
pinC1-88  (42),  -- Overlap C red
pinC1-89  (43),  -- Overlap C yellow
pinC1-90  (44),  -- Overlap C green
pinC1-91  (45),  -- Spare
pinC1-93  (46),  -- Spare
pinC1-94  (47),  -- Overlap B red
pinC1-95  (48),  -- Overlap B yellow
pinC1-96  (49),  -- Overlap B green
pinC1-97  (50),  -- Overlap A red
pinC1-98  (51),  -- Overlap A yellow
pinC1-99  (52),  -- Overlap A green
pinC1-100 (53),  -- Spare
pinC1-101 (54),  -- Flash status
pinC1-102 (55),  -- Detector reset
pinC1-103 (56),  -- Watchdog
```

```
    pinC11-1  (57),  -- all C11 outputs are spare
    pinC11-2  (58),
    pinC11-3  (59),
    pinC11-4  (60),
    pinC11-5  (61),
    pinC11-6  (62),
    pinC11-7  (63),
    pinC11-8  (64) }
```

5.13.12 I/O Map TS1 Pins

```
ascIOmapTS1 OBJECT IDENTIFIER ::= { ascIOmapping 12 }
```

5.13.12.1 I/O TS1 Input Pins

```
AscIOmapTS1inputs ::= INTEGER { -- Enumeration for type 2070-8 NEMA TS1 inputs
    pinA-f  (1),   -- Vehicle Detector 1
    pinA-g  (2),   -- Ped Detector 1
    pinA-h  (3),   -- Hold 1
    pinA-i  (4),   -- Force Off Ring 1
    pinA-j  (5),   -- Min Recall
    pinA-k  (6),   -- Manual Control
    pinA-m  (7),   -- CNA 1
    pinA-n  (8),   -- Test A
    pinA-q  (9),   -- Mode Bit A
    pinA-v  (10),  -- Ped Omit 2
    pinA-w  (11),  -- Omit Red Ring 1
    pinA-x  (12),  -- Red Rest Ring 1
    pinA-y  (13),  -- Mode Bit B
    pinA-z  (14),  -- CNA 2
    pinA-K  (15),  -- Vehicle Detector 2
    pinA-L  (16),  -- Ped Detector 2
    pinA-M  (17),  -- Hold 2
    pinA-N  (18),  -- Stop Time Ring 1
    pinA-P  (19),  -- Inhibit Max Ring 1
    pinA-R  (20),  -- External Start
    pinA-S  (21),  -- Interval Advance
    pinA-T  (22),  -- Lamp Control
    pinA-AA (23),  -- Test B
    pinA-BB (24),  -- Walk Rest Modifier
    pinA-EE (25),  -- Ped Omit 1
    pinA-FF (26),  -- Ped Recycle Ring 1
    pinA-GG (27),  -- Max 2 Ring 1
    pinA-HH (28),  -- Mode Bit C
    pinB-g  (29),  -- Phase Omit 4
    pinB-h  (30),  -- Hold 4
    pinB-i  (31),  -- Hold 3
```

```
pinB-j  (32),  -- Ped Omit 3
pinB-k  (33),  -- Ped Omit 6
pinB-m  (34),  -- Ped Omit 7
pinB-n  (35),  -- Ped Omit 8
pinB-v  (36),  -- Spare
pinB-x  (37),  -- Ped Omit 4
pinB-z  (38),  -- Max 2 Ring 2
pinB-B  (39),  -- Spare
pinB-L  (40),  -- Vehicle Detector 4
pinB-M  (41),  -- Ped Detector 4
pinB-N  (42),  -- Vehicle Detector 3
pinB-P  (43),  -- Ped Detector 3
pinB-R  (44),  -- Phase Omit 3
pinB-S  (45),  -- Phase Omit 2
pinB-T  (46),  -- Ped Omit 5
pinB-U  (47),  -- Phase Omit 1
pinB-V  (48),  -- Ped RECY R2
pinB-W  (49),  -- Spare
pinB-X  (50),  -- Spare
pinC-a  (51),  -- Inhibit Max Ring 2
pinC-b  (52),  -- Ttest C
pinC-m  (53),  -- Hold 5
pinC-n  (54),  -- Phase Omit 5
pinC-p  (55),  -- Hold 6
pinC-q  (56),  -- Phase Omit 6
pinC-r  (57),  -- Phase Omit 7
pinC-s  (58),  -- Phase Omit 8
pinC-t  (59),  -- Vehicle Detector 8
pinC-u  (60),  -- Red Rest Ring 2
pinC-v  (61),  -- Omit Red Ring 2
pinC-P  (62),  -- Vehicle Detector 5
pinC-R  (63),  -- Ped Detector 5
pinC-S  (64),  -- Vehicle Detector 6
pinC-T  (65),  -- Ped Detector 6
pinC-U  (66),  -- Ped Detector 7
pinC-V  (67),  -- Vehicle Detector 7
pinC-W  (68),  -- Ped Detector 8
pinC-X  (69),  -- Hold 8
pinC-Y  (70),  -- Force Off Ring 2
pinC-Z  (71),  -- Stop Time Ring 2
pinC-EE (72),  -- Hold 7
pinD-a  (73),  -- Spare
pinD-b  (74),  -- Alarm 1
pinD-c  (75),  -- Alarm 2
pinD-d  (76),  -- Alarm 3
```

```
pinD-e  (77),  -- Alarm 4
pinD-f  (78),  -- Alarm 5
pinD-g  (79),  -- Flash In
pinD-h  (80),  -- Conflict Monitor Status
pinD-i  (81),  -- Door Open
pinD-j  (82),  -- Special Function 1
pinD-k  (83),  -- Special Function 2
pinD-m  (84),  -- Special Function 3
pinD-n  (85),  -- Special Function 4
pinD-p  (86),  -- Special Function 5
pinD-q  (87),  -- Special Function 6
pinD-r  (88),  -- Special Function 7
pinD-s  (89),  -- Special Function 8
pinD-t  (90),  -- Preempt 1
pinD-u  (91),  -- Preempt 2
pinD-v  (92),  -- Preempt 3
pinD-w  (93),  -- Preempt 4
pinD-x  (94),  -- Preempt 5
pinD-y  (95),  -- Preempt 6
pinD-A  (96),  -- Vehicle Detector  9
pinD-B  (97),  -- Vehicle Detector 10
pinD-C  (98),  -- Vehicle Detector 11
pinD-D  (99),  -- Vehicle Detector 12
pinD-E  (100), -- Vehicle Detector 13
pinD-F  (101), -- Vehicle Detector 14
pinD-G  (102), -- Vehicle Detector 15
pinD-H  (103), -- Vehicle Detector 16
pinD-J  (104), -- Vehicle Detector 17
pinD-K  (105), -- Vehicle Detector 18
pinD-L  (106), -- Vehicle Detector 19
pinD-M  (107), -- Vehicle Detector 20
pinD-N  (108), -- Vehicle Detector 21
pinD-P  (109), -- Vehicle Detector 22
pinD-R  (110), -- Vehicle Detector 23
pinD-S  (111), -- Vehicle Detector 24
pinD-T  (112), -- Clock Update
pinD-U  (113), -- Hardware Control
pinD-V  (114), -- Cycle Advance
pinD-W  (115), -- Max 3 Select
pinD-X  (116), -- Max 4 Select
pinD-Y  (117), -- Free
pinD-Z  (118), -- Spare
pinD-KK (119), -- Spare
pinD-MM (120) } -- Spare
```

5.13.12.2 I/O Map TS1 Output Pins

```
AscIOmapTS1outputs ::= INTEGER {    -- Enumeration for 2070-8 NEMA TS1 outputs
    pinA-a  (1),   -- Ped 1 Yellow
    pinA-b  (2),   -- Phase 2 Yellow
    pinA-c  (3),   -- Phase 2 Green
    pinA-d  (4),   -- Phase Check 2
    pinA-e  (5),   -- Phase On 2
    pinA-r  (6),   -- Status B Ring 1
    pinA-s  (7),   -- Phase 1 Green
    pinA-t  (8),   -- Ped 1 Green
    pinA-u  (9),   -- Phase Check 1
    pinA-A  (10),  -- Fault Monitor
    pinA-C  (11),  -- Voltage Monitor
    pinA-D  (12),  -- Phase 1 Red
    pinA-E  (13),  -- Ped 1 Red
    pinA-F  (14),  -- Phase 2 Red
    pinA-G  (15),  -- Ped 2 Red
    pinA-H  (16),  -- Ped 2 Yellow
    pinA-J  (17),  -- Ped 2 Green
    pinA-X  (18),  -- Flashing Logic
    pinA-Y  (19),  -- Status C Ring 1
    pinA-Z  (20),  -- Phase 1 Yellow
    pinA-CC (21),  -- Status A Ring 1
    pinA-DD (22),  -- Phase ON 1
    pinB-a  (23),  -- Ped 3 Red
    pinB-b  (24),  -- Phase 4 Green
    pinB-c  (25),  -- Phase 4 Yellow
    pinB-d  (26),  -- Ped 4 Green
    pinB-e  (27),  -- Phase On 4
    pinB-f  (28),  -- Phase Next 4
    pinB-p  (29),  -- Overlap A Yellow
    pinB-q  (30),  -- Overlap A Red
    pinB-r  (31),  -- Phase Check 3
    pinB-s  (32),  -- Phase On 3
    pinB-t  (33),  -- Phase Next 3
    pinB-u  (34),  -- Overlap D Red
    pinB-w  (35),  -- Overlap D Green
    pinB-A  (36),  -- Phase Next 1
    pinB-C  (37),  -- Phase Next 2
    pinB-D  (38),  -- Phase 3 Green
    pinB-E  (39),  -- Phase 3 Yellow
    pinB-F  (40),  -- Phase 3 Red
    pinB-G  (41),  -- Phase 4 Red
    pinB-H  (42),  -- Ped 4 Yellow
    pinB-J  (43),  -- Ped 4 Red
```

```
pinB-K  (44),  -- Phase Check 4
pinB-Y  (45),  -- Ped 3 Green
pinB-Z  (46),  -- Ped 3 Yellow
pinB-AA (47),  -- Overlap A Green
pinB-BB (48),  -- Overlap B Yellow
pinB-CC (49),  -- Overlap B Red
pinB-DD (50),  -- Overlap C Red
pinB-EE (51),  -- Overlap D Yellow
pinB-FF (52),  -- Overlap C Green
pinB-GG (53),  -- Overlap B Green
pinB-HH (54),  -- Overlap C Yellow
pinC-c  (55),  -- Status C Ring 2
pinC-d  (56),  -- Ped 8 Green
pinC-e  (57),  -- Phase 8 Yellow
pinC-f  (58),  -- Phase 7 Green
pinC-g  (59),  -- Phase 6 Green
pinC-h  (60),  -- Phase 6 Yellow
pinC-i  (61),  -- Phase 5 Green
pinC-j  (62),  -- Ped 5 Green
pinC-k  (63),  -- Phase Check 5
pinC-w  (64),  -- Ped 8 Yellow
pinC-x  (65),  -- Phase 8  Green
pinC-y  (66),  -- Ped 7 Red
pinC-z  (67),  -- Ped 6 Red
pinC-A  (68),  -- Status A Ring 2
pinC-B  (69),  -- Status B Ring 2
pinC-C  (70),  -- Ped 8 Red
pinC-D  (71),  -- Phase 8 Red
pinC-E  (72),  -- Phase 7 Yellow
pinC-F  (73),  -- Phase 7 Red
pinC-G  (74),  -- Phase 6 Red
pinC-H  (75),  -- Phase 5 Red
pinC-J  (76),  -- Phase 5 Yellow
pinC-K  (77),  -- Ped 5 Yellow
pinC-L  (78),  -- Ped 5 Red
pinC-M  (79),  -- Phase Next 5
pinC-N  (80),  -- Phase On 5
pinC-AA (81),  -- Ped 6 Yellow
pinC-BB (82),  -- Phase Check 6
pinC-CC (83),  -- Phase ON 6
pinC-DD (84),  -- Phase Next 6
pinC-FF (85),  -- Phase Check 8
pinC-GG (86),  -- Phase On 8
pinC-HH (87),  -- Phase Next 8
pinC-JJ (88),  -- Ped 7 Green
```

```
    pinC-KK (89),  -- Ped 7 Yellow
    pinC-LL (90),  -- Ped 6 Green
    pinC-MM (91),  -- Phase Check 7
    pinC-NN (92),  -- Phase On 7
    pinC-PP (93),  -- Phase Next 7
    pinD-z  (94),  -- Alarm 1
    pinD-AA (95),  -- Alarm 2
    pinD-BB (96),  -- Special Function 1
    pinD-CC (97),  -- Special Function 2
    pinD-DD (98),  -- Special Function 3
    pinD-EE (99),  -- Special Function 4
    pinD-FF (100), -- Special Function 5
    pinD-GG (101), -- Special Function 6
    pinD-HH (102), -- Special Function 7
    pinD-JJ (103), -- Special Function 8
    pinD-LL (104) } -- Detector Reset
```

5.13.13 I/O Map TS2 BIU Pins

```
ascIOmapBIU OBJECT IDENTIFIER ::= { ascIOmapping 13 }
```

5.13.13.1 I/O Map TS2 BIU Input Pins

```
AscIOmapBIUinputs ::= INTEGER { -- Enumeration for NEMA TS2 BIU inputs (for each
BIU)
    biuInputIO1   (1),
    biuInputIO2   (2),
    biuInputIO3   (3),
    biuInputIO4   (4),
    biuInputIO5   (5),
    biuInputIO6   (6),
    biuInputIO7   (7),
    biuInputIO8   (8),
    biuInputIO9   (9),
    biuInputIO10  (10),
    biuInputIO11  (11),
    biuInputIO12  (12),
    biuInputIO13  (13),
    biuInputIO14  (14),
    biuInputIO15  (15),
    biuInputIO16  (16),
    biuInputIO17  (17),
    biuInputIO18  (18),
    biuInputIO19  (19),
    biuInputIO20  (20),
    biuInputIO21  (21),
```

```
    biuInputIO22  (22),
    biuInputIO23  (23),
    biuInputIO24  (24),
    biuInputIN1   (25),
    biuInputIN2   (26),
    biuInputIN3   (27),
    biuInputIN4   (28),
    biuInputIN5   (29),
    biuInputIN6   (30),
    biuInputIN7   (31),
    biuInputIN8   (32),
    biuInputOPT1  (33),
    biuInputOPT2  (34),
    biuInputOPT3  (35),
    biuInputOPT4  (36) }
```

5.13.13.2 I/O TS2 BIU Output Pins

```
AscIOmapBIUoutputs ::= INTEGER {     -- Enumeration for NEMA TS2 BIU outputs (for
each BIU)
    biuOutputO1   (1),
    biuOutputO2   (2),
    biuOutputO3   (3),
    biuOutputO4   (4),
    biuOutputO5   (5),
    biuOutputO6   (6),
    biuOutputO7   (7),
    biuOutputO8   (8),
    biuOutputO9   (9),
    biuOutputO10  (10),
    biuOutputO11  (11),
    biuOutputO12  (12),
    biuOutputO13  (13),
    biuOutputO14  (14),
    biuOutputO15  (15),
    biuOutputIO1  (16),
    biuOutputIO2  (17),
    biuOutputIO3  (18),
    biuOutputIO4  (19),
    biuOutputIO5  (20),
    biuOutputIO6  (21),
    biuOutputIO7  (22),
    biuOutputIO8  (23),
    biuOutputIO9  (24),
    biuOutputIO10 (25),
    biuOutputIO11 (26),
```



```
    biuOutputIO12 (27),
    biuOutputIO13 (28),
    biuOutputIO14 (29),
    biuOutputIO15 (30),
    biuOutputIO16 (31),
    biuOutputIO17 (32),
    biuOutputIO18 (33),
    biuOutputIO19 (34),
    biuOutputIO20 (35),
    biuOutputIO21 (36),
    biuOutputIO22 (37),
    biuOutputIO23 (38),
    biuOutputIO24 (39) }
```

5.13.14 I/O Map ATC Cabinet SIU Pins

```
ascIOmapSIU OBJECT IDENTIFIER ::= { ascIOmapping 14 }
```

5.13.14.1 I/O Map ATC

```
AscIOmapSIUinputs ::= INTEGER { -- Enumeration for ATC Cabinet SIU inputs (for
each SIU)
    siuInputIO0  (1),
    siuInputIO1  (2),
    siuInputIO2  (3),
    siuInputIO3  (4),
    siuInputIO4  (5),
    siuInputIO5  (6),
    siuInputIO6  (7),
    siuInputIO7  (8),
    siuInputIO8  (9),
    siuInputIO9  (10),
    siuInputIO10 (11),
    siuInputIO11 (12),
    siuInputIO12 (13),
    siuInputIO13 (14),
    siuInputIO14 (15),
    siuInputIO15 (16),
    siuInputIO16 (17),
    siuInputIO17 (18),
    siuInputIO18 (19),
    siuInputIO19 (20),
    siuInputIO20 (21),
    siuInputIO21 (22),
    siuInputIO22 (23),
    siuInputIO23 (24),
```

```
    siuInputIO24 (25),
    siuInputIO25 (26),
    siuInputIO26 (27),
    siuInputIO27 (28),
    siuInputIO28 (29),
    siuInputIO29 (30),
    siuInputIO30 (31),
    siuInputIO31 (32),
    siuInputIO32 (33),
    siuInputIO33 (34),
    siuInputIO34 (35),
    siuInputIO35 (36),
    siuInputIO36 (37),
    siuInputIO37 (38),
    siuInputIO38 (39),
    siuInputIO39 (40),
    siuInputIO40 (41),
    siuInputIO41 (42),
    siuInputIO42 (43),
    siuInputIO43 (44),
    siuInputIO44 (45),
    siuInputIO45 (46),
    siuInputIO46 (47),
    siuInputIO47 (48),
    siuInputIO48 (49),
    siuInputIO49 (50),
    siuInputIO50 (51),
    siuInputIO51 (52),
    siuInputIO52 (53),
    siuInputIO53 (54),
    siuInputOPT1 (55),
    siuInputOPT2 (56),
    siuInputOPT3 (57),
    siuInputOPT4 (58) }
```

5.13.14.2 I/O Map ATC Cabinet SIU Output Pins

```
AscIOmapSIUoutputs ::= INTEGER {     -- Enumeration for ATC Cabinet SIU outputs
(for each SIU)
    siuOutputIO0  (1),
    siuOutputIO1  (2),
    siuOutputIO2  (3),
    siuOutputIO3  (4),
    siuOutputIO4  (5),
    siuOutputIO5  (6),
```



```
siuOutputIO6  (7),
siuOutputIO7  (8),
siuOutputIO8  (9),
siuOutputIO9  (10),
siuOutputIO10 (11),
siuOutputIO11 (12),
siuOutputIO12 (13),
siuOutputIO13 (14),
siuOutputIO14 (15),
siuOutputIO15 (16),
siuOutputIO16 (17),
siuOutputIO17 (18),
siuOutputIO18 (19),
siuOutputIO19 (20),
siuOutputIO20 (21),
siuOutputIO21 (22),
siuOutputIO22 (23),
siuOutputIO23 (24),
siuOutputIO24 (25),
siuOutputIO25 (26),
siuOutputIO26 (27),
siuOutputIO27 (28),
siuOutputIO28 (29),
siuOutputIO29 (30),
siuOutputIO30 (31),
siuOutputIO31 (32),
siuOutputIO32 (33),
siuOutputIO33 (34),
siuOutputIO34 (35),
siuOutputIO35 (36),
siuOutputIO36 (37),
siuOutputIO37 (38),
siuOutputIO38 (39),
siuOutputIO39 (40),
siuOutputIO40 (41),
siuOutputIO41 (42),
siuOutputIO42 (43),
siuOutputIO43 (44),
siuOutputIO44 (45),
siuOutputIO45 (46),
siuOutputIO46 (47),
siuOutputIO47 (48),
siuOutputIO48 (49),
siuOutputIO49 (50),
siuOutputIO50 (51),
```

```
    siuOutputIO51 (52),
    siuOutputIO52 (53),
    siuOutputIO53 (54) }
```

5.13.15 I/O Map Auxiliary Device Pins

```
ascIOmapAUX OBJECT IDENTIFIER ::= { ascIOmapping 15 }
```

5.13.15.1 I/O Map Auxiliary Device Input Pins

```
AscIOmapAUXinputs ::= INTEGER { -- Enumeration for AUX inputs front panel AUX
switch
    auxInputFPSwitch (1),
    auxInputIO0      (2),
    auxInputIO1      (3),
    auxInputIO2      (4),
    auxInputIO3      (5),
    auxInputIO4      (6),
    auxInputIO5      (7),
    auxInputIO6      (8),
    auxInputIO7      (9) }
```

5.13.15.2 I/O Auxiliary Device Outputs

```
AscIOmapAUXoutputs ::= INTEGER {    -- Enumeration for AUX outputs
    auxOutputIO0  (1),
    auxOutputIO1  (2),
    auxOutputIO2  (3),
    auxOutputIO3  (4),
    auxOutputIO4  (5),
    auxOutputIO5  (6),
    auxOutputIO6  (7),
    auxOutputIO7  (8) }
```

5.14 SIU Port 1 Parameters

```
siuport1  OBJECT IDENTIFIER

    ::= { asc 14 }

-- This object is an identifier used to group all objects for support of ITS
Cabinet V1 Section 4.7.14.7.1 SIU Port 1 Operation.
```

5.14.1 Maximum SIU Port 1 Addresses

```
maxSIUPort1Addresses OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "address"
```



```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The Maximum Number of SIU Port 1 addresses this
Actuated Controller Unit supports. This object indicates the maximum rows which
shall appear in the siuPort1Table object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.14.1
    <Object Type> S"
    ::= { siuport1 1 }
```

5.14.2 SIU Port 1 Table

```
siuport1Table OBJECT-TYPE
    SYNTAX SEQUENCE OF SIUPort1Entry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Actuated Controller Unit port 1
parameters. The number of rows in this table is equal to maxSIUPort1Addresses
object. Address 255 is reserved for the Broadcast All address.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.14.2"
    ::= { siuport1 2 }

siuport1Entry OBJECT-TYPE
    SYNTAX SIUPort1Entry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
    "<Definition> This object defines a conceptual row in the SIU Port 1 Table.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.14.2.1"
    INDEX { siuport1Number }
    ::= { siuport1Table 1 }

SIUPort1Entry ::= SEQUENCE {
    siuport1Number        Integer32,
    siuport1DevicePresent Integer32,
    siuport1Status        INTEGER}
```

5.14.2.1 SIU Port 1 Number

```
siuport1Number OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "address"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The (SIU Port 1 address plus one) for objects in
this row. This value shall not exceed the maxSIUPort1Addresses object value.
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.14.2.1.1
    <Object Type> S"
    ::= { siuport1Entry 1 }
```

5.14.2.2 SIU Port 1 Device Present

```
siuport1DevicePresent OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to program the CU as to the
presence or absence of a device for this SIU Port 1 address. The CU shall
transmit Command Frames only to those devices that are present as determined by
this programming.
True (one) - the device is present.
False (zero) - the device is not present.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.14.2.1.2
    <Object Type> C"
    REFERENCE "ATC 5301 Section 9"
    ::= { siuport1Entry 2 }
```

5.14.2.3 SIU Port 1 Status

```
siuport1Status OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     online (2),
                     responseFault (3)}
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the communications status
with the associated device:
other: This indicates that some other communications fault has been detected.
online: This indicates that at least five of the most recent 10 response
transfers were received correctly.
responseFault: This indicates that more than 5 of the most recent 10 response
transfers were received incorrectly.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.14.2.1.3
    <Object Type> S"
    ::= { siuport1Entry 3 }
```

5.15 RSU Interface

```
ascRsuPort OBJECT IDENTIFIER
    ::= { asc 15 }

-- This defines a node to configure communications for a connected vehicle
environment.
```

5.15.1 RSU Interface Port

```
rsuCommPort OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the ifIndex for a row in the
ifTable and the commPortTable that identifies a communications port on the ASC
that is used to exchange data with an RSU. A value of zero indicates that there
is no RSU port or RSU communications is disabled. The value shall not exceed the
object maxCommPorts value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.1
    <Object Type> P"
    ::= { ascRsuPort 1 }
```

5.15.2 Maximum Number of RSU Ports

```
maxRsuPorts OBJECT-TYPE
    SYNTAX Integer32 (0..16)
    UNITS "port"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of logical
RSU Ports this ASC supports. This object indicates the maximum rows which shall
appear in the rsuPortTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.2
    <Object Type> S"
    ::= { ascRsuPort 2 }
```

5.15.3 Logical RSU Ports Table

```
rsuPortTable OBJECT-TYPE
    SYNTAX SEQUENCE OF RsuPortEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table contains configuration and status
information for logical ports to communicate with RSUs. The number of rows in
this table is equal to the maxRsuPorts object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3"
    ::= { ascRsuPort 3 }

rsuPortEntry OBJECT-TYPE
    SYNTAX RsuPortEntry
    MAX-ACCESS not-accessible
    STATUS current
```

```
    DESCRIPTION "<Definition> The configuration and status of a logical RSU port.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3.1"
    INDEX  { rsuPortIndex }
    ::= { rsuPortTable 1 }

RsuPortEntry ::= SEQUENCE {
    rsuPortIndex         Integer32,
    rsuPortName          SnmpAdminString,
    rsuPortPollingPeriod Integer32,
    rsuPortWatchdogTime  Integer32,
    rsuPortWatchdogTimer Integer32,
    rsuPortNumber        Integer32,
    rsuPortAddress       IpAddress }
```

5.15.3.1 RSU Port Index

```
rsuPortIndex OBJECT-TYPE
    SYNTAX Integer32 (1..16)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This object provides the index into the RSU port
table. This value shall not exceed the maxRsuPorts object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3.1.1
    <Object Type> S"
    ::= { rsuPortEntry 1 }
```

5.15.3.2 RSU Port Name

```
rsuPortName OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A textual string describing the RSU or the location
of the RSU.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3.1.3
    <Object Type> P"
    ::= { rsuPortEntry 3 }
```

5.15.3.3 RSU Interface Polling Period

```
rsuPortPollingPeriod OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "millisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the interval, in milliseconds,
between polls on the port identified by this row. This object assumes that the CU
```

```
behaves as the manager when exchanging data with an RSU. A value of zero
indicates that polling is disabled. This object is not used if the CU is the SNMP
agent between the ASC - RSU interface.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3.1.4
    <Object Type> P"
    ::= { rsuPortEntry 4 }
```

5.15.3.4 RSU Port Watchdog Time

```
rsuPortWatchdogTime OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "millisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the maximum time duration, in
milliseconds, allowable without activity on the port identified by this row. If
the amount of time that activity was last detected exceeds this value, then a RSU
watchdog no activity fault alarm is SET. A value of zero disables the RSU
Watchdog no activity fault alarm. This object is not used if rsuPortPollingPeriod
is zero.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3.1.5
    <Object Type> P"
    ::= { rsuPortEntry 5 }
```

5.15.3.5 RSU Port Watchdog Timer

```
rsuPortWatchdogTimer OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "millisecond"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object defines the watchdog time, in
milliseconds, which is the amount of time since activity was last detected on the
port identified by this row (rsuPortIndex). Activity is defined as any valid data
object received on the port. If this object exceeds rsuPortWatchdogTime, then a
RSU watchdog no activity fault is reported. This object is not used if
rsuPortPollingPeriod is 0.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3.1.6
    <Object Type> S"
    ::= { rsuPortEntry 6 }
```

5.15.3.6 RSU Port Number

```
rsuPortNumber OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION "<Definition> This object defines the port number on the RSU for
the ASC to send data to. An object value of 0 indicates communication to this
port is disabled.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3.1.7
    <Object Type> P"
    DEFVAL { 0 }
    ::= { rsuPortEntry 7 }
```

5.15.3.7 RSU Port Address

```
rsuPortAddress OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the IP address of the RSU for
this port.

    NTCIP 1103 was previously used to identify the IP address in NTCIP 1202 v03.

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.15.3.1.8
    <Object Type> P"
    ::= { rsuPortEntry 8 }
```

5.16 ASC SPaT

```
ascSpat OBJECT IDENTIFIER
    ::= { asc 16 }

-- This defines a node to support signal phase and timing objects for a connected
vehicle environment.
```

5.16.1 SPaT Data Timestamp

```
spatTimestamp OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0 | 5))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> A 5-byte value representing the time of day the
SPAT data is generated by the CU. The 5 bytes are:
    Byte 1: hours (0..23)
    Byte 2: minutes (0..59)
    Byte 3: seconds (0..60). 60 is used to support leap seconds.
    Byte 4-5: milliseconds (0..999)
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.1
    <Object Type> S"
    DEFVAL { "" }
    ::= { ascSpat 1 }
```

5.16.2 SPaT Enabled Lanes Concurrency Table

```
spatEnabledLanesConcurrencyTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SpatEnabledLanesConcurrencyEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing the revocable lanes that are
allowed to be concurrently ACTIVE. The number of rows in this table is equal to
the value of maxEnabledLanesConcurrency.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.3"
    ::= { ascSpat 3 }

spatEnabledLanesConcurrencyEntry OBJECT-TYPE
    SYNTAX SpatEnabledLanesConcurrencyEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The allowable lanes that may be ACTIVE
concurrently.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.3.1"
    INDEX  { enabledLaneIndex }
    ::= { spatEnabledLanesConcurrencyTable 1 }

SpatEnabledLanesConcurrencyEntry::= SEQUENCE {
    enabledLaneIndex        Integer32,
    enabledLaneConcurrency  OCTET STRING }
```

5.16.2.1 Enabled Lane Index

```
enabledLaneIndex OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The number for objects in this row. This value
shall not exceed the maxEnabledLanesConcurrency object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.3.1.1"
    ::= { spatEnabledLanesConcurrencyEntry 1 }
```

5.16.2.2 Enabled Lane Concurrency

```
enabledLaneConcurrency OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet contains a revocable lane identifier
that may be ACTIVE concurrently. The value could be a single octet. This object
```

```
indicates a set of revocable lanes that may be simultaneously enabled. Revocable
lanes are identified in the MAP message for the intersection.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.3.1.2
    <Object Type> P"
    REFERENCE "SAE J2735 DF_EnabledLaneList"
    ::= { spatEnabledLanesConcurrencyEntry 2 }
```

5.16.3 SPaT Message Options

```
spatOptions OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Optional SPAT functions ( 0 = False/Disabled, 1 =
True/Enabled)
    Bit 7: Reserved.
    Bit 6: Reserved.
    Bit 5: Reserved.
    Bit 4: Reserved.
    Bit 3: Reserved.
    Bit 2: MAP Validity. Indicates if the MAP data message broadcasted by the RSU
is valid (1) or invalid (0). When this bit is one, the MAP Message broadcasted by
the CV Application Process is considered valid. When this bit is 0, then the MAP
message broadcasted by the RSU is considered invalid and should not be used
except for test purposes.
    Bit 1: SPaT Validity. Indicates if the SPaT data for this CU is valid (1) or
invalid (0). When this bit is one, the SPaT data provided by the controller is
considered valid. When this bit is 0, then the SPaT data provided by the
controller is considered invalid and should not be used except for test
purposes.
    Bit 0: Enabled SPAT. Provides a means to enable the CU to provide SPAT data
to a management station or a RSU.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.4
    <Object Type> P"
    ::= { ascSpat 4 }
```

5.16.4 SPaT RSU Ports Table

```
spatPortTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SpatPortEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table contains configuration and status
information for exchanging SPAT information with RSUs. The number of rows in this
table is equal to the maxRsuPorts object.
    <TableType> static
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.5"
    ::= { ascSpat 5 }

spatPortEntry OBJECT-TYPE
    SYNTAX SpatPortEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The configuration and status to exchange SPAT data
with a logical RSU port.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.5.1"
    INDEX  { rsuPortIndex }
    ::= { spatPortTable 1 }

SpatPortEntry ::= SEQUENCE {
    spatPortOptions           Integer32,
    spatPortStatus            INTEGER,
    spatPortMapActivationCode MapActivationCode }
```

5.16.4.1 SPaT Port Options

```
spatPortOptions OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Optional SPAT functions for this SPAT port(0 =
False/Disabled, 1 = True/Enabled)
    Bit 7: Reserved
    Bit 6: Reserved
    Bit 5: Reserved
    Bit 4: Reserved
    Bit 3: Reserved
    Bit 2: MAP Validity. Indicates if the MAP data message broadcasted by the RSU
on this RSU port is valid (1) or invalid (0). When this bit is one, the MAP
Message broadcasted by the CV Application Process is considered valid. When this
bit is 0, then the MAP message broadcasted by the RSU is considered invalid and
should not be used except for test purposes.
    Bit 1: SPAT Validity. Indicates if the SPaT data on this RSU port is valid
(1) or invalid (0). When this bit is one, the SPaT data provided by the
controller is considered valid. When this bit is 0, then the SPaT data provided
by the controller is considered invalid and should not be used except for test
purposes.
    Bit 0: Enabled SPAT. Provides a means to enable the CU to exchange SPAT data
on this RSU port.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.5.1.1
    <Object Type> P"
    ::= { spatPortEntry 1 }
```

5.16.4.2 SPaT Port Status

```
spatPortStatus OBJECT-TYPE
    SYNTAX INTEGER { other(1),
                     disabled(2),
                     normal(3),
                     mapError(4),
                     enabledLanesError(5) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The status for exchanging SPAT data for this RSU
port.
    other: A status not defined by this standard.
    disabled: The capability to provide SPAT data is disabled.
    normal: The CU is providing SPAT data.
    mapError: The CU has stopped providing SPAT data because of an inconsistency
in the spatPortMapActivationCode.
    enabledLanesError: The CU has stopped providing SPAT data because of an
inconsistency error with the SPAT enabled lanes concurrency table
(spatEnabledLanesConcurrencyTable). Note: Bit 13 in the spatStatus2 is also
enabled.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.5.1.2
    <Object Type> S"
    ::= { spatPortEntry 2 }
```

5.16.4.3 SPaT Port MAP Activation Code

```
spatPortMapActivationCode OBJECT-TYPE
    SYNTAX MapActivationCode
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Represents the current mapActivationCode for this
RSU port that the signal patterns on the CU are programmed for. The
spatPortMapActivationCode value in this row should match the mapActivatePlan
value for the RSU on this RSU port.
This object allows the CU to compare and confirm that the signal phase and timing
data matches the MAP data that may be broadcast by the RSU.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.5.1.3
    <Object Type> P"
    ::= { spatPortEntry 3 }
```

5.16.5 Current Tick Counter

```
ascCurrentTick2 OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "decisecond"
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object is a counter of the number of elapsed
ASC traffic processing ticks. The value is incremented by 1 (decisecond) every
time the ASC completes its tenth of a second processing and commits a new set of
IO signal states.

The object is represented as a rolling counter with a zero representing the top
of the hour. If the counter were to be incremented beyond 35999, unless a leap
seconds occur, the value shall roll over back to 0. The values 36000..36009 are
used when a leap second occurs. The values 36010..36110 are reserved for future
use. 36111 is to be used when the value is undefined or unknown.
See Annex F.3.3.1 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.8
    <Object Type> S"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { ascSpat 8 }
```

5.16.6 Maximum Signal Groups

```
maxSignalGroups OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "signalGroupID"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the maximum rows which shall
appear in the signalGroupEntryTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.9
    <Object Type> S"
    ::= { ascSpat 9 }
```

5.16.7 Signal Group Entry Table

```
signalGroupEntryTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SignalGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table contains configuration information for a
signalGroupID at a signalized intersection. The number of rows in this table is
equal to the maxSignalGroups object. See Annex F.3.3.5.1 for additional guidance.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10"
    ::= { ascSpat 10 }

signalGroupEntry OBJECT-TYPE
```

```
    SYNTAX SignalGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for specific signal groups at an
intersection.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1"
    INDEX  { signalGroupEntryNumber }
    ::= { signalGroupEntryTable 1 }

SignalGroupEntry ::= SEQUENCE {
    signalGroupEntryNumber            Integer32,
    signalGroupIntersection           Integer32,
    signalGroupID                     Integer32,
    signalGroupControlSource          Integer32,
    signalGroupControlType            INTEGER,
    signalGroupGreenType              INTEGER,
    signalGroupRedType                INTEGER,
    signalGroupPermissiveControlSource Integer32,
    signalGroupPermissiveControlType   Integer32,
    signalGroupEnabledLanes           Integer32 }
```

5.16.7.1 Signal Group Entry Number

```
signalGroupEntryNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The number for objects in this row. This value
shall not exceed the maxSignalGroups object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.1"
    ::= { signalGroupEntry 1 }
```

5.16.7.2 Signal Group Intersection

```
signalGroupIntersection OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The number is used to uniquely define an
intersection within the region. A value of zero indicates that the row is
disabled and all other values in the row are not valid.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.2
    <Object Type> P"
    REFERENCE "SAE J2735 DE_IntersectionID"
    DEFVAL { 0 }
    ::= { signalGroupEntry 2 }
```

5.16.7.3 Signal Group Identifier

```
signalGroupID OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The number used to map between a movement phase
state (and its timings) in the CU and movements defined in a MAP message for the
intersection. A value of 255 is reserved to indicate a permanent green movement
state. A value of zero indicates not available or not known.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.3
    <Object Type> P"
    REFERENCE "SAE J2735 DE_SignalGroupID"
    DEFVAL { 255 }
    ::= { signalGroupEntry 3 }
```

5.16.7.4 Signal Group Control Source Parameter

```
signalGroupControlSource OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines which Phase or Overlap controls
the signal group. The value shall not exceed maxPhases or maxOverlaps as
determined by the signalGroupControlType object:
    Value 00 = No Control (Not In Use)
    Value 01 = Phase 01 or Overlap A
    Value 02 = Phase 02 or Overlap B
    ||
    Value 15 = Phase 15 or Overlap O
    Value 16 = Phase 16 or Overlap P
    etc.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.4
    <Object Type> P"
    DEFVAL { 0 }
    ::= { signalGroupEntry 4 }
```

5.16.7.5 Signal Group Control Type Parameter

```
signalGroupControlType OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     phaseVehicle (2),
                     phasePedestrian (3),
                     overlap (4),
                     pedOverlap (5),
                     queueJump (6) }
    MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION "<Definition> This object defines the type of control for the
signal group:
other: The signal group is not controlled by a type defined by this standard. Per
NTCIP 8004 v02, an attempt to write a value of 'other' shall return a badValue
error. A user- or manufacturer-specific object is permitted to extend the
possible states.
phaseVehicle: The signal group is for a vehicle phase, bicycle phase, or transit
phase.
phasePedestrian: The signal group is for a pedestrian phase.
overlap: The signal group is for an overlap for vehicles, bicycles, or transit.
pedOverlap: The signal group is for a pedestrian overlap.
queueJump: The signal group is for a queue jump, typically transit priority.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.5
    <Object Type> P"
    ::= { signalGroupEntry 5 }
```

5.16.7.6 Signal Group Green Type

```
signalGroupGreenType OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                 preMovement (2),
                 permissiveMovementAllowed (3),
                 protectedMovementAllowed (4),
                 protectedPermissive (5)}
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the type of Green to be
indicated for the signal group in a SPaT message:
other: The signal group green type is not defined by this standard. Per NTCIP
8004 v02, an attempt to write a value of 'other' shall return a badValue error. A
user- or manufacturer-specific object is permitted to extend the possible states.
preMovement: Prepare to proceed. Not common in the United States.
permissiveMovementAllowed: The signal group is a permissive green and may have to
yield to conflicting traffic.
protectedMovementAllowed: The signal group is a protected green and has the
right-of-way.
protectedPermissive: Indicates that the movement for the signal group can be
protected during some portions of the cycle and permissive during other portions
of the cycle. The signalGroupPermissiveControlSource and
signalGroupPermissiveControlType objects determine if the movement is protected
or permissive.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.6
    <Object Type> P"
    REFERENCE "SAE J2735 DE_MovementPhaseState"
```

```
    DEFVAL { 3 }
    ::= { signalGroupEntry 6 }
```

5.16.7.7 Signal Group Red Type

```
signalGroupRedType OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     stopThenProceed (2),
                     stopAndRemain (3)}
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the type of Red to be indicated
for the signal group in a SPaT message:
other: The signal group red type is not defined by this standard. Per NTCIP 8004
v02, an attempt to write a value of 'other' shall return a badValue error. A
user- or manufacturer-specific object is permitted to extend the possible states.
stopThenProceed: Indicates the vehicle must stop at the stop line when the signal
indication is red. It may then proceed if it is safe.
stopAndRemain: Indicates the traveler must stop and remain at the stop line when
the signal indication is red. It may not proceed.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.7
    <Object Type> P"
    REFERENCE "SAE J2735 DE_MovementPhaseState"
    DEFVAL { 3 }
    ::= { signalGroupEntry 7 }
```

5.16.7.8 Permissive Signal Group Control Source Parameter

```
signalGroupPermissiveControlSource OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> If signalGroupGreenType is protectedPermissive(5)
for the row, this object in combination with signalGroupPermissiveControlType
identifies which phase or overlap controls the permissive portion of the
protected-permissive movement. If signalGroupGreenType is any value other than
protectedPermissive(5), then this object is ignored. The value shall not exceed
maxPhases or maxOverlaps as determined by the signalGroupPermissiveControlType
object:
    Value 00 = No Control  (Not In Use)
    Value 01 = Phase 01 or Overlap A
    Value 02 = Phase 02 or Overlap B
    ||
    Value 15 = Phase 15 or Overlap O
    Value 16 = Phase 16 or Overlap P
    etc.
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.8
    <Object Type> P"
    DEFVAL { 0 }
    ::= { signalGroupEntry 8 }
```

5.16.7.9 Permissive Signa Group Control Type Parameter

```
signalGroupPermissiveControlType OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     none (2),
                     phaseVehicle (3),
                     phasePedestrian (4),
                     overlap (5),
                     pedOverlap (6),
                     queueJump (7) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> If signalGroupGreenType is protectedPermissive(5)
for the row, this object in combination with signalGroupPermissiveControlSource
identifies which phase or overlap controls the permissive portion of the
protected-permissive movement. If signalGroupGreenType is any value other than
protectedPermissive(5), then this object is ignored.
    other: The signal group is not controlled by a type defined by this standard.
Per NTCIP 8004 v02, an attempt to write a value of 'other' shall return a
badValue error. A user- or manufacturer-specific object is permitted to extend
the possible states.
    none: The signal group is not dependent on another phase nor overlap.
    phaseVehicle: The signal group is dependent on a vehicle phase, bicycle
phase, or transit phase.
    phasePedestrian: The signal group is dependent on a pedestrian phase.
    Overlap: The signal group is dependent on an overlap for vehicles, bicycles,
or transit.
    pedOverlap: The signal group is dependent on a pedestrian overlap.
    queueJump: The signal group is dependent on a queue jump, typically transit
priority.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.9
    <Object Type> P"
    DEFVAL { 2 }
    ::= { signalGroupEntry 9 }
```

5.16.7.10 Signal Group Enabled Lane

```
signalGroupEnabledLanes OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION "<Definition> The value is equal to the enabledLaneIndex in the
spatEnabledLanesConcurrencyTable. Each octet within the octet string for
enabledLaneConcurrency in that row contains the (revocable) lane number that must
be enabled for this signal group to be enabled. If all specified lanes are
enabled, then this signal group is enabled and in use. Otherwise, the signal
group is disabled and not used. A value of zero indicates that this signal group
is not dependent on a revocable lane. This value shall not exceed the
maxEnabledLanesConcurrency object value.

    This object was named signalGroupLane in NTCIP 1202 v03
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.10.1.10
    <Object Type> P"
    DEFVAL { 0 }
    ::= { signalGroupEntry 10 }
```

5.16.8 Maximum Agency Signal State Entries

```
maxAgencySignalStates OBJECT-TYPE
    SYNTAX Integer32 (3..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the maximum rows which shall
appear in the agencySignalStateTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.11
    <Object Type> S"
    ::= { ascSpat 11 }
```

5.16.9 Agency Signal State Table

```
agencySignalStateTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AgencySignalStateEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table is a lookup table to determine the
movement phase state for a signal group. See Annex F.3.3.5.2 for additional
guidance and see Figure 15 for the default values for the table.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.12"
    ::= { ascSpat 12 }

agencySignalStateEntry OBJECT-TYPE
    SYNTAX AgencySignalStateEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table defines the signal state to provide for
a signal group in the SPaT message based upon the parameters set in the
signalGroupEntryTable.
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.12.1"
    INDEX  { agencySignalStateNumber }
    ::= { agencySignalStateTable 1 }

AgencySignalStateEntry ::= SEQUENCE {
    agencySignalStateNumber      Integer32,
    agencySignalStateColor       INTEGER,
    agencySignalStateControlType INTEGER,
    agencySignalStateOptions     Integer32,
    agencySignalStateValue       INTEGER }
```

5.16.9.1 Agency Signal State Entry Number

```
agencySignalStateNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The number for objects in this row. This value
shall not exceed the maxAgencySignalStates object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.12.1.1"
    ::= { agencySignalStateEntry 1 }
```

5.16.9.2 Agency Signal State Color

```
agencySignalStateColor OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     red (2),
                     yellow (3),
                     green (4),
                     dark (5),
                     flashingRed (6),
                     flashingYellow (7) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the color of the phase or
overlap that the signal group is associated with to determine a signal state to
broadcast in a SPaT message.

    other: The phase or overlap is not defined by this standard.  Per NTCIP 8004
v02, an attempt to write a value of 'other' shall return a badValue error. A
manufacturer-specific object is permitted to extend the possible states.
    red: The phase or overlap is red. If the agencySignalStateControlType object
is pedestrian(3), this value indicates a Steady Don't Walk.
    yellow: The phase or overlap is yellow.
    green: The phase or overlap is green. If the agencySignalStateControlType
object is pedestrian(3), this value indicates a Walk.
```

```
    dark: The phase or overlap is dark.
    flashingRed: The phase or overlap is Flashing Red. If the
agencySignalStateControlType object is pedestrian(3), this value indicates a
Flashing Don't Walk.
    flashingYellow: The phase or overlap is Flashing Yellow.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.12.1.2
    <Object Type> P"
    ::= { agencySignalStateEntry 2 }
```

5.16.9.3 Agency Signal State Control Type

```
agencySignalStateControlType OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     vehicle (2),
                     pedestrian (3) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the type of control for the
signal group (Vehicle Phase, Pedestrian Phase, Overlap or Pedestrian Overlap) to
be used to determine the movement phase state to broadcast in a SPaT message:
other: The signal group is not defined by this standard. Per NTCIP 8004 v02, an
attempt to write a value of 'other' shall return a badValue error. A user- or
manufacturer-specific object is permitted to extend the possible states.
vehicle: The signal group is is configured in the signalGroupEntryTable with a
control type of phaseVehicle (3) or overlap (5) or queueJump (7).
pedestrian: The signal group is configured in the signalGroupEntryTable with a
control type of phasePedestrian (4) or pedOverlap (6).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.12.1.3
    <Object Type> P"
    ::= { agencySignalStateEntry 3 }
```

5.16.9.4 Additional Parameters for Determining Signal State

```
agencySignalStateOptions OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Additional parameters for determining the movement
phase state to broadcast in a SPaT message ( 0 = FALSE/Disabled, 1 =
TRUE/Enabled )
Bit 15: Reserved
Bit 14: Reserved
Bit 13: Reserved
Bit 12: Reserved
Bit 11: Reserved
Bit 10: Reserved
```

Bit 9: Permissive Flashing Red -The signalGroupPermissiveControlSource / signalGroupPermissiveControlType indication is Flashing Red.
Bit 8: Permissive Flashing Yellow -The signalGroupPermissiveControlSource / signalGroupPermissiveControlType indication is Flashing Yellow.
Bit 7: Permissive Green -The signalGroupPermissiveControlSource / signalGroupPermissiveControlType indication is Green.
Bit 6: Permissive Yellow -The signalGroupPermissiveControlSource / signalGroupPermissiveControlType indication is Yellow.
Bit 5: Permissive Red - The signalGroupPermissiveControlSource / signalGroupPermissiveControlType indication is Red.
Bit 4: Protected Green - If Bit 1 is TRUE, the signal group is configured in the signalGroupEntryTable with a signalGroupGreenType of protectedMovementAllowed (4) or protectedPermissive (5).
Bit 3: Permissive Green - If Bit 1 is TRUE, the signal group is configured in the signalGroupEntryTable with a signalGroupGreenType of permissiveMovementAllowed (3) or protectedPermissive (5).
Bit 2: preMovement Green - If Bit 1 is TRUE, the signal group is configured in the signalGroupEntryTable with a signalGroupGreenType is preMovement (2). Not used in the US.
Bit 1: stopAndRemain - If Bit 1 is TRUE, the signal group is configured in the signalGroupEntryTable with a signalGroupRedType of stopAndRemain (3).
Bit 0: stopThenProceed - If Bit 1 is TRUE, the signal group is configured in the signalGroupEntryTable with a signalGroupRedType of stopThenProceed (2).

If both Bits 0 and 1 are FALSE, then the signalGroupRedType is not applicable. Bits 0 and 1 cannot be both TRUE. A SET of both bits 0 and 1 to TRUE shall result return a wrongValue(10) error.

If both Bits 3 and 4 are FALSE, then the signalGroupGreenType is not applicable. If only Bit 3 is TRUE, then the signalGroupGreenType for the the signal group in the signalGroupEntryTable is permissiveMovementAllowed (3). If only Bit 4 is TRUE, then the signalGroupGreenType for the signal group in the signalGroupEntryTable is protectedMovementAllowed (4). If both Bits 3 and 4 are TRUE, that indicates the signalGroupGreenType is protectedPermissive and the signal indication of signalGroupPermissiveControlType and signalGroupPermissiveControlSource determines if the signal green type is permissiveMovementAllowed or protectedMovementAllowed.

Bits 5-9 determine what signal indication(s) of the signalGroupPermissiveControlType / signalGroupPermissiveControlSource makes the movement permissive. If Bits 5-9 are all TRUE, then ANY signal indication of the signalGroupPermissiveControlType / signalGroupPermissiveControlSource determines if the movement is permissive. If Bits 5-9 are all FALSE, then the color of the signalGroupPermissiveControlType / signalGroupPermissiveControlSource is not applicable.

```
See Annex F.3.3.5.2 and F.3.3.5.3 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.12.1.4
    <Object Type> P"
    ::= { agencySignalStateEntry 4 }
```

5.16.9.5 Agency Signal State Indication

```
agencySignalStateValue OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                 unavailable (2),
                 dark (3),
                 stopThenProceed (4),
                 stopAndRemain (5),
                 preMovement (6),
                 permissiveMovementAllowed (7),
                 protectedMovementAllowed (8),
                 permissiveClearance (9),
                 protectedClearance (10),
                 cautionConflictingTraffic (11) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the movement phase state to
be broadcasted in a SPaT message.
    other: The movement phase state is not defined by this standard.
    unavailable: The state of the movement is unknown or an error occurred.
    dark: The signal head is dark (unlit)
    stopThenProceed: Indicates the vehicle must stop at the stop line. It may
proceed if it is safe.
    stopAndRemain: Indicates the vehicle must stop at the stop line. It may not
proceed.
    preMovement: Prepare to proceed.
    permissiveMovementAllowed: Indicates a permissive green.
    protectedMovementAllowed: Indicates a protected green (a steady green arrow).
    permissiveClearance: Prepare to stop. Vehicle may proceed if unable to stop.
Conflicting traffic may be present.
    protectedClearance: Prepare to stop. Vehicle may proceed if unable to stop.
    cautionConflictingTraffic: Proceed with caution. Conflicting traffic may be
present.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.12.1.5
    <Object Type> P"
    REFERENCE "SAE J2735 DE_MovementPhaseState"
    DEFVAL { 2 }
    ::= { agencySignalStateEntry 5 }
```


5.16.10 Maximum Enabled Lanes Concurrency Entries

```
maxEnabledLanesConcurrency OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object indicates the maximum rows which shall
appear in the spatEnabledLanesConcurrencyTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.13
    <Object Type> S"
    ::= { ascSpat 13 }
```

5.16.11 SPaT Enabled Lanes Command

```
spatEnabledLanesCommand2 OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object establishes the Enabled lanes for the
device to be broadcast in a SPAT message. The value is equal to the
enabledLaneIndex in the spatEnabledLanesConcurrencyTable. Each octet within the
octet string for enabledLaneConcurrency in that row contains the (revocable) lane
number(s) that are enabled. The new set of Enabled lanes will be broadcast at the
top of the next cycle.

    A value of zero indicates that there are no enabled lanes. A value of 255
indicates that the device shall cancel this command and shall revert back to the
Enabled lanes defined in the timebaseAscActionTable timebaseAscActionLane.
Otherwise, the Enabled lanes defined is this command shall override the Enabled
lanes defined in timebaseAscActionLane.

    If an unsupported / invalid set of Enabled lanes is called, then the
spatPortStatus object shall be SET to enabledLanesError(5) for all valid entries
in the spatPortTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.16.14
    <Object Type> P"
    REFERENCE "SAE J2735 DF_EnabledLaneList"
    ::= { ascSpat 14 }
```

5.17 RSU – ASC Support

```
rsuAsc OBJECT IDENTIFIER
    ::= { asc 17 }

-- This defines a node for objects to support a CV Application Process in a
connected vehicle environment.

MapActivationCode ::= OCTET STRING (SIZE(3))
```

```
-- The MapActivationCode consists of those parameters required to activate a MAP
Plan message in an ASC. It is defined as an OCTET STRING containing the OER-
encoding of the following ASN.1 structure.

-- mapActivationCodeStructure ::= SEQUENCE {
-- mapPlanIndex INTEGER (1..8),
-- mapPlanCRC OCTET STRING (SIZE (2)) }
-- mapPlanIndex (8 bits) shall indicate the mapPlanIndex requested.
-- mapPlanCRC (16 bits) shall indicate the mapPlanCRC of the requested
mapPlanIndex.
```

5.17.1 RSU Signal Phase and Timing Functions

```
rsuAscSpat OBJECT IDENTIFIER ::= { rsuAsc 1 }

-- This defines a node for objects to support signalized intersection
applications in a CV Application Process in a connected vehicle environment.
```

5.17.1.1 Control Active MAP Plan

```
mapActivatePlan OBJECT-TYPE
    SYNTAX MapActivationCode
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A code indicating the active MAP plan. The value of
this object may be SET by a management station. If a GET is performed on this
object, the device shall respond with the value for the last MAP plan that was
successfully activated.

The mapActivatePlanError object shall be updated appropriately upon any attempt
to update the value of this object. If a MAP plan activation error occurs (e.g.,
mapActivatePlanError is updated to a value other than 'none'), the new MAP plan
shall not be activated and a genErr shall be returned. A management station
should then GET the mapActivatePlanError object as soon as possible to minimize
the chance of additional activation attempts from overwriting the
mapActivatePlanError.

A value of zero0 00 00 indicates no MAP plan is active and thus no MAP data
should be broadcast.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.1.4"
    ::= { rsuAscSpat 4 }
```

5.17.2 Connected Detection Zone

```
ascCvDetector OBJECT IDENTIFIER
```

```
    ::= { rsuAsc 2 }

-- This defines a node for a CV Application Process to support detection of
connected devices in a connected vehicle environment.
```

5.17.2.1 Connected Detection Zone Enable

```
cvDetectionEnable OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is used to enable/disable detection
zones in a connected vehicle environment. If enabled, the CU allows detectors
defined in vehicleDetectorTable and pedestrianDetectorTable to use inputs from
the connected vehicle environment. Inputs may be in the form of actuations,
safety messages and/or detection reports in the detectionReportTable.

    TRUE (one) - CV Detection
    FALSE (zero) - No CV Detection
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.1"
    DEFVAL  { 0 }
    ::= { ascCvDetector 1 }
```

5.17.2.2 Maximum Connected Detection Zones

```
maxCvDetectionZones OBJECT-TYPE
    SYNTAX Integer32(1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The maximum number of connected vehicle detection
zones this RSU supports. This object indicates the maximum rows which appear in
the ascCvDetectorTable object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.2"
    ::= { ascCvDetector 2 }
```

5.17.2.3 Connected Detection Zone Table

```
ascCvDetectorTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AscCvDetectorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing the connected vehicle detection
zone parameters for this RSU. The number of rows in this table is equal to the
maxCvDetectionZones object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3"
```

```
    ::= { ascCvDetector 3 }

ascCvDetectorEntry OBJECT-TYPE
    SYNTAX AscCvDetectorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific connected vehicle
detection zone.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1"
    INDEX { ascCvDetectorNumber }
    ::= { ascCvDetectorTable 1 }

AscCvDetectorEntry ::= SEQUENCE {
    ascCvDetectorNumber          Integer32,
    ascCvDetectorOptions         Integer32,
    ascCvDetectorPort            Integer32,
    ascCvDetectorAssignedInput   Integer32,
    ascCvDetectorAgpDuration     Integer32,
    ascCvDetectorRlvwMinHeading  Integer32,
    ascCvDetectorRlvwMaxHeading  Integer32,
    ascCvDetectorAgpLatitude     Integer32,
    ascCvDetectorAgpLongitude    Integer32 }
```

5.17.2.3.1 Connected Detection Zone Number

```
ascCvDetectorNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The connected vehicle detection zone number for
objects in this row. This value shall not exceed the maxCvDetectionZones object
value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.1"
    ::= { ascCvDetectorEntry 1 }
```

5.17.2.3.2 Connected Detection Zone Parameter

```
ascCvDetectorOptions OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Connected Detection Zone Options Parameter as
follows (0=Disabled, 1=Enabled):
    Bit 7: Reserved
    Bit 6: Enable Processed data    - enables the exchange of detectionReports
across the ASC - RSU interface.
```



```
    Bit 5: Reserved
    Bits 3 & 4: Indicates which detector table the detector number in
ascCvDetectorAssignment is assigned to.
    Bit 3 = 0, Bit 4 = 0    - unknown or not applicable
    Bit 3 = 1, Bit 4 = 0    - vehicleDetectorTable
    Bit 3 = 0, Bit 4 = 1    - pedestrianDetectorTable
    Bit 4 = 1, Bit 4 = 1    - Reserved
    Bit 2: Reserved
    Bit 1: A bit value of 1 enables the processing of Personal Safety messages
(PSMs) detected within the detection zone.
    Bit 0: A bit value of 1 enables the processing of Basic Safety Messages
(BSMs) detected within the detection zone.

    A SET of a both bits 0 and 1 to TRUE shall return a wrongValue(10) error.
    A SET of a both bits 3 and 4 to TRUE shall return a wrongValue(10) error.
    A SET of a 'reserved' bit to a value other than FALSE shall return a
wrongValue(10) error.


    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.2
    <Object Type> P"
    ::= { ascCvDetectorEntry 2 }
```

5.17.2.3.3 Connected Detection Zone Port

```
ascCvDetectorPort OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the ifIndex for a row in the
ifTable and the commPortTable that identifies a communications port on the ASC
that is used to receive connected vehicle messages for placing actuations on
detector defined in ascCvDetectorAssignedInput.
    A value of zero means no port is associated with this connected vehicle
detector.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.14
    <Object Type> P"
    ::= {  ascCvDetectorEntry 14 }
```

5.17.2.3.4 Connected Detection Zone Assigned Input

```
ascCvDetectorAssignedInput OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION "<Definition> This object defines a vehicle detector or
pedestrian detector that is actuated when an actuation is placed on this CV
Detector. An actuation on the assigned detector will call the assigned detectors
call phase. The diagnostic parameters of the assigned detector will also apply to
this CV Detector. Whether this is a vehicle detector or pedestrian detector is
defined in ascCvDetectorOptions. This value shall not exceed maxVehicleDetectors
or maxPedestrianDetectors as configured.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.15
    <Object Type> P"
    ::= {  ascCvDetectorEntry 15 }
```

5.17.2.3.5 Connected Detection Zone AGP Duration

```
ascCvDetectorAgpDuration OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines a FIXED duration of the Assured
Green Period, in tenth seconds. If a connected vehicle is detected at the
location defined by ascCvDetectorAgpLatitude and ascCvDetectorAgpLongitude while
the assigned phase of the detector in ascCvDetectorAssignedInput is GREEN, the
assigned phase is extended by as much as this duration. A phase may not be
extended beyond its programmed maximum green time. A value of zero means Assured
Green Period is not used.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.16
    <Object Type> P"
    ::= {  ascCvDetectorEntry 16 }
```

5.17.2.3.6 Connected Detection Zone RLVW Minimum Heading

```
ascCvDetectorRlvwMinHeading OBJECT-TYPE
    SYNTAX Integer32 (0..28800)
    UNITS "0.0125 degree"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the lower threshold for a
vehicle heading for the Assured Green Period defined in ascCvDetectorAgpDuration
to be applied to the assigned phase of the detector defined in
ascCvDetectorAssignedInput. A value of zero indicates that this heading is not
used for applying Assured Green Period.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.17
    <Object Type> P"
    REFERENCE "SAE J2735 DE_Heading"
    ::= {  ascCvDetectorEntry 17 }
```

5.17.2.3.7 Connected Detection Zone RLVW Maximum Heading

```
ascCvDetectorRlvwMaxHeading OBJECT-TYPE
    SYNTAX Integer32 (0..28800)
    UNITS "0.0125 degree"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the upper threshold for a
vehicle heading for a the Assured Green Period defined in
ascCvDetectorAgpDuration to be applied to the assigned phase of the detector
defined in ascCvDetectorAssignedInput. A value of zero indicates that this
heading is not used for applying Assured Green Period.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.18
    <Object Type> P"
    REFERENCE "SAE J2735 DE_Heading"
    ::= {  ascCvDetectorEntry 18 }
```

5.17.2.3.8 Connected Detection Zone AGP Application Latitude

```
ascCvDetectorAgpLatitude OBJECT-TYPE
    SYNTAX Integer32 (-90000000..90000001)
    UNITS "1/10 microdegree"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the latitude of the reference
point long the upstream edge of stopping distance used to compute the time when
the Assured Green Period is to be applied. This value is ignored if
ascCvDetectorAgpDuration is zero.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.19
    <Object Type> P"
    REFERENCE "SAE J2735 DE_Latitude"
    ::= {  ascCvDetectorEntry 19 }
```

5.17.2.3.9 Connected Detection Zone AGP Application Longitude

```
ascCvDetectorAgpLongitude OBJECT-TYPE
    SYNTAX Integer32 (-179999999..180000001)
    UNITS "1/10 microdegree"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object defines the longitude of the reference
point long the upstream edge of stopping distance used to compute the time when
the Assured Green Period is to be applied. This value is ignored if
ascCvDetectorAgpDuration is zero.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.3.1.20
    <Object Type> P"
    REFERENCE "SAE J2735 DE_Latitude"
```

```
    ::= {  ascCvDetectorEntry 20 }
```

5.17.2.4 Active CV Detectors

```
activeCvDetectors OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The number of active CV detectors in this device.
This object indicates how many rows are in the detectionReportTable object. There
shall be a row for every active CV detector collecting CV data.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.10
    <Object Type> S"
    ::= { ascCvDetector 10 }
```

5.17.2.5 Detection Reports Sequence

```
detectionReportSequence OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "sequence"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object defines a sequence number for detection
reports received from defined connected vehicle detection zones. This object is
used to track where a new detection report received by the CU should be stored.
The value cycles within the limits of 0 to 65535 and is incremented by one when a
detection report is stored in the detectionReportTable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.11
    <Object Type> S"
    ::= { ascCvDetector 11 }
```

5.17.2.6 Connected Detection Reports Table

```
detectionReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF DetectionReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing detector reports for a connected
vehicle detection zone defined for this CU. The number of rows in this table is
equal to the activeCvDetectors object.
    Note: The objects in this table are read-write to allow the RSU to be the
SNMP manager and the ASC to be the SNMP agent across the RSU - ASC interface.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12"
    ::= { ascCvDetector 12 }

detectionReportEntry OBJECT-TYPE
```

```
    SYNTAX DetectionReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Detection reports received for a specific connected
vehicle detection zone.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12.1"
    INDEX { ascCvDetectorNumber }
    ::= { detectionReportTable 1 }

DetectionReportEntry ::= SEQUENCE {
    detectionReportTime        Integer32,
    detectionReportVolume      Integer32,
    detectionReportSpeed       Integer32,
    detectionReportTravelTime Integer32,
    detectionReportQueue       Integer32,
    detectionReportGap         Integer32,
    detectionReportPlatoon     Integer32 }
```

5.17.2.6.1 Connected Detection Zone Report Time

```
detectionReportTime OBJECT-TYPE
    SYNTAX Integer32 (0..3601000)
    UNITS "millisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object indicates the seconds within the hour
the detector report is generated.
<Valid Value Rule> 0 to 3600999 are in milliseconds, with a leap second
represented by 3600000 to 3600999. A value of 3601000 shall represent
unavailable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12.1.1
    <Object Type> P"
    DEFVAL  { 3601000 }
    ::= { detectionReportEntry 1 }
```

5.17.2.6.2 Connected Detection Zone Volume Data

```
detectionReportVolume OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "volume"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A count of the number of connected devices
currently detected in the connected vehicle detection zone at the time the report
was generated. This value shall range from 0 to 254. The value 255 shall indicate
volume overflow.
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12.1.2
    <Object Type> P"
    DEFVAL { 0 }
    ::= { detectionReportEntry 2 }
```

5.17.2.6.3 Connected Detection Zone Speed Data

```
detectionReportSpeed OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "0.5 kilometers/hour"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The average speed, in 0.5 kilometers per hour, of
the connected devices currently detected in the connected vehicle detection zone
at the time the report was generated.
<Valid Value Rule> A value of zero to 253 is the average connected device speed
in 0.5 kilometers per hour units. A value of 254 represents an average speed of
127 kilometers per hour or higher. A value of 255 represents an invalid or
missing value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12.1.3
    <Object Type> P"
    DEFVAL  { 255 }
    ::= { detectionReportEntry 3 }
```

5.17.2.6.4 Connected Detection Zone Travel Time Data

```
detectionReportTravelTime OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The average travel time, in tenths of a second, for
a connected device to traverse the connected vehicle detection zone at the time
the report was generated. A value of 65535 represents an invalid or missing
value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12.1.4
    <Object Type> P"
    DEFVAL { 65535 }
    ::= { detectionReportEntry 4 }
```

5.17.2.6.5 Connected Detection Zone Queue Data

```
detectionReportQueue OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
```

```
    DESCRIPTION "<Definition> A count of the number of connected vehicles
currently queued in the connected vehicle detection zone at the time the report
was generated.
<ValidValueRule> A value of zero to 253 represents the number of vehicles queued.
A value of 254 indicates the queue is 254 or more vehicles. A value of 255
represents an invalid or missing value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12.1.5
    <Object Type> P"
    DEFVAL { 255 }
    ::= { detectionReportEntry 5 }
```

5.17.2.6.6 Connected Detection Zone Gap Data

```
detectionReportGap OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "centimeter"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The average gap, in centimeters, between the
connected vehicles currently detected in the connected vehicle detection zone at
the time the report was generated. The gap is defined as the distance between the
edge of the rear bumper of a connected vehicle and the edge of the front bumper
of a connected vehicle behind it. A value of 65535 indicates an invalid or
missing value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12.1.6
    <Object Type> P"
    DEFVAL { 65535 }
    ::= { detectionReportEntry 6 }
```

5.17.2.6.7 Connected Detection Zone Platoon Data

```
detectionReportPlatoon OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "vehicle"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> A count of the number of connected vehicles in a
platoon currently detected in the connected vehicle detection zone at the time
the report was generated.
<ValidValueRule> A value of zero to 253 represents the number of vehicles in the
platoon. A value of 254 indicates the platoon is 254 or more vehicles. A value of
255 represents an invalid or missing value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.17.2.12.1.7
    <Object Type> P"
    DEFVAL { 255 }
    ::= { detectionReportEntry 7 }
```

5.18 ASC – ECLA Interface

```
ecla OBJECT IDENTIFIER
    ::= { asc 18 }
```

5.18.1 ECLA Communications Parameter

```
eclaCommEnable OBJECT-TYPE
    SYNTAX Integer32 (0..1)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object controls if an ASC should use any data
received from an ECLA to determine when the current phases may terminate and what
phases may be next.
            TRUE (one) - Use data provided by an ECLA.
            FALSE (zero) - Do not use data provided by an ECLA.

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.1
    <Object Type> P"
    DEFVAL { 0 }
    ::= { ecla 1 }
```

5.18.2 ECLA Timestamp Input

```
eclaDataTimestamp OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The most recent time point when the ECLA provided
input into the controller.

    Values from 0 to 35999 cover one hour. Values from 36000 to 36009 are used
when a leap second occurs. Values from 36010 to 36110 are reserved for future
use. A value of 36111 indicates the time point of the event is undefined or
unknown.

    The device shall reset this object to ZERO when in BACKUP Mode. A write to
this object shall reset the Backup timer to ZERO (see unitBackupTime).
    See Annex F.3.3.6.2 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.2
    <Object Type> C"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { ecla 2 }
```

5.18.3 ECLA Input Table

```
eclaInputTable OBJECT-TYPE
    SYNTAX SEQUENCE OF EclaEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing Controller Unit phase
        parameters. The number of rows in this table is equal to the
        maxRings object value.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.3"
    ::= { ecla 3 }

eclaEntry OBJECT-TYPE
    SYNTAX EclaEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Parameters for a specific Controller Unit phase.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.3.1"
    INDEX { sequenceRingNumber }
    ::= { eclaInputTable 1 }

EclaEntry  ::= SEQUENCE {
    eclaRingCurrentPhase        Integer32,
    eclaRingGreenMinEndTime     Integer32,
    eclaRingGreenMaxEndTime     Integer32,
    eclaRingGreenLikelyEndTime  Integer32,
    eclaRingEndTimeConfidence   Integer32,
    eclaRingNextPhase           Integer32 }
```

5.18.3.1 ECLA Ring Current Phase

```
eclaRingCurrentPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The phase number for the phase currently on for
this ring as provided by an ECLA. This is the phase that eclaRingGreenMinEndTime,
eclaRingGreenMaxEndTime, eclaRingGreenLikelyEndTime, and
eclaRingEndTimeConfidence object values correspond to.

    This value shall not exceed value of maxPhases. A value of zero indicates
that the ECLA is currently unable to provide the information.

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.3.1.1
```

```
    <Object Type> C"
    DEFVAL { 0 }
    ::= { eclaEntry 1 }
```

5.18.3.2 ECLA Ring Current Phase Minimum Green End Time

```
eclaRingGreenMinEndTime OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The earliest future time point when the phases in
eclaCurrentDataPhase are expected to terminate, as provided by an ECLA.

    Values from 0 to 35999 cover one hour. Values from 36000 to 36009 are used
when a leap second occurs. Values from 36010 to 36110 are reserved for future
use. A value of 36111 indicates the time point of the event is undefined or
unknown.

    See Annex F.3.3.6.3 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.3.1.2
    <Object Type> C"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { eclaEntry 2 }
```

5.18.3.3 ECLA Ring Current Phase Maximum Green End Time

```
eclaRingGreenMaxEndTime OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The latest future time point when the phases in
eclaCurrentDataPhase are expected to terminate, as provided by an ECLA.

    Values from 0 to 35999 cover one hour. Values from 36000 to 36009 are used
when a leap second occurs. Values from 36010 to 36110 are reserved for future
use. A value of 36111 indicates the time point of the event is undefined or
unknown.

    See Annex F.3.3.6.3 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.3.1.3
    <Object Type> C"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
```



```
    ::= { eclaEntry 3 }
```

5.18.3.4 ECLA Ring Current Phase Likely Green End Time

```
eclaRingGreenLikelyEndTime OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "decisecond"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The most likely future time point when the phases
in eclaCurrentDataPhase are expected to terminate, as provided by an ECLA.

    Values from 0 to 35999 cover one hour. Values from 36000 to 36009 are used
when a leap second occurs. Values from 36010 to 36110 are reserved for future
use. A value of 36111 indicates the time point of the event is undefined or
unknown.

    See Annex F.3.3.6.4 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.3.1.4
    <Object Type> C"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { eclaEntry 4 }
```

5.18.3.5 ECLA Ring Current Phase End Time Confidence

```
eclaRingEndTimeConfidence OBJECT-TYPE
    SYNTAX Integer32 (0..15)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> For NTCIP 1202 user clarification, the statistical
confidence in the predicted value of the eclaCurrentPhaseLikelyEndTime, as
provided by an ECLA.

    See Annex F.3.3.6.5 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.3.1.5
    <Object Type> C"
    REFERENCE "SAE J2735 DE_TimeIntervalConfidence"
    DEFVAL { 0 }
    ::= { eclaEntry 5 }
```

5.18.3.6 ECLA Ring Next Phase

```
eclaRingNextPhase OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
```



```
    STATUS current
    DESCRIPTION "<Definition> The phase number for the phase is expected to be
committed as the next phase in the ring, as provided by an ECLA.

    This value shall not exceed the maxPhases object value. A value of zero
indicates that the ECLA is currently unable to provide the information.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.3.1.6
    <Object Type> C"
    DEFVAL { 0 }
    ::= { eclaEntry 6 }
```

5.18.4 ECLA Input Block

```
--5.19.4 ECLA Input Block
eclaInputBlock OBJECT-TYPE
    SYNTAX       OCTET STRING
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION  "<Definition> An octet string of data required from an ECLA for
an ASC to determine when the current phases are expected to terminate their green
indications and what phases are expected to be serviced next. This object is used
for the ECLA to input into the ASC in a bandwidth efficient manner.

eclaInputData ::= SEQUENCE {
    eclaDataTimestamp.0              Integer32 (0..36611),
    eclaRingCurrentPhase.x           Integer32 (0..255),
    eclaRingGreenMinEndTime.x        Integer32 (0..36611),
    eclaRingGreenMaxEndTime.x        Integer32 (0..36611),
    eclaRingGreenLikelyEndTime.x     Integer32 (0..36611),
    eclaRingEndTimeConfidence.x      Integer32 (0..36611),
    eclaRingNextPhase.x              Integer32 (0..255)
}

where:
    x is the sequenceRingNumber in the eclaInputTable.

    <Object Identifier> 1.3.6.1.4.1.1206.4.2.18.4
    <Object Type> C"
::= { ecla 4 }
```

5.19 ASC SMU Monitoring

```
ascSmu OBJECT IDENTIFIER
    ::= { asc 19 }
```


5.19.1 Signal Monitoring Unit Table

```
scSmuTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AscSmuTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A table containing channel information from the
signal monitoring unit in the cabinet. The number of rows in this table is equal
to three times the value of the maxChannels.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.19.1"
    ::= { ascSmu 1 }

ascSmuTableEntry OBJECT-TYPE
    SYNTAX AscSmuTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> A row of information for each channel output
provided by the signal monitoring unit in the cabinet.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.19.1.1"
    INDEX { channelNumber, ascSmuChannelOutput }
    ::= { ascSmuTable 1 }

AscSmuTableEntry ::= SEQUENCE {
    ascSmuChannelOutput Integer32,
    ascSmuVoltage       Integer32,
    ascSmuCurrent       Integer32 }
```

5.19.1.1 Signal Monitoring Output Index

```
ascSmuChannelOutput OBJECT-TYPE
    SYNTAX Integer32 (1..3)
    UNITS "output"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The different types for a channel.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.19.1.1.1"
    ::= { ascSmuTableEntry 1 }
```

5.19.1.2 Signal Monitoring Unit Channel Voltage

```
ascSmuVoltage OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    UNITS "VAC RMS or VDC"
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION "<Definition> This object shall describe the voltage measured for
the
    channel and output type in this row as reported by the signal monitoring unit
and measured in
    0 to 254 VAC RMS or from 0 to 50 VDC RMS for DC cabinets. A value of 255
indicates
   that the value is unknown or no measurement was made.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.19.1.1.3
    <Object Type> S"
    ::= { ascSmuTableEntry 3 }
```

5.19.1.3 Signal Monitoring Unit Channel Current

```
ascSmuCurrent OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    UNITS "milliamperes"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object shall describe the current measured for
the
    channel and output type in this row as reported by the signal monitoring unit
and measured in
   milliamperes from 0 to 65000 mA. A value of 65535 indicates
   that the value is unknown or no measurement was made.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.19.1.1.4
    <Object Type> S"
    ::= { ascSmuTableEntry 4 }
```

5.20 Signal Phase and Timing

```
saeNtcip OBJECT IDENTIFIER ::= { devices 17 }

spat OBJECT IDENTIFIER
    ::= { saeNtcip 1 }

-- This defines a node for supporting signal phase and timing objects for a
connected vehicle environment.
```

5.20.1 SPaT Enabled Lanes Status

```
spatEnabledLanesStatus OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> Each octet within the octet string contains the
(revocable) lane identifier that is currently enabled. Lanes that may not always
```

```
be ACTIVE (enabled) should be identified as a revocable lane in the MAP message
for the intersection.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.6"
    REFERENCE "SAE J2735 DF_EnabledLaneList"
    ::= { spat 6 }
```

5.20.2 Intersection Status

```
spatStatus2 OBJECT-TYPE
    SYNTAX Integer32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> For NTCIP 1202 user clarification, if a mode is
enabled or active, the associated bit shall be set to (1). Bit 0 is equal to Bit
0 of the BIT STRING of DE_IntersectionStatusObject, Bit 1 is equal to Bit 1, etc.

    Bit 15: Reserved.
    Bit 14: Reserved.
    Bit 13: Invalid SPaT. Enabled if any spatPortStatus object is any value other
than normal (3), if spatOptions Bit 1 is FALSE, or the spatPortOptions Bit 1 is
FALSE.
    Bit 12: Invalid MAP. Enabled if spatOptions Bit 2 is FALSE, or the
spatPortOptions Bit 2 is FALSE.
    Bit 11: Reserved. SET to 1.
    Bit 10: Reserved. SET to 1.
    Bit 9: Reserved. SET to 0.
    Bit 8: Failure Mode. Controller failure or failure in operation.
    Bit 7: Standby. Enabled if the unitFlashStatus is automatic (3), startup (7),
or preempt (8).
    Bit 6: Traffic Dependent. Enabled if the CU is operating in an actuated mode.
This bit and Bit 5 cannot both be simultaneously active.
    Bit 5: Fixed Time. Enabled if the CU is operating in fixedTime mode. This is
true when every phase in the active sequence is running with a Max Vehicle
Recall, a non-actuated split, or is responding to a call to non-actuated input.
This bit and Bit 6 cannot both be simultaneously active.
    Bit 4: Signal Priority. Enabled when Bit 8: Priority Call of
shortAlarmStatusV4 is True.
    Bit 3: Preempt. Enabled when Bit 0: Preempt Active of shortAlarmStatusV4 is
TRUE and the active preempt's preemptState is any value other than notActive (2)
or notActiveWithCall (3). This bit may also be enabled if the active preempt's
preemptState is other (1) and determinate.
    Bit 2: Failure Flash. Enabled when unitFlashStatus is localManual (4),
controllerFaultFlash (5), or smu (6).
    This bit may also be enabled if Bit 0: Preempt Active of shortAlarmStatusV4
is TRUE and the active preempt's preemptState is other (1) and indeterminate.
```

```
    Bit 1: Stop Time. Enabled when Bit 7: Stop Time in shortAlarmStatusV4 is
TRUE.
    Bit 0: Manual Control. Enabled when unitControlStatus is remoteManualControl
(9) or localManualControl (10).

The default bit value for Bits 10 and 11 is enabled (1). The default bit value
for all other Bits is disabled (0).
See Annex F.3.3.4 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.10
    <Object Type> S"
    REFERENCE "SAE J2735 DE_IntersectionStatusObject"
    DEFVAL { 3072 }
    ::= { spat 10 }
```

5.20.3 Maximum Movement Events

```
maxMovementEvents OBJECT-TYPE
    SYNTAX INTEGER (2..16)
    UNITS "movement"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of movement
events for each signal group this CU supports. The first movement event
represents the current interval, the second movement event represents the next
(future) movement event to follow the current interval.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.11
    <Object Type> S"
    ::= { spat 11 }
```

5.20.4 SPaT Signal Status Table

```
signalStatusTable2 OBJECT-TYPE
    SYNTAX SEQUENCE OF SignalStatusEntry2
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table contains signal status information for
signalized intersections controlled by the CU.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12"
    ::= { spat 12 }

signalStatusEntry2 OBJECT-TYPE
    SYNTAX SignalStatusEntry2
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The status of a specific signal group.
```

```
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1"
    INDEX  { signalGroupEntryNumber, movementEventNumber }
    ::= { signalStatusTable2 1 }

SignalStatusEntry2::= SEQUENCE {
    movementEventNumber          Integer32,
    signalState2                 INTEGER,
    signalStateMinEndTick2       Integer32,
    signalStateMaxEndTick2       Integer32,
    signalStateLikelyEndTick2    Integer32,
    signalStateTickConfidence2   Integer32,
    signalStateNextTick2         Integer32,
    signalStateStartTick         Integer32 }
```

5.20.4.1 Movement Event Number

```
movementEventNumber OBJECT-TYPE
    SYNTAX Integer32 (1..16)
    UNITS "number"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The movement index for objects in this row. This
value shall not exceed the maxMovementEvents object value. Index value 1 contains
the value for the current interval and Index value 2 contains the value for the
next interval.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1.1"
    ::= { signalStatusEntry2 1 }
```

5.20.4.2 SPaT Event State

```
signalState2 OBJECT-TYPE
    SYNTAX INTEGER { other (1),
                     unavailable (2),
                     dark (3),
                     stopThenProceed (4),
                     stopAndRemain (5),
                     preMovement (6),
                     permissiveMovementAllowed (7),
                     protectedMovementAllowed (8),
                     permissiveClearance (9),
                     protectedClearance (10),
                     cautionConflictingTraffic (11) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object value defines the movement phase state
for a signal group ID. movementEventNumber.1 indicates the current movement phase
```

```
state, and movementEventNumber.2 indicates the next movement phase state. If the
movement phase state is unknown, a value of 2 is used.

The movement phase state is determined by agencySignalStateValue.

See Annex F.3.3.5 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1.2
    <Object Type> S"
    REFERENCE "SAE J2735 DE_MovementPhaseState"
    ::= { signalStatusEntry2 2 }
```

5.20.4.3 SPaT Signal State Minimum End Time

```
signalStateMinEndTick2 OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "ticks"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The tick count representing the earliest future
time point when the signalState is expected to change, excluding unexpected
events such as a preempt request.  The value of this object will remain constant
if the predicted future time point of the event does not change. Values from 0 to
35999 cover one hour. Values from 36000 to 36009 are used when a leap second
occurs. Values from 36010 to 36110 are reserved for future use. A value of 36111
indicates the time point of the event is undefined or unknown.

This object is used with the ascCurrentTick2 object. If signalStateMinEndTick2 is
greater than ascCurrentTick2 then the end time is in the current hour. If
signalStateMinEndTick2 is less than ascCurrentTick2 then the time is in the next
hour.

See Annex F.3.3.6.2 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1.3
    <Object Type> S"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { signalStatusEntry2 3 }
```

5.20.4.4 SPaT Signal State Maximum End Time

```
signalStateMaxEndTick2 OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "tick"
    MAX-ACCESS read-only
    STATUS current
```

```
    DESCRIPTION "<Definition> The tick count representing the latest future time
point when the signalState is expected to change, excluding unexpected events
such as a preempt request. The value of this object will remain constant if the
predicted future time point of the event does not change. Values from 0 to 35999
cover one hour. Values from 36000 to 36009 are used when a leap second occurs.
Values from 36010 to 36110 are reserved for future use. A value of 36111
indicates the time point of the event is undefined or unknown.

This object is used with the ascCurrentTick2 object. If signalStateMaxEndTick2 is
greater than ascCurrentTick2 then the end time is in the current hour. If
signalStateMaxEndTick2 is less than ascCurrentTick2 then the time is in the next
hour.

See Annex F.3.3.6.3 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1.4
    <Object Type> S"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { signalStatusEntry2 4 }
```

5.20.4.5 SPaT Likely End Time

```
signalStateLikelyEndTick2 OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "tick"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The tick count representing the most likely future
time point when the signalState is expected to change, excluding unexpected
events such as a preempt request. The value of this object will remain constant
if the predicted future time point of the event does not change. Values from 0 to
35999 cover one hour. Values from 36000 to 36009 are used when a leap second
occurs. Values from 36010 to 36110 are reserved for future use. A value of 36111
indicates the time point of the event is undefined or unknown.

This object is used with the ascCurrentTick2 object. If signalStateLikelyEndTick2
is greater than ascCurrentTick2 then the end time is in the current hour. If
signalStateLikelyEndTick2 is less than ascCurrentTick2 then the time is in the
next hour.

See Annex F.3.3.6.4 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1.5
    <Object Type> S"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { signalStatusEntry2 5 }
```

5.20.4.6 SPaT Signal State Time Confidence

```
signalStateTickConfidence2 OBJECT-TYPE
    SYNTAX Integer32 (0..15)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> For NTCIP 1202 user clarification, the statistical
confidence in the predicted value of the signalStateLikelyEndTick2.

See Annex F.3.3.6.5 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1.6
    <Object Type> S"
    REFERENCE "SAE J2735 DE_TimeIntervalConfidence"
    DEFVAL { 0 }
    ::= { signalStatusEntry2 6 }
```

5.20.4.7 SPaT Signal Next Tick

```
signalStateNextTick2 OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "tick"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The tick count representing the most likely future
time point when the movement is expected to be allowed to move again (e.g.,
green), excluding unexpected events such as a preempt request. The value of this
object will remain constant if the predicted future time point of the event does
not change. Values from 0 to 35999 cover one hour. Values from 36000 to 36009 are
used when a leap second occurs. Values from 36010 to 36110 are reserved for
future use. A value of 36111 indicates the time point of the event is undefined
or unknown.

This object is used with the ascCurrentTick2 object. If signalStateNextTick2 is
greater than ascCurrentTick2 then the next time is in the current hour. If
signalStateNextTick2 is less than ascCurrentTick2 then the time is in the next
hour.

See Annex F.3.3.6.6 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1.7
    <Object Type> S"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { signalStatusEntry2 7 }
```

5.20.4.8 SPaT Signal Start Tick

```
signalStateStartTick OBJECT-TYPE
    SYNTAX Integer32 (0..36111)
    UNITS "tick"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The tick count representing the start time of the
next (future) movement state to follow the current movement state for a signal
group. The value of this object will remain constant if the predicted future time
point of the event does not change. Values from 0 to 35999 cover one hour. Values
from 36000 to 36009 are used when a leap second occurs. Values from 36010 to
36110 are reserved for future use. A value of 36111 indicates the time point of
the event is undefined or unknown.

This object is used with the ascCurrentTick2 object. If signalStateStartTick is
greater than ascCurrentTick2 then the start time of the next movement state is in
the current hour. If signalStateStartTick is less than ascCurrentTick2 then the
time is in the next hour.

See Annex F.3.3.6.7 for additional guidance.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.12.1.8
    <Object Type> S"
    REFERENCE "SAE J2735 DE_TimeMark"
    DEFVAL { 36111 }
    ::= { signalStatusEntry2 8 }
```

5.20.5 Maximum SPaT Speed Advisories

```
maxAdvisorySpeeds2 OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "advisory"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of movement
speed advisory entries this CU supports. This object indicates the maximum rows
which shall appear in the advisorySpeedTable2 object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.13
    <Object Type> S"
    ::= { spat 13 }
```

5.20.6 SPaT Speed Advisories Table

```
advisorySpeedTable2 OBJECT-TYPE
    SYNTAX SEQUENCE OF AdvisorySpeedEntry2
    MAX-ACCESS not-accessible
    STATUS current
```

```
    DESCRIPTION "<Definition> This table contains speed advisories for signalized
intersections controlled by this CU. The number of rows in this table is equal to
the maxAdvisorySpeeds2 object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.14"
    ::= { spat 14 }

advisorySpeedEntry2 OBJECT-TYPE
    SYNTAX AdvisorySpeedEntry2
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Speed advisory information for a signal group.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.14.1"
    INDEX  { advisorySpeedIndex2 }
    ::= { advisorySpeedTable2 1 }

AdvisorySpeedEntry2 ::= SEQUENCE {
    advisorySpeedIndex2                 Integer32,
    advisorySpeedSignalGroupEntryNumber Integer32,
    advisorySpeedType2                  INTEGER,
    advisorySpeedAdvice2                Integer32,
    advisorySpeedZoneLength2            Integer32,
    advisorySpeedClass2                 Integer32  }
```

5.20.6.1 SPaT Speed Advisory Index

```
advisorySpeedIndex2 OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "number"
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The advisory speed index for objects in this row.
This value shall not exceed the maxAdvisorySpeeds2 object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.14.1.1"
    ::= { advisorySpeedEntry2 1 }
```

5.20.6.2 SPaT Advisory Signal Group

```
advisorySpeedSignalGroupEntryNumber OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The signalGroupEntryNumber that the speed advisory
is applicable for. A value of zero indicates that the row is disabled and all
other values in the row are not valid.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.14.1.2
```

```
    <Object Type> P"
    ::= { advisorySpeedEntry2 2 }
```

5.20.6.3 SPaT Movement Advisory Speed Type

```
advisorySpeedType2 OBJECT-TYPE
    SYNTAX INTEGER { none (1),
                greenwave (2),
                ecoDrive (3),
                transit (4) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The type of speed advisory provided for this
movement, such as greenwave or ecoDrive. Note: The enumerations do not match
DE_AdvisorySpeedType.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.14.1.3
    <Object Type> P"
    REFERENCE "SAE J2735 DE_AdvisorySpeedType"
    DEFVAL { 1 }
    ::= { advisorySpeedEntry2 3 }
```

5.20.6.4 SPaT Movement Advisory Speed Advice

```
advisorySpeedAdvice2 OBJECT-TYPE
    SYNTAX Integer32 (0..500)
    UNITS "decimeter per second"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The advisory speed provided by the ASC for this
movement, in units of 0.1 meters per second. A value of 500 indicates that speed
is unavailable.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.14.1.4
    <Object Type> P"
    REFERENCE "SAE J2735 DE_SpeedAdvice"
    DEFVAL { 500 }
    ::= { advisorySpeedEntry2 4 }
```

5.20.6.5 SPaT Movement Advisory Speed Zone

```
advisorySpeedZoneLength2 OBJECT-TYPE
    SYNTAX Integer32 (0..10000)
    UNITS "meter"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The distance, in meters, indicates the region
        upstream from the stopbar for which the advisory speed is recommended.
```

```
        A value of zero indicates unknown. A value of 10000 indicates distances
equal to or greater than 10000 meters.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.14.1.5
    <Object Type> P"
    REFERENCE "SAE J2735 DE_ZoneLength"
    ::= { advisorySpeedEntry2 5 }
```

5.20.6.6 SPaT Movement Advisory Speed Restriction Class

```
advisorySpeedClass2 OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The vehicle type(s) to which the advisory speed
        applies to. The value is provided in the MAP message for the
        intersection. A value of zero indicates that advisory speed applies to
        all vehicle types.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.14.1.6
    <Object Type> P"
    REFERENCE "SAE J2735 DE_RestrictionClassID"
    DEFVAL { 0 }
    ::= { advisorySpeedEntry2 6 }
```

5.20.7 Maximum SPaT Movement Maneuvers

```
maxMovementManeuvers2 OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    UNITS "maneuver"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> This object contains the maximum number of rows
which shall appear in the movementManeuverTable2 object.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.15
    <Object Type> S"
    ::= { spat 15 }
```

5.20.8 SPaT Movement Maneuvers Table

```
ovementManeuverTable2 OBJECT-TYPE
    SYNTAX SEQUENCE OF MovementManeuverEntry2
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> This table contains information relevant to a
maneuver that is related to a signal group at a signalized intersection. The
number of rows in this table is equal to the maxMovementManeuvers2 object.
    <TableType> static
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16"
```

```
    ::= { spat 16 }

movementManeuverEntry2 OBJECT-TYPE
    SYNTAX MovementManeuverEntry2
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> Maneuvers information for a specific movement at a
signalized intersection.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16.1"
    INDEX  { movementManeuverIndex2 }
    ::= { movementManeuverTable2 1 }

MovementManeuverEntry2 ::= SEQUENCE {
    movementManeuverIndex2                  Integer32,
    movementManeuverSignalGroupEntryNumber Integer32,
    movementManeuverQueue2                  Integer32,
    movementManeuverStatus2                 Integer32,
    movementManeuverQueueDetector2          OCTET STRING,
    movementManeuverPedPresence2            OCTET STRING,
    movementManeuverBicyclePresence2        OCTET STRING }
```

5.20.8.1 SPaT Movement Maneuver Number

```
movementManeuverIndex2 OBJECT-TYPE
    SYNTAX Integer32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "<Definition> The movement maneuver index for objects in this
row. This value shall not exceed the maxMovementManeuvers2 object value.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16.1.1"
    ::= { movementManeuverEntry2 1 }
```

5.20.8.2 SPaT Movement Maneuver Signal Group

```
movementManeuverSignalGroupEntryNumber OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> The signalGroupEntryNumber that the movement
maneuver is applicable for. A value of zero indicates that the row is disabled
and all other values in the row are not valid.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16.1.2
    <Object Type> P"
    REFERENCE "SAE J2735 DE_SignalGroupID"
    ::= { movementManeuverEntry2 2 }
```

5.20.8.3 SPaT Movement Queue

```
movementManeuverQueue2 OBJECT-TYPE
    SYNTAX Integer32 (0..10000)
    UNITS "meter"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> The distance in meters from the stop line at the
        intersection for this movement to the back edge of the last vehicle in
        the queue as measured along the center line of the lane. A value of zero
        indicates no queue or the queue distance is unknown. A value of 10000
        represents all distance greater or equal to 10000 meters.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16.1.3
    <Object Type> P"
    REFERENCE "SAE J2735 DF_ConnectionManeuverAssist and DE_ZoneLength"
    DEFVAL { 0 }
    ::= { movementManeuverEntry2 3 }
```

5.20.8.4 SPaT Movement Assist Status

```
movementManeuverStatus2 OBJECT-TYPE
    SYNTAX Integer32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> For NTCIP 1202 user clarification, this object
defines the options for traveler connections through an intersection.
    Bit 7:  Reserved
    Bit 6:  Reserved
    Bit 5:  Reserved
    Bit 4:  Reserved
    Bit 3:  Reserved
    Bit 2:  Reserved
    Bit 1:  Vulnerable Road User -Status bit set to TRUE (1) if ANY pedestrians
or bicyclists are detected in a conflicting movement. Set to FALSE (0) if there
is a high certainty no pedestrians or bicyclists are present. The detectors
providing this information is defined by movementManeuverPedPresence2 and
movementManeuverBicyclePresence2
    Bit 0:  Reserved
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16.1.4
    <Object Type> P"
    REFERENCE "SAE J2735 DE_PedestrianBicycleDetect"
    ::= { movementManeuverEntry2 4 }
```

5.20.8.5 SPaT Movement Queue Detector

```
movementManeuverQueueDetector2 OBJECT-TYPE
    SYNTAX OCTET STRING
```

```
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet in this octet string represents a
vehicle detector number (vehicleDetectorNumber) in the vehicleDetectorTable that
provides the data to determine movementManeuverQueue2. A value of zero0 indicates
that no additional vehicle detectors follow in the octet string.
    Each octet cannot exceed the value of maxVehicleDetectors.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16.1.5
    <Object Type> P"
    ::= { movementManeuverEntry2 5 }
```

5.20.8.6 SPaT Movement Pedestrian Presence

```
movementManeuverPedPresence2 OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet in this octet string represents a
pedestrian detector number (pedestrianDetectorNumber) in the
pedestrianDetectorTable (with Bit 0 in the corresponding
pedestrianDetectorOptions Enabled) whose detection status indicates a pedestrian
is detected in the pedestrian crossing that conflicts with this
movementManeuverSignalGroupEntryNumber. For example, an octet string of 01 03
indicates that if pedestrian detector 1 or pedestrian detector 3 is active/ON,
then the presence of a pedestrian has been detected that conflicts with this
movementManeuverSignalGroupEntryNumber. If a pedestrian is detected by any
pedestrianDetectorNumber in the octet string, Bit 1 in movementManeuverStatus2
shall be SET to TRUE (1).
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16.1.6
    <Object Type> P"
    ::= { movementManeuverEntry2 6 }
```

5.20.8.7 SPaT Movement Bicycle Presence

```
movementManeuverBicyclePresence2 OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> Each octet in this octet string represents a
vehicle detector number (vehicleDetectorNumber) in the vehicleDetectorTable that
when active/ON indicates that a bicyclist is detected that conflicts with this
movementManeuverSignalGroupEntryNumber. If a bicyclist is detected by any of the
vehicleDetectorNumber in the octet string, Bit 1 in movementManeuverStatus2 shall
be SET to TRUE (1). For example, an octet string of 02 06 indicates that if
vehicle detector 2 or vehicle detector 6 is active/ON, then the presence of a
```

```
bicyclist has been detected that conflicts with this
movementManeuverSignalGroupEntryNumber.
    Each octet cannot exceed the value of maxVehicleDetectors.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.16.1.7
    <Object Type> P"
    ::= { movementManeuverEntry2 7 }
```

5.20.9 Road Authority Identifier

```
spatRoadAuthorityID OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "<Definition> This object is the unique identifier of the agency
for the SPaT and MAP message. This object shall be either a valid OBJECT
IDENTIFIER or a null string.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.17
    <Object Type> P"
    REFERENCE "SAE J2735 DE_RelativeRoadAuthorityID"
    DEFVAL { zeroDotZero }
    ::= { spat 17 }
```

5.20.10 SPaT Status Signal Block

```
signalStatusBlock2 OBJECT-TYPE
    SYNTAX ITSOerString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "<Definition> An OER encoded string of the signalStatusTable2
        structure as defined below. This object is used for uploading
        configuration data from the ASC in a bandwidth efficient manner.

        signalStatus2 ::= SEQUENCE {
            ascCurrentTick2.0,        -- @NTCIP1202-v03B, Bytes 1 & 2
            spatStatus2.0             -- 2 BYTES @NTCIP1202-v03B
            spatEnabledLanesCount.0   -- 1 BYTE @NTCIP1202-v03B
            spatEnabledLanesStatus.0 -- n BYTES
            signalGroupEntryCount.0   -- 1 BYTE
            signalData2      SEQUENCE OF SignalStatusData2   OPTIONAL }

        SignalStatusData2 ::== SEQUENCE {
            signalGroupEntryNumber.y     -- 1 BYTE
            signalState2.y.x             -- 1 BYTE @NTCIP1202-v03B
            signalStateMinEndTick2.y.x  -- 2 BYTES @NTCIP1202-v03B
            signalStateMaxEndTick2.y.x  -- 2 BYTES @NTCIP1202-v03B
            signalStateNextTick2.y.x     -- 2 BYTES @NTCIP1202-v03B
```

```
        signalStateStartTick.y.x    -- 2 BYTES @NTCIP1202-v03B }

    where:
        n = spatEnabledLanesCount. This is the number of octets in the
            OCTET STRING of spatEnabledLanesStatus
        signalGroupEntryCount is a count of the number of
            signalGroupEntryNumber included in this object. Only
            signalGroupEntryNumbers that are active (in use) are to be
            included.
        y is the signalGroupEntryNumber in the signalGroupEntryTable
        x is the movementEventNumber for the signalStatusTable2. Only a
            value of 1 (current interval) and 2 (next interval) are used.

    For example, if (y = ) signalGroupEntryNumbers 1, 2, 4, 8, 13 and 16
        are included in this object, then signalGroupEntryCount.0 is equal
        to 6 and the SEQUENCE OF SignalStatusData2 is y = 1, 2, 4, 8, 13,
        and 16. x will still be 1, 2.

    Although this object theoretically allows up to 255
        signalGroupEntryNumbers, check that the number does not exceed the
        Maximum Transmission Unit (MTU) of the lower layers used.

    An example of signalStatusBlock2 is provided in Annex F.3.3.3.1.

    See Annex E.2 for the equivalent object definition for the RSU.
    Note: It is necessary that the signalGroupIntersection and
    signalGroupID objects be SET in the RSU before the signalStatusBlock2
    is SET to the RSU.

<Object Identifier> 1.3.6.1.4.1.1206.4.2.17.1.18"
::= { spat 18 }
```

Section 6

Block Object Definitions

BLOCK OBJECT DEFINITIONS

6.1 Block Data Type and ID

All ASC Block Objects shall begin with two octets that define the Data Type and Data ID.

The Data Type octet (ascBlockDataType) provides for the definition of both NTCIP Standard and Device Proprietary data blocks. NTCIP Standard Data Blocks shall utilize an 'ascBlockDataType' of zero. Device Proprietary Data Blocks shall utilize an 'ascBlockDataType' equal to the Private Node Number (PNN) as assigned by NEMA (1.3.6.1.4.1.1206.3.PNN). Each block provided is OER encoded.

dataType	Description
0x00	Standard Data Block
0XPNN	Device Proprietary Data Block

The Data ID octet (ascBlockDataID) provides for definition of included data parameters. NTCIP Standard Data Blocks shall include an 'ascBlockDataID' as listed below:

ascBlockData-dataID Definitions		
dataID	**Name**	**Description**
0x00	AscPhaseBlock	Phase Data (see 6.2)
0x01	AscVehDetectorBlock	Vehicle Detector Data (see 6.3)
0x02	AscPedDetectorBlock	Pedestrian Detector Data (see 6.4)
0x03	AscPatternBlock	Pattern Data (see 6.5)
0x04	AscSplitBlock	Split Data (see 6.6)
0x05	AscOverlapBlock	Overlap Data (see 6.7)
0x06	AscPreemptBlock	Preempt Data (see 6.8)
0x07	AscSequenceBlock	Sequence Data (see 6.9)
0x08	AscChannelBlock	Channel Data (see 6.10)
0x09 – 0x36	Deprecated	Reserved For NTCIP ASC Usage
0x37 - 0xFF		Reserved For NTCIP ASC Usage

New versions of this Standard shall NOT change the structure (content or definition) for any dataID block. New dataID blocks may be added for ascBlockData for expansion to cover other parameters. When a dataID block needs to be revised, the standard writers shall deprecate ascBlockData and establish a new OID (i.e., ascBlockData1) for all the current dataID blocks.

Proprietary Device Blocks shall include an 'ascBlockDataID' as defined in their separate documentation

6.2 Phase Block Data

```
-- ascBlockData values for standard Block
-- Phase Data shall be as follows:

AscPhaseBlock ::= SEQUENCE
      {
      ascBlockDataType  INTEGER (0..255),  -- 0x00 standard block
```

```
      ascBlockDataID    INTEGER (0..255),  -- 0x00 phase data
      ascBlockIndex1    INTEGER (0..255),  -- phaseNumber
      ascBlockQuantity1 INTEGER (0..255),  -- ## of phases
      ascBlockIndex2    INTEGER (0..255),  -- phaseSetNumber
      ascBlockQuantity2 INTEGER (0..255),  -- ## of phase set

--For ASCs that do no support multiple phase sets, the value of
ascBlockIndex2 is 1, the value of ascBlockQuanity2 is 1, and the phaseTable
is used instead of the phaseSetTable.

-- for (
 --      y = ascBlockIndex2;
 --      y < (ascBlockIndex2 + ascBlockQuantity2);
 --      y++)
 --      for (
 --     x = ascBlockIndex1;
 --     x < (ascBlockIndex1 + ascBlockQuantity1);
 --     x++)

 data    SEQUENCE OF AscPhaseBlockData
}

AscPhaseBlockData ::= SEQUENCE
      {
      phaseSetWalk.y.x                         Integer32 (0..255),
      phaseSetPedestrianClear.y.x              Integer32 (0..255),
      phaseSetMinimumGreen.y.x                 Integer32 (0..255),
      phaseSetPassage.y.x                      Integer32 (0..255),
      phaseSetMaximum1.y.x                     Integer32 (0..999),
      phaseSetMaximum2.y.x                     Integer32 (0..999),
      phaseSetYellowChange.y.x                 Integer32 (0..255),
      phaseSetRedClear.y.x                     Integer32 (0..255),
      phaseSetRedRevert.y.x                    Integer32 (0..255),
      phaseSetAddedInitial.y.x                 Integer32 (0..255),
      phaseSetMaximumInitial.y.x               Integer32 (0..255),
      phaseSetTimeBeforeReduction.y.x          Integer32 (0..255),
      phaseSetCarsBeforeReduction.y.x          Integer32 (0..255),
      phaseSetTimeToReduce.y.x                 Integer32 (0..255),
      phaseSetReduceBy.y.x                     Integer32 (0..255),
      phaseSetMinimumGap.y.x                   Integer32 (0..255),
      phaseSetDynamicMaxLimit.y.x              Integer32 (0..255),
      phaseSetDynamicMaxStep.y.x               Integer32 (0..255),
      phaseSetOptions.y.x                      Integer32 (0..65535),
      phaseSetMaximum3.y.x                     Integer32 (0..999),
      phaseSetPedClearDuringVehicleClear.y.x   Integer32 (0..255),
      phaseSetPedServiceLimit.y.x              Integer32 (0..255),
      phaseSetDontWalkRevert.y.x               Integer32 (0..255),
      phaseSetPedAlternateClearance.y.x        Integer32 (0..255),
      phaseSetPedAlternateWalk.y.x             Integer32 (0..255),
      phaseSetPedAdvanceWalkTime.y.x           Integer32 (0..255),
      phaseSetPedDelayWalkTime.y.x             Integer32 (0..255),
      phaseSetAdvWarnGrnStartTime.y.x          Integer32 (0..255),
      phaseSetAdvWarnRedStartTime.y.x          Integer32 (0..255),
      phaseSetWalkTimeTransition.y.x           Integer32 (0..255),
      phaseSetPedClearDurationTransition.y.x   Integer32 (0..255)
      }
```


6.2.1 Phase Block Example

```
-- The following provides an example octet string value for
-- a set or get of a phase block.
--
--    SEQUENCE
-- 00       ascBlockDataType (standard block)
-- 00       ascBlockDataID (phase data)
-- 02       ascBlockIndex1 (start with phaseNumber=2)
-- 02       ascBlockQuantity1 (## of phases = 2)
-- 01       ascBlockIndex2 (start with phaseSetNumber=1)
-- 01       ascBlockQuantity2 (## of phase sets = 1)

--    SEQUENCE OF
-- 02 01    quantity of items (ascBlockQuantity1 * ascBlockQuantity2)
--    SEQUENCE # 1 (phaseNumber=2 / phaseSetNumber=1)
-- 17       phaseSetWalk.1.2      (23 sec)
-- 0E       phaseSetPedestrianClear.1.2 (14 sec)
-- 0A       phaseSetMinimumGreen.1.2 (10 sec)
-- 1E       phaseSetPassageTime.1.2 (30 decisecond)
-- 00 25    phaseSetMaximum1.1.2 (37 sec)
--  |       etc, etc, to:
-- 0C       phaseSetPedClearDurationTransition.1.2 (12 sec)

--    SEQUENCE # 2 (phaseNumber=3 / phaseSetNumber=1)
-- 00       phaseWalk.1.3      (0 sec)
-- 00       phasePedestrianClear.1.3 (0 sec)
-- 07       phaseSetMinimumGreen.1.2 (7 sec)
-- 1E       phaseSetPassageTime.1.2 (30 decisecond)
-- 00 0D    phaseSetMaximum1.1.2 (13 sec)

--  |       etc, etc, to:
-- 00       phaseSetPedClearDurationTransition.1.2 (0 sec)
```

6.3 Vehicle Detector Block Data

```
-- ascBlockData values for standard Block
-- Vehicle Detector Data shall be as follows:

AscVehDetectorBlock ::= SEQUENCE
      {
      ascBlockDataType  INTEGER (0..255),  -- 0x00 standard block
      ascBlockDataID    INTEGER (0..255),  -- 0x01 veh detector data
      ascBlockIndex1    INTEGER (0..255),  -- vehicleDetectorNumber
      ascBlockQuantity1 INTEGER (0..255),  -- ## of veh detectors
      ascBlockIndex2    INTEGER (0..255),  -- vehicleDetectorSetNumber
      ascBlockQuantity2 INTEGER (0..255),  -- ## of veh detector sets

--For ASCs that do no support multiple vehicle detector sets, the value of
ascBlockIndex2 is 1 and the vehicleDetectorTable is used instead of the
vehicleDetectorSetTable.

-- for (
 --      y = ascBlockIndex2;
 --      y < (ascBlockIndex2 + ascBlockQuantity2);
 --      y++)
 --      for (
```

```
 --     x = ascBlockIndex1;
 --     x < (ascBlockIndex1 + ascBlockQuantity1);
 --     x++)

 data    SEQUENCE OF AscVehDetectorBlockData
}

AscVehDetectorBlockData ::= SEQUENCE
      {
      vehicleDetectorSetOptions.x         Integer32 (0..255),
      vehicleDetectorSetCallPhase.x       Integer32 (0..255),
      vehicleDetectorSetSwitchPhase.x     Integer32 (0..255),
      vehicleDetectorSetDelay.x           Integer32 (0..65535),
      vehicleDetectorSetExtend.x          Integer32 (0..255),
      vehicleDetectorSetQueueLimit.x      Integer32 (0..255),
      vehicleDetectorSetNoActivity.x      Integer32 (0..65535),
      vehicleDetectorSetMaxPresence.x     Integer32 (0..255),
      vehicleDetectorSetErraticCounts.x   Integer32 (0..255),
      vehicleDetectorSetFailTime.x        Integer32 (0..255)
      }
```

6.3.1 Vehicle Detector Block Example

```
-- The following provides an example octet string value for
-- a set or get of a vehicle detector block.
--
--    SEQUENCE
-- 00       ascBlockDataType (standard block)
-- 01       ascBlockDataID (veh detector data)
-- 02       ascBlockIndex1 (start with vehicleDetectorNumber=2)
-- 02       ascBlockQuantity1 (## of veh det=2)
-- 01       ascBlockIndex2 (start with vehicleDetectorSetNumber=1)
-- 01       ascBlockQuantity2 (## of veh det sets = 1)

--    SEQUENCE OF
-- 02 01    quantity of items (ascBlockQuantity1 * ascBlockQuantity2)
--    SEQUENCE # 1 (vehicleDetectorNumber=2 / vehicleDetectorSetNumber =1)
-- B4       vehicleDetectorSetOptions.2 (bits)
-- 02       vehicleDetectorSetCallSetPhase.2  (ph 2)
--  |       etc, etc, to:
-- 00       vehicleDetectorSetErraticCounts.2 (0 cpm)
-- FF       vehicleDetectorSetFailTime.2   (255 sec)
--    SEQUENCE # 2 (vehicleDetectorNumber=3 / vehicleDetectorSetNumber =1)
-- B4       vehicleDetectorSetOptions.3 (bits)
-- 03       vehicleDetectorSetCallPhase.3   (ph 3)
--  |       etc, etc, to:
-- 00       vehicleDetectorSetErraticCounts.3 (0 cpm)
-- FF       vehicleDetectorSetFailTime.3      (255 sec)
```

6.4 Pedestrian Detector Block Data

```
-- ascBlockData values for standard Block
-- Pedestrian Detector Data shall be as follows:

AscPedDetectorBlock ::= SEQUENCE
      {
```



```
      ascBlockDataType  INTEGER (0..255),  -- 0x00 standard block
      ascBlockDataID    INTEGER (0..255),  -- 0x02 ped detector data
      ascBlockIndex1    INTEGER (0..255),  -- pedestrianDetectorNumber
      ascBlockQuantity1 INTEGER (0..255),  -- ## of ped detectors
      ascBlockIndex2    INTEGER (0..255),  -- pedestrianDetectorSetNumber
      ascBlockQuantity2 INTEGER (0..255),  -- ## of ped detector sets

--For ASCs that do no support multiple pedestrian detector sets, the value of
ascBlockIndex2 is 1 and the pedestrianDetectorTable is used instead of the
pedestrianDetectorSetTable.

-- for (
 --      y = ascBlockIndex2;
 --      y < (ascBlockIndex2 + ascBlockQuantity2);
 --      y++)
 --      for (
 --     x = ascBlockIndex1;
 --     x < (ascBlockIndex1 + ascBlockQuantity1);
 --     x++)

 data    SEQUENCE OF AscPedDetectorBlockData
}

AscPedDetectorBlockData ::= SEQUENCE
      {
      pedestrianSetDetectorCallPhase.x          Integer32 (0..255),
      pedestrianSetDetectorNoActivity.x         Integer32 (0..65535),
      pedestrianSetDetectorMaxPresence.x        Integer32 (0..255),
      pedestrianSetDetectorErraticCounts.x      Integer32 (0..255),
      pedestrianSetDetectorOptions.x            Integer32 (0..255)
      }
```

6.4.1 Pedestrian Detector Block Example

```
-- The following provides an example octet string value for
-- a set or get of a pedestrian detector block.
--
--    SEQUENCE
-- 00       ascBlockDataType (standard block)
-- 02       ascBlockDataID (ped detector data)
-- 02       ascBlockIndex1 (start with pedestrianDetectorNumber=2)
-- 02       ascBlockQuantity1 (## of ped det=2)
-- 01       ascBlockIndex2 (start with pedestrianDetectorSetNumber=1)
-- 01       ascBlockQuantity2 (## of ped det sets = 1)

--    SEQUENCE OF
-- 01 02    quantity of items (ascBlockQuantity1)
--    SEQUENCE # 1 (pedestrianDetectorNumber =2)
-- 02       pedestrianDetectorCallPhase.2     (ph 2)
-- 00 00    pedestrianDetectorNoActivity.2    (0 min)
-- 00       pedestrianDetectorMaxPresence.2   (0 min)
-- 00       pedestrianDetectorErraticCounts.2 (0 cpm)
-- 00       pedestrianDetectorOptions.2 (bits)

--    SEQUENCE # 2 (pedestrianDetectorNumber =3)
-- 03       pedestrianDetectorCallPhase.3     (ph 3)
-- 00 00    pedestrianDetectorNoActivity.3    (0 min)
```

```
-- 00       pedestrianDetectorMaxPresence.3   (0 min)
-- 00       pedestrianDetectorErraticCounts.3 (0 cpm)
-- 00       pedestrianDetectorOptions.3 (bits)
```

6.5 Pattern Block Data

```
-- ascBlockData values for standard Block
-- Pattern Data shall be as follows:

AscPatternBlock ::= SEQUENCE
      {
       ascBlockDataType       INTEGER (0..255),  -- 0x00 standard block
       ascBlockDataID         INTEGER (0..255),  -- 0x03 pattern data
       ascBlockIndex1         INTEGER (0..255),  -- patternNumber
       ascBlockQuantity1      INTEGER (0..255),  -- ## of patterns

       -- for (
       --      x = ascBlockIndex1;
       --      x < (ascBlockIndex1 + ascBlockQuantity1);
       --      x++)

       data    SEQUENCE OF AscPatternBlockData
      }

AscPatternBlockData ::= SEQUENCE
      {
       patternCycleTime.x           Integer32 (0..999),
       patternOffsetTime.x          Integer32 (0..999),
      patternSplitNumber.x          Integer32 (0..255)
      patternSequenceNumber.x       Integer32 (0..255),
      patternReferencePoint.x       INTEGER,
      patternMaximumMode.x          INTEGER,
      patternReferencePhase         Integer32 (0..255),
      patternPhaseSet.x             Integer32 (1..255),
      patternOverlapSet.x           Integer32 (1..255),
      patternVehicleDetectorSet.x   Integer32 (1..255),
      patternPedestrianDetectorSet.x Integer32 (1..255),
      patternSpecialFunction.x      Unsigned32 (0..4294967295)
      }
```

6.5.1 Pattern Block Example

```
-- The following provides an example octet string value for
-- a set or get of a pattern block.
--
--    SEQUENCE
-- 00       ascBlockDataType (standard block)
-- 03       ascBlockDataID (pattern data)
-- 02       ascBlockIndex1 (start with patternNumber=2)
-- 02       ascBlockQuantity1 (## of patterns=2)
--    SEQUENCE OF
-- 01 02    quantity of items (ascBlockQuantity1)
--    SEQUENCE # 1 (patternNumber =2)
-- 00 5A          patternCycleTime.2      (90 sec)
-- 00 2D          patternOffsetTime.2     (45 sec)
            etc, etc, to:
-- 00 00 00 01    patternSpecialFunction.2   (special function 1)
```

```
--    SEQUENCE # 2 (patternNumber =3)
-- 00 78          patternCycleTime.3      (120 sec)
-- 00 71          patternOffsetTime.3     (113 sec)
etc, etc, to:

-- 00 00 00 02    patternSpecialFunction.3   (special function 2)
```

6.6 Split Block Data

```
-- ascBlockData values for standard Block
-- Split Data shall be as follows:

AscSplitBlock ::= SEQUENCE
      {
       ascBlockDataType       INTEGER (0..255),  -- 0x00 standard block
       ascBlockDataID         INTEGER (0..255),  -- 0x04 split data
       ascBlockIndex1         INTEGER (0..255),  -- splitPhase
       ascBlockQuantity1      INTEGER (0..255),  -- ## of phases
       ascBlockIndex2         INTEGER (0..255),  -- splitNumber
       ascBlockQuantity2      INTEGER (0..255),  -- ## of splits

       -- for (
       --      y = ascBlockIndex2;
       --      y < (ascBlockIndex2 + ascBlockQuantity2);
       --      y++)
       --      for (
       --     x = ascBlockIndex1;
       --     x < (ascBlockIndex1 + ascBlockQuantity1);
       --     x++)

       data    SEQUENCE OF AscSplitBlockData
      }

AscSplitBlockData ::= SEQUENCE
      {
       splitTime.y.x          Integer32 (0..999),
       splitMode.y.x          INTEGER,
       splitCoordPhase.y.x    INTEGER ,
      splitOptions.y.x        Integer32 (0..255),
      splitCoordForceMode     INTEGER
      }
```

6.6.1 Split Block Example

```
-- The following provides an example octet string value for
-- a set or get of a split block.
--
--    SEQUENCE
-- 00       ascBlockDataType (standard block)
-- 04       ascBlockDataID (split data)
-- 01       ascBlockIndex1 (start with splitPhase=1)
-- 02       ascBlockQuantity1 (## of phases=2)
-- 01       ascBlockIndex2 (start with splitNumber=1)
-- 02       ascBlockQuantity2 (## of splits=2)
--    SEQUENCE OF
-- 01 04    quantity of items (ascBlockQuantity1 * ascBlockQuantity2)
--    SEQUENCE # 1 (splitNumber=1 / splitPhase=1)
```



```
-- 00 14    splitTime.1.1 (20 sec)
-- 02       splitMode.1.1 (none)
-- 00       splitCoordPhase.1.1 (false)
-- 00       splitOptions.1.1 (bits)
-- 03       splitCoordForceMode.1.1 (floating)
--     SEQUENCE # 2 (splitNumber=1 / splitPhase=2)
-- 00 14    splitTime.1.2 (20 sec)
-- 02       splitMode.1.2 (none)
-- 01       splitCoordPhase.1.2 (true)
-- 00       splitOptions.1.2 (bits)
-- 02       splitCoordForceMode.2.2 (unitCoordForceMode)

--     SEQUENCE # 3 (splitNumber=2 / splitPhase=1)
-- 00 19    splitTime.2.1 (25 sec)
-- 02       splitMode.2.1 (none)
-- 00       splitCoordPhase.2.1 (false)
-- 00       splitOptions.2.1 (bits)
-- 03       splitCoordForceMode.2.1 (floating)

--     SEQUENCE # 4 (splitNumber=2 / splitPhase=2)
-- 00 19    splitTime.2.2 (25 sec)
-- 02       splitMode.2.2 (none)
-- 01       splitCoordPhase.2.2 (true)
-- 00       splitOptions.2.2 (bits)
-- 02       splitCoordForceMode.2.2 (unitCoordForceMode)
```

6.7 Overlap Block Data

```
-- ascBlockData values for standard Block
-- Overlap Data shall be as follows:

AscOverlapBlock ::= SEQUENCE
      {
       ascBlockDataType       Integer32 (0..255),  -- 0x00 standard block
       ascBlockDataID         Integer32 (0..255),  -- 0x09 overlap data
       ascBlockIndex1         Integer32 (0..255),  -- overlapNumber
       ascBlockQuantity1      Integer32 (0..255),  -- ## of overlaps
       ascBlockIndex2         Integer32 (0..255),  -- overlapSetNumber
       ascBlockQuantity2      Integer32 (0..255),  -- ## of overlap sets

--For ASCs that do no support multiple overlap sets, the value of
ascBlockIndex2 is 1 and the overlapTable is used instead of the
overlapSetTable.

-- for (
 --      y = ascBlockIndex2;
 --      y < (ascBlockIndex2 + ascBlockQuantity2);
 --      y++)
 --      for (
 --     x = ascBlockIndex1;
 --     x < (ascBlockIndex1 + ascBlockQuantity1);
 --     x++)

 data    SEQUENCE OF AscOverlapBlockData
}

AscOverlapBlockData ::= SEQUENCE
```

```
      {
      overlapSetIncludedPhases.y.x        OCTET STRING,
      overlapSetModifierPhases.y.x        OCTET STRING,
      overlapSetTrailGreen.y.x            Integer32 (0..255),
      overlapSetTrailYellow.y.x           Integer32 (0..255),
      overlapSetTrailRed.y.x              Integer32 (0..255),
      overlapSetWalk.y.x                  Integer32 (0..255),
      overlapSetPedClearance.y.x          Integer32 (0..255),
      overlapSetConflictingPedPhases.y.x  OCTET STRING
      }
```

6.7.1 Overlap Block Example

```
-- The following provides an example octet string value for
-- a SET or GET of a overlap block.
--
--    SEQUENCE
-- 00       ascBlockDataType (standard block)
-- 09       ascBlockDataID (overlap data)
-- 02       ascBlockIndex1 (start with overlapNumber=2)
-- 02       ascBlockQuantity1 (## of overlaps=2)
--    SEQUENCE OF
-- 01 02    quantity of items (ascBlockQuantity1)
--    SEQUENCE # 1 (overlapNumber=2)
-- 02 02 03 overlapSetIncludedPhases.2 (ph 2 & 3)
-- 00       overlapSetModifierPhases.2 (none)
-- 00       overlapSetTrailGreen.2     (0 sec)
-- 23       overlapSetTrailYellow.2    (35 deciseconds)
-- 05       overlapSetTrailRed.2 (0.5 sec)
--          etc, etc, to:
--01 06     overlapSetConflictingPedPhases (ph 6)

--    SEQUENCE # 2 (overlapNumber=3)
-- 02 04 05 overlapSetIncludedPhases.3 (ph 4 & 5)
-- 00       overlapSetModifierPhases.3 (none)
-- 00       overlapSetTrailGreen.3     (0 sec)
-- 23       overlapSetTrailYellow.3    (35 deciseconds)
-- 05       overlapSetTrailRed.3 (0.5 sec)
--          etc, etc, to:
--01 08     overlapSetConflictingPedPhases (ph 8)
```

6.8 Preempt Block Data

```
-- ascBlockData values for standard Block
-- Preempt Data shall be as follows:

AscPreemptBlock ::= SEQUENCE
      {
       ascBlockDataType       Integer32 (0..255),  -- 0x00 standard block
       ascBlockDataID         Integer32 (0..255),  -- 0x06 preempt data
       ascBlockIndex1         Integer32 (0..255),  -- preemptNumber
       ascBlockQuantity1      Integer32 (0..255),  -- ## of preempts

       -- for (
       --      x = ascBlockIndex1;
       --      x < (ascBlockIndex1 + ascBlockQuantity1);
       --      x++)
```

```
        data    SEQUENCE OF AscPreemptBlockData
       }

AscPreemptBlockData ::= SEQUENCE
       {
        preemptControl.x             Integer32 (0..255),
        preemptLink.x                Integer32 (0..255),
        preemptDelay.x               Integer32 (0..65535),
        preemptMinimumDuration.x     Integer32 (0..65535),
        preemptMinimumGreen.x        Integer32 (0..255),
        preemptMinimumWalk.x         Integer32 (0..255),
        preemptEnterPedClear.x       Integer32 (0..255),
        preemptTrackGreen.x          Integer32 (0..255),
        preemptDwellGreen.x          Integer32 (0..255),
        preemptMaximumPresence.x     Integer32 (0..65535),
        preemptTrackPhase.x          OCTET STRING,
        preemptDwellPhase.x          OCTET STRING,
        preemptDwellPed.x            OCTET STRING,
        preemptExitPhase.x           OCTET STRING,
        preemptTrackOverlap.x        OCTET STRING,
        preemptDwellOverlap.x        OCTET STRING,
        preemptCyclingPhase.x        OCTET STRING,
        preemptCyclingPed.x          OCTET STRING,
        preemptCyclingOverlap.x      OCTET STRING,
        preemptEnterYellowChange.x   Integer32 (0..255),
        preemptEnterRedClear.x       Integer32 (0..255),
        preemptTrackYellowChange.x   Integer32 (0..255),
        preemptTrackRedClear.x       Integer32 (0..255),
       preemptSequenceNumber.x       Integer32 (0..255),
       preemptExitType.x             INTEGER
       }
```

6.8.1 Preempt Block Example

```
-- The following provides an example octet string value for
-- a set or get of a preempt block.
--
--     SEQUENCE
-- 00        ascBlockDataType (standard block)
-- 06        ascBlockDataID (preempt data)
-- 02        ascBlockIndex1 (start with preemptNumber =2)
-- 02        ascBlockQuantity1 (## of preempts=2)

--     SEQUENCE OF
-- 01 02     quantity of items (ascBlockQuantity1)
--     SEQUENCE # 1 (preemptNumber =2)
-- 05        preemptControl.2  (bits)
-- 00        preemptLink.2     (none)
--  |        etc, etc, to:
-- 28        preemptTrackYellowChange.2 (4.0 Sec)
-- 00        preemptTrackRedClear.2     (  0 Sec)
--     SEQUENCE # 2 (preemptNumber =3)
-- 05        preemptControl.3  (bits)
-- 01        preemptLink.3     (pe 1)
--  |        etc, etc, to:
-- 28        preemptTrackYellowChange.3 (4.0 Sec)
```

```
-- 00       preemptTrackRedClear.3     (  0 Sec)
```

6.9 Sequence Block Data

```
-- ascBlockData values for standard Block
-- Sequence Data shall be as follows:

AscSequenceBlock ::= SEQUENCE
      {
       ascBlockDataType       Integer32 (0..255),  -- 0x00 standard block
       ascBlockDataID         Integer32 (0..255),  -- 0x07 sequence data
       ascBlockIndex1         Integer32 (0..255),  -- sequenceRingNumber
       ascBlockQuantity1      Integer32 (0..255),  -- ## of rings
       ascBlockIndex2         Integer32 (0..255),  -- sequenceNumber
       ascBlockQuantity2      Integer32 (0..255),  -- ## of sequences

       -- for (
       --      y = ascBlockIndex2;
       --      y < (ascBlockIndex2 + ascBlockQuantity2);
       --      y++)
       --      for (
       --     x = ascBlockIndex1;
       --     x < (ascBlockIndex1 + ascBlockQuantity1);
       --     x++)

       data    SEQUENCE OF AscSequenceBlockData
      }

AscSequenceBlockData ::= SEQUENCE
      {
       sequenceData.y.x       OCTET STRING
      }
```

6.9.1 Sequence Block Example

```
-- The following provides an example octet string value for
-- a set or get of a sequence block.
--
--    SEQUENCE
-- 00       ascBlockDataType (standard block)
-- 07       ascBlockDataID (sequence data)
-- 01       ascBlockIndex1 (start with sequenceRingNumber=1)
-- 02       ascBlockQuantity1 (## of rings=2)
-- 01       ascBlockIndex2 (start with sequenceNumber=1)
-- 02       ascBlockQuantity2 (## of sequences =2)
--    SEQUENCE OF
-- 01 04    quantity of items (ascBlockQuantity1 * ascBlockQuantity2)
--    SEQUENCE # 1 (sequenceNumber=1 / sequenceRingNumber=1)
-- 04 01 02 03 04       sequenceData.1.1 (ph 1-2-3-4)
--    SEQUENCE # 2 (sequenceNumber=1 / sequenceRingNumber=2)
-- 04 05 06 07 08       sequenceData.1.2 (ph 5-6-7-8)
--    SEQUENCE # 3 (sequenceNumber=2 / sequenceRingNumber=1)
-- 04 02 01 04 03       sequenceData.2.1 (ph 1-2-3-4)
--    SEQUENCE # 4 (sequenceNumber=2 / sequenceRingNumber=2)
-- 04 06 05 08 07       sequenceData.2.2 (ph 5-6-7-8)
```


6.10 Channel Block Data

```
-- ascBlockData values for standard Block
-- Channel Data shall be as follows:

AscChannelBlock ::= SEQUENCE
      {
       ascBlockDataType       Integer32 (0..255),  -- 0x00 standard block
       ascBlockDataID         Integer32 (0..255),  -- 0x08 channel data
       ascBlockIndex1         Integer32 (0..255),  -- channelNumber
       ascBlockQuantity1      Integer32 (0..255),  -- ## of channels

       -- for (
       --      x = ascBlockIndex1;
       --      x < (ascBlockIndex1 + ascBlockQuantity1);
       --      x++)

       data    SEQUENCE OF AscChannelBlockData
      }

AscChannelBlockData ::= SEQUENCE
      {
       channelControlSource.x Integer32 (0..255),
       channelControlType.x   INTEGER,
       channelFlash.x         Integer32 (0..255)
      }
```

6.10.1 Channel Block Example

```
-- The following provides an example octet string value for
-- a SET or GET of a channel block.
--
--    SEQUENCE
-- 00       ascBlockDataType (standard block)
-- 08       ascBlockDataID (channel data)
-- 02       ascBlockIndex1 (start with channelNumber=2)
-- 02       ascBlockQuantity1 (## of channels=2)
--    SEQUENCE OF
-- 01 02    quantity of items (ascBlockQuantity1)
--    SEQUENCE # 1 (channelNumber=2)
-- 02       channelControlSource.2 (ph 2)
-- 02       channelControlType.2   (phaseVehicle)
-- 02       channelFlash.2   (bits)
--    SEQUENCE # 2 (channelNumber=3)
-- 03       channelControlSource.3 (ph 3)
-- 02       channelControlType.3   (phaseVehicle)
-- 04       channelFlash.3   (bits)
```

Annex A

Requirements Traceability Matrix (RTM) [Normative]

The Requirements Traceability Matrix (RTM) links the Functional Requirements as presented in Section 3 with the corresponding Dialogs (Section 4.2) on the same (gray) line. Each Functional Requirement/Dialog relates/uses one or more groups of Objects. The Objects (also known as Data Elements) are listed to the side; the formal definition of each object is contained within Section 5. Using this table, each Functional Requirement can thus be traced in a standardized way.

Note: The INDEX objects into any of the tables are not explicitly exchanged but are used as index values for other objects that are exchanged.

The audience for this table is implementers (vendors and central system developers) and conformance testers. Additionally, other interested parties might use this table to determine how particular functions are to be implemented using the standardized dialogs, interfaces, and object definitions.

To conform to a requirement, an ASC system shall implement all objects traced from that requirement; and unless otherwise indicated, shall implement all dialogs traced from the requirement. To be consistent with a requirement, an ASC system shall be able to fulfill the requirement using only objects that a conforming ASC system is required to support.

Section 3 defines Supplemental Requirements, which are refining other functional requirements. These functional requirements in turn are generally traced to design elements (e.g., rather than being directly traced to design elements).

Note: Visit www.ntcip.org for information on availability of electronic copies of the RTM.

A.1 Notation [Informative]

A.1.1 Functional Requirement Columns

The functional requirements are defined within Section 3 and the RTM is based upon the requirements within that Section. The section number and the functional requirement name are indicated within these columns.

A.1.2 Dialog Column

The standardized dialogs are defined in ISO 26048-1 (26048-1) and within Section 4; and the RTM references the traces from requirements to this dialog. The section number of the dialog is indicated within this column.

A.1.3 Object Columns

The objects are primarily defined within Section 5 of NTCIP 1202. If the data object is not defined by NTCIP 1202, the standard or the name of the MIB where the object definition is found is listed under the Object ID column. The object identifier (if applicable) and object name are indicated within these columns. The RTM references the data objects that are referenced by the dialog. For example, ISO26048-1-Transaction is ISO26048-1-Transaction.mib.

A.1.4 Additional Specifications

The "Additional Specifications" column may (and should) be used to provide additional notes and requirements about the dialog or may be used by an implementer to provide any additional details about the implementation.

A.2 Instructions For Completing the RTM [Informative]

To find the standardized design content for a functional requirement, search for the requirement identification number and functional requirement under the functional requirements columns. Next to the functional requirements column is a dialog identification number, identifying either a generic dialog (defined in ISO 26048-1) or a specified dialog (found in Section 4.2) to be used to fulfill that requirement. To the right of the dialog identification number are the identification number and name of the data objects that are referenced or used by the dialog to fulfill the functional requirement. Object definitions specific to NTCIP 1202 v04 can be found in Section 5. If an object is defined in a different standard, that standard shall be listed first, followed by the section number where the object definition can be found. The "Additional Specifications" column provides additional notes or details about the design content.

A.3 Requirements Traceability Matrix (RTM) Table

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.4	Architectural Requirements				
3.4.1	Support Basic Communications Requirements				
3.4.1.1	Retrieve Data	26048-1§9.2.1			
3.4.1.2	Deliver Data	26048-1§9.2.2			
3.4.1.3	Explore Data	26048-1§9.2.3			
3.4.1.4	Monitor SNMP Requirements				Also see ISO 26048-1 Section 8.5
3.4.1.4.1	Monitor SNMP Information	26048-1§9.2.1			
			RFC 3418	snmpInPkts	
			RFC 3418	snmpInBadVersions	
			RFC 3418	snmpInBadCommunityNames	
			RFC 3418	snmpInBadCommunityUses	
			RFC 3418	snmpInASNParseErrs	
			RFC 3418	snmpSilentDrops	
			RFC 3418	snmpProxyDrops	
3.4.2	Manage Data Blocks Requirements				Also see ISO 26048-1 Section 8.7
3.4.2.1	Store Pre-Defined Compressed Data Blocks				

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.4.3	Support Logged Data Requirements				Also see ISO 26048-1 Sections 8.9 & 8.10.
3.4.4	Support Database Management Requirements				Also see ISO 26048-1 Section 8.21
3.4.5	Support Condition-based Exception Reporting Requirements				Also see ISO 26048-1 Sections 8.1, 8.11, and 8.16
3.5	Data Exchange and Operational Environment Requirements				
3.5.1	ASC Configuration Management Requirements				
3.5.1.1	Manage ASC Information Requirements				Also see ISO 26048-1 Sections 8.5, 8.8, and 8.12.
3.5.1.1.1	Configure ASC Location - Antenna Offset	26048-1§9.2.2			
			5.4.13	ascElevationOffset	ISO supports fdGnssLatitude, fdGnssLongitude, fdGnssElevation
3.5.1.2	Manage Communications Requirements				
3.5.1.2.1	Configure Communications Requirements				
3.5.1.2.1.1	Enable/Disable Communications Port	26048-1§9.2.6			
			5.4.15.1	maxCommPorts	
			5.4.15.2	commPortsTable	
			RFC 1213	ifIndex	
			5.4.15.2.1	commPortType	
			5.4.15.2.2	commPortTypeIndex	
			5.4.15.2.3	commPortEnable	
3.5.1.2.2	Retrieve Communications Requirements				
3.5.1.2.2.1	Determine Number of ASC Communications Ports	26048-1§9.2.1			
			5.4.15.1	maxCommPorts	
3.5.1.3	Manage Cabinet Environment Requirements				Also see ISO 26048-1 Section 8.8

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.1.4	Monitor Power Source Requirements				Also see ISO 26048-1 Section 8.8
3.5.1.5	Manage Operational Performance Data Requirements				Also see ISO 26048-1 Section 8.13
3.5.1.6	Manage Auxiliary External Inputs/Outputs Requirements				Also see ISO 26048-1 Section 8.18
3.5.1.7	Manage Database Operations				Also see ISO 26048-1 Section 8.5
3.5.1.7.1	Determine Configuration Identifier Parameter Content				Also see ISO 26048-1 Section 8.5.2.14
3.5.1.8	Manage Interface with External Detectors Requirements				Also see NTCIP 1209
3.5.1.9	Manage ASC Clock Requirements				See ISO 26048-1 Section 8.2
3.5.1.10	Manage External Control Local Application State Requirements				See NTCIP 1209
3.5.1.10.1	Manage ECLA Interface Requirements				
3.5.1.10.1.1	Enable ECLA Communications	26048-1§9.2.2			
			5.18.1	eclaCommEnable	Value = 1
3.5.1.10.1.2	Disable ECLA Communications	26048-1§9.2.2			
			5.4.19	unitManualBackup	Value = 1
			5.18.1	eclaCommEnable	Value = 0
3.5.1.10.2	Monitor ECLA Data Input Time	26048-1§9.2.1			
			5.18.2	eclaDataTimestamp	
3.5.2	Manage Signal Operations Management Requirements				
3.5.2.1	Manage Signal Configuration Requirements				
3.5.2.1.1	Manage Unit Configuration Requirements				
3.5.2.1.1.1	Manage Start-Up Flash Requirements				

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.1.1.1	Configure Start-Up Flash Mode	26048-1§9.2.2			
			5.4.14	unitStartUpFlashMode	
3.5.2.1.1.1.2	Configure Start-Up Flash Time	26048-1§9.2.2			
			5.4.1	unitStartUpFlash	
3.5.2.1.1.2	Configure Backup Time	26048-1§9.2.2			
			5.4.3	unitBackupTime	
3.5.2.1.2	Manage Phase Configuration Requirements				
3.5.2.1.2.1	Configure Phase Requirements				
3.5.2.1.2.1.1	Enable/Disable Phase	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 0
3.5.2.1.2.1.2	Configure Phase Minimum Green Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.4	phaseMinimumGreen	
3.5.2.1.2.1.3	Configure Phase Passage Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.5	phasePassage	
3.5.2.1.2.1.4	Configure Two Fixed Phase Maximum Green Times	26048-1§9.2.6			
			5.2.1	maxPhases	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.6	phaseMaximum1	
			5.2.2.7	phaseMaximum2	
3.5.2.1.2.1.5	Configure Three Fixed Phase Maximum Green Times	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.6	phaseMaximum1	
			5.2.2.7	phaseMaximum2	
			5.2.2.24	phaseMaximum3	
3.5.2.1.2.1.6	Configure Phase Yellow Change Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.8	phaseYellowChange	
3.5.2.1.2.1.7	Configure Phase Red Clearance Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.9	phaseRedClear	
3.5.2.1.2.1.8	Configure Phase Red Revert Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.10	phaseRedRevert	
3.5.2.1.2.1.9	Configure Unit Red Revert Time	26048-1§9.2.2			
			5.4.4	unitRedRevert	
3.5.2.1.2.1.10	Configure Phase Added Initial Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.2.2.11	phaseAddedInitial	
3.5.2.1.2.1.11	Configure Phase Maximum Initial Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.12	phaseMaximumInitial	
3.5.2.1.2.1.12	Configure Phase Time Before Reduction	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.13	phaseTimeBeforeReduction	
3.5.2.1.2.1.13	Configure Phase Time to Reduce	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.15	phaseTimeToReduce	
3.5.2.1.2.1.14	Configure Phase Cars Before Reduction	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.14	phaseCarsBeforeReduction	
3.5.2.1.2.1.15	Configure Phase Reduce By Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.16	phaseReduceBy	
3.5.2.1.2.1.16	Configure Phase Minimum Gap Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.17	phaseMinimumGap	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.2.1.17	Configure Phase Dynamic Maximum Limit	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.18	phaseDynamicMaxLimit	
3.5.2.1.2.1.18	Configure Phase Dynamic Maximum Step	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.19	phaseDynamicMaxStep	
3.5.2.1.2.1.19	Configure Phase Start-Up State	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.20	phaseStartup	
3.5.2.1.2.1.20	Configure Automatic Flash Entry Phase	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 1
3.5.2.1.2.1.21	Configure Automatic Flash Exit Phase	4.2.2			

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 2
3.5.2.1.2.1.22	Configure Call to Non-Actuated 1	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 3
3.5.2.1.2.1.23	Configure Call to Non-Actuated 2	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 4
3.5.2.1.2.1.24	Configure Non-Lock Detector Memory	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 5
3.5.2.1.2.1.25	Configure Phase Minimum Vehicle Recall	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 6
3.5.2.1.2.1.26	Configure Phase Maximum Vehicle Recall	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 7
3.5.2.1.2.1.27	Configure Phase Soft Vehicle Recall	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 9
3.5.2.1.2.1.28	Configure Dual Phase Entry	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 10
3.5.2.1.2.1.29	Configure Simultaneous Gap Disable	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 11
3.5.2.1.2.1.30	Configure Guaranteed Passage	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 12
3.5.2.1.2.1.31	Configure Actuated Rest-in-Walk	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 13
3.5.2.1.2.1.32	Configure Conditional Service Enable	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 14
3.5.2.1.2.1.33	Configure Added Initial Calculation	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.2.2.21	phaseOptions	Bit = 15
3.5.2.1.2.1.34	Configure Phase-to-Ring Association	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.22	phaseRing	
3.5.2.1.2.1.35	Configure Phase Concurrency	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.23	phaseConcurrency	
3.5.2.1.2.1.36	Configure Pedestrian Clearance Time Allowed During Vehicle Clearance	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.25	phasePedClearDuringVehicleClear	
3.5.2.1.2.1.37	Configure Pedestrian Walk Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.2	phaseWalk	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.2.1.38	Configure Pedestrian Clearance Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.3	phasePedestrianClear	
3.5.2.1.2.1.39	Configure Pedestrian Phase Walk Service Limit	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.26	phasePedServiceLimit	
3.5.2.1.2.1.40	Configure Pedestrian Phase Don't Walk Revert Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.27	phaseDontWalkRevert	
3.5.2.1.2.1.41	Configure Non-Lock Ped Detector Memory	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 5
3.5.2.1.2.1.42	Configure Pedestrian Phase Recall	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.21	phaseOptions	Bit = 8
3.5.2.1.2.1.43	Configure Phase Alternate Pedestrian Clearance Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.28	phasePedAlternateClearance	
3.5.2.1.2.1.44	Configure Phase Alternate Pedestrian Walk Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.29	phasePedAlternateWalk	
3.5.2.1.2.1.45	Configure Pedestrian Phase Advanced Walk Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.30	phasePedAdvanceWalkTime	
3.5.2.1.2.1.46	Configure Pedestrian Phase Delayed Walk Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.31	phasePedDelayTime	
3.5.2.1.2.1.47	Configure Phase Advance Warning Green	26048-1§9.2.6			See Requirement 3.5.2.1.11.1.2.3.3, Object ID 5.14.10, ascIOmapOutputFunctions - advWarnGrn (4)
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.32	phaseAdvWarnGrnStartTime	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.2.1.48	Configure Phase Advance Warning Red				See Requirement 3.5.2.1.11.1.2.3.3, Object ID 5.14.10, ascIOmapOutputFunctions - advWarnRed (5)
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.33	phaseAdvWarnRedStartTime	
3.5.2.1.2.1.49	Configure Flashing Yellow Arrow Associated Vehicle Phase	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	
			5.10.2.3	overlapIncludedPhases	
			5.10.2.4	overlapModifierPhases	
3.5.2.1.2.1.50	Configure Flashing Red Arrow Associated Vehicle Phase	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	
			5.10.2.3	overlapIncludedPhases	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.10.2.4	overlapModifierPhases	
3.5.2.1.2.1.51	Configure Alternate Minimum Green Time during Transition	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.34	phaseAltMinTimeTransition	
3.5.2.1.2.1.52	Configure Alternate Minimum Walk Time during Transition	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.35	phaseWalkDuringTransition	
3.5.2.1.2.1.53	Configure Alternate Minimum Pedestrian Clearance Time during Transition	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.36	phasePedClearDuringTransition	
3.5.2.1.2.2	Configure Multiple Phase Sets	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.6	maxPhaseSets	
			5.2.7	phaseSetTable	
			5.2.7.1	phaseSetNumber	
			5.2.2.1	phaseNumber	
			5.2.7.2	phaseSetWalk	
			5.2.7.3	phaseSetPedestrianClear	
			5.2.7.4	phaseSetMinimumGreen	
			5.2.7.5	phaseSetPassage	
			5.2.7.6	phaseSetMaximum1	
			5.2.7.7	phaseSetMaximum2	
			5.2.7.8	phaseSetYellowChange	
			5.2.7.9	phaseSetRedClear	
			5.2.7.10	phaseSetRedRevert	
			5.2.7.11	phaseSetAddedInitial	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.2.7.12	phaseSetMaximumInitial	
			5.2.7.13	phaseSetTimeBeforeReduction	
			5.2.7.14	phaseSetCarsBeforeReduction	
			5.2.7.15	phaseSetTimeToReduce	
			5.2.7.16	phaseSetReduceBy	
			5.2.7.17	phaseSetMinimumGap	
			5.2.7.18	phaseSetDynamicMaxLimit	
			5.2.7.19	phaseSetDynamicMaxStep	
			5.2.7.20	phaseSetOptions	
			5.2.7.21	phaseSetMaximum3	
			5.2.7.22	phaseSetPedClearDuringVehicleClear	
			5.2.7.23	phaseSetPedServiceLimit	
			5.2.7.24	phaseSetDontWalkRevert	
			5.2.7.25	phaseSetPedAlternateClearance	
			5.2.7.26	phaseSetPedAlternateWalk	
			5.2.7.27	phaseSetPedAdvanceWalkTime	
			5.2.7.28	phaseSetPedDelayTime	
			5.2.7.29	phaseSetAdvWarnGrnStartTime	
			5.2.7.30	phaseSetAdvWarnRedStartTime	
			5.2.7.31	phaseSetAltMinTimeTransition	
			5.2.7.32	phaseSetWalkDuringTransition	
			5.2.7.33	phaseSetPedClearDuringTransition	
3.5.2.1.2.3	Retrieve Phase Configuration Requirements				
3.5.2.1.2.3.1	Determine Maximum Number of Phases	26048-1§9.2.1			
			5.2.1	maxPhases	
3.5.2.1.2.3.2	Determine Maximum Number of Phase Sets	26048-1§9.2.1			
			5.2.6	maxPhaseSets	
3.5.2.1.3	Manage Coordination Configuration Requirements				
3.5.2.1.3.1	Configure Operational Mode for Coordination	26048-1§9.2.2			
			5.5.1	coordOperationalMode	
3.5.2.1.3.2	Configure Correction Mode for Coordination	26048-1§9.2.2			
			5.5.2	coordCorrectionMode	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.3.3	Configure Maximum Mode for Coordination	26048-1§9.2.2			
			5.5.3	coordMaximumMode	
3.5.2.1.3.4	Configure Unit-Level Force Mode for Coordination	26048-1§9.2.2			
			5.5.4	unitCoordForceMode	
3.5.2.1.3.5	Configure Phase-Level Force Mode for Coordination	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.5.7	maxSplits	
			5.5.8	splitTable	
			5.5.8.1	splitNumber	
			5.5.8.2	splitPhase	
			5.5.8.7	splitCoordForceMode	
3.5.2.1.3.6	Configure Pattern Reference Phase	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.8	patternReferencePhase	Max Value = maxPhases
3.5.2.1.3.7	Configure Pattern Reference Point	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.9	patternReferencePoint	
3.5.2.1.3.8	Configure Omit Phases During Transitions	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.5.7	maxSplits	
			5.5.8	splitTable	
			5.5.8.1	splitNumber	Max Value = maxSplits
			5.5.8.2	splitPhase	Max Value = maxPhases
			5.5.8.6	splitOptions	Bit = 0
3.5.2.1.3.9	Configure Pattern Synchronization Time	26048-1§9.2.2			
			5.6.1	timebaseAscPatternSync	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.4	Manage Timing Patterns Requirements				
3.5.2.1.4.1	Configure Timing Patterns Requirements				
3.5.2.1.4.1.1	Configure Pattern to Run Free	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.2	patternCycleTime	Value = 0
3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.5	patternSequenceNumber	Value = 0
3.5.2.1.4.1.3	Configure Pattern Cycle Time for Coordination	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.2	patternCycleTime	Value = 1..999
3.5.2.1.4.1.4	Configure Pattern Offset Time	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.3	patternOffsetTime	
3.5.2.1.4.1.5	Configure Pattern Split Association	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.4	patternSplitTime	
3.5.2.1.4.1.6	Configure Pattern Sequence Association	26048-1§9.2.6			
			5.5.5	maxPatterns	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.5	patternSequenceNumber	
3.5.2.1.4.1.7	Configure Pattern Maximum Mode				
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.6	patternMaximumMode	
3.5.2.1.4.1.8	Configure Pattern Phase Set	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.10	patternPhaseSet	Max Value = maxPhaseSets
3.5.2.1.4.1.9	Configure Pattern Overlap Set	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.11	patternOverlapSet	Max Value = maxOverlapSets
3.5.2.1.4.1.10	Configure Pattern Vehicle Detector Set	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.12	patternVehicleDetectorSet	Max Value = maxVehicleDetectorSets
3.5.2.1.4.1.11	Configure Pattern Pedestrian Detector Set	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.13	patternPedestrianDetectorSet	Max Value = maxPedestrianDetectorSets
3.5.2.1.4.1.12	Configure Pattern Special Functions	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.14	patternSpecialFunctions	
3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	26048-1§9.2.1			
			5.5.5	maxPatterns	
3.5.2.1.5	Manage Splits Configuration Requirements				
3.5.2.1.5.1	Configure Split Requirements				
3.5.2.1.5.1.1	Configure Phase Split Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.5.7	maxSplits	
			5.5.8	splitTable	
			5.5.8.1	splitNumber	Max Value = maxSplits
			5.5.8.2	splitPhase	Max Value = maxPhases
			5.5.8.3	splitTime	
3.5.2.1.5.1.2	Configure Phase Split Mode	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.5.7	maxSplits	
			5.5.8	splitTable	
			5.5.8.1	splitNumber	Max Value = maxSplits
			5.5.8.2	splitPhase	Max Value = maxPhases
			5.5.8.4	splitMode	
3.5.2.1.5.1.3	Configure Split Coordination Phase	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.5.7	maxSplits	
			5.5.8	splitTable	
			5.5.8.1	splitNumber	Max Value = maxSplits
			5.5.8.2	splitPhase	Max Value = maxPhases
			5.5.8.5	splitCoordPhase	
3.5.2.1.5.2	Determine Maximum Number of Phase Splits	26048-1§9.2.1			
			5.5.7	maxSplits	
3.5.2.1.6	Manage Ring Configuration Requirements				
3.5.2.1.6.1	Configure Sequence Data	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	

Requirements Traceability Matrix (RTM)					
FR ID	Functional Requirement	Dialog ID	Object ID	Object Name	Additional Specifications
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.8.1	maxRings	
			5.8.2	maxSequences	
			5.8.3	sequenceTable	
			5.8.3.1	sequenceNumber	Max Value = maxSequences
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.8.3.3	sequenceData	Max Value = maxPhases
3.5.2.1.6.2	Determine Maximum Number of Rings	26048-1§9.2.1			
			5.8.1	maxRings	
3.5.2.1.6.3	Determine Maximum Number of Sequences	26048-1§9.2.1			
			5.8.2	maxSequences	
3.5.2.1.7	Manage Channel Configuration Requirements				
3.5.2.1.7.1	Configure Channel Requirements				
3.5.2.1.7.1.1	Configure Channel Control Source	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.9.1	maxChannels	
			5.9.2	channelTable	
			5.9.2.1	channelNumber	Max Value = maxChannels
			5.9.2.2	channelControlSource	Max Value = maxPhases; Max Value = maxOverlaps
			5.9.2.3	channelControlType	
3.5.2.1.7.1.2	Configure Channel Control Type	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.9.1	maxChannels	
			5.9.2	channelTable	
			5.9.2.1	channelNumber	Max Value = maxChannels
			5.9.2.3	channelControlType	
3.5.2.1.7.1.3	Configure Channel Flash Requirements				
3.5.2.1.7.1.3.1	Configure Channel Flash Yellow	26048-1§9.2.6			
			5.9.1	maxChannels	
			5.9.2	channelTable	
			5.9.2.1	channelNumber	Max Value = maxChannels
			5.9.2.4	channelFlash	Bit = 1
3.5.2.1.7.1.3.2	Configure Channel Flash Red	26048-1§9.2.6			
			5.9.1	maxChannels	
			5.9.2	channelTable	
			5.9.2.1	channelNumber	Max Value = maxChannels
			5.9.2.4	channelFlash	Bit = 2
3.5.2.1.7.1.3.3	Configure Channel Flash Alternate Half Hertz	26048-1§9.2.6			
			5.9.1	maxChannels	
			5.9.2	channelTable	
			5.9.2.1	channelNumber	Max Value = maxChannels
			5.9.2.4	channelFlash	Bit = 3
3.5.2.1.7.1.3.4	Configure Channel Flash Alternate First or Second	26048-1§9.2.6			
			5.9.1	maxChannels	
			5.9.2	channelTable	
			5.9.2.1	channelNumber	Max Value = maxChannels
			5.9.2.4	channelFlash	Bit = 4
3.5.2.1.7.2	Determine Maximum Number of Channels	26048-1§9.2.1			
			5.9.1	maxChannels	
3.5.2.1.8	Manage Overlap Configuration Requirements				

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.8.1	Configure Overlap Requirements				
3.5.2.1.8.1.1	Configure Overlap Type Requirements				
3.5.2.1.8.1.1.1	Configure Overlap Type - Vehicle Normal	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Values = maxOverlaps
			5.10.2.2	overlapType	Value = normal (2)
3.5.2.1.8.1.1.2	Configure Overlap Type - Vehicle Minus Green and Yellow	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	Value = minusGreenYellow (3)
3.5.2.1.8.1.1.3	Configure Overlap Type - Pedestrian Normal	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	Value = pedestrianNormal (4)
3.5.2.1.8.1.1.4	Configure Overlap Type - Flashing Yellow Arrow - 3 Section Head	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	Value = fYAThreeSection (5)
3.5.2.1.8.1.1.5	Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	Value = fYAFourSection (6)
3.5.2.1.8.1.1.6	Configure Overlap Type - Flashing Red Arrow - 3 Section Head	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	Value = fRAThreeSection (7)
3.5.2.1.8.1.1.7	Configure Overlap Type - Flashing Red Arrow - 4 Section Head	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	Value = fRAFourSection (8)
3.5.2.1.8.1.1.8	Configure Overlap Type - 2 Section Transit Specific Signal Head	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	Value = transit-2 (9)
3.5.2.1.8.1.1.9	Configure Overlap Type - Minus Green Yellow Alternate	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.2	overlapType	Value = minusGreenYellowAlternate (10)
3.5.2.1.8.1.2	Configure Overlap Included Phases	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.3	overlapIncludedPhases	Max Value = maxPhases
3.5.2.1.8.1.3	Configure Overlap Modifier Phases	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.4	overlapModifierPhases	Max Value = maxPhases
3.5.2.1.8.1.4	Configure Pedestrian Modifier Phases	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.10	overlapConflictingPedPhases	Max Value = maxPhases
3.5.2.1.8.1.5	Configure Overlap Trailing Green	26048-1§9.2.6			
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.5	overlapTrailGreen	
3.5.2.1.8.1.6	Configure Overlap Trailing Yellow	26048-1§9.2.6			
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.6	overlapTrailYellow	
3.5.2.1.8.1.7	Configure Overlap Trailing Red Clearance	26048-1§9.2.6			
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.7	overlapTrailRed	
3.5.2.1.8.1.8	Configure Overlap Walk	26048-1§9.2.6			
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.8	overlapWalk	
3.5.2.1.8.1.9	Configure Overlap Pedestrian Clearance	26048-1§9.2.6			
			5.10.1	maxOverlaps	
			5.10.2	overlapTable	
			5.10.2.1	overlapNumber	Max Value = maxOverlaps
			5.10.2.9	overlapPedClearance	
3.5.2.1.8.2	Configure Multiple Overlap Sets	4.2.2			

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.10.1	maxOverlaps	
			5.10.5	maxOverlapSets	
			5.10.6	overlapSetTable	
			5.10.6.1	overlapSetNumber	
			5.10.2.1	overlapNumber	
			5.10.6.2	overlapSetIncludedPhases	
			5.10.6.3	overlapSetModifierPhases	
			5.10.6.4	overlapSetTrailGreen	
			5.10.6.5	overlapSetTrailYellow	
			5.10.6.6	overlapSetTrailRed	
			5.10.6.7	overlapSetWalk	
			5.10.6.8	overlapSetPedClearance	
			5.10.6.9	overlapSetConflictingPedPhases	
3.5.2.1.8.3	Retrieve Overlaps Requirements				
3.5.2.1.8.3.1	Determine Maximum Number of Overlaps	26048-1§9.2.1			
			5.10.1	maxOverlaps	
3.5.2.1.8.3.2	Determine Maximum Number of Overlap Sets	26048-1§9.2.1			
			5.10.5	maxOverlapSets	
3.5.2.1.9	Manage Preempt Configuration Requirements				
3.5.2.1.9.1	Configure Preempt Requirements				
3.5.2.1.9.1.1	Enable/Disable Preempt Inputs	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.2	preemptControl	Bit = 4

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.9.1.2	Configure Preempt Control - Non-Locking Memory	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.2	preemptControl	Bit = 0
3.5.2.1.9.1.3	Configure Preempt Control - Override Automatic Flash	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.2	preemptControl	Bit = 1
3.5.2.1.9.1.4	Configure Preempt Control - Override Preempt	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.2	preemptControl	Bit = 2
3.5.2.1.9.1.5	Configure Preempt Control - Flash Dwell	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.2	preemptControl	Bit = 3
3.5.2.1.9.1.6	Configure Preempt Control - All Red Entry	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.2	preemptControl	Bit = 6
3.5.2.1.9.1.7	Configure Preempt Link	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.3	preemptLink	
3.5.2.1.9.1.8	Configure Preempt Delay	26048-1§9.2.6			
			5.7.1	maxPreempts	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.4	preemptDelay	
3.5.2.1.9.1.9	Configure Preempt Minimum Duration	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.5	preemptMinimumDuration	
3.5.2.1.9.1.10	Preempt Entry Configuration Requirements				
3.5.2.1.9.1.10.1	Configure Preempt Enter Minimum Green Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.6	preemptMinimumGreen	
3.5.2.1.9.1.10.2	Configure Preempt Enter Minimum Walk Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.7	preemptMinimumWalk	
3.5.2.1.9.1.10.3	Configure Preempt Enter Pedestrian Clearance Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.8	preemptEnterPedClear	
3.5.2.1.9.1.10.4	Configure Preempt Enter Yellow Change Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.22	preemptEnterYellowChange	
3.5.2.1.9.1.10.5	Configure Preempt Enter Red Clearance Time	26048-1§9.2.6			
			5.7.1	maxPreempts	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.23	preemptEnterRedClear	
3.5.2.1.9.1.11	Configure Preempt Track Clearance Requirements				
3.5.2.1.9.1.11.1	Configure Preempt Track Clearance Green Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.9	preemptTrackGreen	
3.5.2.1.9.1.11.2	Configure Preempt Track Clearance Yellow Change Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.24	preemptTrackYellowChange	
3.5.2.1.9.1.11.3	Configure Preempt Track Red Clearance Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.25	preemptTrackRedClear	
3.5.2.1.9.1.11.4	Configure Preempt Track Clearance Phases	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.12	preemptTrackPhase	Max Value = maxPhases
3.5.2.1.9.1.11.5	Configure Preempt Track Clearance Overlaps	4.2.2			

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.17	preemptTrackOverlap	Max Value = maxOverlaps
3.5.2.1.9.1.12	Configure Preempt Dwell Requirements				
3.5.2.1.9.1.12.1	Configure Preempt Minimum Green Dwell Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
3.5.2.1.9.1.12.2	Configure Preempt Dwell Phases		5.7.2.10	preemptDwellGreen	
		4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.13	preemptDwellPhase	Max Value = maxPhases
3.5.2.1.9.1.12.3	Configure Preempt Dwell Pedestrian Movements	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.14	preemptDwellPed	Max Value = maxPhases
3.5.2.1.9.1.12.4	Configure Preempt Dwell Overlaps	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.18	preemptDwellOverlap	Max Value = maxOverlaps
3.5.2.1.9.1.12.5	Configure Preempt Cycling Phases	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.19	preemptCyclingPhase	Max Value = maxPhases
3.5.2.1.9.1.12.6	Configure Preempt Cycling Pedestrian Movements	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.20	preemptCyclingPed	Max Value = maxPhases
3.5.2.1.9.1.12.7	Configure Preempt Cycling Phases Sequence	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.26	preemptSequenceNumber	Max Value = maxSequences
3.5.2.1.9.1.12.8	Configure Preempt Cycling Overlaps	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.21	preemptCyclingOverlap	Max Value = maxOverlaps
3.5.2.1.9.1.13	Configure Preempt Exit Requirements				
3.5.2.1.9.1.13.1	Configure Preempt Exit Phases	4.2.2			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.7.1	maxPreempts	
			5.7.2	preemptTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.15	preemptExitPhase	Max Value = maxPhases
3.5.2.1.9.1.13.2	Configure Preempt Exit Phase Strategy	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.27	preemptExitType	
3.5.2.1.9.1.13.3	Configure Preempt Exit Priority Levels	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.3.1	maxVehicleDetectors	
			5.7.7	preemptQueueDelayTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.7.7.1	preemptDetectorWeight	
3.5.2.1.9.1.14	Configure Preempt Max Presence Exceeded Requirements				
3.5.2.1.9.1.14.1	Configure Preempt Maximum Presence Time	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.11	preemptMaximumPresence	
3.5.2.1.9.1.14.2	Configure Preempt Maximum Presence Action	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.2	preemptControl	Bit = 5
3.5.2.1.9.1.15	Configure Preempt Gate Description	26048-1§9.2.6			
			5.7.8	maxPreemptGates	
			5.7.9	preemptGateTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.7.9.1	preemptGateNumber	Max Value = maxPreemptGates
			5.7.9.3	preemptGateDescription	
3.5.2.1.9.2	Determine Maximum Number of Preempts				
			5.7.1	maxPreempts	
3.5.2.1.10	Manage ASC Scheduler Requirements				See ISO 26048-1 Section 8.1 & 8.6
3.5.2.1.10.1	Configure ASC Timebased Action Requirements				
3.5.2.1.10.1.1	Configure Timebased Action - Pattern	26048-1§9.2.6			
			5.6.2	maxTimebaseAscActions	
			5.6.3	timebaseAscActionTable	
			5.6.3.1	timebaseAscActionNumber	Max Value = maxTimebaseAscActions
			5.6.3.2	timebaseAscPattern	Max Value = maxPatterns
3.5.2.1.10.1.2	Configure Timebased Action - Special Functions	26048-1§9.2.6			
			5.6.2	maxTimebaseAscActions	
			5.6.3	timebaseAscActionTable	
			5.6.3.1	timebaseAscActionNumber	Max Value = maxTimebaseAscActions
			5.6.3.4	timebaseAscSpecialFunction	
3.5.2.1.10.1.3	Determine Maximum Number of Timebased Actions	26048-1§9.2.1			
			5.6.2	maxTimebaseAscActions	
3.5.2.1.10.1.4	Determine Action In Effect	26048-1§9.2.1			
			5.6.4	timebaseAscActionStatus	
3.5.2.1.10.1.5	Activate Action Plan Remotely	26048-1§9.2.2			
			5.6.5	actionPlanControl	
3.5.2.1.11	Manage I/O Mapping Requirements				
3.5.2.1.11.1	Configure I/O Mapping Requirements				
3.5.2.1.11.1.1	Set Active I/O Map	4.2.5			

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.13.1.2	ascIOactiveMap	
			5.13.1.3	ascIOactivateRequirement	
3.5.2.1.11.1.2	Configure I/O Map Requirements				
3.5.2.1.11.1.2.1	Configure I/O Map Description	4.2.4			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.13.1.1	ascIOmaxMaps	
			5.13.8	ascIOmapDescriptionTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.8.1	ascIOmapDescription	
3.5.2.1.11.1.2.2	Configure I/O Map Input Requirements				
3.5.2.1.11.1.2.2.1	Configure I/O Map Input Device	4.2.4			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.13.1.1	ascIOmaxMaps	
			5.13.2	ascIOmapMaxInputs	
			5.13.4	ascIOinputMapTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.4.2	ascIOinputMapIOindex	Max Value = ascIOmapMaxInputs

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.13.4.3	ascIOinputMapDeviceType	
			5.13.4.4	ascIOinputMapDevicePNN	
			5.13.4.5	ascIOinputMapDevicePtype	
			5.13.4.6	ascIOinputMapDeviceAddr	
3.5.2.1.11.1.2.2.2	Configure I/O Map Input Device Pin	4.2.4			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.13.1.1	ascIOmaxMaps	
			5.13.2	ascIOmapMaxInputs	
			5.13.4	ascIOinputMapTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.4.2	ascIOinputMapIOindex	Max Value = ascIOmapMaxInputs
			5.13.4.3	ascIOinputMapDeviceType	
			5.13.4.4	ascIOinputMapDevicePNN	
			5.13.4.5	ascIOinputMapDevicePtype	
			5.13.4.7	ascIOinputMapDevicePin	
3.5.2.1.11.1.2.2.3	Configure I/O Map Input Function	4.2.4			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.13.1.1	ascIOmaxMaps	
			5.13.2	ascIOmapMaxInputs	
			5.13.4	ascIOinputMapTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.4.2	ascIOinputMapIOindex	Max Value = ascIOmapMaxInputs
			5.13.4.8	ascIOinputMapFuncType	
			5.13.4.9	ascIOinputMapFuncPtype	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.13.4.10	ascIOinputMapFunction	
			5.13.4.11	ascIOinputMapFuncIndex	
3.5.2.1.11.1.2.3	Configure I/O Map Output Requirements				
3.5.2.1.11.1.2.3.1	Configure I/O Map Output Device	4.2.4			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.13.1.1	ascIOmaxMaps	
			5.13.3	ascIOmapMaxOutputs	
			5.13.6	ascIOoutputMapTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.6.1	ascIOoutputMapIOindex	Max Value = ascIOmapMaxOutputs
			5.13.6.2	ascIOoutputMapDeviceType	
			5.13.6.3	ascIOoutputMapDevicePNN	
			5.13.6.4	ascIOoutputMapDevicePtype	
			5.13.6.5	ascIOoutputMapDeviceAddr	
3.5.2.1.11.1.2.3.2	Configure I/O Map Output Device Pin	4.2.4			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.13.1.1	ascIOmaxMaps	
			5.13.3	ascIOmapMaxOutputs	
			5.13.6	ascIOoutputMapTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.6.1	ascIOoutputMapIOindex	Max Value = ascIOmapMaxOutputs
			5.13.6.2	ascIOoutputMapDeviceType	
			5.13.6.3	ascIOoutputMapDevicePNN	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.13.6.4	ascIOoutputMapDevicePtype	
			5.13.6.6	ascIOoutputMapDevicePin	
3.5.2.1.11.1.2.3.3	Configure I/O Map Output Function	4.2.4			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.13.1.1	ascIOmaxMaps	
			5.13.3	ascIOmapMaxOutputs	
			5.13.6	ascIOoutputMapTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.6.1	ascIOoutputMapIOindex	Max Value = ascIOmapMaxOutputs
			5.13.6.7	ascIOoutputMapFuncType	
			5.13.6.8	ascIOoutputMapFuncPtype	
			5.13.6.9	ascIOoutputMapFunction	
			5.13.6.10	ascIOoutputMapFuncIndex	
3.5.2.1.11.2	Determine I/O Mapping Requirements				
3.5.2.1.11.2.1	Retrieve Maximum Number of I/O Maps	26048-1§9.2.1			
			5.13.1.1	ascIOmaxMaps	
3.5.2.1.11.2.2	Retrieve Maximum Number of I/O Map Inputs	26048-1§9.2.1			
			5.13.2	ascIOmapMaxInputs	
3.5.2.1.11.2.3	Retrieve Maximum Number of I/O Map Outputs	26048-1§9.2.1			
			5.13.3	ascIOmapMaxOutputs	
3.5.2.1.11.2.4	Retrieve I/O Mapping Activate Conditions	26048-1§9.2.1			
			5.13.1.3	ascIOactivateRequirement	
3.5.2.1.11.2.5	Retrieve I/O Mapping Input Functions	26048-1§9.2.5			
			5.13.9.1	ascIOmapMaxInputFunctions	
			5.13.9.2	ascIOmapInputFuncTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.13.9.2.1	ascIOinputIndex	Max Value = ascIOmapMaxInputFunctions
			5.13.9.2.2	ascIOinputMaxFuncIndex	
			5.13.9.2.3	ascIOinputFunctionName	
3.5.2.1.11.2.6	Retrieve I/O Mapping Output Functions	26048-1§9.2.5			
			5.13.10.1	ascIOmapMaxOutputFunctions	
			5.13.10.2	ascIOmapOutputFuncTable	
			5.13.10.2.1	ascIOoutputIndex	Max Value = ascIOmapMaxOutputFunctions
			5.13.10.2.2	ascIOoutputMaxFuncIndex	
			5.13.10.2.3	ascIOoutputFunctionName	
3.5.2.1.11.2.7	Retrieve I/O Map Input Device Pin Status	26048-1§9.2.5			
			5.13.1.1	ascIOmaxMaps	
			5.13.2	ascIOmapMaxInputs	
			5.13.5	ascIOinputMapStatusTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.4.2	ascIOinputMapIOindex	Max Value = ascIOmapMaxInputs
			5.13.5.1	ascIOinputMapDevPinDescr	
			5.13.5.2	ascIOinputMapDevPinStatus	
3.5.2.1.11.2.8	Retrieve I/O Map Output Device Pin Status	26048-1§9.2.5			
			5.13.1.1	ascIOmaxMaps	
			5.13.3	ascIOmapMaxOutputs	
			5.13.7	ascIOoutputMapStatusTable	
			5.13.4.1	ascIOmapNumber	Max Value = ascIOmaxMaps
			5.13.6.1	ascIOoutputMapIOindex	Max Value = ascIOmapMaxOutputs
			5.13.7.1	ascIOoutputMapDevPinDescr	
			5.13.7.2	ascIOoutputMapDevPinStatus	
3.5.2.1.11.2.9	Enumerate I/O Mapping Device Pin Requirements				
3.5.2.1.11.2.9.1	Enumerate I/O Map - FIO Inputs				5.13.11.1 - AscIOmapFIOinputs
3.5.2.1.11.2.9.2	Enumerate I/O Map - FIO Outputs				5.13.11.2 - AscIOmapFIOoutputs

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.1.11.2.9.3	Enumerate I/O Map - TS1 Inputs				5.13.12.1 - AscIOmapTS1inputs
3.5.2.1.11.2.9.4	Enumerate I/O Map - TS1 Outputs				5.13.12.2 - AscIOmapTS1outputs
3.5.2.1.11.2.9.5	Enumerate I/O Map - TS2 BIU Inputs				5.13.13.1 - AscIOmapBIUinputs
3.5.2.1.11.2.9.6	Enumerate I/O Map - TS2 BIU Outputs				5.13.13.2 - AscIOmapBIUoutputs
3.5.2.1.11.2.9.7	Enumerate I/O Map - ATC Cabinet SIU Inputs				5.13.14.1 - AscIOmapSIUinputs
3.5.2.1.11.2.9.8	Enumerate I/O Map - ATC Cabinet SIU Outputs				5.13.14.2 - AscIOmapSIUoutputs
3.5.2.1.11.2.9.9	Enumerate I/O Map - Auxiliary Device Inputs				5.13.15.1 - AscIOmapAUXinputs
3.5.2.1.11.2.9.10	Enumerate I/O Map - Auxiliary Device Outputs				5.13.15.2 - AscIOmapAUXoutputs
3.5.2.1.12	Manage Intra-Cabinet Communications Requirements				
3.5.2.1.12.1	Manage Intra-Cabinet Communications Requirements - ATC				
3.5.2.1.12.1.1	Determine Serial Bus 1 Device Present	26048-1§9.2.5			
			5.14.1	maxSIUPort1Addresses	
			5.14.2	siuport1Table	
			5.14.2.1	siuport1Number	Max Value = maxSIUPort1Addresses
			5.14.2.2	siuport1DevicePresent	
3.5.2.1.12.2	Manage Intra-Cabinet Communications Requirements - TS2				
3.5.2.1.12.2.1	Determine TS2 Port 1 Device Present	26048-1§9.2.5			
			5.11.1	maxPort1Addresses	
			5.11.2	port1Table	
			5.11.2.1	port1Number	Max Value = maxPort1Addresses

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.11.2.2	port1DevicePresent	
3.5.2.1.12.2.2	Enable/Disable TS2 Port 1 Frame 40 Messages	26048-1§9.2.6			
			5.11.1	maxPort1Addresses	
			5.11.2	port1Table	
			5.11.2.1	port1Number	Max Value = maxPort1Addresses
			5.11.2.3	port1Frame40Enable	
3.5.2.1.13	Manage ADA Support Requirements				
3.5.2.1.13.1	Configure ADA Support Requirements				
3.5.2.1.13.1.1	Configure APS Push Button Minimum Press Time	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.8	pedestrianButtonPushTime	
3.5.2.1.13.1.2	Configure APS Push Button to Phase Association	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.2	pedestrianDetectorCallPhase	Max Value = maxPhases
3.5.2.1.13.1.3	Configure APS Extra Crossing Time	26048-1§9.2.6			
			5.2.1	maxPhases	
			5.2.2	phaseTable	
			5.2.2.1	phaseNumber	Max Value = maxPhases
			5.2.2.28	phasePedAlternateClearance	
			5.2.2.29	phasePedAlternateWalk	
3.5.2.1.13.1.4	Configure Pedestrian Detector for Alternate Pedestrian Timing	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.9	pedestrianDetectorOptions	Bit = 1
3.5.2.1.14	Manage Block Object Requirements				
3.5.2.1.14.1	Configure Block Object Get Control Requirements				
3.5.2.1.14.1.1	Configure Block Object Get Control - Phase Data	4.2.3			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.2 - AscPhaseBlock
3.5.2.1.14.1.2	Configure Block Object Get Control - Vehicle Detector Data	4.2.3			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.3 - AscVehDetectorBlock
3.5.2.1.14.1.3	Configure Block Object Get Control - Pedestrian Detector Data	4.2.3			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.4 - AscPedDetectorBlock
3.5.2.1.14.1.4	Configure Block Object Get Control - Pattern Data	4.2.3			

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.5 - AscPatternBlock
3.5.2.1.14.1.5	Configure Block Object Get Control - Split Data	4.2.3			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.6 - AscSplitBlock
3.5.2.1.14.1.6	Configure Block Object Get Control - Overlap Data	4.2.3			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.7 - AscOverlapBlock
3.5.2.1.14.1.7	Configure Block Object Get Control - Preempt Data	4.2.3			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.8 - AscPreemptBlock
3.5.2.1.14.1.8	Configure Block Object Get Control - Sequence Data	4.2.3			
			ISO26048-1-Transaction	fdTransactionMode	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.9 - AscSequenceBlock
3.5.2.1.14.1.9	Configure Block Object Get Control - Channel Data	4.2.3			
			ISO26048-1-Transaction	fdTransactionMode	
			ISO26048-1-Transaction	fdTransactionStatus	
			ISO26048-1-Transaction	fdTransactionError	
			5.12.2	ascBlockData	6.10 - AscChannelBlock
3.5.2.1.14.2	Monitor Block Error Status - Error-causing Data Element	26048-1§9.2.1			
			ISO26048-1-Transaction	fdTransactionError	
			5.12.3	ascBlockErrorStatus	
3.5.2.2	Monitor Signal Operations Requirements				
3.5.2.2.1	Determine Controller Health Requirements				See ISO Section 8.6.2
3.5.2.2.1.1	Monitor External Alarm Input States	26048-1§9.2.5			
			5.4.7	maxAlarmGroups	
			5.4.8	alarmGroupTable	
			5.4.8.1	alarmGroupNumber	Max Value = maxAlarmGroups
			5.4.8.2	alarmGroupState	
3.5.2.2.1.2	Monitor External Alarm Active	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 6
3.5.2.2.1.3	Monitor Flash Status	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 1
			5.4.6	unitFlashStatus	
3.5.2.2.1.4	Monitor Local Override	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 3
3.5.2.2.1.5	Monitor Coordination Alarm	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 4

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.2.1.6	Monitor Detector Fault	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 5
3.5.2.2.1.7	Monitor Stop Time Input Alarm	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 7
3.5.2.2.1.8	Monitor Cycle Fault Alarm	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 0
3.5.2.2.1.9	Monitor Coordination Fault	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 1
3.5.2.2.1.10	Monitor Coordination Fail Alarm	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 2
3.5.2.2.1.11	Monitor Cycle Fail Alarm	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 3
3.5.2.2.1.12	Monitor Cabinet IO Link Alarm	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 6
3.5.2.2.1.13	Monitor SMU Communications Error	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 7
3.5.2.2.1.14	Monitor Preempt Maximum Presence Alarm	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 10
3.5.2.2.2	Retrieve Current Operation Requirements				
3.5.2.2.2.1	Monitor Unit Control Status	26048-1§9.2.1			
			5.4.5	unitControlStatus	
3.5.2.2.2.2	Monitor Preempt Active	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 0
3.5.2.2.2.3	Monitor Offset Transitioning	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 2
3.5.2.2.2.4	Monitor Priority Call Active	26048-1§9.2.1			
			5.4.17	shortAlarmStatusV4	Bit = 8
3.5.2.2.2.5	Monitor Local Free Status	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 4
			5.5.10	localFreeStatus	
3.5.2.2.2.6	Monitor Coordination Active	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 5

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.2.2.7	Monitor ECLA Control Active	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 11
3.5.2.2.2.8	Monitor Current Timing Pattern Requirements				
3.5.2.2.2.8.1	Monitor Current Pattern Status	26048-1§9.2.1			
			5.5.9	coordPatternStatus	
3.5.2.2.2.8.2	Monitor Current Pattern Command Source	26048-1§9.2.1			
			5.5.16	coordPatternSource	
3.5.2.2.2.8.3	Monitor Current Pattern Fault Status	26048-1§9.2.1			
			5.5.17	coordPatternFaultStatus	
3.5.2.2.2.9	Monitor Current Cycle Requirements				
3.5.2.2.2.9.1	Monitor Coordination Cycle Status	26048-1§9.2.1			
			5.5.11	coordCycleStatus	
3.5.2.2.2.9.2	Monitor Coordination Synchronization Status	26048-1§9.2.1			
			5.5.12	coordSyncStatus	
3.5.2.2.2.9.3	Monitor Current Offset	26048-1§9.2.1			
			5.5.15	coordCurrentOffset	
3.5.2.2.3	Retrieve Current Signal Indication Requirements				
3.5.2.2.3.1	Monitor Active Red Phases	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.2	phaseStatusGroupReds	
3.5.2.2.3.2	Monitor Active Yellow Phases	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.3	phaseStatusGroupYellows	
3.5.2.2.3.3	Monitor Active Green Phases	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.4	phaseStatusGroupGreens	
3.5.2.2.3.4	Monitor Active Don't Walk Phases	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.5	phaseStatusGroupDontWalks	
3.5.2.2.3.5	Monitor Active Pedestrian Clearance Phases	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.6	phaseStatusGroupPedClears	
3.5.2.2.3.6	Monitor Active Walk Phases	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.7	phaseStatusGroupWalks	
3.5.2.2.3.7	Monitor Active On Phases	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.10	phaseStatusGroupPhaseOns	
3.5.2.2.3.8	Monitor Next Phases	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.11	phaseStatusGroupPhaseNexts	
3.5.2.2.3.9	Monitor Phase Vehicle Calls	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.8	phaseStatusGroupVehCalls	
3.5.2.2.3.10	Monitor Phase Pedestrian Calls	26048-1§9.2.5			
			5.2.3	maxPhaseGroups	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.2.4	phaseStatusGroupTable	
			5.2.4.1	phaseStatusGroupNumber	Max Value = maxPhaseGroups
			5.2.4.9	phaseStatusGroupPedCalls	
3.5.2.2.4	Retrieve Current Ring Requirements				
3.5.2.2.4.1	Monitor Ring Status	26048-1§9.2.5			
			5.8.1	maxRings	
			5.8.6	ringStatusTable	
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.8.6.1	ringStatus	
3.5.2.2.4.2	Monitor Ring Termination Cause	26048-1§9.2.5			
			5.8.1	maxRings	
			5.8.6	ringStatusTable	
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.8.6.1	ringStatus	
3.5.2.2.4.3	Monitor Current Phase On Time	26048-1§9.2.5			
			5.8.1	maxRings	
			5.8.6	ringStatusTable	
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.8.6.2	ringCurrentPhase	
			5.8.6.3	ringCurrentOnTime	
3.5.2.2.5	Retrieve Current Channel Status Requirements				
3.5.2.2.5.1	Monitor Active Red Channels	26048-1§9.2.5			
			5.9.3	maxChannelStatusGroups	
			5.9.4	channelStatusGroupTable	
			5.9.4.1	channelStatusGroupNumber	Max Value = maxChannelStatusGroups
			5.9.4.2	channelStatusGroupReds	
3.5.2.2.5.2	Monitor Active Yellow Channels	26048-1§9.2.5			
			5.9.3	maxChannelStatusGroups	
			5.9.4	channelStatusGroupTable	
			5.9.4.1	channelStatusGroupNumber	Max Value = maxChannelStatusGroups
			5.9.4.3	channelStatusGroupYellows	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.2.5.3	Monitor Active Green Channels	26048-1§9.2.5			
			5.9.3	maxChannelStatusGroups	
			5.9.4	channelStatusGroupTable	
			5.9.4.1	channelStatusGroupNumber	Max Value = maxChannelStatusGroups
			5.9.4.4	channelStatusGroupGreens	
3.5.2.2.6	Retrieve Current Overlap Status Requirements				
3.5.2.2.6.1	Monitor Active Red Overlaps	26048-1§9.2.5			
			5.10.3	maxOverlapStatusGroups	
			5.10.4	overlapStatusGroupTable	
			5.10.4.1	overlapStatusGroupNumber	Max Value = maxOverlapStatusGroups
			5.10.4.2	overlapStatusGroupReds	
3.5.2.2.6.2	Monitor Active Yellow Overlaps	26048-1§9.2.5			
			5.10.3	maxOverlapStatusGroups	
			5.10.4	overlapStatusGroupTable	
			5.10.4.1	overlapStatusGroupNumber	Max Value = maxOverlapStatusGroups
			5.10.4.3	overlapStatusGroupYellows	
3.5.2.2.6.3	Monitor Active Green Overlaps	26048-1§9.2.5			
			5.10.3	maxOverlapStatusGroups	
			5.10.4	overlapStatusGroupTable	
			5.10.4.1	overlapStatusGroupNumber	Max Value = maxOverlapStatusGroups
			5.10.4.4	overlapStatusGroupGreens	
3.5.2.2.6.4	Monitor Active Flashing Yellow Arrow Overlaps	26048-1§9.2.5			
			5.10.3	maxOverlapStatusGroups	
			5.10.4	overlapStatusGroupTable	
			5.10.4.1	overlapStatusGroupNumber	Max Value = maxOverlapStatusGroups
			5.10.4.3	overlapStatusGroupYellows	Note: whether this object or overlapStatusGroupGreens is

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
					used is dependent on where the FYA is wired.
			5.10.4.4	overlapStatusGroupGreens	
3.5.2.2.6.5	Monitor Active Flashing Red Arrow Overlaps	26048-1§9.2.5			
			5.10.3	maxOverlapStatusGroups	
			5.10.4	overlapStatusGroupTable	
			5.10.4.1	overlapStatusGroupNumber	Max Value = maxOverlapStatusGroups
			5.10.4.2	overlapStatusGroupReds	Note: whether this object or overlapStatusGroupGreens is used is dependent on where the FRA is wired.
			5.10.4.4	overlapStatusGroupGreens	
3.5.2.2.7	Retrieve Current Preempt Status Requirements				
3.5.2.2.7.1	Monitor Currently Active Preempt	26048-1§9.2.1			
			5.7.4	preemptStatus	
3.5.2.2.7.2	Monitor Current Preempt Inputs	26048-1§9.2.5			
			5.7.5	maxPreemptGroups	
			5.7.6	preemptStatusGroupTable	
			5.7.6.1	preemptStatusGroupNumber	Max Value = maxPreemptGroups
			5.7.6.2	preemptStatusGroup	
3.5.2.2.7.3	Monitor Current Preempt State	26048-1§9.2.5			
			5.7.1	maxPreempts	
			5.7.2	preemptTable	
			5.7.2.1	preemptNumber	Max Value = maxPreempts
			5.7.2.16	preemptState	
3.5.2.2.7.4	Monitor Current Gate Status	26048-1§9.2.5			
			5.7.8	maxPreemptGates	
			5.7.9	preemptGateTable	
			5.7.9.1	preemptGateNumber	Max Value = maxPreemptGates
			5.7.9.2	preemptGateStatus	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.7.9.3	preemptGateDescription	
3.5.2.2.8	Retrieve Special Function Outputs Requirements				
3.5.2.2.8.1	Determine Maximum Number of Special Functions	26048-1§9.2.1			
			5.4.9	maxSpecialFunctionOutputs	
3.5.2.2.8.2	Monitor Special Function Status	26048-1§9.2.5			
			5.4.9	maxSpecialFunctionOutputs	
			5.4.10	specialFunctionOutputTable	
			5.4.10.1	specialFunctionOutputNumber	Max Value = maxSpecialFunctionOutputs
			5.4.10.3	specialFunctionOutputStatus	
3.5.2.2.8.3	Monitor Special Function Status	26048-1§9.2.5			
			5.4.9	maxSpecialFunctionOutputs	
			5.4.10	specialFunctionOutputTable	
			5.4.10.1	specialFunctionOutputNumber	Max Value = maxSpecialFunctionOutputs
			5.4.10.4	specialFunctionOutputControlSource	
3.5.2.2.9	Monitor Intra-Cabinet Communications Requirements				
3.5.2.2.9.1	Monitor TS2 Port 1 Status	26048-1§9.2.5			
			5.11.1	maxPort1Addresses	
			5.11.2	port1Table	
			5.11.2.1	port1Number	Max Value = maxPort1Addresses
			5.11.2.4	port1Status	
3.5.2.2.9.2	Monitor TS2 Port 1 Fault Frame	26048-1§9.2.5			
			5.11.1	maxPort1Addresses	
			5.11.2	port1Table	
			5.11.2.1	port1Number	Max Value = maxPort1Addresses
			5.11.2.5	port1FaultFrame	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.2.9.3	Monitor ATC Serial Bus 1 Status	26048-1§9.2.5			
			5.14.1	maxSIUPort1Addresses	
			5.14.2	siuport1Table	
			5.14.2.1	siuport1Number	Max Value = maxSIUPort1Addresses
			5.14.2.2	siuport1DevicePresent	
			5.14.2.3	siuport1Status	
3.5.2.2.10	Monitor Signal Monitoring Unit Requirements				
3.5.2.2.10.1	Monitor Signal Monitoring Unit Channel Voltage	26048-1§9.2.5			
			5.9.1	maxChannels	
			5.19.1	ascSmuTable	
			5.9.2.1	channelNumber	Max Value = maxChannels
			5.19.1.1	ascSmuChannelOutput	
			5.19.1.2	ascSmuVoltage	
3.5.2.2.10.2	Monitor Signal Monitoring Unit Channel Current	26048-1§9.2.5			
			5.9.1	maxChannels	
			5.19.1	ascSmuTable	
			5.9.2.1	channelNumber	Max Value = maxChannels
			5.19.1.1	ascSmuChannelOutput	
			5.19.1.3	ascSmuCurrent	
3.5.2.3	Manage Signal Operations Control Requirements				
3.5.2.3.1	Control ASC Function Requirements				
3.5.2.3.1.1	Enable/Disable Manual Backup	26048-1§9.2.2			
			5.4.19	unitManualBackup	
3.5.2.3.1.2	Control Global Minimum Recall	26048-1§9.2.2			
			5.4.18	unitControlV4	Bit = 0
3.5.2.3.1.3	Control Call to Non-Actuated 1	26048-1§9.2.2			
			5.4.18	unitControlV4	Bit = 1

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.3.1.4	Control Call to Non-Actuated 2	26048-1§9.2.2			
			5.4.18	unitControlV4	Bit = 2
3.5.2.3.1.5	Control Walk Rest Modifier	26048-1§9.2.2			
			5.4.18	unitControlV4	Bit = 3
3.5.2.3.1.6	Control Interconnect	26048-1§9.2.2			
			5.4.18	unitControlV4	Bit = 4
3.5.2.3.2	Command Timing Plan Requirements				
3.5.2.3.2.1	Activate System Timing Pattern Remotely	26048-1§9.2.2			
			5.5.13	systemPatternControl	
3.5.2.3.2.2	Control System Reference Point	26048-1§9.2.2			
			5.5.14	systemSyncControl	
3.5.2.3.3	Control Phases Requirements				
3.5.2.3.3.1	Control Phase Omits	26048-1§9.2.6			
			5.2.3	maxPhaseGroups	
			5.2.5	phaseControlGroupTable	
			5.2.5.1	phaseControlGroupNumber	Max Value = maxPhaseGroups
			5.2.5.2	phaseControlGroupPhaseOmit	
3.5.2.3.3.2	Control Pedestrian Phase Omits	26048-1§9.2.6			
			5.2.3	maxPhaseGroups	
			5.2.5	phaseControlGroupTable	
			5.2.5.1	phaseControlGroupNumber	Max Value = maxPhaseGroups
			5.2.5.3	phaseControlGroupPedOmit	
3.5.2.3.3.3	Control Phase Holds	26048-1§9.2.6			
			5.2.3	maxPhaseGroups	
			5.2.5	phaseControlGroupTable	
			5.2.5.1	phaseControlGroupNumber	Max Value = maxPhaseGroups
			5.2.5.4	phaseControlGroupHold	
3.5.2.3.3.4	Control Phase Force Offs	26048-1§9.2.6			
			5.2.3	maxPhaseGroups	
			5.2.5	phaseControlGroupTable	
			5.2.5.1	phaseControlGroupNumber	Max Value = maxPhaseGroups
			5.2.5.5	phaseControlGroupForceOff	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.2.3.3.5	Control Phase Vehicle Calls	26048-1§9.2.6			
			5.2.3	maxPhaseGroups	
			5.2.5	phaseControlGroupTable	
			5.2.5.1	phaseControlGroupNumber	Max Value = maxPhaseGroups
			5.2.5.6	phaseControlGroupVehCall	
3.5.2.3.3.6	Control Phase Pedestrian Calls	26048-1§9.2.6			
			5.2.3	maxPhaseGroups	
			5.2.5	phaseControlGroupTable	
			5.2.5.1	phaseControlGroupNumber	Max Value = maxPhaseGroups
			5.2.5.7	phaseControlGroupPedCall	
3.5.2.3.4	Activate Preempt Remotely	26048-1§9.2.6			
			5.7.1	maxPreempts	
			5.7.3	preemptControlTable	
			5.7.3.1	preemptControlNumber	Max Value = maxPreempts
			5.7.3.2	preemptControlState	
3.5.2.3.5	Control Ring Requirements				
3.5.2.3.5.1	Control Ring Stop Time	26048-1§9.2.6			
			5.8.4	maxRingControlGroups	
			5.8.5	ringControlGroupTable	
			5.8.5.1	ringControlGroupNumber	Max Value = maxRingControlGroups
			5.8.5.2	ringControlGroupStopTime	
3.5.2.3.5.2	Control Ring Force Offs	26048-1§9.2.6			
			5.8.4	maxRingControlGroups	
			5.8.5	ringControlGroupTable	
			5.8.5.1	ringControlGroupNumber	Max Value = maxRingControlGroups
			5.8.5.3	ringControlGroupForceOff	
3.5.2.3.5.3	Control Ring Maximum 2 Time Settings	26048-1§9.2.6			
			5.8.4	maxRingControlGroups	
			5.8.5	ringControlGroupTable	
			5.8.5.1	ringControlGroupNumber	Max Value = maxRingControlGroups
			5.8.5.4	ringControlGroupMax2	
3.5.2.3.5.4	Control Ring Maximum 3 Time Settings	26048-1§9.2.6			

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.8.4	maxRingControlGroups	
			5.8.5	ringControlGroupTable	
			5.8.5.1	ringControlGroupNumber	Max Value = maxRingControlGroups
			5.8.5.9	ringControlGroupMax3	
3.5.2.3.5.5	Control Ring Maximum Inhibit Settings	26048-1§9.2.6			
			5.8.4	maxRingControlGroups	
			5.8.5	ringControlGroupTable	
			5.8.5.1	ringControlGroupNumber	Max Value = maxRingControlGroups
			5.8.5.5	ringControlGroupMaxInhibit	
3.5.2.3.5.6	Control Ring Pedestrian Recycle Settings	26048-1§9.2.6			
			5.8.4	maxRingControlGroups	
			5.8.5	ringControlGroupTable	
			5.8.5.1	ringControlGroupNumber	Max Value = maxRingControlGroups
			5.8.5.6	ringControlGroupPedRecycle	
3.5.2.3.5.7	Control Ring Red Rest Settings	26048-1§9.2.6			
			5.8.4	maxRingControlGroups	
			5.8.5	ringControlGroupTable	
			5.8.5.1	ringControlGroupNumber	Max Value = maxRingControlGroups
			5.8.5.7	ringControlGroupRedRest	
3.5.2.3.5.8	Control Ring Red Clearance Omit Settings	26048-1§9.2.6			
			5.8.4	maxRingControlGroups	
			5.8.5	ringControlGroupTable	
			5.8.5.1	ringControlGroupNumber	Max Value = maxRingControlGroups
			5.8.5.8	ringControlGroupOmitRedClear	
3.5.2.3.6	Activate Special Function Remotely	26048-1§9.2.6			
			5.4.9	maxSpecialFunctionOutputs	
			5.4.10	specialFunctionOutputTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.4.10.1	specialFunctionOutputNumber	Max Value = maxSpecialFunctionOutputs
			5.4.10.2	specialFunctionOutputControl	
3.5.2.3.7	Remote Manual Control Requirements				
3.5.2.3.7.1	Enable Remote Manual Control	26048-1§9.2.2			
			5.4.11	unitMCETimeout	
3.5.2.3.7.2	Advance Interval During Remote Manual Control	26048-1§9.2.2			
			5.4.11	unitMCETimeout	
			5.4.12	unitMCEIntAdv	
3.5.2.3.7.3	Configure Manual Control Timeout	26048-1§9.2.2			
			5.4.11	unitMCETimeout	
3.5.2.3.7.4	Enable/Disable Automatic Pedestrian Clearance Setting	26048-1§9.2.2			
			5.4.2	unitAutoPedestrianClear	
3.5.3	Detector Management Requirements				
3.5.3.1	Manage Detector Configuration Requirements				
3.5.3.1.1	Configure Vehicle Detector Requirements				
3.5.3.1.1.1	Configure Vehicle Detector Travel Mode	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.21	vehicleDetectorTravelMode	
3.5.3.1.1.2	Configure Vehicle Detector Description	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.3	vehicleDetectorDescription	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.3.1.1.3	Configure Vehicle Detector Yellow Lock Call Enabled	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.2	vehicleDetectorOptions	Bit = 2
3.5.3.1.1.4	Configure Vehicle Detector Red Lock Call Enabled	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.2	vehicleDetectorOptions	Bit = 3
3.5.3.1.1.5	Configure Vehicle Detector Passage Enabled	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.2	vehicleDetectorOptions	Bit = 4
3.5.3.1.1.6	Configure Vehicle Detector Added Initial Time Enabled	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.2	vehicleDetectorOptions	Bit = 5
3.5.3.1.1.7	Configure Vehicle Detector Queue Enabled	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.2	vehicleDetectorOptions	Bit = 6
3.5.3.1.1.8	Configure Vehicle Detector Call Enabled	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.2	vehicleDetectorOptions	Bit = 7
3.5.3.1.1.9	Configure Vehicle Detector Call Phase	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.4	vehicleDetectorCallPhase	Max Value = maxPhases
3.5.3.1.1.10	Configure Vehicle Detector Switch Phase	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.5	vehicleDetectorSwitchPhase	Max Value = maxPhases
3.5.3.1.1.11	Configure Vehicle Detector Delay Time	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.6	vehicleDetectorDelay	
3.5.3.1.1.12	Configure Vehicle Detector Extend Time	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.7	vehicleDetectorExtend	
3.5.3.1.1.13	Configure Vehicle Detector Queue Limit Time	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.3.2.8	vehicleDetectorQueueLimit	
3.5.3.1.1.14	Configure Vehicle Detector No Activity Fault Time	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.9	vehicleDetectorNoActivity	
3.5.3.1.1.15	Configure Vehicle Detector Maximum Presence Fault Time	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.10	vehicleDetectorMaxPresence	
3.5.3.1.1.16	Configure Vehicle Detector Erratic Counts	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.11	vehicleDetectorErraticCounts	
3.5.3.1.1.17	Configure Vehicle Detector Fail Time	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.12	vehicleDetectorFailTime	
3.5.3.1.2	Configure Multiple Vehicle Detector Sets for Actuation	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.13	maxVehicleDetectorSets	
			5.3.14	vehicleDetectorSetTable	
			5.3.14.1	vehicleDetectorSetNumber	Max Value = maxVehicleDetectorSets

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.14.2	vehicleDetectorSetOptions	
			5.3.14.3	vehicleDetectorSetCallPhase	
			5.3.14.4	vehicleDetectorSetSwitchPhase	
			5.3.14.5	vehicleDetectorSetDelay	
			5.3.14.6	vehicleDetectorSetExtend	
			5.3.14.7	vehicleDetectorSetQueueLimit	
			5.3.14.8	vehicleDetectorSetNoActivity	
			5.3.14.9	vehicleDetectorSetMaxPresence	
			5.3.14.10	vehicleDetectorSetErraticCounts	
			5.3.14.11	vehicleDetectorSetFailTime	
			5.3.14.14	vehicleDetectorSetTravelMode	
3.5.3.1.3	Configure Pedestrian Detector Requirements				
3.5.3.1.3.1	Configure Pedestrian Detector Description	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.10	pedestrianDetectorDescription	
3.5.3.1.3.2	Configure Pedestrian Detector Call Phase	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.2	pedestrianDetectorCallPhase	Max Value = maxPhases
3.5.3.1.3.3	Configure Pedestrian Detector No Activity Fault Time	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.3	pedestrianDetectorNoActivity	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.3.1.3.4	Configure Pedestrian Detector Maximum Presence Fault Time	26048-1§9.2.6	5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.4	pedestrianDetectorMaxPresence	
3.5.3.1.3.5	Configure Pedestrian Detector Erratic Counts	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.5	pedestrianDetectorErraticCounts	
3.5.3.1.3.6	Configure Pedestrian Detector Non-Lock Calls	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.9	pedestrianDetectorOptions	Bit = 2
3.5.3.1.3.7	Configure Pedestrian Detector for Presence Detection	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.9	pedestrianDetectorOptions	Bit = 0
3.5.3.1.3.8	Configure Pedestrian Detector for Delayed Walk	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.9	pedestrianDetectorOptions	Bit = 3
3.5.3.1.3.9	Configure Pedestrian Detector for Advanced Walk	26048-1§9.2.6			

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.9	pedestrianDetectorOptions	Bit = 4
3.5.3.1.4	Configure Multiple Pedestrian Detector Sets for Actuation	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.15	maxPedestrianDetectorSets	
			5.3.16	pedestrianDetectorSetTable	
			5.3.16.1	pedestrianDetectorSetNumber	Max Value = maxPedestrianDetectorSets
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
				pedestrianDetectorSetCallPhase	
				pedestrianDetectorSetNoActivity	
				pedestrianDetectorSetMaxPresence	
				pedestrianDetectorSetErraticCounts	
				pedestrianDetectorSetOptions	
3.5.3.1.5	Retrieve Detector Configuration Requirements				
3.5.3.1.5.1	Determine Maximum Number of Vehicle Detectors	26048-1§9.2.1			
			5.3.1	maxVehicleDetectors	
3.5.3.1.5.2	Determine Maximum Number of Vehicle Detector Sets	26048-1§9.2.1			
			5.3.6	maxPedestrianDetectors	
3.5.3.1.5.3	Determine Maximum Number of Pedestrian Detectors	26048-1§9.2.1			
			5.3.13	maxVehicleDetectorSets	
3.5.3.1.5.4	Determine Maximum Number of Pedestrian Detector Sets	26048-1§9.2.1			
			5.3.15	maxPedestrianDetectorSets	
3.5.3.2	Retrieve Detector Status Requirements				
3.5.3.2.1	Monitor Active Vehicle Detector Actuations	26048-1§9.2.5			
			5.3.3	maxVehicleDetectorGroups	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.3.4	vehicleDetectorStatusGroupTable	
			5.3.4.1	vehicleDetectorStatusGroupNumber	Max Value = maxVehicleDetectorGroups
			5.3.4.2	vehicleDetectorStatusGroupActive	
3.5.3.2.2	Monitor Active Pedestrian Detector Actuations	26048-1§9.2.5			
			5.3.8	maxPedestrianDetectorGroups	
			5.3.9	pedestrianDetectorStatusGroupTable	
			5.3.9.1	pedestrianDetectorStatusGroupNumber	Max Value = maxPedestrianDetectorGroups
			5.3.9.2	pedestrianDetectorStatusGroupActive	
3.5.3.3	Retrieve Detector Health Requirements				
3.5.3.3.1	Retrieve Vehicle Detector Health Requirements				
3.5.3.3.1.1	Monitor Vehicle Detector Alarm Status	26048-1§9.2.5			
			5.3.3	maxVehicleDetectorGroups	
			5.3.4	vehicleDetectorStatusGroupTable	
			5.3.4.1	vehicleDetectorStatusGroupNumber	Max Value = maxVehicleDetectorGroups
			5.3.4.3	vehicleDetectorStatusGroupAlarms	
3.5.3.3.1.2	Monitor Vehicle Detector Faults from Controller	26048-1§9.2.5			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.13	vehicleDetectorAlarms	
3.5.3.3.1.3	Monitor Vehicle Detector Faults from Detector	26048-1§9.2.5			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.14	vehicleDetectorReportedAlarms	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.3.3.2	Retrieve Pedestrian Detector Health Requirements				
3.5.3.3.2.1	Monitor Pedestrian Detector Alarm Status	26048-1§9.2.5			
			5.3.8	maxPedestrianDetectorGroups	
			5.3.9	pedestrianDetectorStatusGroupTable	
			5.3.9.1	pedestrianDetectorStatusGroupNumber	Max Value = maxPedestrianDetectorGroups
			5.3.9.3	pedestrianDetectorStatusGroupAlarms	
3.5.3.3.2.2	Monitor Pedestrian Detector Faults	26048-1§9.2.5			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.6	pedestrianDetectorAlarms	
3.5.3.4	Control Detector Requirements				
3.5.3.4.1	Control Vehicle Detector Reset	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.15	vehicleDetectorReset	
3.5.3.4.2	Control Pedestrian Detector Reset	26048-1§9.2.6			
			5.3.6	maxPedestrianDetectors	
			5.3.7	pedestrianDetectorTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.7.7	pedestrianDetectorReset	
3.5.3.4.3	Control Detector Diagnostic Reset	26048-1§9.2.2			
			5.3.3.8	unitDetectorDiagnosticReset	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.3.4.4	Control Vehicle Detector Actuation	26048-1§9.2.6			
			5.3.3	maxVehicleDetectorGroups	
			5.3.11	vehicleDetectorControlGroupTable	
			5.3.11.1	vehicleDetectorControlGroupNumber	Max Value = maxVehicleDetectorGroups
			5.3.11.2	vehicleDetectorControlGroupActuation	
3.5.3.4.5	Control Pedestrian Detector Actuation	26048-1§9.2.6			
			5.3.8	maxPedestrianDetectorGroups	
			5.3.12	pedestrianDetectorControlGroupTable	
			5.3.12.1	pedestrianDetectorControlGroupNumber	Max Value = maxPedestrianDetectorGroups
			5.3.12.2	pedestrianDetectorControlGroupActuation	
3.5.3.5	Manage Detector Data Collection Requirements				
3.5.3.5.1	Monitor Vehicle Detector Data Requirements				
3.5.3.5.1.1	Monitor Vehicle Detector Data Sequence	26048-1§9.2.1			
			5.3.5.1	volumeOccupancySequence	
3.5.3.5.1.2	Monitor Vehicle Volume Data	26048-1§9.2.5			
			5.3.1	maxVehicleDetectors	
			5.3.5.3	volumeOccupancyTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.5.3.1	detectorVolume	
3.5.3.5.1.3	Monitor Vehicle Occupancy Data	26048-1§9.2.5			
			5.3.1	maxVehicleDetectors	
			5.3.5.3	volumeOccupancyTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.5.3.2	detectorOccupancy	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.3.5.1.4	Monitor Vehicle Average Speed	26048-1§9.2.5			
			5.3.1	maxVehicleDetectors	
			5.3.5.3	volumeOccupancyTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.5.3.3	detectorAvgSpeed	
3.5.3.5.1.5	Monitor Vehicle Detector Data Sample Time	26048-1§9.2.1			
			5.3.5.4	detectorSampleTime	
3.5.3.5.1.6	Monitor Vehicle Detector Data Sample Duration	26048-1§9.2.1			
			5.3.5.5	detectorSampleDuration	
3.5.3.5.2	Monitor Pedestrian Detector Data Requirements				
3.5.3.5.2.1	Monitor Pedestrian Detector Data Sequence	26048-1§9.2.1			
			5.3.10.1	pedestrianDetectorSequence	
3.5.3.5.2.2	Monitor Pedestrian Counts	26048-1§9.2.5			
			5.3.6	maxPedestrianDetectors	
			5.3.10.3	pedestrianSampleTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.10.3.1	pedestrianDetectorVolume	
3.5.3.5.2.3	Monitor Pedestrian Detector Actuations	26048-1§9.2.5			
			5.3.6	maxPedestrianDetectors	
			5.3.10.3	pedestrianSampleTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.10.3.2	pedestrianDetectorActuations	
3.5.3.5.2.4	Monitor Pedestrian Services	26048-1§9.2.5			
			5.3.6	maxPedestrianDetectors	
			5.3.10.3	pedestrianSampleTable	
			5.3.7.1	pedestrianDetectorNumber	Max Value = maxPedestrianDetectors
			5.3.10.3.3	pedestrianDetectorServices	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.3.5.2.5	Monitor Pedestrian Detector Data Sample Time	26048-1§9.2.1			
			5.3.10.4	pedestrianDetectorSampleTime	
3.5.3.5.2.6	Monitor Pedestrian Detector Data Sample Duration	26048-1§9.2.1			
			5.3.10.5	pedestrianDetectorSampleDuration	
3.5.3.5.3	Configure Detector Data Collection Requirements				
3.5.3.5.3.1	Configure Vehicle Detector Data Sample Period	26048-1§9.2.2			
			5.3.5.2	volumeOccupancyPeriod	
3.5.3.5.3.2	Configure Pedestrian Detector Data Sample Period	26048-1§9.2.2			
			5.3.10.2	pedestrianDetectorPeriod	
3.5.3.5.3.3	Configure Vehicle Speed Detectors	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.16	vehicleDetectorOptions2	Bit = 0
3.5.3.5.3.4	Configure Single Detector Speed Mode	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.16	vehicleDetectorOptions2	Bit = 2
3.5.3.5.3.5	Configure Paired Detector	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.17	vehicleDetectorPairedDetector	
3.5.3.5.3.6	Configure Paired Detector Placement	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.16	vehicleDetectorOptions2	Bit = 1
3.5.3.5.3.7	Configure Paired Detector Spacing	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.18	vehicleDetectorPairedDetectorSpacing	
3.5.3.5.3.8	Configure Average Vehicle Length	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.19	vehicleDetectorAvgVehicleLength	
3.5.3.5.3.9	Configure Vehicle Detection Zone Length	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.20	vehicleDetectorLength	
3.3.3.5.4	Configure Multiple Vehicle Detector Sets for Data Collection	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.13	maxVehicleDetectorSets	
			5.3.14	vehicleDetectorSetTable	
			5.3.14.1	vehicleDetectorSetNumber	Max Value = maxVehicleDetectorSets
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.14.12	vehicleDetectorSetOptions2	
			5.3.2.19	vehicleDetectorAvgVehicleLength	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4	Connected Vehicles Interface Management				
3.5.4.1	Manage ASC - RSU Interface Requirements				
3.5.4.1.1	Configure ASC Communications Port for RSU	26048-1§9.2.2			
			5.15.1	rsuCommPort	Max Value = maxCommPorts
3.5.4.1.2	Configure Logical RSU Ports and Address	26048-1§9.2.6			
			5.15.2	maxRsuPorts	
			5.15.3	rsuPortTable	
			5.15.3.1	rsuPortIndex	Max Value = maxRsuPorts
			5.15.3.2	rsuPortName	
			5.15.3.6	rsuPortNumber	
			5.15.3.7	rsuPortAddress	
3.5.4.1.3	Configure RSU Interface Polling Period	26048-1§9.2.6			
			5.15.2	maxRsuPorts	
			5.15.3	rsuPortTable	
			5.15.3.1	rsuPortIndex	Max Value = maxRsuPorts
			5.15.3.3	rsuPortPollingPeriod	
3.5.4.1.4	Configure RSU Interface Watchdog	26048-1§9.2.6			
			5.15.2	maxRsuPorts	
			5.15.3	rsuPortTable	
			5.15.3.1	rsuPortIndex	Max Value = maxRsuPorts
			5.15.3.4	rsuPortWatchdogTime	
3.5.4.1.5	Monitor RSU Interface Watchdog Timer	26048-1§9.2.5			
			5.15.2	maxRsuPorts	
			5.15.3	rsuPortTable	
			5.15.3.1	rsuPortIndex	Max Value = maxRsuPorts
			5.15.3.5	rsuPortWatchdogTimer	
3.5.4.1.6	Monitor RSU Interface Watchdog Alarm	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 8

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4.2	Manage ASC Process Requirements				
3.5.4.2.1	Manage Signal Phase and Timing Requirements				
3.5.4.2.1.1	Enable Signal Phase and Timing Data	26048-1§9.2.2			
			5.16.3	spatOptions	Bit = 0
3.5.4.2.1.2	Retrieve Signal Phase and Timing Generation Time	26048-1§9.2.1			
			5.16.1	spatTimestamp	
3.5.4.2.1.3	Exchange Movement Status Requirements				
3.5.4.2.1.3.1	Retrieve Movement Timing Requirements				
3.5.4.2.1.3.1.1	Monitor Movement Minimum End Time	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	
			5.20.4.1	movementEventNumber	Row = 1
				signalStateMinEndTick2	
3.5.4.2.1.3.1.2	Monitor Movement Maximum End Time	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	
			5.20.4.1	movementEventNumber	Row = 1
			5.20.4.4	signalStateMaxEndTick2	
3.5.4.2.1.3.1.3	Monitor Movement Likely End Time	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	
			5.20.4.1	movementEventNumber	Row = 1
			5.20.4.5	signalStateLikelyEndTick2	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4.2.1.3.1.4	Monitor Movement Likely End Time Confidence	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	
			5.20.4.1	movementEventNumber	Row = 1
			5.20.4.6	signalStateTickConfidence2	
3.5.4.2.1.3.1.5	Monitor Movement Next Occurrence	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	
			5.20.4.1	movementEventNumber	Row = 1
			5.20.4.7	signalStateNextTick2	
3.5.4.2.1.3.1.6	Monitor Movement Start Time	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	
			5.20.4.1	movementEventNumber	Row = 1
			5.20.4.8	signalStateStartTick	Value = 36111
3.5.4.2.1.3.1.7	Monitor Next Movement Minimum End Time	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	
			5.20.4.1	movementEventNumber	Row = 2
			5.20.4.3	signalStateMinEndTick2	
3.5.4.2.1.3.1.8	Monitor Next Movement Maximum End Time	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.20.4.1	movementEventNumber	Row = 2
			5.20.4.4	signalStateMaxEndTick2	
3.5.4.2.1.3.1.9	Monitor Next Movement Start Time	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	
			5.20.4.1	movementEventNumber	Row = 2
			5.20.4.8	signalStateStartTick	
3.5.4.2.1.3.1.10	Determine Maximum Number of Movement Events	26048-1§9.2.1			
			5.20.3	maxMovementEvents	
3.5.4.2.1.3.2	Configure Movement Assistance Requirements				
3.5.4.2.1.3.2.1	Configure Queue Detectors for Movement Assistance	26048-1§9.2.6			
			5.20.7	maxMovementManeuvers2	
			5.20.8	movementManeuverTable2	
			5.20.8.1	movementManeuverIndex2	Max Value = maxMovementManeuvers2
			5.20.8.2	movementManeuverSignalGroupEntryNumber	
			5.20.8.5	movementManeuverQueueDetector2	Max Value = maxVehicleDetectors
3.5.4.2.1.3.2.2	Configure Pedestrian Detectors for Movement Conflict Assistance	26048-1§9.2.6			
			5.20.7	maxMovementManeuvers2	
			5.20.8	movementManeuverTable2	
			5.20.8.1	movementManeuverIndex2	Max Value = maxMovementManeuvers2
			5.20.8.2	movementManeuverSignalGroupEntryNumber	
			5.20.8.6	movementManeuverPedPresence2	Max Value = maxPedestrianDetectors

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4.2.1.3.2.3	Configure Bicycle Detectors for Movement Conflict Assistance	26048-1§9.2.6			
			5.20.7	maxMovementManeuvers2	
			5.20.8	movementManeuverTable2	
			5.20.8.1	movementManeuverIndex2	Max Value = maxMovementManeuvers2
			5.20.8.2	movementManeuverSignalGroupEntryNumber	
			5.20.8.7	movementManeuverBicyclePresence2	Max Value = maxVehicleDetectors
3.5.4.2.1.3.3	Retrieve Movement Assistance Requirements				
3.5.4.2.1.3.3.1	Monitor Lane Connection Queue Length	26048-1§9.2.5			
			5.20.7	maxMovementManeuvers2	
			5.20.8	movementManeuverTable2	
			5.20.8.1	movementManeuverIndex2	Max Value = maxMovementManeuvers2
			5.20.8.2	movementManeuverSignalGroupEntryNumber	
			5.20.8.3	movementManeuverQueue2	
3.5.4.2.1.3.3.2	Monitor Lane Connection Vulnerable Road User Detection	26048-1§9.2.5			
			5.20.7	maxMovementManeuvers2	
			5.20.8	movementManeuverTable2	
			5.20.8.1	movementManeuverIndex2	Max Value = maxMovementManeuvers2
			5.20.8.2	movementManeuverSignalGroupEntryNumber	
			5.20.8.4	movementManeuverStatus2	Bit = 1
3.5.4.2.1.3.4	Manage Advisory Speed Requirements				
3.5.4.2.1.3.4.1	Configure Advisory Speed Type	26048-1§9.2.6			
			5.20.5	maxAdvisorySpeeds2	
			5.20.6	advisorySpeedTable2	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.20.6.1	advisorySpeedIndex2	Max Value = maxAdvisorySpeeds2
			5.20.6.2	advisorySpeedSignalGroupEntryNumber	
			5.20.6.3	advisorySpeedType2	
3.5.4.2.1.3.4.2	Configure Advisory Speed	26048-1§9.2.6			
			5.20.5	maxAdvisorySpeeds2	
			5.20.6	advisorySpeedTable2	
			5.20.6.1	advisorySpeedIndex2	Max Value = maxAdvisorySpeeds2
			5.20.6.2	advisorySpeedSignalGroupEntryNumber	
			5.20.6.4	advisorySpeedAdvice2	
3.5.4.2.1.3.4.3	Configure Advisory Speed Zone	26048-1§9.2.6			
			5.20.5	maxAdvisorySpeeds2	
			5.20.6	advisorySpeedTable2	
			5.20.6.1	advisorySpeedIndex2	Max Value = maxAdvisorySpeeds2
			5.20.6.2	advisorySpeedSignalGroupEntryNumber	
			5.20.6.5	advisorySpeedZoneLength2	
3.5.4.2.1.3.4.4	Configure Advisory Speed Vehicle Type	26048-1§9.2.6			
			5.20.5	maxAdvisorySpeeds2	
			5.20.6	advisorySpeedTable2	
			5.20.6.1	advisorySpeedIndex2	Max Value = maxAdvisorySpeeds2
			5.20.6.2	advisorySpeedSignalGroupEntryNumber	
			5.20.6.6	advisorySpeedClass2	
3.5.4.2.1.3.5	Monitor Movement State	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	Max Value = maxSignalGroups
			5.20.4.1	movementEventNumber	Row = 1
			5.20.4.2	signalState2	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4.2.1.3.6	Monitor Next Movement State	26048-1§9.2.5			
			5.16.6	maxSignalGroups	
			5.20.3	maxMovementEvents	
			5.20.4	signalStatusTable2	
			5.16.7.1	signalGroupEntryNumber	Max Value = maxSignalGroups
			5.20.4.1	movementEventNumber	Row = 2
			5.20.4.2	signalState2	
3.5.4.2.1.3.7	Monitor Movement Status	26048-1§9.2.1			
			5.20.10	signalStatusBlock2	
3.5.4.2.1.4	Manage Enabled Lane Requirements				
3.5.4.2.1.4.1	Configure Concurrent Enabled Lanes	26048-1§9.2.6			
			5.16.10	maxEnabledLanesConcurrency	
			5.16.2	spatEnabledLanesConcurrencyTable	
			5.16.2.1	enabledLaneIndex	Max Value = maxEnabledLanesConcurrency
			5.16.2.2	enabledLaneConcurrency	
3.5.4.2.1.4.2	Configure Enabled Lanes by Time of Day	26048-1§9.2.6			
			5.5.5	maxPatterns	
			5.5.6	patternTable	
			5.5.6.1	patternNumber	Max Value = maxPatterns
			5.5.6.7	patternSpatEnabledLanes	Max Value = maxEnabledLanesConcurrency
3.5.4.2.1.4.3	Determine Lanes Enabled	26048-1§9.2.1			
			5.20.1	spatEnabledLanesStatus	
3.5.4.2.1.4.4	Command Enabled Lanes	4.2.8			
			5.15.2	maxRsuPorts	
			5.15.3	rsuPortTable	
			5.15.3.1	rsuPortIndex	
			5.16.4	spatPortTable	
			5.16.4.2	spatPortStatus	
			5.16.11	spatEnabledLanesCommand2	
3.5.4.2.1.5	Enable Signal Phase and Timing Data Exchange	26048-1§9.2.6			
			5.15.2	maxRsuPorts	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.16.4	spatPortTable	
			5.15.3.1	rsuPortIndex	Max Value = maxRsuPorts
			5.16.4.1	spatPortOptions	Bit = 0
3.5.4.2.1.6	Configure Road Authority Identifier	26048-1§9.2.2			
			5.20.9	spatRoadAuthorityID	
3.5.4.2.1.7	Retrieve Signal Phase and Timing Intersection Status Requirements				
3.5.4.2.1.7.1	Monitor Manual Control Indication	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 0
3.5.4.2.1.7.2	Monitor Stop Indication	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 1
3.5.4.2.1.7.3	Monitor Failure Flash Indication	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 2
3.5.4.2.1.7.4	Monitor Preemption Operation Indication	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 3
3.5.4.2.1.7.5	Monitor Priority Operation Indication	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 4
3.5.4.2.1.7.6	Monitor Fixed Time Control Indication	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 5
3.5.4.2.1.7.7	Monitor Non-Fixed Time Control Indication	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 6
3.5.4.2.1.7.8	Monitor Standby Operation Indication	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 7
3.5.4.2.1.7.9	Monitor Controller Failure	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 8
3.5.4.2.1.7.10	Monitor MAP Message Validity	26048-1§9.2.1			
			5.20.2	spatStatus2	Bit = 12
3.5.4.2.1.7.11	Monitor SPaT Data Validity	26048-1§9.2.1			

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.20.2	spatStatus2	Bit = 13
3.5.4.2.1.8	Mark SPaT Invalid - Controller	26048-1§9.2.2			
			5.16.3	spatOptions	Bit = 1
3.5.4.2.1.9	Mark SPaT Invalid - Port	26048-1§9.2.6			
			5.15.2	maxRsuPorts	
			5.15.3.1	rsuPortIndex	
			5.16.4	spatPortTable	
			5.16.4.1	spatPortOptions	Bit = 1
3.5.4.2.1.10	Mark MAP Message Invalid - Controller	26048-1§9.2.2			
			5.16.3	spatOptions	Bit = 2
3.5.4.2.1.11	Mark MAP Message Invalid - Port	26048-1§9.2.6			
			5.15.2	maxRsuPorts	
			5.15.3.1	rsuPortIndex	
			5.16.4	spatPortTable	
			5.16.4.1	spatPortOptions	Bit = 2
3.5.4.2.1.12	Manage Signal Group Requirements				
3.5.4.2.1.12.1	Determine Maximum Number of Signal Groups	26048-1§9.2.1			
			5.16.6	maxSignalGroups	
3.5.4.2.1.12.2	Configure Signal Group Intersection Mapping	26048-1§9.2.6			
			5.16.6	maxSignalGroups	
			5.16.7	signalGroupEntryTable	
			5.16.7.1	signalGroupEntryNumber	Max Value = maxSignalGroups
			5.16.7.2	signalGroupIntersection	
			5.16.7.3	signalGroupID	
3.5.4.2.1.12.3	Configure Signal Group Control Source	26048-1§9.2.6			
			5.16.6	maxSignalGroups	
			5.16.7	signalGroupEntryTable	
			5.16.7.1	signalGroupEntryNumber	Max Value = maxSignalGroups
			5.16.7.3	signalGroupID	
			5.16.7.4	signalGroupControlSource	
			5.16.7.5	signalGroupControlType	Value = 2..6

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4.2.1.12.4	Configure Signal Group Indication Types	26048-1§9.2.6			
			5.16.6	maxSignalGroups	
			5.16.7	signalGroupEntryTable	
			5.16.7.1	signalGroupEntryNumber	Max Value = maxSignalGroups
			5.16.7.3	signalGroupID	
			5.16.7.6	signalGroupGreenType	Value = 2..5
			5.16.7.7	signalGroupRedType	Value = 2..3
3.5.4.2.1.12.5	Configure Signal Group Protected or Permissive State	26048-1§9.2.6			
			5.16.6	maxSignalGroups	
			5.16.7	signalGroupEntryTable	
			5.16.7.1	signalGroupEntryNumber	Max Value = maxSignalGroups
			5.16.7.3	signalGroupID	
			5.16.7.6	signalGroupGreenType	Value = 5
			5.16.7.8	signalGroupPermissiveControlSource	
			5.16.7.9	signalGroupPermissiveControlType	Value = 2..7
3.5.4.2.1.12.6	Configure Signal Group Revocable Lanes	26048-1§9.2.6			
			5.16.6	maxSignalGroups	
			5.16.7	signalGroupEntryTable	
			5.16.7.1	signalGroupEntryNumber	Max Value = maxSignalGroups
			5.16.7.3	signalGroupID	
			5.16.7.10	signalGroupEnabledLanes	Max Value = maxEnabledLanesConcurrency
3.5.4.2.1.12.7	Determine Maximum Number of Signal State Entries	26048-1§9.2.1			
			5.16.8	maxAgencySignalStates	
3.5.4.2.1.12.8	Configure Customized Signal State Parameters	26048-1§9.2.6			
			5.16.8	maxAgencySignalStates	
			5.16.9	agencySignalStateTable	
			5.16.9.1	agencySignalStateNumber	Max Value = AgencySignalStates
			5.16.9.2	agencySignalStateColor	Value = 2..8
			5.16.9.3	agencySignalStateControlType	Value = 2..3

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.16.9.4	agencySignalStateOptions	
			5.16.9.5	agencySignalStateValue	Value = 2..11
3.5.4.2.1.13	Retrieve Signal Phase and Timing Time Point	26048-1§9.2.1			
			5.16.5	ascCurrentTick2	
3.5.4.2.2	Manage Connected Vehicle Detector Requirements				
3.5.4.2.2.1	Enabled Connected Device Connection	26048-1§9.2.1			
			5.17.2.1	cvDetectionEnable	
3.5.4.2.2.2	Configure Vehicle Detector for Connected Vehicle Applications	26048-1§9.2.6			
			5.3.1	maxVehicleDetectors	
			5.3.2	vehicleDetectorTable	
			5.3.2.1	vehicleDetectorNumber	Max Value = maxVehicleDetectors
			5.3.2.16	vehicleDetectorOptions2	Bit = 0
3.5.4.2.2.3	Configure Connected Vehicle Detector Input Assignment	26048-1§9.2.6			
			5.17.2.2	maxCvDetectionZones	
			5.17.2.3	ascCvDetectorTable	
			5.17.2.3.1	ascCvDetectorNumber	Max Value = maxCvDetectionZones
			5.17.2.3.2	ascCvDetectorOptions	Bit = 3
			5.17.2.3.4	ascCvDetectorAssignedInput	
3.5.4.2.2.4	Configure Connected Vehicle Detector Port Assignment	26048-1§9.2.6			
			5.17.2.2	maxCvDetectionZones	
			5.17.2.3	ascCvDetectorTable	
			5.17.2.3.1	ascCvDetectorNumber	Max Value = maxCvDetectionZones
			5.17.2.3.3	ascCvDetectorPort	
3.5.4.2.2.5	Configure Assured Green Period Duration	26048-1§9.2.6			
			5.17.2.2	maxCvDetectionZones	
			5.17.2.3	ascCvDetectorTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.17.2.3.1	ascCvDetectorNumber	Max Value = maxCvDetectionZones
			5.17.2.3.5	ascCvDetectorAgpDuration	
3.5.4.2.2.6	Configure Red Light Violation Warning Application Parameters	26048-1§9.2.6			
			5.17.2.2	maxCvDetectionZones	
			5.17.2.3	ascCvDetectorTable	
			5.17.2.3.1	ascCvDetectorNumber	Max Value = maxCvDetectionZones
			5.17.2.3.6	ascCvDetectorRlvwMinHeading	
			5.17.2.3.7	ascCvDetectorRlvwMaxHeading	
			5.17.2.3.8	ascCvDetectorAgpLatitude	
			5.17.2.3.9	ascCvDetectorAgpLongitude	
3.5.4.3	Manage ASC - CV Application Process Interface Requirements				
3.5.4.3.1	ASC - External CV Application Process Requirements				
3.5.4.3.1.1	Provide Movement Information Requirements				
3.5.4.3.1.1.1	Provide Movement Time Point	4.2.10			
			1218v01A - 5.22.1	rsuAscCurrentTick	
3.5.4.3.1.1.2	Provide Movement State	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 1

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	
			1218v01A - 5.22.4.4	rsuSignalGroupID	
			1218v01A - 5.22.4.5	rsuSignalState	
3.5.4.3.1.1.3	Provide Movement Minimum End Time	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 1
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	
			1218v01A - 5.22.4.4	rsuSignalGroupID	
			1218v01A - 5.22.4.6	rsuSignalStateMinEndTick	
3.5.4.3.1.1.4	Provide Movement Maximum End Time	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 1
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			1218v01A - 5.22.4.4	rsuSignalGroupID	
			1218v01A - 5.22.4.7	rsuSignalStateMaxEndTick	
3.5.4.3.1.1.5	Provide Movement Likely End Time	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 1
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	
			1218v01A - 5.22.4.4	rsuSignalGroupID	
			1218v01A - 5.22.4.8	rsuSignalStateLikelyEndTick	
3.5.4.3.1.1.6	Provide Movement Likely End Time Confidence	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 1
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	
			1218v01A - 5.22.4.4	rsuSignalGroupID	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			1218v01A - 5.22.4.9	rsuSignalStateTickConfidence	
3.5.4.3.1.1.7	Provide Next Movement State	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 2
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	
			1218v01A - 5.22.4.4	rsuSignalGroupID	
			1218v01A - 5.22.4.5	rsuSignalState	
3.5.4.3.1.1.8	Provide Next Movement Minimum End Time	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 2
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	
			1218v01A - 5.22.4.4	rsuSignalGroupID	
			1218v01A - 5.22.4.6	rsuSignalStateMinEndTick	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4.3.1.1.9	Provide Next Movement Maximum End Time	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 2
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	
			1218v01A - 5.22.4.4	rsuSignalGroupID	
			1218v01A - 5.22.4.7	rsuSignalStateMaxEndTick	
3.5.4.3.1.1.10	Provide Next Movement Start Time	4.2.10			
			1218v01A - 5.22.2	maxRsuSignalGroups	
			1218v01A - 5.22.3	maxRsuMovementEvents	
			1218v01A - 5.22.4	rsuSignalStatusTable	
			1218v01A - 5.22.4.1	rsuSignalGroupEntryNumber	
			1218v01A - 5.22.4.2	rsuMovementIndex	Row = 2
			1218v01A - 5.22.4.3	rsuSignalGroupIntersection	
			1218v01A - 5.22.4.4	rsuSignalGroupID	
			1218v01A - 5.22.4.11	rsuSignalStateStartTick	
3.5.4.3.1.1.11	Provide Movement Next Occurrence				

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4.3.1.2	Provide Movement Assistance Requirements				
3.5.4.3.1.2.1	Provide Lane Connection Queue Length	4.2.10			
			1218v01A - 5.22.5	maxRsuMovementManeuver	
			1218v01A - 5.22.6	rsuMovementManeuverTable	
			1218v01A - 5.22.6.1	rsuMovementManeuverIndex	
			1218v01A - 5.22.6.2	rsuMovementManeuverIntersection	
			1218v01A - 5.22.6.3	rsuMovementManeuverSignalGroupID	
			1218v01A - 5.22.6.4	rsuMovementManeuverQueue	
3.5.4.3.1.2.2	Provide Lane Connection Vulnerable Road User Detection	4.2.10			
			1218v01A - 5.22.5	maxRsuMovementManeuver	
			1218v01A - 5.22.6	rsuMovementManeuverTable	
			1218v01A - 5.22.6.1	rsuMovementManeuverIndex	
			1218v01A - 5.22.6.2	rsuMovementManeuverIntersection	
			1218v01A - 5.22.6.3	rsuMovementManeuverSignalGroupID	
			1218v01A - 5.22.6.5	rsuMovementManeuverStatus	Bit = 1
3.5.4.3.1.3	Provide Advisory Speed Requirements				
3.5.4.3.1.3.1	Provide Advisory Speed Type	4.2.10			
			1218v01A - 5.22.7	maxRsuAdvisorySpeeds	
			1218v01A - 5.22.8	rsuAdvisorySpeedTable	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			1218v01A - 5.22.8.1	rsuAdvisorySpeedIndex	
			1218v01A - 5.22.8.2	rsuAdvisorySpeedIntersection	
			1218v01A - 5.22.8.3	rsuAdvisorySpeedSignalGroupID	
			1218v01A - 5.22.8.4	rsuAdvisorySpeedType	
3.5.4.3.1.3.2	Provide Advisory Speed	4.2.10			
			1218v01A - 5.22.7	maxRsuAdvisorySpeeds	
			1218v01A - 5.22.8	rsuAdvisorySpeedTable	
			1218v01A - 5.22.8.1	rsuAdvisorySpeedIndex	
			1218v01A - 5.22.8.2	rsuAdvisorySpeedIntersection	
			1218v01A - 5.22.8.3	rsuAdvisorySpeedSignalGroupID	
			1218v01A - 5.22.8.5	rsuAdvisorySpeedAdvice	
3.5.4.3.1.3.3	Provide Advisory Speed Zone	4.2.10			
			1218v01A - 5.22.7	maxRsuAdvisorySpeeds	
			1218v01A - 5.22.8	rsuAdvisorySpeedTable	
			1218v01A - 5.22.8.1	rsuAdvisorySpeedIndex	
			1218v01A - 5.22.8.2	rsuAdvisorySpeedIntersection	
			1218v01A - 5.22.8.3	rsuAdvisorySpeedSignalGroupID	
			1218v01A - 5.22.8.6	rsuAdvisorySpeedZoneLength	
3.5.4.3.1.3.4	Provide Advisory Speed Vehicle Type	4.2.10			
			1218v01A - 5.22.7	maxRsuAdvisorySpeeds	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			1218v01A - 5.22.8	rsuAdvisorySpeedTable	
			1218v01A - 5.22.8.1	rsuAdvisorySpeedIndex	
			1218v01A - 5.22.8.2	rsuAdvisorySpeedIntersection	
			1218v01A - 5.22.8.3	rsuAdvisorySpeedSignalGroupID	
			1218v01A - 5.22.8.7	rsuAdvisorySpeedZoneClass	
3.5.4.3.1.4	Provide Road Authority ID	4.2.10			
			1218v01A - 5.22.10	rsuSpatRoadAuthorityID	
3.5.4.3.1.5	Provide Signal Phase and Timing Intersection Status	4.2.10			
			1218v01A - 5.22.9	rsuSpatStatus	
3.5.4.3.1.6	Provide Compressed SPaT Information to External CV Application Process	4.2.10			
			1218v01A - 5.22.12	rsuSignalStatusBlock	
3.5.4.3.2	ASC - Internal CV Application Process Requirements				
3.5.4.3.2.1	Provide UPER-encoded SPaT Message	4.2.10			
			1218v01A - 5.5.1	maxRsuIFMs	
			1218v01A - 5.5.2	rsuIFMStatusTable	
			1218v01A - 5.5.2.1	rsuIFMIndex	Max Value = maxRsuIFMs
			1218v01A - 5.5.2.2	rsuIFMPsid	Value = 0x8002
			1218v01A - 5.5.2.3	rsuIFMTxChannel	Value = 183
			1218v01A - 5.5.2.4	rsuIFMEnable	Value = 1

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			1218v01A - 5.5.2.5	rsuIFMStatus	Value = 4
			1218v01A - 5.5.2.6	rsuIFMPriority	
			1218v01A - 5.5.2.7	rsuIFMOptions	
			1218v01A - 5.5.2.8	rsuIFMPayload	
3.5.4.3.2.2	Retrieve BSMs				
3.5.4.3.2.3	Retrieve PSMs				
3.5.4.3.2.4	Retrieve Actuation Report	26048-1§9.2.5			
			5.3.3	maxVehicleDetectorGroups	
			5.3.4	vehicleDetectorStatusGroupTable	
			5.3.4.1	vehicleDetectorStatusGroupNumber	Max Value = maxVehicleDetectorGroups
			5.3.4.2	vehicleDetectorStatusGroupActive	
3.5.4.3.2.5	Retrieve Detection Report				
			5.17.2.4	activeCvDetectors	
			5.17.2.5	detectionReportSequence	
			5.17.2.6	detectionReportTable	
			5.17.2.3.1	ascCvDetectorNumber	Max Value = activeCvDetectors
			5.17.2.6.1	detectionReportTime	
			5.17.2.6.2	detectionReportVolume	
			5.17.2.6.3	detectionReportSpeed	
			5.17.2.6.4	detectionReportTravelTime	
			5.17.2.6.5	detectionReportQueue	
			5.17.2.6.6	detectionReportGap	
			5.17.2.6.7	detectionReportPlatoon	
3.5.4.3.3	Exchange Roadway Geometrics Information Requirements				
3.5.4.3.3.1	Retrieve MAP Plan in Effect	26048-1§9.2.1			
			5.17.1.1	mapActivatePlan	
3.5.4.3.3.2	Confirm MAP Plan Compatibility	4.2.7			
			5.15.2	maxRsuPorts	
			5.15.3.1	rsuPortIndex	Max Value = maxRsuPorts

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
			5.16.4	spatPortTable	
			5.16.4.3	spatPortMapActivationCode	
			5.17.1.1	mapActivatePlan	
3.5.4.3.4	Monitor CV Certificate Faults	26048-1§9.2.1			
			5.4.16	unitAlarmStatus	Bit = 9
3.5.4.4	Manage ASC - ECLA Interface Requirements				
3.5.4.4.1	Receive Current Phase Minimum End Time from an ECLA	26048-1§9.2.6			
			5.8.1	maxRings	
			5.18.3	eclaInputTable	
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.19.3.1	eclaRingCurrentPhase	Max Value = maxPhases
			5.19.3.2	eclaRingCurrentMinEndTime	
3.5.4.4.2	Receive Current Phase Maximum End Time from an ECLA	26048-1§9.2.6			
			5.8.1	maxRings	
			5.18.3	eclaInputTable	
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.19.3.1	eclaRingCurrentPhase	Max Value = maxPhases
			5.19.3.3	eclaRingCurrentMaxEndTime	
3.5.4.4.3	Receive Current Phase Likely End Time from an ECLA	26048-1§9.2.6			
			5.8.1	maxRings	
			5.18.3	eclaInputTable	
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.19.3.1	eclaRingCurrentPhase	Max Value = maxPhases
			5.19.3.4	eclaRingCurrentLikelyEndTime	
3.5.4.4.4	Receive Current Phase Likely End Time Confidence from an ECLA	26048-1§9.2.6			
			5.8.1	maxRings	
			5.18.3	eclaInputTable	
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.19.3.5	eclaRingCurrentEndTimeConfidence	

Requirements Traceability Matrix (RTM)					
FR ID	**Functional Requirement**	**Dialog ID**	**Object ID**	**Object Name**	**Additional Specifications**
3.5.4.4.5	Receive Next Phase from an ECLA	26048-1§9.2.6			
			5.8.1	maxRings	
			5.18.3	eclaInputTable	
			5.8.3.2	sequenceRingNumber	Max Value = maxRings
			5.19.3.6	eclaRingNextPhase	Max Value = maxPhases
3.5.4.4.6	Receive Compressed ECLA Input Data	26048-1§9.2.2			
			5.18.4	eclaInputBlock	
3.5.5	Backward Compatibility Requirements				
3.6	Supplemental Non-communications Requirements				
3.6.1	Response Time for Requests				See Requirement 3.6.1 in the PRL
3.6.2	Condition-based Maximum Start Time				
3.6.3	Signal Phase and Timing Data Performance Requirements				
3.6.3.1	SPaT Maximum Transmission Start Time				See Requirement 3.6.3.1 in the PRL
3.6.3.2	Movement Time Point Minimum Transmission Rate				See Requirement 3.6.3.2 in the PRL
3.6.3.3	SPaT Maximum Transmission Rate				
3.6.3.4	SPaT Time Accuracy				

Annex B
Object View [Informative]

B.1 Object Tree

Figure 11 and Figure 12 provide a pictorial representation of the Actuated Signal Controller Object Tree Structure.
UPDATE OBJECT TREE. SAMPLE FROM 1203 UCD
The tree structure identifies how the object definitions are combined under specific nodes.

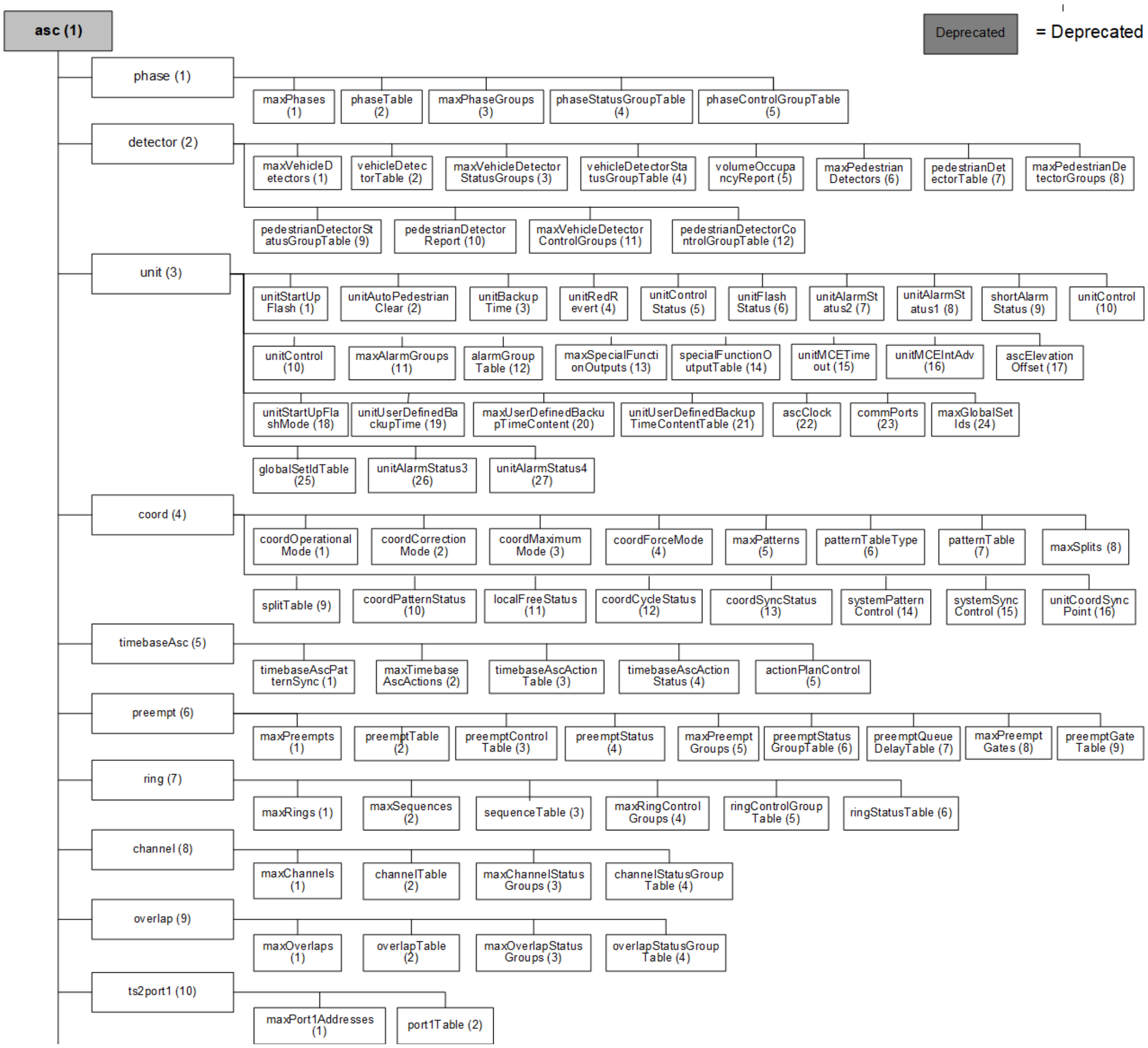

Figure 11 Object Tree for NTCIP 1202 v03B (continued in Figure 11)

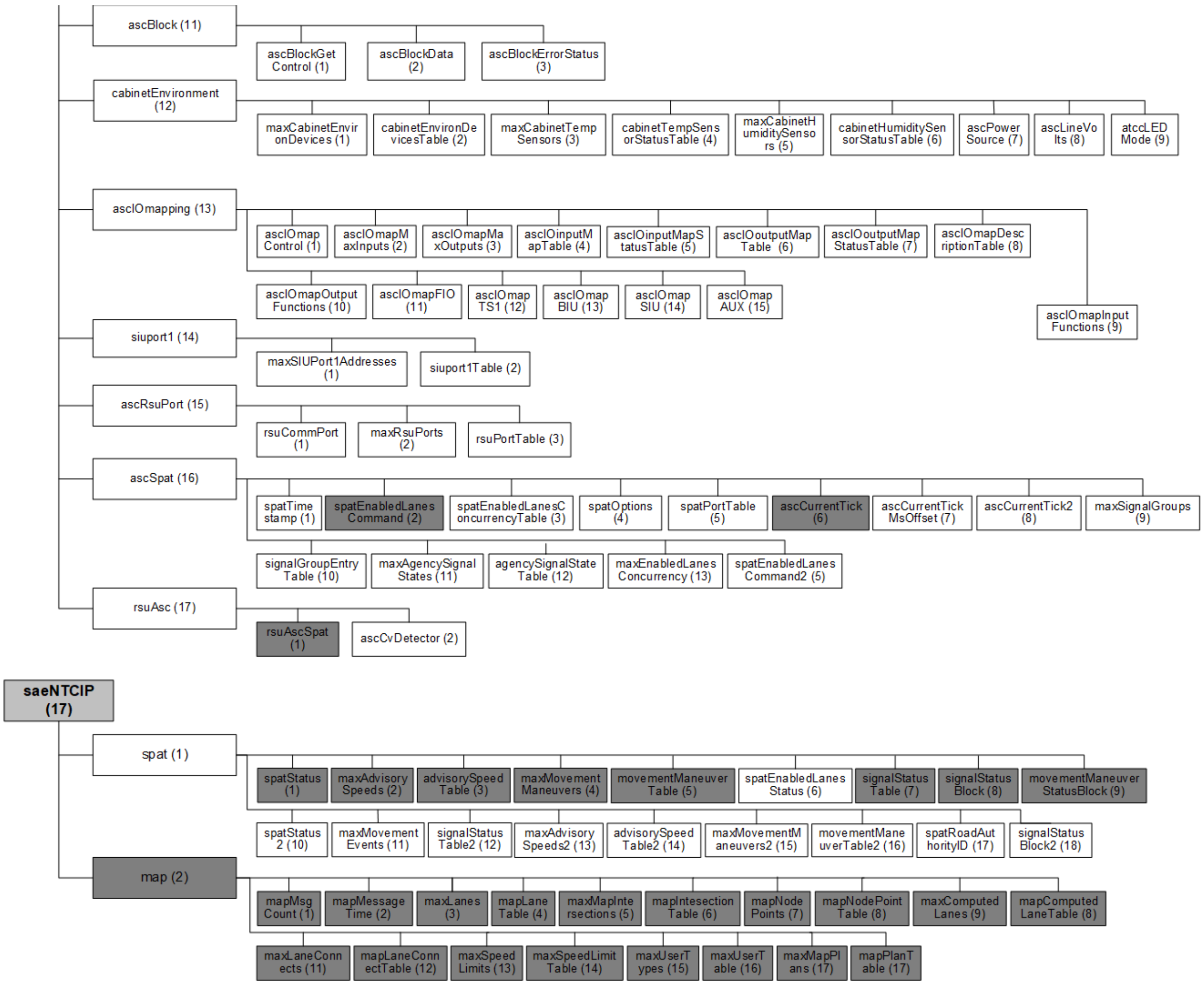

Figure 12 Object Tree for NTCIP 1202 v03B (Continued)

B.2 Profile Implementation Conformance Statement

Still requested Supported Values and Object Support

Object Name	Object Type	Allowed Values
maxPhases	S	2..255
phaseTable		
phaseNumber	NA	1..255
phaseWalk	P	0..255
phasePedestrianClear	P	0..255
phaseMinimumGreen	P	0..255
phasePassage	P	0..255
phaseMaximum1	P	0..999
phaseMaximum2	P	0..999
phaseYellowChange	P	0..255
phaseRedClear	P	0..255
phaseRedRevert	P	0..255
phaseAddedInitial	P	0..255
phaseMaximumInitial	P	0..255
phaseTimeBeforeReduction	P	0..255
phaseCarsBeforeReduction	P	0..255
phaseTimeToReduce	P	0..255
phaseReduceBy	P	0..255
phaseMinimumGap	P	0..255
phaseDynamicMaxLimit	P	0..255
phaseDynamicMaxStep	P	0..255
phaseStartup	P2	1..6
other(1)		
phaseNotOn(2)		
greenWalk(3)		
greenNoWalk(4)		
yellowChange(5)		
redClear(6)		
phaseOptions	P2	0..65535
Bit 0: Enabled Phase		
Bit 1: Automatic Flash Entry Phase		
Bit 2: Automatic Flash Exit Phase		
Bit 3: Non-Actuated 1		
Bit 4: Non-Actuated 2		
Bit 5: Non Lock Detector Memory		
Bit 6: Min. Vehicle Recall		
Bit 7: Max Vehicle Recall		
Bit 8: Ped. Recall		
Bit 9: Soft Vehicle Recall		
Bit 10: Dual Entry Phase		

Bit 11: Simultaneous Gap Disable		
Bit 12: Guranteed Passage		
Bit 13: Actuated Rest-in-Walk		
Bit 14: Conditional Service Enable		
Bit 15: AddedInitialCalculation		
phaseRing	P2	0..255
phaseConcurrency	P2	Octet String
phaseMaximum3	P	0..999
phasePedClearDuringVehicleClear	P	0..255
phasePedServiceLimit	P	0..255
phaseDontWalkRevert	P	0..255
phasePedAlternateClearance	P	0..255
phasePedAlternateWalk	P	0..255
phasePedAdvanceWalkTime	P	0..255
phasePedDelayWalkTime	P	0..255
phaseAdvWarnGrnStartTime	P	0..255
phaseAdvWarnRedStartTime	P	0..255
phaseAltMinTimeTransition	P	0..255
phaseWalkDuringTransition	P	0..255
phasePedClearDuringTransition	P	0..255
maxPhaseGroups	S	1..32
phaseStatusGroupTable		
phaseStatusGroupNumber	NA	1..32
phaseStatusGroupReds	S	0..255
phaseStatusGroupYellows	S	0..255
phaseStatusGroupGreens	S	0..255
phaseStatusGroupDontWalks	S	0..255
phaseStatusGroupPedClears	S	0..255
phaseStatusGroupWalks	S	0..255
phaseStatusGroupVehCalls	S	0..255
phaseStatusGroupPedCalls	S	0..255
phaseStatusGroupPhaseOns	S	0..255
phaseStatusGroupPhaseNexts	S	0..255
phaseControlGroupTable		
phaseControlGroupNumber	NA	1..32
phaseControlGroupPhaseOmit	C	0..255
phaseControlGroupPedOmit	C	0..255
phaseControlGroupHold	C	0..255
phaseControlGroupForceOff	C	0..255
phaseControlGroupVehCall	C	0..255
phaseControlGroupPedCall	C	0..255
maxPhaseSets	S	1..255
phaseSetTable		
phaseSetNumber	NA	1..255

phaseSetWalk	P	0..255
phaseSetPedestrianClear	P	0..255
phaseSetMinimumGreen	P	0..255
phaseSetPassage	P	0..255
phaseSetMaximum1	P	0..999
phaseSetMaximum2	P	0..999
phaseSetYellowChange	P	0..255
phaseSetRedClear	P	0..255
phaseSetRedRevert	P	0..255
phaseSetAddedInitial	P	0..255
phaseSetMaximumInitial	P	0..255
phaseSetTimeBeforeReduction	P	0..255
phaseSetCarsBeforeReduction	P	0..255
phaseSetTimeToReduce	P	0..255
phaseSetReduceBy	P	0..255
phaseSetMinimumGap	P	0..255
phaseSetDynamicMaxLiit	P	0..255
phaseSetOptions	P	0..65535
Bit 0: Reserved		
Bit 1: Reserved		
Bit 2: Reserved		
Bit 3: Non-Actuated 1		
Bit 4: Non-Actuated 2		
Bit 5: Non Lock Detector Memory		
Bit 6: Min. Vehicle Recall		
Bit 7: Max Vehicle Recall		
Bit 8: Ped. Recall		
Bit 9: Soft Vehicle Recall		
Bit 10: Dual Entry Phase		
Bit 11: Simultaneous Gap Disable		
Bit 12: Reserved		
Bit 13: Actuated Rest-in-Walk		
Bit 14: Conditional Service Enable		
Bit 15: AddedInitialCalculation		
phaseSetMaximum3	P	0..999
phaseSetPedClearDuringVehicleClear	P	0..255
phaseSetPedServiceLimit	P	0..255
phaseSetDontWalkRevert	P	0..255
phaseSetPedAlternateClearance	P	0..255
phaseSetPedAlternateWalk	P	0..255
phaseSetPedAdvanceWalkTime	P	0..255
phaseSetPedDelayTime	P	0..255
phaseSetAdvWarnGrnStartTime	P	0..255
phaseSetAdvWarnRedStartTime	P	0..255

phaseSetAltMinTimeTransition	P	0..255
phaseSetWalkDuringTransition	P	0..255
phaseSetPedClearDuringTransition	P	0..255
maxVehicleDetectors		
vehicleDetectorTable		
vehicleDetectorNumber	NA	1..255
vehicleDetectorOptions	P	0..65535
Bit 0: Reserved		
Bit 1: Reserved		
Bit 2: Yellow Lock Call		
Bit 3: Red Lock Call		
Bit 4: Passage		
Bit 5: Added Initial		
Bit 6: Queue		
Bit 7: Call		
vehicleDetectorCallPhase	P	0..255
vehicleDetectorSwitchPhase	P	0..255
vehicleDetectorDelay	P	0..2550
vehicleDetectorExtend	P	0..255
vehicleDetectorQueueLimit	P	0..255
vehicleDetectorNoActivity	P	0..65535
vehicleDetectorMaxPresence	P	0..255
vehicleDetectorErraticCounts	P	0..255
vehicleDetectorFailTime	P	0..255
vehicleDetectorAlarms	S	0..255
vehicleDetectorReportedAlarms	S	0..255
vehicleDetectorReset	C	0..1
vehicleDetectorOptions2	P	0..255
Bit 0: Enable Speed Data		
Bit 1 Detector Placement Option		
Bit 2 Default Detector Speed Mode		
Bit 3: Reserved		
Bit 4: Reserved		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		
vehicleDetectorPairedDetector	P	0..255
vehicleDetectorPairedDetectorSpacing	P	0..65535
vehicleDetectorAvgVehicleLength	P	1..4000
vehicleDetectorLength	P	1..65535
vehicleDetectorTravelMode	P	1..4
other(1)		
vehicle(2)		
transit(2)		

bicycle(4)		
vehicleDetectorDescription	P	String
maxVehicleDetectorGroups	S	1..32
vehicleDetectorStatusGroupTable		
vehicleDetectorStatusGroupNumber	NA	1..32
vehicleDetectorStatusGroupActive	S	0..255
vehicleDetectorStatusGroupAlarms	S	0..255
volumeOccupancySequence	S	0..255
volumeOccupancyPeriod	P	0..65535
volumeOccupancyTable		
detectorVolume	S	0..65535
detectorOccupancy	S	0..255
detectorAvgSpeed	S	0..511
detectorSampleTime	S	Counter
detectorSampleDuration	S	0..3600
maxPedestrianDetectors	S	1..255
pedestrianDetectorTable		
pedestrianDetectorNumber	NA	1..255
pedestrianDetectorCallPhase	P	0..255
pedestrianDetectorNoActivity	P	0..65535
pedestrianDetectorMaxPresence	P	0..255
pedestrianDetectorErraticCounts	P	0..255
pedestrianDetectorAlarms	S	0..255
pedestrianDetectorReset	C	0..1
pedestrianButtonPushTime	P	0..255
pedestrianDetectorOptions	P	0..255
Bit 0: Presence		
Bit 1: Alternate Timing		
Bit 2: Non-locking		
Bit 3: Delayed Walk		
Bit 4: Advanced Walk		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		
pedestrianDetectorDescription	P	String
maxPedestrianDetectorGroups	S	1..32
pedestrianDetectorStatusGroupTable		
pedestrianDetectorStatusGroupActive	S	0..255
pedestrianDetectorStatusGroupAlarms	S	0..255
pedestrianDetectorSequence	S	0..255
pedestrianDetectorPeriod	P	0..65535
pedestrianSampleTable		
pedestrianDetectorVolume	S	0..255
pedestrianDetectorActuations	S	0..255

pedestrianDetectorServices	S	0..255
pedestrianDetectorSampleTime	S	Counter
pedestrianDetectorSampleDuration	S	0..65535
vehicleDetectorControlGroupTable		
vehicleDetectorControlGroupNumber	NA	1..32
vehicleDetectorControlGroupActuation	C	0..255
pedestrianDetectorControlGroupTable		
pedestrianDetectorControlGroupNumber	NA	1..32
pedestrianDetectorControlGroupActuation	C	0..255
maxVehicleDetectorSets		
vehicleDetectorSetNumber	NA	1..255
vehicleDetectorSetOptions	P	0..65535
Bit 0: Reserved		
Bit 1: Reserved		
Bit 2: Yellow Lock Call		
Bit 3: Red Lock Call		
Bit 4: Passage		
Bit 5: Added Initial		
Bit 6: Queue		
Bit 7: Call		
vehicleDetectorSetCallPhase	P	0..255
vehicleDetectorSetSwitchPhase	P	0..255
vehicleDetectorSetDelay	P	0..2550
vehicleDetectorSetExtend	P	0..255
vehicleDetectorSetQueueLimit	P	0..255
vehicleDetectorSetNoActivity	P	0..65535
vehicleDetectorSetMaxPresence	P	0..255
vehicleDetectorSetErraticCounts	P	0..255
vehicleDetectorSetFailTime	P	0..255
vehicleDetectorSetOptions2	P	0..255
Bit 0: Enable Speed Data		
Bit 1 Detector Placement Option		
Bit 2 Default Detector Speed Mode		
Bit 3: Connected Vehicle Detector		
Bit 4: Reserved		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		
vehicleDetectorSetAvgVehicleLength	P	1..4000
vehicleDetectorSetTravelMode	P	1..4
other(1)		
vehicle(2)		
transit(2)		
bicycle(4)		

maxPedestrianDetectorSets	S	1..255
pedestrianDetectorSetTable		
pedestrianDetectorSetNumber	NA	1..255
pedestrianDetectorSetCallPhase	S	0..255
pedestrianDetectorSetNoActivity	P	0..65535
pedestrianDetectorSetMaxPresence	P	0..255
pedestrianDetectorSetErraticCounts	P	0..255
pedestrianDetectorOptions	P	0..255
Bit 0: Presence		
Bit 1: Alternate Timing		
Bit 2: Non-locking		
Bit 3: Delayed Walk		
Bit 4: Advanced Walk		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		
unitDetectorDiagnosticReset	C	0..1
unitStartUpFlash	P	0..255
unitAutoPedestrianClear	P	0..1
unitBackUpTime	P	0..16777216
unitRedRevert	P	0..255
unitControlStatus	S	1..10
other(1)		
systemControl(2)		
systemStandby(3)		
backupMode(4)		
manual(5)		
localTimebase(6)		
interconnect(7)		
interconnectBackup(8)		
remoteManualControl(9)		
localManualControl(10)		
unitFlashStatus	S	1..9
other(1)		
notFlash(2)		
automatic(3)		
localManual(4)		
controllerFaultFlash(5)		
smu(6)		
startup(7)		
preempt(8)		
remoteManual(9)		
maxAlarmGroups	S	1..32
alarmGroupTable		

alarmGroupNumber	NA	1..32
alarmGroupState	S	0..255
maxSpecialFunctionOutputs	S	1..255
specialFunctionOutputTable		
specialFunctionOutputNumber	NA	1..255
specialFunctionOutputControl	C	0..1
specialFunctionOutputStatus	S	0..1
specialFunctionOutputControlSource	S	Bits
Bit 0: Other		
Bit 1: Remote		
Bit 2: Timebase		
Bit 3: Front Panel		
unitMCETimeout	C	0..65535
unitMCEIntAdv	C	0..1
unitStartUpFlashMode	P	1..3
other(1)		
automaticFlash(2)		
cabinetFlash(3)		
allRedControllerFlash(4)		
maxCommPorts	S	0..16
commPortTable		
commPortType	S	1..3
other(1)		
ethernet(2)		
rs232like(3)		
commPortTypeIndex		
commPortEnable	S	1..2
enabled(1)		
disabled(2)		
port1TimeoutFault	S	0..255
serialBus1Fault	S	0..255
unitAlarmStatus	S	Bits
Bit 0: Cycle Fault		
Bit 1: Coord Fault		
Bit 2: Coord Fail		
Bit 3: Cycle Fault		
Bit 4: Local Free		
Bit 5: Coordination Active		
Bit 6: IO Link Error		
Bit 7: SMU Link Error		
Bit 8: RSU Link Error		
Bit 9: CV Certificate Error		
Bit 10: Preempt Maximum Presence		
Bit 11: ECLA Active		

shortAlarmStatusV4	S	Bits
Bit 0: Preempt Actie		
Bit 1: Flash Active		
Bit 2: Transitioning		
Bit 3: Local Overide		
Bit 4: Coordination Alarm		
Bit 5: Detector Fault		
Bit 6: External Alarm		
Bit 7: Stop Time		
Bit 8: Priority Call		
unitControlV4	C	Bits
Bit 0: Global Minimum Recall		
Bit 1: Call to Non-Actuated 1		
Bit 2: Call to Non-Actuated 1		
Bit 3: Walk Rest Modifier		
Bit 4: Interconnect		
unitManualBackup	P	0..1
coordOperationalMode	P	0..255
coordCorrectionMode	P	1..5
other(1)		
dwell(2)		
shortway(3)		
addOnly(4)		
subtract(5)		
coordMaximumMode	P	1..5
other(1)		
maximum1(2)		
maximum2(3)		
maxInhibit(4)		
maximum3(5)		
unitCoordForceMode	P	1..3
other(1)		
floating(2)		
fixed(3)		
maxPatterns	S	1..253
patternTable		
patternNumber	NA	1..253
patternCycleTime	P	0..999
patternOffsetTime	P	0..998
patternSplitNumber	P	0..255
patternSequenceNumber	P	0..255
patternReferencePoint	P	1..5
other(1)		
greenBegin(2)		

yellowBegin(2)		
redBegin(4)		
redEnd(5)		
patternMaximumMode		1..6
other(1)		
coordMaximum(2)		
maxInhibit(3)		
maximum1(4)		
maximum2(5)		
maximum3(6)		
patternSpatEnabledLanes	P	0..255
patternReferencePhase	P	0..255
patternReferencePoint	P	1..5
other(1)		
greenBegin(2)		
yellowBegin(3)		
redBegin(4)		
redEnd(5)		
patternOverlapSet	P	1.255
patternVehicleDetectorSet	P	1.255
patternPedestrianDetectorSet	P	1.255
patternSpecialFunction	P	0..4294967295
patternStatus	S	Bits
Bit 0: Running		
Bit 1: Bad Cycle Tie		
Bit 2: Split Overrun		
Bit 3: Invalid Offset		
Bit 4: Invalid Reference Point		
maxSplits	S	1..255
splitTable		
splitNumber	NA	1..255
splitPhase	NA	1..255
splitTime	P	0..999
splitMode	P	1..8
other(1)		
none(2)		
minimumVehicleRecall(3)		
maximumVehicleRecall(4)		
pedestrianRecall(5)		
maximumVehicleAndPedestrianRecall(6)		
phaseOmitted(7)		
nonActuated(8)		
splitCoordPhase	P	0..1
splitOptions	P	0..255
Bit 0: Transiton Phase Omit		

Bit 1: Reserved		
Bit 2: Reserved		
Bit 3: Reserved		
Bit 4: Reserved		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		
splitCoordForceMode	P	1..4
other(1)		
unitCoordForceMode(2)		
floating(3)		
fixed(4)		
coordPatternStatus	S	1..255
localFreeStatus	S	1..8
other(1)		
notFree(2)		
commandFree(3)		
transitionFree(4)		
inputFree(5)		
coordFree(6)		
badPattern(7)		
failed(11)		
coordCycleStatus	S	0..65535
coordSyncStatus	S	0..65535
systemPatternControl	C	0..255
systemSyncControl	C	0..65535
coordCurrentOffset	S	0..65535
coordPatternSource	S	Bits
Bit 0: Other		
Bit 1: Remote		
Bit 2: Timebase		
Bit 3: Front Panel		
Bit 4: Backup		
coordPatternFaultStatus	S	Bits
Bit 0: Other		
Bit 1: Bad Cycle Time		
Bit 2: Split Overrun		
Bit 3: Invalid Offset		
Bit 4: Invalid Reference Point		
timebaseAscPatternSync	P	0..65535
maxTimebaseAscActions	S	1..255
timebaseAscActionTable		
timebaseAscActionNumber	NA	1..255
timebaseAscPattern	P	0..255

timebaseAscSpecialFunction	P	0..4294967295
timebaseAscEnabledLane	P	0..255
timebaseAscActionStatus	S	0..255
actionPlanControl	C	0..255
maxPreempts	S	1..255
preemptTable		
preemptNumber	NA	0..255
preemptControl	P	0..255
Bit 0: Non-Locking Memory		
Bit 1: Preempt Override Flash Locl		
Bit 2: Preempt Overide Preempt Lock		
Bit 3: Flash Dwell		
Bit 4: Preempt Enable		
Bit 5: All Red Flash Exit		
Bit 6: All Red Entry		
Bit 7: Reserved		
preemptLink	P	0..255
preemptDelay	P	0..255
preemptMinimumDuration	P	0..65535
preemptMinimumGreen	P	0..255
preemptMinimumWalk	P	0..255
preemptEnterPedClear	P	0..255
preemptTrackGreen	P	0..255
preemptDwellGreen	P	0..255
preemptMaximumPresence	P	0..65535
preemptTrackPhase	P2	string
preemptDwellPhase	P2	string
preemptDwellPed	P2	string
preemptExitPhase	P2	string
preemptState	S	1..10
other(1)		
notActive(2)		
notActiveWithCall(3)		
entryStarted(4)		
trackService(5)		
dwell(6)		
linkActive(7)		
exitStarted(8)		
maxPresence(9)		
adancedPreempt(10)		
preemptTrackOverlap	P2	string
preemptDwellOverlap	P2	string
preemptCyclingPhase	P2	string
preemptCyclingOverlap	P2	string

preemptEnterYellowChange	P	0..255
preemptRedClear	P	0..255
preemptTrackYellowChange	P	0..255
preemptTrackRedClear	P	0..255
preemptSequenceNumber	P	0..255
preemptExitType	P	1..4
exitPhases(1)		
queueDelayRecovery(2)		
shortService(3)		
exitCoord(4)		
preemptControlTable		
preemptControlNumber	NA	1..255
preemptControlState	C	0..1
preemptStatus	S	0..255
maxPreemptGroups	S	1..32
preemptStatusGroupTable		
preemptStatusGroupNumber	S	1..32
preemptStatusGroup	S	0..255
preemptQueueDelayTable		
preemptDetectorWeight	P	0..1000
maxPreemptGates	S	1..255
preemptGateTable		
preemptGateNumber	NA	1..255
preemptGateStatus	S	1..4
other(1)		
unknown(2)		
up(3)		
down(4)		
preemptGateDescription	P	string
maxRings	S	1..255
maxSequences	S	1..255
sequenceTable		
sequenceNumber	NA	1..255
sequenceRingNumber	NA	1..255
sequenceData	P2	string
maxRingControlGroups	S	1..32
ringControlGroupTable		
ringControlGroupNumber	NA	1..32
ringControlGroupStopTime	C	0..255
ringControlGroupForceOff	C	0..255
ringControlGroupMax2	C	0..255
ringControlGroupMaxInhibit	C	0..255
ringControlGroupPedRecycle	C	0..255
ringControlGroupRedRest	C	0..255

ringControlGroupOmitRedClear	C	0..255
ringControlGroupMax3	C	0..255
ringStatusTable		
ringStatus	S	0..65535
Bit 0: Coded Status Bit		
Bit 1: Coded Status Bit		
Bit 2: Coded Status Bit		
Bit 3: Gap Out		
Bit 4: Max Out		
Bit 5: Force Off		
Bit 6: Advanced Walk		
Bit 7: Walk		
Bit 8: Delayed Ped Waiting		
Bit 9: Rest-in-Walk		
Bit 10: Flashing Don't Walk		
Bit 11: Don't Walk		
Bit 12: Reserved		
Bit 13: Reserved		
Bit 14: Reserved		
Bit 15: Reserved		
ringOnPhase	S	0..255
ringOnPhaseDuration	S	0..4294967295
maxChannels	S	2..255
channelTable		
channelNumber	NA	1..255
channelControlSource	P2	0..255
channelControlType	P2	1..6
other(1)		
phaseVehicle(2)		
phasePedestrian(3)		
overlap(4)		
pedOverlap(5)		
queueJump(6)		
channelFlash	P	0..255
Bit 0: Reserved		
Bit 1: Flash Yellow		
Bit 2: Flash Red		
Bit 3: Flash Alternate Half Hertz		
Bit 4: Flash Alternate Second		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		
maxChannelStatusGroups	S	1..32
channelStatusGroupTable		

channelStatusGroupNumber	NA	1..32
channelStatusGroupReds	S	0..255
channelStatusGroupYellows	S	0..255
channelStatusGroupGreens	S	0..255
maxOverlaps	S	1..255
overlapTable		
overlapNumber	NA	1..255
overlapType	P2	1..10
other(1)		
normal(2)		
minusGreenYellow(3)		
pedestrianNormal(4)		
fYAThreeSection(5)		
fYAFourSection(6)		
fRAThreeSection(7)		
fRAFourSection(8)		
transit-2(9)		
minusGreenYellowAlternate(10)		
overlapIncludedPhases	P2	string
overlapModifierPhases	P2	string
overlapTrailGreen	P	0..255
overlapTrailYellow	P	0..255
overlapTrailRed	P	0..255
overlapWalk	P	0..255
overlapPedClearance	P	0..255
overlapConflictingPedPhases	P2	string
maxOverlapStatusGroups	S	1..32
overlapStatusGroupTable		
overlapStatusGroupNumber	NA	1..32
overlapStatusGroupReds	S	0..255
overlapStatusGroupYellows	S	0..255
overlapStatusGroupGreens	S	0..255
maxOverlapSets	S	1..255
overlapSetTable		
overlapSetNumber	NA	1..255
overlapSetIncludedPhases	P2	string
overlapSetModifierPhases	P2	string
overlapSetTrailGreen	P	0..255
overlapSetTrailYellow	P	0..255
overlapSetTrailRed	P	0..255
overlapSetWalk	P	0..255
overlapSetPedClearance	P	0..255
overlapSetConflictingPedPhases	P2	string
maxPort1Addresses	S	1..255

port1Table		
port1Number	NA	1..255
port1DevicePresent	P	0..1
port1Frame40Enable	P	0..1
port1Status	S	1..3
other(1)		
online(2)		
responseFault(3)		
port1FaultFrame	S	0..255
ascBlockGetControl	C	string
ascBlockData	C	string
ascBlockErrorStatus	S	0..65535
IO		
maxSIUPort1Addresses	S	1..255
siuport1Table		
siuPort1Number	NA	1..255
siuPort1DevicePresent	C	0..1
siuport1Status	S	1..3
other(1)		
online(2)		
responseFault(3)		
rsuCommPort	P	0..255
maxRsuPorts	S	0..16
rsuPortTable		
rsuPortIndex	S	1..16
rsuPortName	P	string
rsuPortPollingPeriod	P	0..65535
rsuPortWatchdogTime	P	0..65535
rsuPortWatchdogTimer	S	0..65535
rsuPortNumber	P	0..65535
rsuPortAddress	P	IpAddress
spatTimestamp	S	string
spatEnabledLanesConcurrencyTable		
enabledLaneIndex	NA	1..255
enabledLaneConcurrency	P	string
spatOptions	P	0..255
Bit 0: Enabled SPaT		
Bit 1: SPaT Validity		
Bit 2: MAP Validity		
Bit 3: Reserved		
Bit 4: Reserved		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		

spatPortTable		
spatPortOptions	P	0..255
Bit 0: Enabled SPaT		
Bit 1: SPaT Validity		
Bit 2: MAP Validity		
Bit 3: Reserved		
Bit 4: Reserved		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		
spatPortStatus	S	1..5
other(1)		
disabled(2)		
normal(3)		
mapError(4)		
enabledLanesError(5)		
spatPortMapActivationCode	P	MapActivationCode
ascCurrentTick2	S	0..36111
maxSignalGroups	S	1..255
signalGroupEntryTable		
signalGroupEntryNumber	P	0..65535
signalGroupIntersection	P	0..255
signalGroupID	P	0..255
signalGroupControlSource	P	0..255
signalGroupControlType	P	1..6
other(1)		
phaseVehicle(2)		
phasePedestrian(3)		
overlap(4)		
pedOverlap(5)		
queueJump(6)		
signalGroupGreenType	P	1..5
other(1)		
preMovement(2)		
permissiveMovementAllowed(3)		
protectedMovementAllowed(4)		
protectedPermissive (5)		
signalGroupRedType	P	1..3
other(1)		
stopThenProceed(2)		
stopAndRemain(3)		
signalGroupPermissiveControlSource	P	0..255
signalGroupPermissiveControlType	P	1..6
other(1)		

phaseVehicle(2)		
phasePedestrian(3)		
overlap(4)		
pedOverlap(5)		
queueJump(6)		
signalGroupEnabledLanes	P	0..255
maxAgencySignalStates	S	3..255
agencySignalStateTable	NA	1..255
agencySignalStateColor	P	1..7
other(1)		
red(2)		
yellow(3)		
green(4)		
dark(5)		
flashingRed(6)		
flashingYellow(7)		
agencySignalStateControlType	P	1..3
other(1)		
vehicle(2)		
pedestrian(3)		
agencySignalStateOptions	P	0..65535
Bit 0: stopThenProceed		
Bit 1: stopAndRemain		
Bit 2: preMovement Green		
Bit 3: Permissive Green		
Bit 4: Protected Green		
Bit 5: Permissive Red		
Bit 6: Permissive Yellow		
Bit 7: Permissive Green		
Bit 8: Permissive Flashing Yellow		
Bit 9: Permissive Flashing Red		
Bit 10: Reserved		
Bit 11: Reserved		
Bit 12: Reserved		
Bit 13: Reserved		
Bit 14: Reserved		
Bit 15: Reserved		
agencySignalStateValue	P	1..11
other(1)		
unavailable(2)		
dark(3)		
stopThenProceed(4)		
stopAndRmeain(5)		
preMovement(6)		

permissiveMovementAllowed(8)		
permissiveClearance(9)		
protectedClearance(10)		
cautionConflictingTraffic(11)		
maxEnabledLanesConcurrency	S	1..255
spatEnabledLanesCommand2	P	string
mapActivatePlanError		
cvDetectionEnable		
enabled(1)		
disabled(2)		
maxCvDetectionZones	S	1..255
ascCvDetectorTable		
ascCvDetectorNumber	NA	1..255
ascCvDetectorOptions	P	0..255
Bit 0: BSM		
Bit 1: PSM		
Bit 2: Reserved		
Bit 3: Vehicle Detector		
Bit 4: Pedestrian Detector		
Bit 5: Reserved		
Bit 6: Enable Processed Data		
Bit 7: Reserved		
ascCvDetectorPort	P	0..255
ascCvDetectorAssignmentInput	P	1..255
ascCvDetectorAgpDuration	P	0..255
ascCvDetectorRlvwMinHeading	P	0..28800
ascCvDetectorRlvwMaxHeading	P	0..28800
ascCvDetectorAgpLatitude	P	-90000000..90000001
ascCvDetectorAgpLongitude	P	-179999999..180000001
eclaCommEnable	P	0..1
eclaDataTimestamp	C	0..36111
eclaInputTable		
eclaRingCurrentPhase	C	0..255
eclaRingGreenMinEndTime	C	0..36111
eclaRingGreenMaxEndTime	C	0..36111
eclaRingGreenLikelyEndTime	C	0..15
eclaRingEndTimeConfidence	C	0..36111
eclaRingNextPhase	C	0..255
eclaInputBlock	C	string
ascSmuTable		
ascSmuChannelOutput	NA	1..3
ascSmuVoltage	S	0..255
ascSmuChannel	S	0..65535
spatStatus2	S	

Bit 0: Manual Control		
Bit 1: Stop Time		
Bit 2: Failure Flash		
Bit 3: Preempt		
Bit 4: Signal Priority		
Bit 5: Fixed Time		
Bit 6: Traffic Dependent		
Bit 7: Standby		
Bit 8: Failure Mode		
Bit 9: 0		
Bit 10: 1		
Bit 11: 1		
Bit 12: Invalid MAP		
Bit 13: Invalid SPaT		
Bit 14: Reserved		
Bit 15: Reserved		
maxMovementEvents	S	2..16
signalStatusTable2		
movementEventNumber	NA	1..16
signalState2	S	1..11
other(1)		
unavailable(2)		
dark(3)		
stopThenProceed(4)		
stopAndRmeain(5)		
preMovement(6)		
permissiveMovementAllowed(8)		
permissiveClearance(9)		
protectedClearance(10)		
cautionConflictingTraffic(11)		
signalStateMinEndTick2	S	0..36111
signalStateMaxEndTick2	S	0..36111
signalStateLikelyEndTick2	S	0..36111
signalStateTickConfidence2	S	0..15
signalStateNextTick2	S	0..36111
maxAdvisorySpeeds2	S	1..255
advisorySpeedTable2		
advisorySpeedIndex2	NA	1..255
advisorySpeedSignalGroupEntryNumber	P	1..255
advisorySpeedType2	P	1..4
none(1)		
greenWave(2)		
ecoDrive(3)		
transit(4)		

advisorySpeedAdvice2	P	0..500
advisorySpeedZoneLength2	P	0..10000
advisorySpeedClass2	P	0..255
maxMovementManeuvers2	S	1..255
movementManeuverTable2		
movementManeuverIndex2	NA	1..255
movementManeuverSignalGroupEntryNumber	P	0..255
movementManeuverQueue2	P	0..10000
movementManeuverStatus2	S	0..255
Bit 0: Reserved		
Bit 1: Vulnerable Road User		
Bit 2: Reserved		
Bit 3: Reserved		
Bit 4: Reserved		
Bit 5: Reserved		
Bit 6: Reserved		
Bit 7: Reserved		
movementManeuverQueueDetector2	P	string
movementManeuverPedPresence2	P	string
movementManeuverBicyclePresence2	P	string
spatRoadAuhorityID	P	OID
signalStatusBlock2	S	string

Annex C
Test Procedures [Normative]

See Separate Report for Annex C.

Annex D
Documentation of Revisions [Informative]

Error! Reference source not found. identifies the changes that have been made to NTCIP 1202. The NTCIP effort makes reasonable efforts to ensure that standards are as backward compatible as possible, but the primary purpose of NTCIP 1202 is to provide interoperability by developing standards in a consensus environment. When changes are required to meet this objective, the problematic objects are refined (if the issue is primarily editorial) or deprecated and, in most cases, replaced with new objects. **Error! Reference source not found.** identifies why each of these changes has been made. New implementations should support the new/replacement objects; they may also support deprecated objects.

D.1 NTCIP 1202 v02 to NTCIP 1202 v03

The following identify changes that were made from NTCIP 1202 v02 to NTCIP 1202 v03.

D.1.1 Added Systems Engineering Process

The major change is the structure of the document. NTCIP 1202 v03 adds systems engineering process (SEP) content defined in NTCIP 8002, Annex B1, version 01, which was published on September 2016. The SEP includes the definition of the user needs, functional requirements, dialogs and a requirements traceability matrix in addition to the already existing MIB. The conformance group definitions and the conformance statement contained in NTCIP 1202 v02 were replaced by the PRL (Table 5), which allows a user to specify which functions are supposed to be supported by an ASC.

D.1.2 General MIB Changes

General edits were made to the MIB header in NTCIP 1202 v03 to reflect updates to other MIBs from which the NTCIP 1202 v03 MIB imports data.

All DESCRIPTION fields were updated to conform to NTCIP 8004 v02. Additionally, many DESCRIPTION fields have received additional clarifications and explanations to remove ambiguities.

The STATUS of all objects was changed to "mandatory" to reflect that conformance for NTCIP 1202 v03 is now measured through the use of the PRL as contained in Section 3 and the RTM contained in Annex A.

References to objects defined in NTCIP 1201 are now made through the RTM rather than through comments in the MIB.

Default values (DEFVAL) were added to certain object definitions (e.g., status objects, control objects).

Several objects were added to reflect new user needs.

D.1.3 New User Needs

In addition to the systems engineering content (user needs, requirements) developed that trace to existing objects in NTCIP 1202 v02, new user needs were defined and supported in NTCIP 1202 v03. From these new user needs, new requirements and design content (in the form of dialogs and object definitions) were developed and included in NTCIP 1202 v03. The following identifies these new user needs supported.

D.1.3.1 Added Support for Connected Vehicle Environment

NTCIP 1202 v03 supports a new user need to provide a standardized way to exchange signal phasing and timing data (SPaT), MAP (roadway geometry) data, and the presence of connected devices with a

RoadSide Unit (RSU) in a connected vehicle environment. This user need includes support to manage the interface between an ASC and a RSU.

D.1.3.2 Added Support to Manage the Cabinet Environment

NTCIP 1202 v03 supports a new user need to provide a standardized way to monitor and manage the operating environment that the ASC is located in. The operating environment includes the temperature and humidity within the ASC cabinet, the status of the cabinet doors (open or closed), and if the cabinet is flooding.

D.1.3.3 Added Support to Manage the Power Sources

NTCIP 1202 v03 supports a new user need to provide a standardized way to monitor and manage the power sources for the ASC.

D.1.3.4 Added Support to Retrieve Operational Performance Data

NTCIP 1202 v03 supports a new user need to provide a standardized way to manage and retrieve the collection of signal operations and detector data for the analysis of signal timing efficacy. An example of this operational data is the Indiana Traffic Signal Hi Resolution Data Logger Enumerations.

D.1.3.5 Added Support to Manage I/O Mapping

NTCIP 1202 v03 supports a new user need to provide a standardized way to manage and retrieve the Input / Output mapping for the ASC. The user need also allows a manager to reset the input/output mapping to a default configuration.

D.1.3.6 Added Support for Accessible Pedestrian Signals (APS)

NTCIP 1202 v03 supports a new user need to support the configuration of APS push buttons.

D.1.3.7 Added Support to Activate an Action Plan

NTCIP 1202 v03 supports a new user need to provide a standardized way to call a pre-configured action plan. In NTCIP 1202 v02, each action plan defines a group of functions, such as activating an output, that may be activated from a scheduler. NTCIP 1202 v03 extends this feature so the action plan may be manually called outside the scheduler.

D.1.3.8 Added Support to Manually Advance the Controller Remotely

NTCIP 1202 v03 supports a new user need to provide a standardized way to remotely and manually advance the ASC through the phase or interval.

D.1.3.9 Added Support for Condition Based Exception Reporting

NTCIP 1202 v03 supports a new user need to provide a standardized way to manage exception-based reporting for the ASC. Under this scenario, a manager can configure an ASC to automatically transmit data to a management station when specific conditions are satisfied.

D.1.4 New Requirements

In addition to new requirements and objects to support the user needs expressed in Annex D.1.3, new requirements (and objects) were added to extend user needs already supported in NTCIP 1202 v02. The following identifies these new requirements and supported in NTCIP 1202 v03.

D.1.4.1 Manage ASC Location

NTCIP 1202 v03 adds requirements to manage the location of the ASC.

D.1.4.2 Manage Communications Ports

NTCIP 1202 v03 adds requirements to manage the communications ports on the ASC.

D.1.4.3 Manage ASC Clock

NTCIP 1202 v03 adds requirements to manage the clock of an ASC.

D.1.4.4 Manage User-Defined Backup Time

NTCIP 1202 v03 adds requirements to configure a user-defined backup time, based on user-defined functions in addition to the existing backup time defined in NTCIP 1202 v02. The backup time defines the period of time to be exceeded when no SET operations are received for a set of system control parameters before the ASC reverts to Backup Mode. The set of system control parameters are defined in NTCIP 1202 v02 for the existing backup time, and is user defined for the user-defined backup time.

D.1.4.5 Support for Advanced Warning Signal Indications

NTCIP 1202 v03 adds requirements to support warnings in advanced of a green or red signal indication for a phase.

D.1.4.6 Support for Phase Maximum 3

NTCIP 1202 v03 added requirements to support a Phase Maximum 3 mode, which has a range from 0 to 6000 seconds.

D.1.4.7 Support for Bicycle Phases

NTCIP 1202 v03 adds requirements to support bicycle phases for the ASC.

D.1.4.8 Support for Transit Phases

NTCIP 1202 v03 adds requirements to support transit phases for the ASC.

D.1.4.9 Manage Alternate Times for Transitions

NTCIP 1202 v03 adds requirements to manage alternate minimum times that can be used during transitions.

D.1.4.10 Manage Coordination Point

NTCIP 1202 v03 adds requirements to manage the coordination point for the unit and for a specific pattern for the ASC.

D.1.4.11 Support for Additional Overlaps

NTCIP 1202 v03 adds support for several additional overlap configurations, including for flashing yellow arrows, flashing red arrows, and transit-specific overlaps.

D.1.4.12 Manage Preempt Exit Strategy

NTCIP 1202 v03 adds requirements to manage the exit strategy out of a preempt sequence.

D.1.4.13 Manage Additional Alarms

NTCIP 1202 v03 adds requirements to configure and monitor additional alarms for the ASC.

D.1.4.14 Support for Paired Detectors

NTCIP 1202 v03 adds requirements to support paired detectors (e.g., speed traps) for the ASC.

D.1.4.15 Improved Support for Vehicle Detectors

NTCIP 1202 v03 adds requirements to improve support for vehicle detectors for the ASC. Examples include support to define average vehicle lengths, collect speed data, define detector types, data collection periods up to 3600 seconds, and data collection periods based on the current cycle length (instead of a fixed period).

D.1.4.16 Improved Support for Pedestrian Detectors

NTCIP 1202 v03 adds requirements to improve support for detector data for the ASC. Examples include support to reset a pedestrian detector, define a pedestrian pushbutton for accessible pedestrian signal (APS) features, monitor pedestrian detector status, and generate pedestrian detector reports.

D.1.4.17 Block Objects for New NTCIP 1202 v03 Objects

NTCIP 1202 v03 adds new block objects to support the new object definitions added in NTCIP 1202 v03.

D.1.5 Changes to Existing Objects

In addition new enumerations were added to several existing object definitions in NTCIP 1202 v02. The following identifies the new enumerations supported in NTCIP 1202 v03.

D.1.5.1 Additional Coordination Correction Mode

NTCIP 1202 v03 added an enumeration for a subtractOnly mode for the coordCorrectionMode object. This mode support coordination correction by subtracting only from the timings.

D.2 NTCIP 1202 v03 to NTCIP 1202 v03A

Updated with an errata for Flashing Yellow Arrow (FYA) functionality described in Sections 5.9.2.7, 7.2.5.3, 7.2.5.11, and 7.2.7.1.

D.3 NTCIP 1202 v03A to NTCIP 1202 v03B

The following identify changes that were made from NTCIP 1202 v02 to NTCIP 1202 v03.

D.3.1 Fixes to RTM

Following the City of Anaheim's testing of NTCIP 1202, numerous errors were discovered with the RTM in NTCIP 1202 v03A. The following errors were corrected in NTCIP 1202 v03B.

- 3.5.1.3.7 Configure ASC Temperature Threshold is updated to follow the H.2.7 dialog instead of the H.2.5 dialog.
- 3.5.1.3.8 Configure ASC Humidy Thresholds is updated to follow the H.2.7 dialog instead of the H.2.5 dialog.

- 3.5.2.1.2.1.51 Configure Flashing Yellow Arrow Associated Vehicle Phase is updated to follow the 4.2.2 dialog instead of the H.2.7 dialog.
- 3.5.2.1.2.1.52 Configure Flashing Red Arrow Associated Vehicle Phase is updated to follow the 4.2.2 dialog instead of the H.2.7 dialog.
- 3.5.2.1.4.1.5 Configure Pattern Maximum Mode is updated to not specify any bits as patternOptions is an enumerated object.
- 3.5.2.1.8.1 Configure Overlap Type Requirements are updated with the correct enumerations and object traceability.
- Several requirements that reference the preemptTable are updated to follow the 4.2.2 dialog instead of the H.2.7 dialog. The requirements are:
    - 3.5.2.1.9.1.12-3.5.2.1.9.1.15
    - 3.5.2.1.9.1.17-3.5.2.1.9.1.21
    - 3.5.2.1.9.1.28
    - 3.5.2.1.9.1.34

D.3.2 Deprecated User Needs

D.3.2.1 Connected Vehicle Manager – Management Station – CV Application Process Interface

NTCIP 1202 v03B deprecates user need 2.5.4.2 Connected Vehicle Manager: Management Station – CV Application Process Interface and all of its subsequent user needs. These needs are superceeded by NTCIP 1218.

D.3.3 New Requirements

New requirements and objects to support the user needs expressed in Annex D.3.3. The following identifies these new requirements and supported in NTCIP 1202 v03.

D.3.3.1 Support SNMP Conformance Group

NTCIP 1202 v03B adds requirements to support the SNMP Conformance Group that was mandatory in NTCIP 1202 v02 and missed in 1202 v03.

D.3.3.2 Support Cabinet Flash (Rewrite this)

NTCIP 1202 v03B adds a requirement that allows an ASC to know how a cabinet is wired to flash.

D.3.3.3 Support Unit Coordination Point

NTCIP 1202 v03B adds a requirement that was missed in 1202 v03 to support the Coordination Point being the unitCoordSyncPoint object.

D.3.3.4 Support Minus Green Yellow Alternate Overlap

NTCIP 1202 v03B adds a requirement that was missed in 1202 v03 to support the Overlap Type being mnusGreenYellowAlternate.

D.3.3.5 Support Preempt Gate Description Configuration

NTCIP 1202 v03B adds a requirement to allow the preemptGateDescription object to be SET. Note: preemptGateDescription has been changed to read-write.

D.3.3.6 Configure Intersection Identifier

NTCIP 1202 v03B deprecates 3.5.4.1.3.2 Retrieve Intersection Identifier and adds new requirements, 3.5.4.2.1.12.2, Configure Signal Group Intersection Mapping, which allows the intersection identifier to be configured. Requirement 0,
Configure Road Authority allows the the road authority identifier to be configured.

D.3.3.7 New SPaT Requirements to Support CTI 4501

To support conformance to CTI 4501, NTCIP 1202 v03B adds several new requirements. These requirements include:

- Indiciating SPaT data as invalid.
- Providing SPaT data about the next movement.
- Provide more details regarding SPaT data broadcasted to other connected devices.

D.3.3.8 Security

NTCIP 1202 v03B adds new security requirements, including support for DTLS.

D.3.4 Deprecated Requirements

NTCIP 1202 v03B deprecates several requirements as follows:

D.3.4.1 Drop Support for Management Station – CV Application Process Interface

NTCIP 1202 v03B deprecates requirements for the interface between a management station and a CV Application Process Interface. These requirements are superseded by NTCIP 1218. The following requirements are deprecated.

- 3.5.4.2 Manage Management Station – CV Application Process Interface Requirements and all its subsequent requirements

D.3.4.2 Drop Support for RSU SNMP Managers

NTCIP 1202 v03B deprecates requirements for the RSU. The ASC Working Group came to a consensus that the ASC shall act as the SNMP manager. The following requirements are deprecated as they were written for RSUs acting as SNMP managers.

- 3.5.4.3.1.2 Retrieve Current and Next Movement Information Requirements and all its subsequent requirements.
- 3.5.4.3.2.2 Retrieve Movement Next Occurrence.
- 3.5.4.3.3.2 Provide Connected Devices Presence Information Requirements and all its subsequent requirements.
- 3.5.4.3.4.2 Provide Roadway Geometry Plan Requirements and all its subsequent requirements.
- 3.6.3.3 SPaT-data Request Transmission Rate.
- 3.6.3.4 Condition-based SPaT Maximum Transmission Start Time.

D.4 NTCIP 1202 v03B to NTCIP 1202 v04

D.4.1 Ported MIB to SMI V2 and SNMP v3

Previous versions of NTCIP 1202 used SNMP v1 as the communications protocol. Per recommendations of NTCIP 9014, NTCIP 1202 v04 has been ported to SNMP v3 using the communications stack defined in ISO 15074.

D.4.2 Global Object Definitions Moved to ISO 26048-1

Previous versions of NTCIP 1202 included normative references to NTCIP 1201 and NTCIP 1103. Those normative references have been replaced with normative references to ISO 26048-1.

D.4.3 Generic Dialogs Moved to ISO 26084-1

Previous versions of NTCIP 1202 defined dialogs for the GET and SET interface for scalar and tabular objects. The dialogs have been replaced with references to dialogs defined in ISO 26048-1.

D.4.4 New Requirements

D.4.4.1 Support for Multiple Phase Sets

NTCIP 1202 v04 adds object definitions for multiple phase sets that can be activated by a timing pattern. Select parameters including phaseStartUp, phaseRing, and phaseConcurrency are the same for all phase sets.

D.4.4.2 Support for Multiple Overlap Sets

NTCIP 1202 v04 adds object definitions for multiple overlap sets that can be activated by a timing pattern. The overlap type remains the same for all overlap sets.

D.4.4.3 Support for Multiple Vehicle Detector Sets

NTCIP 1202 v04 adds object definitions for multiple vehicle detector sets that can be activated by a timing pattern.

D.4.4.4 Support for Multiple Pedestrian Detector Sets

NTCIP 1202 v04 adds object definitions for multiple pedestrian detector sets that can be activated by a timing pattern.

D.4.4.5 Support for Alternate Walk and Ped Clearance Times During Transition

NTCIP 1202 v04 adds support for truncating pedestrian walk and clearance times during transition.

D.4.4.6 Support for Pedestrian Phase Advanced and Delayed Walk Activating by Detector

NTCIP 1202 v04 adds support for activating advanced pedestrian walks (leading pedestrian intervals) or delayed pedestrian walks based on pedestrian detector input.

D.4.4.7 Extra Capabilities for Timing Patterns

Timing Patterns in NTCIP 1202 v04 are capable of functions previously only driven by Actions. This includes the activating special functions, putting the controller into Automatic Flash, and enabling SPaT revocable lanes. A timing pattern may also report on the status of its parameters and why it may be a bad plan.

D.4.4.8 Support for Phase – Level Coord Force Mode.

NTCIP 1202 v04 adds support for the ability to specify whether each non-coordinated phase in a timing pattern is forced off in a fixed mode or a floating mode. Previously, this setting was the same for all non-coordinated phases in all timing patterns.

D.4.4.9 Support for Channel Alternate Flash First or Second

NTCIP 1202 v04 adds support for channels to flash in an alternating manner during Automatic Flash. This is known as a Wig-Wag.

D.4.4.10 Enhanced Ring Status

NTCIP 1202 v04 adds support for extra states that a ring could actively have regarding pedestrian indications and the ability to view how long a phase has been on for.

D.4.4.11 Support for Current Offset Monitoring

NTCIP 1202 v04 adds support to monitor the offset currently in effect.

D.4.4.12 Support for Detector Diagnostic Reset

NTCIP 1202 v04 adds support to reset the diagnostic counters for vehicle and pedestrian detectors.

D.4.4.13 Support for Signal Monitoring Unit Diagnostics

NTCIP 1202 v04 adds support for viewing voltage and current levels as reported by the Signal Monitoring Unit. This allows the user to determine if voltage and current levels are low enough to indicate possible malfunctions.

D.4.4.14 Support for External Local Control Application Data

NTCIP 1202 v04 adds support viewing minimum, maximum, and expected end times for currently active phases from ECLAs that control the controller by placing SNMP SETs to objects that directly impact the controller operations. This allows the controller using to broadcast connected vehicle messages while under control of an ECLA.

D.4.4.15 Enhanced Connected Vehicle and Device Detectors

NTCIP 1202 v04 adds support for connected vehicle detectors to support an Assured Green Period for a Red Light Violation Warning. Connected vehicle detectors are tied to vehicle or pedestrian detectors in NTCIP, allowing the connected vehicle detectors to inherit the diagnostic parameters associated with vehicle or pedestrian detectors. This also allows the connected vehicle application parameters, such as Assured Green Period, to be applied in connected intersections where the CV Application Process is external to the controller.

D.4.4.16 Support for Enabling Manual BackUp Mode

NTCIP 1202 v04 adds support for manually enabling BackUp Mode regardless of the BackUp timer and disables the system control parameters. This allows a manager to clear any control object immediately after disconnecting a device that may have set those objects.

D.4.4.17 Support for Detector Descriptions

NTCIP 1202 v04 adds support for adding user descriptions to vehicle and pedestrian detectors that can be used to specify detector location or call phases.

D.4.5 Deprecated Requirements

D.4.5.1 Pattern Table Type

NTCIP 1202 v04 deprecates to specify a pattern table type for controllers that used dial split offsetting.

D.4.5.2 Channel Dimming

NTCIP 1202 v04 deprecates support for channel diming. At the time of publication, signals heads use LEDs instead of incandescent lights. Signal heads are also programmable to support dimming.

D.4.5.3 Auxiliary Functions

NTCIP 1202 v04 deprecates support for auxiliary functions. Auxiliary functions can be activated at an intersection. NTCIP 1202 v04 adds support for more time based special functions.

D.4.5.4 Enable Vehicle Detector Volume and Occupancy Collection

It is expected that all vehicle detectors are capable of volume and occupancy collection. Therefore, NTCIP 1202 v04 deprecates the ability to enable or disable volume and occupancy data collection from vehicle collection.

D.4.5.5 User-Defined Backup Functions

NTCIP 1202 v04 deprecates support for user-defined backup timers and functions.

D.4.5.6 Disable Remote Commands

NTCIP 1202 v04 deprecates support for allowing a management station to disable remote commands. This is replaced with support for activating backup mode remotely for not allowing writes to system control parameters while the setting is enabled. Field users who may want to disable remote commands may do so by physically disconnecting the ASC from the network.

D.4.6 Changes to Existing Objects

D.4.6.1 Changes to Maximum Phase Green Times

In NTCIP 1202 v04, all phases may have three different maximum green times of 999 seconds.

D.4.6.2 Increased Pattern Cycle Time

In NTCIP 1202 v04, a pattern cycle time may be as long as 999 seconds, up from 255 seconds.

D.4.6.3 Increased Number of Time-Based Special Functions

In NTCIP 1202 v04, up to 32 special functions may be activated on a time based schedule, up from eight special functions.

D.4.6.4 Increased Detector No Activity Time

In NTCIP 1202 v04, a vehicle or pedestrian detector may report no actuations for up to 65535 minutes before it is classified as failed, up from 255 minutes.

D.4.6.5 Increased Remote Manual Control Time

In NTCIP 1202 v04, a manager can activate Remote Manual Control for up to 65535 seconds before having to reactivate it, up from 255 seconds.

D.4.6.6 Increased Remote Manual Control Time

In NTCIP 1202 v04, a manager can activate Remote Manual Control for up to 65535 seconds before having to reactivate it, up from 255 seconds.

D.4.6.7 Updated Alarm Status Objects

The alarm status objects have been redesigned to remove alarms that are applicable to any ITS device. Those alarms have been moved to ISO 26048-1. Additionally, support for the Local Cycle Zero alarm is deprecated as central systems no longer rely on it to record the cycle zero point. The Response Fault alarm is now an IO Link Error alarm that is no longer specific to NEMA cabinets. Alarms for active ECLAs and Priority Calls are now supported.

D.4.6.8 Updated Pre-Defined Block Definitions

NTCIP 1202 v04 includes new pre-defined block definitions that consolidate several definitions from NTCIP 1202 v03. The new pre-defined block definitions also add support for multiple sets for phases, overlaps, vehicle detectors, and pedestrians.

D.4.6.9 Updated Cycle Zero Point Objects

NTCIP 1202 v03 added the unitCoordSyncPoint and patternCoordSyncPoint objects allowing users to specify the reference point where pattern cycle begins and ends. To clarify ambiguities, these objects are deprecated. NTCIP 1202 v04 adds the patternReferencePhase and patternReferencePoint objects for the user to specify a single phase and interval as the reference point where pattern cycle begins and ends.

Annex E
User Requests [Informative]

Error! Reference source not found. identifies features that were suggested for NTCIP 1202 v04, but are either supported by mechanisms that may not be readily obvious, or are not supported by NTCIP 1202 v04.

E.1 Features Not Supported by This Version

Features considered for inclusion in NTCIP 1202 v04 but not included are as follows:

E.1.1 Interval Based Controllers

The NTCIP ASC Working Group (WG) considered including support for interval-based controllers (See Section 2.3.2) in NTCIP 1202 v03, but due to time and schedule considerations, was deferred to a future version of NTCIP 1202. User needs and requirements were developed for interval-based controllers, and draft design content to fulfill some of the requirements for interval-based controllers were developed but are not included in NTCIP 1202 v03. The developed content has been saved by NTCIP for consideration for future versions of NTCIP 1202.

E.1.2 Non-Persistent Timing Patterns

The NTCIP ASC WG considered including support for non-persistent traffic patterns in NTCIP 1202 v03, but due to time and schedule considerations, was deferred to a future version of NTCIP 1202. Non-persistent timing patterns are temporary timing patterns that are not retained in the ASC through a power loss. User needs were developed to support non-persistent traffic patterns but were not included in NTCIP 1202 v03.

E.1.3 Traffic Adaptive Algorithm

The NTCIP ASC WG considered including support for traffic adaptive strategies in NTCIP 1202 v03, but due to time and schedule considerations, was deferred to a future version of NTCIP 1202. Traffic adaptive is used to continuously adjust the signal timing pattern (cycle, offset and splits) based on traffic demand as determined by detector inputs and other inputs, such as from an adjacent signalized intersection. User needs and requirements were developed to support traffic adaptive strategies but are not included in NTCIP 1202 v03. The developed content has been saved by NTCIP for consideration for future versions of NTCIP 1202.

E.1.4 Peer-to-Peer

The NTCIP ASC WG considered including support for peer-to-peer communications in NTCIP 1202 v03, but due to time and schedule considerations and complexity of the topic, was deferred to a future version of NTCIP 1202. User needs, requirements, and draft design content were developed to support peer-to-peer communications were developed but are not included in NTCIP 1202 v03. The developed content has been saved by NTCIP for consideration for future versions of NTCIP 1202.

E.1.5 Signal Control Priority

The NTCIP ASC WG considered including support for signal control priority, including for a connected vehicle environment, in NTCIP 1202 v03, but the ASC WG that the user needs, requirements and design were better handled by the Signal Control and Priority (SCP) WG. During the development phase, the ASC WG identified several new requirements related to signal control priority, especially as it relates to the connected vehicle environment. New design content was also developed to support most of the new requirements. These new requirements and design content have been saved by NTCIP and provided to the SCP WG for consideration for future versions of NTCIP 1211.

E.1.6 Additional Support for ADA

The NTCIP ASC WG considered including support for ADA pedestrians using non-visual formats, such as audible tones, verbal messages, and/or vibrating surfaces, but due to time and schedule considerations, were deferred to a future version of NTCIP 1202. User needs, requirements and draft design content were developed to support non-visual formats, but consensus was not reached on the design content, and there were higher priority user needs. The developed content has been saved by NTCIP for consideration for future versions of NTCIP 1202.

E.1.7 Programmable Logic Gates and Functions

The NTCIP ASC WG considered and developed user needs, requirements, and system detailed design (SDD) to support programmable logic functions and gates. However, following the User Comment period, the ASC WG agreed to remove programmable logic functions and gates. While the proposed NTCIP 1202 v03 design incorporated most of the desired functions, custom and unique functions and operations (that were not addressed by the proposed SDD) already exist and are implemented. For that reason, non-NTCIP-standardized objects are still needed and used. In this instance, standardizing programmable logic and gates in NTCIP 1202 v03 offered little benefit to managing this functionality. A Section 3.3.2.1 provision was added to require the provision of a proprietary MIB for ASC devices that wish to claim conformance to NTCIP 1202 v03.

E.1.8 Advanced Preempt Inputs

The NTCIP ASC WG considered support for advanced preempt inputs, such as provided by IEEE 1570 was considered, but due to time and schedule considerations and complexity of the topic, was deferred to a future version of NTCIP 1202.

E.1.9 Signal Monitoring Unit Programmed Conflicts

NTCIP 1202 v04 added support for voltage and current levels from the Signal Monitoring Unit. There a request to be able to view the programmed conflicts but from the signal monitoring unit but a design could not be established for NTCIP 1202 v04 due to limitations in what is reported by a signal monitoring unit.

E.1.10 Clarified Overlap Functionality

NTCIP 1202 v03 added support for overlap parameters such as additional overlap types and overlap walk time. However, these parameters have been implemented differently by manufacturers. It has been requested to clarify how controllers should operate based on the overlap parameters. However, a design could not be established in a timely manner.

E.1.11 Advisory Speed by Lane

The ASC WG considered a proposal

- The RSU should populate the J2735 message by repeating the same advisory speed for each movement event associated with the signal group, since NTCIP only provides the advisory speeds at the signal group (channel) level.
- Recommend conferring with TOSCo and any others that are providing Eco-Driving to see if it is necessary to change NTCIP 1202 to provide advisory speeds at the movement event (interval) level, like J2735 does, rather than only at the signal group (channel) level.
- NOTE: TOSCo provides on a per lane basis – provide a green window.

E.1.12 Enabled Lanes During Preemption

The ASC WG considered requirements to support enabling lanes during a preempt, particularly to indicate no right turn on red is permitted during railroad preempts.

E.1.13 MAP Consistency Code

The ASC WG deprecated several requirements regarding the configuration of a MAP Message. These requirements are superseded by NTCIP 1218. The ASC WG needs more input from RSU vendors regarding how to do checksums.

E.1.14 Maneuver by Lane

NTCIP 1202 v03 is designed to be compatible with SAE J2735 and as result considers maneuvers by movements. It was requested to add requirement to support maneuvers by lane.

E.1.15 Support for DTLS

The ASC WG considered utilization of the Datagram Transport Layer Security (DTLS) Protocol Version 1.3 to protect all communications between the ASC and the RSU, with the exception that if a standardized interaction explicitly identified in this standard does not support the use of DTLS 1.3 then a different security protocol shall be used for that interaction. At the time of publication, the Base Standards and Protocols 2 Working Group (BSP2 WG) were considering recommendations for security, so this topic was deferred. The new user needs and requirements have been saved by NTCIP for consideration for future versions of NTCIP 1202.

Annex F
Generic Concepts and Definitions [Informative]

F.1 Meaning of 'Other' as a Value

To obtain the goals of interoperability and interchangeability among devices complying with NTCIP standards, the standard has been written to allow the introduction and use of values and methods other than those defined in the standard, but with the caveat that the setting of values within standardized functions and objects does not allow to set a value to 'other', while the retrieval of values within standardized functions and objects includes a value of 'other'. This approach allows non-standardized approaches to address special operations, while maintaining the integrity of the standard.

Should the use of a value of 'other' be allowed to be set for very specific standardized functions and objects, the intended operations follow:

a) SET value of standardized object to a value of 'other' indicating that none of the standardized values are to be use.
b) Use the vendor's detailed documentation to determine the vendor-specific function or object that is associated with the standardized function or object whose value has been set to 'other'.
c) SET of vendor-specific function or object to the desired value
d) When performing a GET on the standardized function or object, the ASC returns a value of 'other' and it is assumed that the vendor-specific function or object is known and can be queried to return the configured value to address the desired function or object.

F.2 Manufacturer-Specific Consistency Checks

There are functional differences between CU's manufactured by different vendors. It is assumed that manufacturers will use consistency checks, beyond those specified here, to prevent accidental corruption of the CU database. Any such consistency checks have to utilize the error reporting mechanism defined by this standard. These consistency checks and associated error messages should be clearly described and documented.

This Annex provides:

a) Examples of how a management station may interface with an ASC complying with this standard as envisioned by the authors. Any ASC claiming conformance with the subject features depicted in these figures shall support the exchanges as shown. However, the flexible design of the NTCIP protocols allows a large number of other possibilities and these figures do not limit any other requirements of these standards. These diagrams are merely provided to promote a common understanding of how systems may be designed in order to increase the likelihood of interoperability in deployed systems.
b) Supplemental information on overlap sequences based on programming data for 'overlapIncludedPhases' and 'overlapModifierPhases'.

F.3 Connected Intersections

A Connected Intersection (CI) is defined as an infrastructure system that broadcasts Signal, Phase, and Timing (SPaT), mapping information, and positional correction data to directly vehicles via a Roadside Unit (RSU). At a CI, a RSU broadcasts intersection and approach geometry (MAP) and SPAT messages to connected devices near the intersection. The source of the MAP data is from an agency or map provider, who loads the MAP data to the RSU for broadcasting. The source of the SPAT data is the traffic controller, who provides the data to the RSU for broadcasting. Connected devices, such as a connected vehicle or a connected pedestrian, uses the MAP and SPAT data to traverse through the intersection. For additional information about a connected intersection, refer to CTI 4501, Connected Intersections (CI) Implementation Guide.

From the context of NTCIP 1202, the RSU is viewed as a functional entity called the CV Application Process in the context of NTCIP 1202. The CV Application Process may reside in a separate physical device (such as the RSU) or as a functional process within the ASC Controller Unit. The CV Application Process is responsible for managing and processing connected vehicle applications, and to receive and broadcast connected vehicle (e.g., SAE J2735) messages. The ASC Process is a functional entity that resides within the ASC Controller Unit and is responsible for managing signal operations for a signalized intersection.

Part of the following guidance was developed by a Traffic Controller Vendors Subcommittee from the Connected Intersections project, sponsored by USDOT JPO. This subcommittee was part of the Traffic Controller Issues Task Force (TCI TF) of the Connected Intersections (CI) Steering Committee. The Subcommittee was tasked to establish a consistent set of assumptions and criteria that should be adhered to when generating SPaT data from within NTCIP 1202-based traffic controller firmware for all operational modes of traffic control. This commonality aims to reduce the vendor-specific ambiguity of controller generated SPaT data as well as seeks to improve the determinism and quality of SPaT data when sourced by NTCIP 1202 based traffic controllers. The full guidance by the Subcommittee can be found in Annex A of CTI 4501 v01. It was a goal of CTI 4501 to provide SPaT data in a nationally interoperable manner for a connected intersection during all possible times and conditions of signalized operation including non-standard scenarios. This includes, but is not limited to, startup, flashing, manual control, and preemptive operation.

F.3.1 Base Assumptions

This section provides the base assumptions and input parameters that all NTCIP 1202 traffic controllers should apply when generating SPaT data.

F.3.1.1 Active TOD Control

SPaT data shall be generated by the traffic control firmware under assumption that the currently active time of day plans (Event Plan), input driven commands, or active pages will remain currently active. This includes coordination patterns, timing plans, sequence tables, and other “pages” that are run on a time-of-day basis and/or triggered from a variety of potential sources. A traffic controller's SPaT data can be enhanced to “look ahead” into pending plans, patterns, sequences, and pages that are soon to become active, but this is not a requirement of the control firmware.

F.3.1.2 Active Preemption Control

SPaT data shall be generated under assumption that any inactive preemption, NTCIP 1211, or SRM requests will remain inactive and any active priority requests will remain active until served. The controller firmware shall update SPaT values within one second upon receiving changes to priority control inputs. One second was selected to allow the controller sufficient time to determine various analog inputs (e.g., high / low for a preemption input). The controller still must update the SPaT values as required in Sections 3.6.3.1 and 3.6.3.3.

F.3.1.3 Active Demand

SPaT data shall be generated under assumption of any active demand for phase or pedestrian service. This demand can be sourced from cabinet inputs, central system commands, controller configuration, internal logic, or various other means. The controller shall generate SPaT data upon current demand (Phase/Ped calls) and be updated within one tenth of a second upon changes to this demand (See Sections 3.6.3.1 and 3.6.3.3). Note that the demand need not be actively serviceable to be considered in the SPaT message. As example, a coordinator may omit a phase until its permissive window opens, however the SPaT data generated shall consider this phase to have serviceable demand with expectation of future opening of this permissive window.

F.3.1.4 NTCIP 1202 Configuration

SPaT data shall be generated with consideration of the NTCIP configuration of the traffic controller for those objects listed below. These objects form the minimal basis of consideration for SPaT generation. Vendors are encouraged to improve the accuracy of SPaT by incorporation of additional features or algorithmic logic, however, this minimal set of features, if supported by the traffic controller, shall be factored into SPaT generation. There is not a documented basis for “factoring” in these features. It is expected that each traffic controller vendor maintains active timers for these features and can compute the expected phase sequence and timing intervals using these internal mechanisms. SPaT data shall be updated within one tenth of a second after any consistency checks and transactional database changes are applied to the traffic controller.

Example: If the traffic controller supports options to include MUTCD 3 second all red pedestrian clearance, pedestrian change through yellow, and pedestrian change through red, then the TimeMark generated for the end of green estimation must be adjusted based upon the pedestrian timing to include these 3 features if enabled as follows:

```
TimeMark >=   Current Time
              + Pedestrian Walk Time Remaining
              + Pedestrian Change Time Remaining
              + MUTCD 3 Second Pedestrian Clearance (if enabled)
              – Pedestrian Change Through Yellow (if enabled)
              – Pedestrian Change Through Red (if enabled)
```

Some of these objects and how they affect SPaT should fully self-explanatory (e.g., dual entry is highlighted below so that type of call should be treated as demand when applicable). Others may require more detailed guidance for how the object is to be applied.

```
PhaseEntry ::= SEQUENCE {
        phaseNumber             INTEGER,
        phaseWalk               INTEGER,
        phasePedestrianClear    INTEGER,
        phaseMinimumGreen       INTEGER,
        phasePassage            INTEGER,
        phaseMaximum1           INTEGER,
        phaseMaximum2           INTEGER,
        phaseYellowChange       INTEGER,
        phaseRedClear           INTEGER,
        phaseRedRevert          INTEGER,
        phaseAddedInitial       INTEGER,
        phaseMaximumInitial     INTEGER,
        phaseDynamicMaxLimit    INTEGER,
        phaseDynamicMaxStep     INTEGER,
        phaseOptions            INTEGER,
        phaseRing               INTEGER,
        phaseConcurrency        OCTET STRING,
        phaseMaximum3           INTEGER,
        phaseYellowandRedChangeTimeBeforeEndPedClear     INTEGER,
        phasePedWalkService     INTEGER,
        phaseDontWalkRevert     INTEGER,
        phasePedAlternateClearance      INTEGER,
        phasePedAlternateWalk           INTEGER,
        phasePedAdvanceWalkTime         INTEGER,
        phasePedDelayTime               INTEGER,
        phaseAdvWarnGrnStartTime        INTEGER,
        phaseAdvWarnRedStartTime        INTEGER,
        phaseAltMinTimeTransition       INTEGER }
```

PhaseStatusGroupEntry ::= SEQUENCE { }
PhaseControlGroupEntry ::= SEQUENCE { }

unitStartUpFlash
unitAutoPedestrianClear
unitRedRevert
unitFlashStatus SYNTAX INTEGER { notFlash(2),
    automatic(3),
    localManual(4),
    faultMonitor(5),
    mmu(6),
    startup(7),
    preempt (8)}
unitAlarmStatus2
    Bit 5: Offset Transitioning – Whenever the CU is performing an offset transition (correction in process)
    Bit 4: Stop Time – When either CU Stop Time Input becomes active.
    Bit 3: External Start – When the CU External Start becomes active.
unitAlarmStatus1
    Bit 7: CoordActive – When coordination is active and not preempted or overridden.
    Bit 6: Local Free – When any of the CU inputs and/or programming cause it not to run coordination.
    Bit 5: Local Flash – When the Controller Unit Local Flash input becomes active, MMU Flash input is not active, and Flash is not commanded by the system.
    Bit 4: MMU Flash – When the Controller Unit MMU Flash input remains active for a period of time exceeding the Start-Up Flash time.
    shortAlarmStatus
    Bit 1: T&F Flash – When either the Local Flash or MMU Flash input becomes active.
    Bit 0: Preempt – When any of the CU Preempt inputs become active.
unitControl
    Bit 5: Walk Rest Modifier
    Bit 4: Call to Non-Actuated 2
    Bit 3: Call to Non-Actuated 1
    Bit 2: External Minimum Recall

MCE/Manual Advance

PreemptEntry ::= SEQUENCE { }
RingControlGroupEntry ::= SEQUENCE { }
channelGreenType SYNTAX INTEGER { other (1),
    protected (2),
    permissive (3),
    flashYellow (4),
    flashRed (5) }
OverlapEntry ::= SEQUENCE { }

F.3.1.5 Proprietary Features

Vendors are encouraged, but not required to implement SPaT for proprietary controller features. Those features that impact the features listed above, shall be assumed to be incorporated into SPaT when these dependent features are affected (e.g., Minimum Green affected by proprietary queue measurement techniques).

F.3.1.6 Externally Coordinated and/or Adaptively Controlled Intersections

In certain cases, the resultant signal state durations are not being decided by the traffic controller but rather an external process that is asserting a higher-level control over the traffic controller. The traffic controller may be commanded to run free, hold-on-line, or run specific coordination patterns by the external control process that are then manipulated in real time by hold/force-off/omit or other remote commands. CTI 4501 describes this process as an External Control Local Application (ECLA). SPaT messages generated by the traffic controller in these cases will not be accurate to the future state control commands that are offered by the external process.

As example, an intersection running under an external command to free or hold on-line, may not have awareness of serviceable side street demand or pending force off commands, that are being managed by the external process. In these cases, the external process should carry the responsibility to distribute the SPaT messages since it alone knows the likely future state of the intersection. It, however, may not have awareness of cabinet I/O, preemptive control, or other higher priority actions within the traffic controller.

The ASC WG discussed if the ASC or the ECLA should be responsible for generating the SPaT data needed for a J2735 SPaT message. Contrary to CTI 4501, the ASC WG believes that the controller needs to provide the SPaT data because the controller ultimately controls the signal phases and timing for the signalized intersection (except for cabinet flash). The ASC WG agreed that exchanging data between the controller and the ECLA are needed to properly support the generation of SPaT data, such as the ECLA providing the minimum and maximum end times. ~~However, due to time and schedule considerations, the requirements and design to support these data exchanges are deferred to a future version of NTCIP 1202.~~

F.3.1.7 Post-processed SPaT Generation

There are expected cases where the SPaT message is not generated by the traffic controller, but rather a separate device that is observing the operation of the traffic controller. This could include MMU/CMUs that are observing the actual field outputs, or a process that is performing NTCIP polling of the traffic controller status and issuing SPaT messages on behalf of the traffic controller. These methods will be limited in their knowledge of the inner-workings of the traffic controller, however, offer some means to retro-fit SPaT functionality to intersections running legacy control equipment which cannot support SPaT message generation.

In these cases, the device that is generating the SPaT message is not likely to be able to handle more detailed cases, such as timed overlaps, preemptions, and/or flash events. Given the critical safety nature of SPaT messages, it is recommended that this approach be taken only when the signals are running simple sequences and conditions that the post-processed SPaT generator can accurately represent.

F.3.2 Conformance

For an ASC to claim conformance to NTCIP 1202 for a connected vehicle environment, the ASC shall provide the connected vehicle interface for a management station including all mandatory requirements identified in the PRL Table (

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.3	Reference Physical Architecture [Informative]					
2.3.1	ASC Characteristics – Cabinet Specifications [Normative]			M	Yes	
2.3.1.a (332)	Model 332 Cabinet			O.1 (1)	Yes / No	
2.3.1.b (TS1)	NEMA TS 1 Cabinet			O.1 (1)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.3.1.c (TS2-2)	NEMA TS 2 Type 2 Cabinet			O.1 (1)	Yes / No	
2.3.1.d (TS2-1)	NEMA TS 2 Type 1 Cabinet			O.1 (1)	Yes / No	
2.3.1.e (ATC)	ATC Cabinet			O.1 (1)	Yes / No	
2.3.2	ASC Characteristics – Controller Types [Normative]			M	Yes	
2.3.2.a	Phase-based controller			M	Yes	
2.3.2.b	Interval-based controller			NA	NA	Interval-based controllers are not supported by NTCIP 1202 v04
2.3.2.c	Stage-based controller			NA	NA	Stage-based controllers are not supported by NTCIP 1202 v04
2.4	Architectural Needs					
2.4.1	Provide Live Data			M	Yes	
		3.4.1.1	Retrieve Data	M	Yes	
		3.4.1.2	Deliver Data	M	Yes	
		3.4.1.3	Explore Data	M	Yes	
		3.4.1.4.1	Monitor SNMP Information	M	Yes	
		ISO 26048-1§ 8.5.2.6	Determine basic capabilities of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.7	Determine SNMP capabilities of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.8	Determine the SNMP engine identifier	M	Yes	
		ISO 26048-1§ 8.5.2.9	Monitor when the SNMP capabilities last changed	M	Yes	
		ISO 26048-1§ 8.5.4	Controller performance requirements	M	Yes	
		3.6.1	Response Time for Requests	M	Yes	The Response Time for all requests shall be ___ milliseconds (5-500: Default=25). Note: Redundant with the previous requirement.
		~~3.6.2~~	Atomic Operations	M	Yes	Note: should 3.6.2 be deleted?
2.4.2	Provide Data Blocks			O	Yes / No	
		3.4.2.1	Store Pre-defined Compressed Data Blocks	M	Yes	
		3.5.2.1.14.1.1	Configure Block Object Get Control - Phase Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.2	Configure Block Object Get Control - Vehicle Detector Data	O.2 (1..*)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.2.1.14.1.3	Configure Block Object Get Control - Pedestrian Detector Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.4	Configure Block Object Get Control - Pattern Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.5	Configure Block Object Get Control - Split Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.6	Configure Block Object Get Control - Overlap Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.7	Configure Block Object Get Control - Preempt Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.8	Configure Block Object Get Control - Sequence Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.9	Configure Block Object Get Control - Channel Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.2	Monitor Block Error Status - Error-causing Data Element	M	Yes	
		ISO 26048-1§ 8.7	Dynamic object feature	O	Yes / No	All requirements that trace to this feature are also inherited. See the Features To Requirements Traceability (FTRT) in ISO 26048-1.
2.4.3	Provide for Log Data Local Storage and Retrieval			O.3 (1..*)	Yes / No	Note: Either Logging or Exception Reporting is Mandatory in ISO 26048-1.
		ISO 26048-1§ 8.9	File Feature	M	Yes	
		ISO 26048-1§ 8.10	Logging feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.4.4	Provide for Database Management			M	Yes	
		ISO 26048-1§ 8.21	Transaction feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.4.5	Condition-based Exception Reporting			O.3 (1..*)	Yes / No	Note: Either Logging or Exception Reporting is Mandatory in ISO 26048-1.

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.1	Action feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.11	Notification feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.16	SNMP target feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5	Features					
2.5.1	Manage the ASC Configuration			M	Yes	
2.5.1.1	Manage Device Identity			M	Yes	
		3.5.1.1.1	Configure ASC Location - Antenna Offset	C	Yes / NA	Mandatory if an external GNSS device is attached to the ASC.
		ISO 26048-1§ 8.5.2.1	Configure the controller's identity	M	Yes	
		ISO 26048-1§ 8.5.2.2	Configure the default language	M	Yes	
		ISO 26048-1§ 8.5.2.5	Control remote reset of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.14	Obtain the field device configuration identifier	M	Yes	
		ISO 26048-1§ 8.5.3.1	Support maximum message size	M	Yes	The maximum SNMP message size shall be _____ bytes (Default: 484 bytes).
		ISO 26048-1§ 8.5.3.2	Support total memory	O	Yes / No	The controller shall support a minimum of _____ kilobytes of total memory.
		ISO 26048-1§ 8.5.3.3	Support changeable memory	O	Yes / No	The controller shall support a minimum of _____ kilobytes of changeable memory.
		ISO 26048-1§ 8.8.2.1.1	Configure the field device's location by providing coordinates	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.8.2.1.2	Configure the field device's location by copying coordinates	O	Yes / No	
		ISO 26048-1§ 8.8.2.1.3	Configure the field device's location by GNSS reading	O	Yes / No	
		ISO 26048-1§ 8.8.2.1.4	Configure the field device's physical components	M	Yes	
		ISO 26048-1§ 8.12	Owner feature	M	Yes	
2.5.1.2	Manage Communications			O	Yes / No	Propose to delete.
		3.5.1.2.1.1	Enable/Disable Communications Port	M	Yes	The ASC shall not be allowed to enable/disable the following ports numbers: ________
		3.5.1.2.2.1	Determine Number of ASC Communications Ports	M	Yes	
2.5.1.3	Manage Cabinet Environment			O	Yes / No	
		ISO 26048-1§ 8.8.2.1.4	Configure the cabinet's physical components	M	Yes	
		ISO 26048-1§ 8.8.12	Field device doors	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of doors to be supported (Default: 2) ____
		ISO 26048-1§ 8.8.13	Field device fans	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of fans supported (Default: 2) _____
		ISO 26048-1§ 8.8.15	Field device heaters	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of heaters supported (Default: 1) _____
		ISO 26048-1§ 8.8.16	Field device humidity	O	Yes / No	All requirements that trace to this

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of humidity sensors supported (Default: 1) _____
		ISO 26048-1§ 8.8.21	Field device temperature	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of temperature sensors supported (Default: 1) _____
2.5.1.4 (Power)	Monitor Power			O	Yes / No	
		ISO 26048-1§ 8.8.2.1.9	Monitor current power source	M	Yes	
		ISO 26048-1§ 8.8.2.2.1.a	Support power sources – mainline (alternating current) power	M	Yes	
		ISO 26048-1§ 8.8.2.2.1.b	Support power sources – battery power	UPS:M	Yes / NA	
		ISO 26048-1§ 8.8.2.2.1.c (Generator)	Support power sources – generator power	O	Yes / No	
		ISO 26048-1§ 8.2.2.2.1.d (Solar)	Support power sources – solar power	O	Yes / No	
		ISO 26048-1§ 8.8.2.2.2 (UPS)	Support UPS power	O	Yes / No	
		ISO 26048-1§ 8.8.10	Field device battery	UPS:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet battery supported (Default: 1) _____
		ISO 26048-1§ 8.8.14	Field device generator	Generator:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
						1. Minimum number of voltage sensors for the cabinet generator supported (Default: 1) _____
		ISO 26048-1§ 8.8.17	Field device mains power	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet mains power supported (Default: 1) _____
		ISO 26048-1§ 8.8.20	Field device solar power	Solar:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet solar power system supported (Default: 1) _____
2.5.1.5	Retrieve Operational Performance Data			O	Yes / No	
		ISO 26048-1§ 8.13	Recording feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.1.6	Manage Auxiliary External Inputs/Outputs			O	Yes / No	
		ISO 26048-1§ 8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Number of additional SRSA ports to support: _____. This value is exclusive of ports used to monitor cabinet power,

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
						cabinet doors, fans, heaters, temperature sensor and humidity sensors.
2.5.1.7	Manage Database			M	Yes	
		3.5.1.7.1	Determine Configuration Identifier Parameter Content	M	Yes	
		ISO 26048-1§ 8.5.2.14	Obtain the field device configuration identifier	M	Yes	
2.5.1.8	Manage Interface with External Detectors			O	Yes / No	See 2.5.3.5 for detector data collected by the ASC through external inputs. Requirements to view advanced detector data are found in NTCIP 1209.
2.5.1.9	Manage ASC Clock			M	Yes	
		ISO 26048-1§ 8.2.1	UTC clock	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.2.2	Local clock	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.2.3	Daylight saving time	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.1.10	Manage External Control Local Application State			O	Yes / No	
		3.5.1.10.1.1	Enable ECLA Communications	M	Yes	
		3.5.1.10.1.2	Disable ECLA Communications	M	Yes	
		3.5.1.10.2	Monitor ECLA Data Input Time	M	Yes	
2.5.2	Manage Signal Operations			M	Yes	
2.5.2.1	Manage Signal Configuration			M	Yes	
2.5.2.1.1	Manage Controller Startup Functions			M	Yes	
		3.5.2.1.1.1.1	Configure Start-Up Flash Mode	O	Yes / No	
		3.5.2.1.1.1.2	Configure Start-Up Flash Time	M	Yes	
		3.5.2.1.1.2	Configure Backup Time	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
2.5.2.1.2	Manage Phase Configurations			M	Yes	
		3.5.2.1.2.1.1	Enable/Disable Phase	M	Yes	
		3.5.2.1.2.1.2	Configure Phase Minimum Green Time	M	Yes	
		3.5.2.1.2.1.3	Configure Phase Passage Time	M	Yes	
		3.5.2.1.2.1.4	Configure Two Fixed Phase Maximum Green Times	M	Yes	
		3.5.2.1.2.1.5	Configure Three Fixed Phase Maximum Green Times	O	Yes / No	
		3.5.2.1.2.1.6	Configure Phase Yellow Change Time	M	Yes	
		3.5.2.1.2.1.7	Configure Phase Red Clearance Time	M	Yes	
		3.5.2.1.2.1.8	Configure Phase Red Revert Time	O	Yes / No	
		3.5.2.1.2.1.9	Configure Unit Red Revert Time	Unit:M	Yes / NA	
		3.5.2.1.2.1.10	Configure Phase Added Initial Time	M	Yes	
		3.5.2.1.2.1.11	Configure Phase Maximum Initial Time	M	Yes	
		3.5.2.1.2.1.12	Configure Phase Time Before Reduction	M	Yes	
		3.5.2.1.2.1.13	Configure Phase Time to Reduce	M	Yes	
		3.5.2.1.2.1.14	Configure Phase Cars Before Reduction	O	Yes / No	
		3.5.2.1.2.1.15	Configure Phase Reduce By Time	O	Yes / No	
		3.5.2.1.2.1.16	Configure Phase Minimum Gap Time	M	Yes	
		3.5.2.1.2.1.17	Configure Phase Dynamic Maximum Limit	O	Yes / No	
		3.5.2.1.2.1.18	Configure Phase Dynamic Maximum Step	O	Yes / No	
		3.5.2.1.2.1.19	Configure Phase Start-Up State	M	Yes	
		3.5.2.1.2.1.20	Configure Automatic Flash Entry Phase	O	Yes / No	
		3.5.2.1.2.1.21	Configure Automatic Flash Exit Phase	O	Yes / No	
		3.5.2.1.2.1.22	Configure Call to Non-Actuated 1	O	Yes / No	
		3.5.2.1.2.1.23	Configure Call to Non-Actuated 2	O	Yes / No	
		3.5.2.1.2.1.24	Configure Non-Lock Detector Memory	O	Yes / No	
		3.5.2.1.2.1.25	Configure Phase Minimum Vehicle Recall	M	Yes	Should this be mandatory?

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.2.1.2.1.26	Configure Phase Maximum Vehicle Recall	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.27	Configure Phase Soft Vehicle Recall	O	Yes / No	
		3.5.2.1.2.1.28	Configure Dual Phase Entry	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.29	Configure Simultaneous Gap Disable	O	Yes / No	
		3.5.2.1.2.1.30	Configure Guaranteed Passage	O	Yes / No	
		3.5.2.1.2.1.31	Configure Actuated Rest-in-Walk	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.32	Configure Conditional Service Enable	O	Yes / No	
		3.5.2.1.2.1.33	Configure Added Initial Calculation	O	Yes / No	
		3.5.2.1.2.1.34	Configure Phase-to-Ring Association	M	Yes	
		3.5.2.1.2.1.35	Configure Phase Concurrency	M	Yes	
		3.5.2.1.2.1.36	Configure Pedestrian Clearance Time Allowed During Vehicle Clearance	O	Yes / No	
		3.5.2.1.2.1.37	Configure Pedestrian Walk Time	M	Yes	
		3.5.2.1.2.1.38	Configure Pedestrian Clearance Time	M	Yes	
		3.5.2.1.2.1.39	Configure Pedestrian Phase Walk Service Limit	M	Yes	
		3.5.2.1.2.1.40	Configure Pedestrian Phase Don't Walk Revert Time	M	Yes	
		3.5.2.1.2.1.41	Configure Non-Lock Ped Detector Memory	M	Yes	
		3.5.2.1.2.1.42	Configure Pedestrian Phase Recall	M	Yes	
		3.5.2.1.2.1.43	Configure Phase Alternate Pedestrian Clearance Time	O	Yes / No	
		3.5.2.1.2.1.44	Configure Phase Alternate Pedestrian Walk Time	O	Yes / No	
		3.5.2.1.2.1.45	Configure Pedestrian Phase Advanced Walk Time	O	Yes / No	
		3.5.2.1.2.1.46	Configure Pedestrian Phase Delayed Walk Time	O	Yes / No	
		3.5.2.1.2.1.47	Configure Phase Advance Warning Green	O	Yes / No	
		3.5.2.1.2.1.48	Configure Phase Advance Warning Red	O	Yes / No	
		3.5.2.1.2.1.49	Configure Flashing Yellow Arrow Associated Vehicle Phase	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.2.1.50	Configure Flashing Red Arrow Associated Vehicle Phase	O	Yes / No	
		3.5.2.1.2.1.51	Configure Alternate Minimum Green Time during Transition	O	Yes / No	
		3.5.2.1.2.1.52	Configure Alternate Minimum Walk Time during Transition	O	Yes / No	
		3.5.2.1.2.1.53	Configure Alternate Minimum Pedestrian Clearance Time during Transition	O	Yes / No	
		3.5.2.1.2.2(PhaseSet)	Configure Multiple Phase Sets	O	Yes / No	
		3.5.2.1.2.3.1	Determine Maximum Number of Phases	M	Yes	The ASC shall support at least _____ phases.
		3.5.2.1.2.3.2	Determine Maximum Number of Phase Sets	PhaseSet:M	Yes / NA	The ASC shall support at least _____ phase sets.
2.5.2.1.3 (Coord)	Manage Coordination Configurations			O	Yes / No	
		3.5.2.1.3.1	Configure Operational Mode for Coordination	M	Yes	The ASC shall support the following values: ___ automatic ___ pattern ___ manual free ___ manual flash
		3.5.2.1.3.2	Configure Correction Mode for Coordination	M	Yes	The ASC shall support the following values: __ dwell ___ shortway __ add only ___ subtract only
		3.5.2.1.3.3	Configure Maximum Mode for Coordination	M	Yes	The ASC shall support the following values: ___ maxInhibit ___maximum1 ___ maximum2 ___maximum3
		3.5.2.1.3.4	Configure Unit-level Force Mode for Coordination	O	Yes / No	
		3.5.2.1.3.5	Configure Phase-level Force Mode for Coordination	O	Yes / No	
		3.5.2.1.3.6	Configure Pattern Reference Phase	M	Yes	
		3.5.2.1.3.7	Configure Pattern Reference Point	M	Yes	
		3.5.2.1.3.9	Configure Pattern Synchronization Time	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.2.1.4	Manage Timing Patterns			Coord:M	Yes / NA	
		3.5.2.1.4.1.1	Configure Pattern to Run Free	M	Yes	
		3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	O	Yes / No	
		3.5.2.1.4.1.3	Configure Pattern Cycle Time for Coordination	M	Yes	
		3.5.2.1.4.1.4	Configure Pattern Offset Time	M	Yes	
		3.5.2.1.4.1.5	Configure Pattern Split Association	M	Yes	
		3.5.2.1.4.1.6	Configure Pattern Sequence Association	M	Yes	
		3.5.2.1.4.1.7	Configure Pattern Maximum Mode	O	Yes / No	
		3.5.2.1.4.1.8	Configure Pattern Phase Set	O	Yes / No	
		3.5.2.1.4.1.9	Configure Pattern Overlap Set	O	Yes / No	
		3.5.2.1.4.1.10	Configure Pattern Vehicle Detector Set	O	Yes / No	
		3.5.2.1.4.1.11	Configure Pattern Pedestrian Detector Set	O	Yes / No	
		3.5.2.1.4.1.12	Configure Pattern Special Functions	O	Yes / No	
		3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	M	Yes	The ASC shall support at least _____ timing patterns.
2.5.2.1.5	Manage Splits Configurations			O	Yes / No	
		3.5.2.1.3.8	Configure Omit Phases During Transitions	O	Yes / No	
		3.5.2.1.5.1.1	Configure Phase Split Time	M	Yes	
		3.5.2.1.5.1.2	Configure Phase Split Mode	M	Yes	
		3.5.2.1.5.1.3	Configure Split Coordination Phase	M	Yes	
		3.5.2.1.5.2	Determine Maximum Number of Phase Splits	M	Yes	The ASC shall support at least _____ splits
2.5.2.1.6 (Ring)	Manage Ring Configurations			O	Yes / No	
		3.5.2.1.6.1	Configure Sequence Data	M	Yes	
		3.5.2.1.6.2	Determine Maximum Number of Rings	M	Yes	The ASC shall support at least _____ rings
		3.5.2.1.6.3	Determine Maximum Number of Sequences	M	Yes	The ASC shall support at least _____ sequences

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.2.1.7 (Channel)	Manage Channel Configurations			O	Yes / No	
		3.5.2.1.7.1.1	Configure Channel Control Source	M	Yes	
		3.5.2.1.7.1.2	Configure Channel Control Type	M	Yes	
		3.5.2.1.7.1.3.1	Configure Channel Flash Yellow	M	Yes	
		3.5.2.1.7.1.3.2	Configure Channel Flash Red	M	Yes	
		3.5.2.1.7.1.3.3	Configure Channel Flash Alternate Half Hertz	O	Yes / No	
		3.5.2.1.7.1.3.4	Configure Channel Flash Alternate First or Second	O	Yes / No	
		3.5.2.1.7.2	Determine Maximum Number of Channels	M	Yes	The ASC shall support at least ____ channels (See appropriate hardware specification such as NEMA TS 2 to determine maximum number of supported channels)
2.5.2.1.8 (Overlap)	Manage Overlap Configurations			O	Yes / No	
		3.5.2.1.8.1.1.1	Configure Overlap Type - Vehicle Normal	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.2	Configure Overlap Type - Vehicle Minus Green and Yellow	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.3	Configure Overlap Type - Pedestrian Normal	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.4	Configure Overlap Type - Flashing Yellow Arrow - 3 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.5	Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.6	Configure Overlap Type - Flashing Red Arrow - 3 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.7	Configure Overlap Type - Flashing Red Arrow - 4 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.8	Configure Overlap Type - 2 Section Transit Specific Signal Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.9	Configure Overlap Type - Minus Green Yellow Alternate	O.4 (1..*)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.8.1.2	Configure Overlap Included Phases	M	Yes	
		3.5.2.1.8.1.3	Configure Overlap Modifier Phases	M	Yes	
		3.5.2.1.8.1.4	Configure Pedestrian Modifier Phases	O	Yes / No	
		3.5.2.1.8.1.5	Configure Overlap Trailing Green	M	Yes	
		3.5.2.1.8.1.6	Configure Overlap Trailing Yellow	M	Yes	
		3.5.2.1.8.1.7	Configure Overlap Trailing Red Clearance	M	Yes	
		3.5.2.1.8.1.8	Configure Overlap Walk	O	Yes / No	
		3.5.2.1.8.1.9	Configure Overlap Pedestrian Clearance	O	Yes / No	
		3.5.2.1.8.2 (OverlapSet)	Configure Multiple Overlap Sets	O	Yes / No	
		3.5.2.1.8.3.1	Determine Maximum Number of Overlaps	M	Yes	The ASC shall support at least ____ overlaps
		3.5.2.1.8.3.2	Determine Maximum Number of Overlap Sets	Overlap Set:M	Yes / NA	The ASC shall support at least ____ overlap sets.
2.5.2.1.9 (Preempt)	Manage Preempt Configurations			O	Yes / No	
		3.5.2.1.9.1.1	Enable/Disable Preempt Inputs	M	Yes	
		3.5.2.1.9.1.2	Configure Preempt Control - Non-Locking Memory	O	Yes / No	
		3.5.2.1.9.1.3	Configure Preempt Control - Override Automatic Flash	O	Yes / No	
		3.5.2.1.9.1.4	Configure Preempt Control - Override Preempt	O	Yes / No	
		3.5.2.1.9.1.5	Configure Preempt Control - Flash Dwell	O	Yes / No	
		3.5.2.1.9.1.6	Configure Preempt Control - All Red Entry	O	Yes / No	
		3.5.2.1.9.1.7	Configure Preempt Link	M	Yes	
		3.5.2.1.9.1.8	Configure Preempt Delay	M	Yes	
		3.5.2.1.9.1.9	Configure Preempt Minimum Duration	M	Yes	
		3.5.2.1.9.1.10.1	Configure Preempt Enter Minimum Green Time	O	Yes / No	
		3.5.2.1.9.1.10.2	Configure Preempt Enter Minimum Walk Time	O	Yes / No	
		3.5.2.1.9.1.10.3	Configure Preempt Enter Pedestrian Clearance Time	O	Yes / No	
		3.5.2.1.9.1.10.4	Configure Preempt Enter Yellow Change Time	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.2.1.9.1.10.5	Configure Preempt Enter Red Clearance Time	O	Yes / No	
		3.5.2.1.9.1.11.1	Configure Preempt Track Clearance Green Time	M	Yes	
		3.5.2.1.9.1.11.2	Configure Preempt Track Clearance Yellow Change Time	O	Yes / No	
		3.5.2.1.9.1.11.3	Configure Preempt Track Red Clearance Time	O	Yes / No	
		3.5.2.1.9.1.11.4	Configure Preempt Track Clearance Phases	M	Yes	
		3.5.2.1.9.1.11.5	Configure Preempt Track Clearance Overlaps	O	Yes / No	
		3.5.2.1.9.1.12.1	Configure Preempt Minimum Green Dwell Time	M	Yes	
		3.5.2.1.9.1.12.2	Configure Preempt Dwell Phases	M	Yes	
		3.5.2.1.9.1.12.3	Configure Preempt Dwell Pedestrian Movements	O	Yes / No	
		3.5.2.1.9.1.12.4	Configure Preempt Dwell Overlaps	O	Yes / No	
		3.5.2.1.9.1.12.5	Configure Preempt Cycling Phases	O	Yes / No	
		3.5.2.1.9.1.12.6	Configure Preempt Cycling Pedestrian Movements	O	Yes / No	
		3.5.2.1.9.1.12.7	Configure Preempt Cycling Phases Sequence	O	Yes / No	
		3.5.2.1.9.1.12.8	Configure Preempt Cycling Overlaps	O	Yes	
		3.5.2.1.9.1.13.1	Configure Preempt Exit Phases	O	Yes / No	
		3.5.2.1.9.1.13.2	Configure Preempt Exit Phase Strategy	O	Yes / No	
		3.5.2.1.9.1.13.3	Configure Preempt Exit Priority Levels	C	Yes / No	Mandatory for Conformance if 'Exit to Queue Delay Recovery' is supported as a Preempt Exit Phase (See 3.5.2.1.9.1.13.2)
		3.5.2.1.9.1.14.1	Configure Preempt Maximum Presence Time	M	Yes	
		3.5.2.1.9.1.14.2	Configure Preempt Maximum Presence Action	M	Yes	
		3.5.2.1.9.1.15	Configure Preempt Gate Description	O	Yes / No	
		3.5.2.1.9.2	Determine Maximum Number of Preempts	M	Yes	The ASC shall support at least ____ preempts

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.2.1.10	Manage Timing Pattern Scheduler			M	Yes / No	
		ISO 26048-1§ 8.1.2.3	Configure an action owner	M	Yes	
		ISO 26048-1§ 8.1.2.4	Configure an action group	M	Yes	
		ISO 26048-1§ 8.1.2.5	Configure an action	M	Yes	
		ISO 26048-1§ 8.6	Day plan feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.2.1.11	Manage Action Scheduler			M	Yes / No	
		ISO 26048-1§ 8.1.2.3	Configure an action owner	M	Yes	
		ISO 26048-1§ 8.1.2.4	Configure an action group	M	Yes	
		ISO 26048-1§ 8.1.2.5	Configure an action	M	Yes	
		ISO 26048-1§ 8.6	Day plan feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		3.5.2.1.4.1.1	Configure Pattern to Run Free	patternFunction: M	Yes / NA	
		3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	patternFunction: M	Yes / NA	
		3.5.2.1.4.1.12(patternFunction)	Configure Pattern Special Functions	O.5 (1..*)	Yes / No	
		3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	patternFunction: M	Yes	
		3.5.2.1.10.1.1	Configure Timebased Action - Pattern	actionFunction:M	Yes / NA	
		3.5.2.1.10.1.2(actionFunction)	Configure Timebased Action - Special Functions	O.5 (1..*)	Yes / No	
		3.5.2.1.10.1.3	Determine Maximum Number of Timebased Actions	actionFunction:M	Yes / NA	
2.5.2.1.12	Manage I/O Mapping			O	Yes / No	
		3.5.2.1.11.1.1	Set Active I/O Map	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.2.1.11.1.2.1	Configure I/O Map Description	M	Yes	
		3.5.2.1.11.1.2.2.1	Configure I/O Map Input Device	M	Yes	
		3.5.2.1.11.1.2.2.2	Configure I/O Map Input Device Pin	M	Yes	
		3.5.2.1.11.1.2.2.3	Configure I/O Map Input Function	M	Yes	
		3.5.2.1.11.1.2.3.1	Configure I/O Map Output Device	M	Yes	
		3.5.2.1.11.1.2.3.2	Configure I/O Map Output Device Pin	M	Yes	
		3.5.2.1.11.1.2.3.3	Configure I/O Map Output Function	M	Yes	
		3.5.2.1.11.2.1	Retrieve Maximum Number of I/O Maps	M	Yes	
		3.5.2.1.11.2.2	Retrieve Maximum Number of I/O Map Inputs	M	Yes	
		3.5.2.1.11.2.3	Retrieve Maximum Number of I/O Map Outputs	M	Yes	
		3.5.2.1.11.2.4	Retrieve I/O Mapping Activate Conditions	M	Yes	The following conditions shall be satisfied before a new I/O map can be activated: ___ Cabinet Door Open ___ in any flash state ___ programmed all red flash ___ in CVM flash ___ ASC restart
		3.5.2.1.11.2.5	Retrieve I/O Mapping Input Functions	M	Yes	
		3.5.2.1.11.2.6	Retrieve I/O Mapping Output Functions	M	Yes	
		3.5.2.1.11.2.7	Retrieve I/O Map Input Device Pin Status	M	Yes	
		3.5.2.1.11.2.8	Retrieve I/O Map Output Device Pin Status	M	Yes	
		3.5.2.1.11.2.9.1	Enumerate I/O Map - FIO Inputs	332:M	Yes / NA	
		3.5.2.1.11.2.9.2	Enumerate I/O Map - FIO Outputs	332:M	Yes / NA	
		3.5.2.1.11.2.9.3	Enumerate I/O Map - TS1 Inputs	TS1, TS2-2:M	Yes / NA	
		3.5.2.1.11.2.9.4	Enumerate I/O Map - TS1 Outputs	TS1, TS2-2:M	Yes / NA	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.2.1.11.2.9.5	Enumerate I/O Map - TS2 BIU Inputs	TS2-1:M	Yes / NA	
		3.5.2.1.11.2.9.6	Enumerate I/O Map - TS2 BIU Outputs	TS2-1:M	Yes / NA	
		3.5.2.1.11.2.9.7	Enumerate I/O Map - ATC Cabinet SIU Inputs	ATC:M	Yes / NA	
		3.5.2.1.11.2.9.8	Enumerate I/O Map - ATC Cabinet SIU Outputs	ATC:M	Yes / NA	
		3.5.2.1.11.2.9.9	Enumerate I/O Map - Auxiliary Device Inputs	O	Yes / No	
		3.5.2.1.11.2.9.10	Enumerate I/O Map - Auxiliary Device Outputs	O	Yes / No	
2.5.2.1.13	Manage Intra-Cabinet Communications Configuration			O	Yes / No	
		3.5.2.1.12.1.1	Determine Serial Bus 1 Device Present	ATC:M	Yes / NA	The ASC shall support at least ____ Serial Bus 1 Addresses (between 1 and 255).
		3.5.2.1.12.2.1	Determine TS2 Port 1 Device Present	TS2-2:M	Yes / NA	The ASC shall support at least ____ TS2 Port1 Addresses (between 1 and 255).
		3.5.2.1.12.2.2	Enable/Disable TS2 Port 1 Frame 40 Messages	TS2-2:M	Yes / NA	
2.5.2.1.14	Manage Accessible Pedestrian Support			O	Yes / No	
		3.5.2.1.13.1.1	Configure APS Push Button Minimum Press Time	M	Yes	
		3.5.2.1.13.1.2	Configure APS Push Button to Phase Association	M	Yes	
		3.5.2.1.13.1.3	Configure APS Extra Crossing Time	M	Yes	
		3.5.2.1.13.1.4	Configure Pedestrian Detector for Alternate Pedestrian Timing	M	Yes	
2.5.2.2	Monitor Signal Operations Status					
2.5.2.2.1	Determine Controller Health			M	Yes	
		ISO 26048-1§ 8.5.2.9	Monitor when the SNMP capabilities last changed	M	Yes	
		ISO 26048-1§ 8.5.2.10	Monitor controller communications	M	Yes	
		ISO 26048-1§ 8.5.2.11	Monitor controller operational status	M	Yes	
		ISO 26048-1§ 8.5.2.12	Monitor controller up time	M	Yes	
		ISO 26048-1§ 8.5.2.13	Monitor watchdog failure count	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.1.1	Monitor External Alarm Input States	M	Yes	The ASC shall support at least ____ Alarm Groups (between 1 and 255).
		3.5.2.2.1.2	Monitor External Alarm Active	M	Yes	
		3.5.2.2.1.3	Monitor Flash Status	M	Yes	
		3.5.2.2.1.4	Monitor Local Override	M	Yes	
		3.5.2.2.1.5	Monitor Coordination Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.6	Monitor Detector Fault	Detector: M	Yes / NA	
		3.5.2.2.1.7	Monitor Stop Time Input Alarm	M	Yes	
		3.5.2.2.1.8	Monitor Cycle Fault Alarm	M	Yes	
		3.5.2.2.1.9	Monitor Coordination Fault	Coord:M	Yes / NA	
		3.5.2.2.1.10	Monitor Coordination Fail Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.11	Monitor Cycle Fail Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.12	Monitor Cabinet IO Link Alarm	M	Yes	
		3.5.2.2.1.13	Monitor SMU Communications Error	O	Yes / No	
		3.5.2.2.1.14	Monitor Preempt Maximum Presence Alarm	Preempt: O	Yes / No / NA	
2.5.2.2.2	Determine Mode of Operation					
2.5.2.2.2.1 (Unit)	Monitor Unit-wide General Operations			O	Yes / No	
		3.5.2.2.2.1	Monitor Unit Control Status	M	Yes	
		3.5.2.2.2.2	Monitor Preempt Active	Preempt: M	Yes / NA	
		3.5.2.2.2.3	Monitor Offset Transitioning	Coord:M	Yes / NA	
		3.5.2.2.2.4	Monitor Priority Call Active	O	Yes / No	
		3.5.2.2.2.5	Monitor Local Free Status	M	Yes	
		3.5.2.2.2.6	Monitor Coordination Active	Coord:M	Yes / NA	
		3.5.2.2.2.7	Monitor ECLA Control Active	O	Yes / No	
2.5.2.2.2.2	Monitor Flashing			M	Yes	
		3.5.2.2.1.3	Monitor Flash Status	M	Yes	
2.5.2.2.2.3	Monitor Current Timing Pattern			Coord:M	Yes / NA	
		3.5.2.2.2.5	Monitor Local Free Status	M	Yes	
		3.5.2.2.2.8.1	Monitor Current Pattern Status	M	Yes	
		3.5.2.2.2.8.2	Monitor Current Pattern Command Source	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.2.8.3	Monitor Current Pattern Fault Status	O	Yes / No	
2.5.2.2.2.4	Monitor Current Cycle			Coord:M	Yes / NA	
		3.5.2.2.2.9.1	Monitor Coordination Cycle Status	M	Yes	
		3.5.2.2.2.9.2	Monitor Coordination Synchronization Status	M	Yes	
		3.5.2.2.2.9.3	Monitor Current Offset	M	Yes	
2.5.2.2.3	Monitor Phase Status			M	Yes	
		3.5.2.2.3.1	Monitor Active Red Phases	M	Yes	
		3.5.2.2.3.2	Monitor Active Yellow Phases	M	Yes	
		3.5.2.2.3.3	Monitor Active Green Phases	M	Yes	
		3.5.2.2.3.4	Monitor Active Don't Walk Phases	M	Yes	
		3.5.2.2.3.5	Monitor Active Pedestrian Clearance Phases	M	Yes	
		3.5.2.2.3.6	Monitor Active Walk Phases	M	Yes	
		3.5.2.2.3.7	Monitor Active On Phases	M	Yes	
		3.5.2.2.3.8	Monitor Next Phases	M	Yes	
		3.5.2.2.3.9	Monitor Phase Vehicle Calls	M	Yes	
		3.5.2.2.3.10	Monitor Phase Pedestrian Calls	M	Yes	
		3.5.2.2.7.4	Monitor Current Gate Status	O	Yes / No	
2.5.2.2.4	Monitor Ring Status			Ring:M	Yes / NA	
		3.5.2.2.4.1	Monitor Ring Status	M	Yes	
		3.5.2.2.4.2	Monitor Ring Termination Cause	M	Yes	
		3.5.2.2.4.3	Monitor Current Phase On Time	M	Yes	
2.5.2.2.5	Monitor Channel Status			Channel: M	Yes / NA	
		3.5.2.2.5.1	Monitor Active Red Channels	M	Yes	
		3.5.2.2.5.2	Monitor Active Yellow Channels	M	Yes	
		3.5.2.2.5.3	Monitor Active Green Channels	M	Yes	
2.5.2.2.6	Monitor Overlap Status			Overlap: M	Yes / NA	
		3.5.2.2.6.1	Monitor Active Red Overlaps	M	Yes	
		3.5.2.2.6.2	Monitor Active Yellow Overlaps	M	Yes	
		3.5.2.2.6.3	Monitor Active Green Overlaps	M	Yes	
		3.5.2.2.6.4	Monitor Active Flashing Yellow Arrow Overlaps	O	Yes / No	
		3.5.2.2.6.5	Monitor Active Flashing Red Arrow Overlaps	O	Yes / No	
2.5.2.2.7	Monitor Preempt Status			Preempt: M	Yes / NA	
		3.5.2.2.7.1	Monitor Currently Active Preempt	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.7.2	Monitor Current Preempt Inputs	M	Yes	
		3.5.2.2.7.3	Monitor Current Preempt State	M	Yes	
		3.5.2.2.7.4	Monitor Current Gate Status	O	Yes / No	
2.5.2.2.8 (SpecialFunc)	Monitor Special Function Outputs			O	Yes / No	
		3.5.2.2.8.1	Determine Maximum Number of Special Functions	M	Yes	The ASC shall support at least ____ Special Functions (between 1 and 255).
		3.5.2.2.8.2	Monitor Special Function Status	M	Yes	
		3.5.2.2.8.3	Monitor Special Function Control Source	O	Yes / No	
2.5.2.2.9	Monitor Timebase Action Status			M	Yes	
		3.5.2.2.2.8.1	Monitor Current Pattern Status	patternFunction:M	Yes / NA	
		3.5.2.1.10.1.4	Determine Action in Effect	actionFunction:M	Yes / NA	
2.5.2.2.10	Monitor Intra-Cabinet Communications Configuration			O	Yes / No	
		3.5.2.2.9.1	Monitor TS2 Port 1 Status	TS2-2:M	Yes / NA	
		3.5.2.2.9.2	Monitor TS2 Port 1 Fault Frame	TS2-2:M	Yes / NA	
		3.5.2.2.9.3	Monitor Serial Bus 1 Status	ATC:M	Yes / NA	
2.5.2.2.11	Monitor Signal Monitoring Unit			O	Yes / No	
		3.5.2.2.10.1	Monitor Signal Monitoring Unit Channel Voltage	ATC:M	Yes	It's optional for any cabinet, but format is standard on an ATC.
		3.5.2.2.10.2	Monitor Signal Monitoring Unit Channel Current	ATC:M	Yes	
2.5.2.3	Control Signal Operations			M	Yes	
2.5.2.3.1	Control ASC-wide General Operations			M	Yes	
		3.5.2.3.1.1	Enable/Disable Manual Backup	O	Yes / No	
		3.5.2.3.1.2	Control Global Minimum Recall	M	Yes	
		3.5.2.3.1.3	Control Call to Non-Actuated 1	M	Yes	
		3.5.2.3.1.4	Control Call to Non-Actuated 2	M	Yes	
		3.5.2.3.1.5	Control Walk Rest Modifier	M	Yes	
		3.5.2.3.1.6	Control Interconnect	O	Yes / No	
2.5.2.3.2	Activate Timing Pattern			Coord:M	Yes / NA	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.3.2.1	Activate System Timing Pattern Remotely	M	Yes	
		3.5.2.3.2.2	Control System Reference Point	M	Yes	
2.5.2.3.3	Phase Requests			O	Yes / No	
		3.5.2.3.3.1	Control Phase Omits	M	Yes	
		3.5.2.3.3.2	Control Pedestrian Phase Omits	M	Yes	
		3.5.2.3.3.3	Control Phase Holds	M	Yes	
		3.5.2.3.3.4	Control Phase Force Offs	O	Yes / No	
		3.5.2.3.3.5	Control Phase Vehicle Calls	M	Yes	
		3.5.2.3.3.6	Control Phase Pedestrian Calls	M	Yes	
2.5.2.3.4	Activate Preempt			Preempt: O	Yes / No / NA	
		3.5.2.3.4	Activate Preempt Remotely	M	Yes	
2.5.2.3.5	Control Ring Operations			Ring:O	Yes / No / NA	
		3.5.2.3.5.1	Control Ring Stop Time	M	Yes	
		3.5.2.3.5.2	Control Ring Force Offs	M	Yes	
		3.5.2.3.5.3	Control Ring Maximum 2 Settings	M	Yes	
		3.5.2.3.5.4	Control Ring Maximum 3 Settings	O	Yes / No	
		3.5.2.3.5.5	Control Ring Maximum Inhibit Settings	M	Yes	
		3.5.2.3.5.6	Control Ring Pedestrian Recycle Settings	M	Yes	
		3.5.2.3.5.7	Control Ring Red Rest Settings	M	Yes	
		3.5.2.3.5.8	Control Ring Red Clearance Omit Settings	M	Yes	
2.5.2.3.6	Activate Special Function Output			SpecialFunc:O	Yes / No / NA	
		3.5.2.3.6	Activate Special Function Remotely	M	Yes	
2.5.2.3.7	Activate Action Plan			O	Yes / No	
		3.5.2.1.10.5	Activate Action Plan Remotely	actionFunction:M	Yes / NA	
		3.5.2.3.2.1	Activate System Timing Pattern Remotely	patternFunction: M	Yes / NA	
2.5.2.3.8	Remote Manual Control			O	Yes / No	
		3.5.2.3.7.1	Enable Remote Manual Control	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.3.7.2	Advance Interval During Remote Manual Control	M	Yes	
		3.5.2.3.7.3	Configure Manual Control Timeout	M	Yes	
		3.5.2.3.7.4	Enable/Disable Automatic Pedestrian Clearance Setting	M	Yes	
2.5.3	Manage Detectors					
2.5.3.1 (Detector)	Manage Detector Configuration			M	Yes	
		3.5.3.1.1.1	Configure Vehicle Travel Mode	O	Yes / No	
		3.5.3.1.1.2	Configure Vehicle Detector Description	O	Yes / No	
		3.5.3.1.1.3	Configure Vehicle Detector Yellow Lock Call Enabled	O	Yes / No	
		3.5.3.1.1.4	Configure Vehicle Detector Red Lock Call Enabled	O	Yes / No	
		3.5.3.1.1.5	Configure Vehicle Detector Passage Enabled	O	Yes / No	
		3.5.3.1.1.6	Configure Vehicle Detector Added Initial Time Enabled	O	Yes / No	
		3.5.3.1.1.7	Configure Vehicle Detector Queue Enabled	O	Yes / No	
		3.5.3.1.1.8	Configure Vehicle Detector Call Enabled	M	Yes	
		3.5.3.1.1.9	Configure Vehicle Detector Call Phase	M	Yes	
		3.5.3.1.1.10	Configure Vehicle Detector Switch Phase	M	Yes	
		3.5.3.1.1.11	Configure Vehicle Detector Delay Time	M	Yes	
		3.5.3.1.1.12	Configure Vehicle Detector Extend Time	M	Yes	
		3.5.3.1.1.13	Configure Vehicle Detector Queue Limit Time	O	Yes / No	
		3.5.3.1.1.14	Configure Vehicle Detector No Activity Fault Time	M	Yes	
		3.5.3.1.1.15	Configure Vehicle Detector Maximum Presence Fault Time	M	Yes	
		3.5.3.1.1.16	Configure Vehicle Detector Erratic Counts	M	Yes	
		3.5.3.1.1.17	Configure Vehicle Detector Fail Time	O	Yes / No	
		3.5.3.1.2 (VehDetectSet)	Configure Multiple Vehicle Detector Sets for Actuation	O	Yes / No	
		3.5.3.1.3.1	Configure Pedestrian Detector Description	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.3.1.3.2	Configure Pedestrian Detector Call Phase	M	Yes	
		3.5.3.1.3.3	Configure Pedestrian Detector No Activity Fault Time	M	Yes	
		3.5.3.1.3.4	Configure Pedestrian Detector Maximum Presence Fault Time	M	Yes	
		3.5.3.1.3.5	Configure Pedestrian Detector Erratic Counts	M	Yes	
		3.5.3.1.3.6	Configure Pedestrian Detector Non-Lock Calls	O	Yes / No	
		3.5.3.1.3.7	Configure Pedestrian Detector for Presence Detection	O	Yes / No	
		3.5.3.1.3.8	Configure Pedestrian Detector for Delayed Walk	O	Yes / No	
		3.5.3.1.3.9	Configure Pedestrian Detector for Advanced Walk	O	Yes / No	
		3.5.3.1.4 (PedDetectSet)	Configure Multiple Pedestrian Detector Sets for Actuations	O	Yes / No	
		3.5.3.1.5.1	Determine Maximum Number of Vehicle Detectors	M	Yes	The ASC shall support at least ____ vehicle detectors (between 1 and 255).
		3.5.3.1.5.2	Determine Maximum Number of Vehicle Detector Sets	VehDetectSet:M	Yes / NA	The ASC shall support at least ____ vehicle detector sets.
		3.5.3.1.5.3	Determine Maximum Number of Pedestrian Detectors	M	Yes	The ASC shall support at least ____ pedestrian detectors (between 1 and 255).
		3.5.3.1.5.4	Determine Maximum Number of Pedestrian Detector Sets	PedDetectSet:M	Yes / NA	The ASC shall support at least ____ pedestrian detector sets.
2.5.3.2	Monitor Detector Status			O	Yes / No	
		3.5.3.2.1	Monitor Active Vehicle Detector Actuations	M	Yes	
		3.5.3.2.2	Monitor Active Pedestrian Detector Actuations	M	Yes	
2.5.3.3	Monitor Detector Health			O	Yes / No	
		3.5.3.3.1.1	Monitor Vehicle Detector Alarm Status	M	Yes	
		3.5.3.3.1.2	Monitor Vehicle Detector Faults from Controller	M	Yes	
		3.5.3.3.1.3	Monitor Vehicle Detector Faults from Detector	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.3.3.3.1	Monitor Pedestrian Detector Alarm Status	M	Yes	
		3.5.3.3.3.2	Monitor Pedestrian Detector Faults	M	Yes	
2.5.3.4	Control Detectors			O	Yes / No	
		3.5.3.4.1	Control Vehicle Detector Reset	M	Yes	
		3.5.3.4.2	Control Pedestrian Detector Reset	M	Yes	
		3.5.3.4.3	Control Detector Diagnostic Reset	O	Yes / No	
		3.5.3.4.4	Control Vehicle Detector Actuation	O	Yes / No	
		3.5.3.4.5	Control Pedestrian Detector Actuation	O	Yes / No	
2.5.3.5	Manage Detector Data Collection			O	Yes / No	
		3.5.3.5.3.1	Configure Vehicle Detector Data Sample Period	M	Yes	
		3.5.3.5.3.2	Configure Pedestrian Detector Sample Period	M	Yes	
		3.5.3.5.3.3 (Speed)	Configure Vehicle Speed Detectors	O	Yes / No	
		3.5.3.5.3.4	Configure Single Detector Speed Mode	Speed:M	Yes / NA	
		3.5.3.5.3.5	Configure Paired Detector	Speed:M	Yes / NA	
		3.5.3.5.3.6	Configure Paired Detector Placement	Speed:M	Yes / NA	
		3.5.3.5.3.7	Configure Paired Detector Spacing	Speed:M	Yes / NA	
		3.5.3.5.3.8	Configure Average Vehicle Length	O	Yes / No	
		3.5.3.5.3.9	Configure Vehicle Detection Zone Length	O	Yes / No	
		3.3.3.5.4	Configure Multiple Vehicle Detector Sets for Data Collection	O	Yes / No	
2.5.3.6	Monitor Detector Data from Controller			O	Yes / No	
		3.5.3.5.1.1	Monitor Vehicle Detector Data Sequence	M	Yes	
		3.5.3.5.1.2	Monitor Vehicle Volume Data	O	Yes / No	
		3.5.3.5.1.3	Monitor Vehicle Occupancy Data	O	Yes / No	
		3.5.3.5.1.4	Monitor Vehicle Average Speed	Speed:M	Yes / NA	
		3.5.3.5.1.5	Monitor Vehicle Detector Data Sample Time	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.3.5.2.1	Monitor Pedestrian Detector Data Sequence	O	Yes / No	
		3.5.3.5.2.2	Monitor Pedestrian Counts	O	Yes / No	
		3.5.3.5.2.3	Monitor Pedestrian Actuations	O	Yes / No	
		3.5.3.5.2.4	Monitor Pedestrian Services	O	Yes / No	
2.5.4 (CV)	Manage Connected Vehicles Interface			O	Yes / No	
2.5.4.1	Connected Vehicle Interface: Management Station – ASC Interface			M	Yes / No	
2.5.4.1.1	Manage ASC - RSU Interface			M	Yes	
		3.5.4.1.1	Configure ASC Communications Port for RSU	M	Yes	
		3.5.4.1.2	Configure Logical RSU Ports and Address	M	Yes	
		3.5.4.1.3	Configure RSU Interface Polling Period	O	Yes / No	
2.5.4.1.2	Manage ASC - RSU Interface Watchdog			O	Yes / No	
		3.5.4.1.4	Configure RSU Interface Watchdog	M	Yes	
		3.5.4.1.5	Monitor RSU Interface Watchdog Timer	M	Yes	
		3.5.4.1.6	Monitor RSU Interface Watchdog Alarm	M	Yes	
2.5.4.1.3	Manage Signal Phase and Timing Data			M	Yes	
		3.5.4.2.1.1	Enable Signal Phase and Timing Data	M	Yes	
		3.5.4.2.1.2	Retrieve Signal Phase and Timing Generation Time	O	Yes / No	
		3.5.4.2.1.3.1.1	Monitor Movement Minimum End Time	M	Yes	
		3.5.4.2.1.3.1.2	Monitor Movement Maximum End Time	M	Yes	
		3.5.4.2.1.3.1.3	Monitor Movement Likely End Time	O	Yes / No	
		3.5.4.2.1.3.1.4	Monitor Movement Likely End Time Confidence	O	Yes / No	
		3.5.4.2.1.3.1.5	Monitor Movement Next Occurrence	M	Yes	
		3.5.4.2.1.3.1.6	Monitor Movement Start Time	M	Yes	
		3.5.4.2.1.3.1.7	Monitor Next Movement Minimum End Time	M	Yes	
		3.5.4.2.1.3.1.8	Monitor Next Movement Maximum End Time	M	Yes	
		3.5.4.2.1.3.1.9	Monitor Next Movement Start Time	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.4.2.1.3.1.10	Determine Maximum Number of Movement Events			
		3.5.4.2.1.3.2.1	Configure Queue Detectors for Movement Assistance	MvtQueue:M	Yes / NA	
		3.5.4.2.1.3.2.2	Configure Pedestrian Detectors for Movement Conflict Assistance	MvtConflict:O.4 (1..*)	Yes / No / NA	
		3.5.4.2.1.3.2.3	Configure Bicycle Detectors for Movement Conflict Assistance	MvtConflict:O.4 (1..*)	Yes / No / NA	
		3.5.4.2.1.3.3.1 (MvtQueue)	Monitor Lane Connection Queue Length	O	Yes / No	
		3.5.4.2.1.3.3.2 (MvtConflict)	Monitor Lane Connection Vulnerable Road User Detection	O	Yes / No	
		3.5.4.2.1.3.4.1 (SpdAdvice)	Configure Advisory Speed Type	O	Yes / No	
		3.5.4.2.1.3.4.2	Configure Advisory Speed	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.4.3	Configure Advisory Speed Zone	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.4.4	Configure Advisory Speed Vehicle Type	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.5	Monitor Movement State	M	Yes	
		3.5.4.2.1.3.6	Monitor Next Movement State	M	Yes	
		3.5.4.2.1.3.7	Monitor Movement Status	M	Yes	
		3.5.4.2.1.4.1	Configure Concurrent Enabled Lanes	M	Yes	
		3.5.4.2.1.4.2	Configure Enabled Lanes by Time of Day	M	Yes	
		3.5.4.2.1.4.3	Determine Lanes Enabled	M	Yes	
		3.5.4.2.1.4.4	Command Enabled Lanes	M	Yes	
		3.5.4.2.1.5	Enable Signal Phase and Timing Exchange	M	Yes	
		3.5.4.2.1.6	Configure Road Authority Identifier	M	Yes	
		3.5.4.2.1.7.1	Monitor Manual Control Indication	M	Yes	
		3.5.4.2.1.7.2	Monitor Stop Indication	M	Yes	
		3.5.4.2.1.7.3	Monitor Failure Flash Indication	M	Yes	
		3.5.4.2.1.7.4	Monitor Preemption Operation Indication	M	Yes	
		3.5.4.2.1.7.5	Monitor Priority Operation Indication	M	Yes	
		3.5.4.2.1.7.6	Monitor Fixed Time Control Indication	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.4.2.1.7.7	Monitor Non-Fixed Time Control Indication	M	Yes	
		3.5.4.2.1.7.8	Monitor Standby Operation Indication	M	Yes	
		3.5.4.2.1.7.9	Monitor Controller Failure	M	Yes	
		3.5.4.2.1.7.10	Monitor MAP Message Validity	M	Yes	
		3.5.4.2.1.7.11	Monitor SPaT Message Validity	M	Yes	
		3.5.4.2.1.8	Mark SPaT Invalid - Controller	M	Yes	
		3.5.4.2.1.9	Mark SPaT Invalid - Port	O	Yes / No	
		3.5.4.2.1.10	Mark MAP Message Invalid - Controller	M	Yes	
		3.5.4.2.1.11	Mark MAP Message Invalid - Port	O	Yes / No	
		3.5.4.2.1.12.1	Determine Maximum Number of Signal Groups	M	Yes	
		3.5.4.2.1.12.2	Configure Signal Groups Intersection Mapping	M	Yes	
		3.5.4.2.1.12.3	Configure Signal Group Control Source	M	Yes	
		3.5.4.2.1.12.4	Configure Signal Group Indication Types	M	Yes	
		3.5.4.2.1.12.5	Configure Signal Group Protected or Permissive State	M	Yes	
		3.5.4.2.1.12.6	Configure Signal Group Revocable Lanes	M	Yes	
		3.5.4.2.1.12.7	Determine Maximum Number of Signal State Entries	M	Yes	
		3.5.4.2.1.12.8	Configure Customized Signal State Parameters	M	Yes	
		3.5.4.2.1.13	Retrieve Signal Phase and Timing Time Point	M	Yes	
2.5.4.1.4	Manage Assured Green Period			O	Yes / No	
		3.5.4.2.2.1	Enabled Connected Device Connection	M	Yes	
		3.5.4.2.2.2	Configure Vehicle Detector for Connected Vehicle Application	M	Yes	
		3.5.4.2.2.3	Configure Connected Vehicle Detector Assigned Input	M	Yes	
		3.5.4.2.2.4	Configure Connected Vehicle Detector Port Assignment	O	Yes / No	
		3.5.4.2.2.5	Configure Assured Green Period Duration	M	Yes	
		3.5.4.2.2.6	Configure Red Light Violation Warning Parameters	O	Yes / No	
2.5.4.2	Connected Vehicle Interface: ASC – CV Application Process Interface			M	Yes	
2.5.4.2.1	Exchange Current and Next Movement Information			M	Yes	
		3.5.4.2.1.1.1	Provide Movement Time Point	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.4.2.1.1.2	Provide Movement State	M	Yes	
		3.5.4.2.1.1.3	Provide Movement Minimum End Time	M	Yes	
		3.5.4.2.1.1.4	Provide Movement Maximum End Time	M	Yes	
		3.5.4.2.1.1.5	Provide Movement Likely End Time	O	Yes / No	
		3.5.4.2.1.1.6	Provide Movement Likely End Time Confidence	O	Yes / No	
		3.5.4.2.1.1.7	Provide Next Movement State	M	Yes	
		3.5.4.2.1.1.8	Provide Next Movement Minimum End Time	M	Yes	
		3.5.4.2.1.1.9	Provide Next Movement Maximum End Time	M	Yes	
		3.5.4.2.1.1.10	Provide Next Movement Start Time	M	Yes	
		3.5.4.2.1.2.1	Provide Lane Connection Queue Length	O	Yes / No	
		3.5.4.2.1.2.2	Provide Lane Connection Traveler Detection	O	Yes / No	
		3.5.4.2.1.3.1	Provide Advisory Speed Type	O	Yes / No	
		3.5.4.2.1.3.2	Provide Advisory Speed	O	Yes / No	
		3.5.4.2.1.3.3	Provide Advisory Speed Zone	O	Yes / No	
		3.5.4.2.1.3.4	Provide Advisory Speed Vehicle Type	O	Yes / No	
		3.5.4.2.1.4	Provide Road Authority ID	M	Yes	
		3.5.4.2.1.5	Provide Signal Phase and Timing Intersection Status	M	Yes	
		3.5.4.3.1.6	Provide Compressed SPaT Information to External CV Application Process	M	Yes	
		3.5.4.3.2.1	Provide UPER-encoded SPaT Message	M	Yes	
		3.6.5.1	SPaT Maximum Transmission Start Time	M	Yes	The Maximum Transmission Start Time for all SPAT data shall be ___ milliseconds (Default=10).
		3.6.5.2	Movement Time Point Minimum Transmission Rate	M	Yes	The Movement Time Point Minimum Transmission Rate shall be once per _____ milliseconds (Default=100).
		3.6.5.3	SPaT Maximum Transmission Rate	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.6.5.4	SPaT Time Accuracy	M	Yes	
2.5.4.2.2	Exchange Next Occurrence of a Movement			M	Yes	
		3.5.4.3.1.1.11	Provide Movement Next Occurrence	M	Yes	
		3.5.4.3.4	Monitor CV Certificate Faults	O	Yes / No	
		3.6.5.1	SPaT Maximum Transmission Start Time	M	Yes	The Maximum Transmission Start Time for all SPAT data that changed shall be ___ milliseconds (Default=10).
		3.6.5.2	Movement Time Point Minimum Transmission Rate	M	Yes	The Movement Time Point Minimum Transmission Rate shall be once per _____ milliseconds (Default=100).
		3.6.5.3	SPaT Maximum Transmission Rate	M	Yes	
		3.6.5.4	SPaT Time Accuracy	M	Yes	
2.5.4.2.3	Exchange Presence of Connected Devices			O	Yes / No	
		3.5.4.3.2.2	Retrieve BSMs	O.12(1..*)	Yes / No	
		3.5.4.3.2.3	Retrieve PSMs	O	Yes / No	
		3.5.4.3.2.4	Retrieve Actuation Report	O.12(1..*)	Yes / No	
		3.5.4.3.2.5	Retrieve Detection Report	O	Yes / No	
2.5.4.2.4	Exchange Roadway Geometrics Information			O	Yes / No	
		3.5.4.3.3.1	Retrieve MAP Plan in Effect	M	Yes	
		3.5.4.3.3.2	Confirm MAP Plan Compatibility	M	Yes	
2.5.4.3	ASC – ECLA Interface			O	Yes / No	
		3.5.4.4.1	Receive Current Phase Minimum End Time from an ECLA	M	Yes	
		3.5.4.4.2	Receive Current Phase Maximum End Time from an ECLA	M	Yes	
		3.5.4.4.3	Receive Current Phase Likely End Time from an ECLA	O	Yes / No	
		3.5.4.4.4	Receive Current Phase Likely End Time Confidence from an ECLA	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.4.4.5	Receive Next Phase from an ECLA	M	Yes	
		3.5.4.4.6	Receive Compressed ECLA Input Data	M	Yes	
2.5.5	Backward Compatibility Features					
2.6	Security			M	Yes	
2.6.1	Manage Authentication			M	Yes	
2.6.2	Manage Accessibility			M	Yes	
2.6.3	Manage Users			M	Yes	
		ISO 26048-1§ 8.22	View-based access control model (VACM) feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.6.4	Log User Access			O	Yes / No	
		ISO 26048-1§ 8.1.3.1	Validate access upon action activation	M	Yes	
		ISO 26048-1§ 8.1.3.2	Validate access upon action being called	M	Yes	
		3.6.1	Response Time for Requests	M	Yes	The Response Time for all requests shall be ___ milliseconds (5-500: Default=25).
2.6.5	Manage ASC Interface Security			O	Yes / No	
2.6.5.1	Manage Security for the ASC to RSU Interface					
2.6.5.2	Manage Security for Other ASC Interfaces					

) for the interface between the management station and the ASC Process. As noted in Section 3.3.2.1, a conformant device may offer additional (optional) features, as long as they are conformant with the requirements of NTCIP 1202 v04 and the standards it references (e.g., ISO 26048-1).

Support of the interface between the ASC Process and CV Application Process is also required to claim conformance with NTCIP 1202 v04. The ASC shall provide for that interface by fulfilling all mandatory requirements identified in the PRL Table (

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.3	Reference Physical Architecture [Informative]					
2.3.1	ASC Characteristics – Cabinet Specifications [Normative]			M	Yes	
2.3.1.a (332)	Model 332 Cabinet			O.1 (1)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.3.1.b (TS1)	NEMA TS 1 Cabinet			O.1 (1)	Yes / No	
2.3.1.c (TS2-2)	NEMA TS 2 Type 2 Cabinet			O.1 (1)	Yes / No	
2.3.1.d (TS2-1)	NEMA TS 2 Type 1 Cabinet			O.1 (1)	Yes / No	
2.3.1.e (ATC)	ATC Cabinet			O.1 (1)	Yes / No	
2.3.2	ASC Characteristics – Controller Types [Normative]			M	Yes	
2.3.2.a	Phase-based controller			M	Yes	
2.3.2.b	Interval-based controller			NA	NA	Interval-based controllers are not supported by NTCIP 1202 v04
2.3.2.c	Stage-based controller			NA	NA	Stage-based controllers are not supported by NTCIP 1202 v04
2.4	Architectural Needs					
2.4.1	Provide Live Data			M	Yes	
		3.4.1.1	Retrieve Data	M	Yes	
		3.4.1.2	Deliver Data	M	Yes	
		3.4.1.3	Explore Data	M	Yes	
		3.4.1.4.1	Monitor SNMP Information	M	Yes	
		ISO 26048-1§ 8.5.2.6	Determine basic capabilities of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.7	Determine SNMP capabilities of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.8	Determine the SNMP engine identifier	M	Yes	
		ISO 26048-1§ 8.5.2.9	Monitor when the SNMP capabilities last changed	M	Yes	
		ISO 26048-1§ 8.5.4	Controller performance requirements	M	Yes	
		3.6.1	Response Time for Requests	M	Yes	The Response Time for all requests shall be ___ milliseconds (5-500: Default=25). Note: Redundant with the previous requirement.
		~~3.6.2~~	Atomic Operations	M	Yes	Note: should 3.6.2 be deleted?
2.4.2	Provide Data Blocks			O	Yes / No	
		3.4.2.1	Store Pre-defined Compressed Data Blocks	M	Yes	
		3.5.2.1.14.1.1	Configure Block Object Get Control - Phase Data	O.2 (1..*)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.14.1.2	Configure Block Object Get Control - Vehicle Detector Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.3	Configure Block Object Get Control - Pedestrian Detector Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.4	Configure Block Object Get Control - Pattern Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.5	Configure Block Object Get Control - Split Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.6	Configure Block Object Get Control - Overlap Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.7	Configure Block Object Get Control - Preempt Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.8	Configure Block Object Get Control - Sequence Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.1.9	Configure Block Object Get Control - Channel Data	O.2 (1..*)	Yes / No	
		3.5.2.1.14.2	Monitor Block Error Status - Error-causing Data Element	M	Yes	
		ISO 26048-1§ 8.7	Dynamic object feature	O	Yes / No	All requirements that trace to this feature are also inherited. See the Features To Requirements Traceability (FTRT) in ISO 26048-1.
2.4.3	Provide for Log Data Local Storage and Retrieval			O.3 (1..*)	Yes / No	Note: Either Logging or Exception Reporting is Mandatory in ISO 26048-1.
		ISO 26048-1§ 8.9	File Feature	M	Yes	
		ISO 26048-1§ 8.10	Logging feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.4.4	Provide for Database Management			M	Yes	
		ISO 26048-1§ 8.21	Transaction feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.4.5	Condition-based Exception Reporting			O.3 (1..*)	Yes / No	Note: Either Logging or Exception Reporting

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						is Mandatory in ISO 26048-1.
		ISO 26048-1§ 8.1	Action feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.11	Notification feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.16	SNMP target feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5	Features					
2.5.1	Manage the ASC Configuration			M	Yes	
2.5.1.1	Manage Device Identity			M	Yes	
		3.5.1.1.1	Configure ASC Location - Antenna Offset	C	Yes / NA	Mandatory if an external GNSS device is attached to the ASC.
		ISO 26048-1§ 8.5.2.1	Configure the controller's identity	M	Yes	
		ISO 26048-1§ 8.5.2.2	Configure the default language	M	Yes	
		ISO 26048-1§ 8.5.2.5	Control remote reset of the controller	M	Yes	
		ISO 26048-1§ 8.5.2.14	Obtain the field device configuration identifier	M	Yes	
		ISO 26048-1§ 8.5.3.1	Support maximum message size	M	Yes	The maximum SNMP message size shall be _____ bytes (Default: 484 bytes).
		ISO 26048-1§ 8.5.3.2	Support total memory	O	Yes / No	The controller shall support a minimum of _____ kilobytes of total memory.
		ISO 26048-1§ 8.5.3.3	Support changeable memory	O	Yes / No	The controller shall support a minimum of _____ kilobytes of changeable memory.

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.8.2.1.1	Configure the field device's location by providing coordinates	M	Yes	
		ISO 26048-1§ 8.8.2.1.2	Configure the field device's location by copying coordinates	O	Yes / No	
		ISO 26048-1§ 8.8.2.1.3	Configure the field device's location by GNSS reading	O	Yes / No	
		ISO 26048-1§ 8.8.2.1.4	Configure the field device's physical components	M	Yes	
		ISO 26048-1§ 8.12	Owner feature	M	Yes	
2.5.1.2	Manage Communications			O	Yes / No	Propose to delete.
		3.5.1.2.1.1	Enable/Disable Communications Port	M	Yes	The ASC shall not be allowed to enable/disable the following ports numbers: ________
		3.5.1.2.2.1	Determine Number of ASC Communications Ports	M	Yes	
2.5.1.3	Manage Cabinet Environment			O	Yes / No	
		ISO 26048-1§ 8.8.2.1.4	Configure the cabinet's physical components	M	Yes	
		ISO 26048-1§ 8.8.12	Field device doors	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of doors to be supported (Default: 2) ___
		ISO 26048-1§ 8.8.13	Field device fans	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of fans supported (Default: 2) _____
		ISO 26048-1§ 8.8.15	Field device heaters	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						heaters supported (Default: 1) ____
		ISO 26048-1§ 8.8.16	Field device humidity	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of humidity sensors supported (Default: 1) ____
		ISO 26048-1§ 8.8.21	Field device temperature	O	Yes / No	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of temperature sensors supported (Default: 1) ____
2.5.1.4 (Power)	Monitor Power			O	Yes / No	
		ISO 26048-1§ 8.8.2.1.9	Monitor current power source	M	Yes	
		ISO 26048-1§ 8.8.2.2.1.a	Support power sources – mainline (alternating current) power	M	Yes	
		ISO 26048-1§ 8.8.2.2.1.b	Support power sources – battery power	UPS:M	Yes / NA	
		ISO 26048-1§ 8.8.2.2.1.c (Generator)	Support power sources – generator power	O	Yes / No	
		ISO 26048-1§ 8.2.2.2.1.d (Solar)	Support power sources – solar power	O	Yes / No	
		ISO 26048-1§ 8.8.2.2.2 (UPS)	Support UPS power	O	Yes / No	
		ISO 26048-1§ 8.8.10	Field device battery	UPS:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet battery supported (Default: 1) ____

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.8.14	Field device generator	Generator:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet generator supported (Default: 1) _____
		ISO 26048-1§ 8.8.17	Field device mains power	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet mains power supported (Default: 1) _____
		ISO 26048-1§ 8.8.20	Field device solar power	Solar:M	Yes / NA	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Minimum number of voltage sensors for the cabinet solar power system supported (Default: 1) _____
2.5.1.5	Retrieve Operational Performance Data			O	Yes / No	
		ISO 26048-1§ 8.13	Recording feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.1.6	Manage Auxiliary External Inputs/Outputs			O	Yes / No	
		ISO 26048-1§ 8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1. Number of additional SRSA

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
						ports to support: ____. This value is exclusive of ports used to monitor cabinet power, cabinet doors, fans, heaters, temperature sensor and humidity sensors.
2.5.1.7	Manage Database			M	Yes	
		3.5.1.7.1	Determine Configuration Identifier Parameter Content	M	Yes	
		ISO 26048-1§ 8.5.2.14	Obtain the field device configuration identifier	M	Yes	
2.5.1.8	Manage Interface with External Detectors			O	Yes / No	See 2.5.3.5 for detector data collected by the ASC through external inputs. Requirements to view advanced detector data are found in NTCIP 1209.
2.5.1.9	Manage ASC Clock			M	Yes	
		ISO 26048-1§ 8.2.1	UTC clock	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.2.2	Local clock	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		ISO 26048-1§ 8.2.3	Daylight saving time	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.1.10	Manage External Control Local Application State			O	Yes / No	
		3.5.1.10.1.1	Enable ECLA Communications	M	Yes	
		3.5.1.10.1.2	Disable ECLA Communications	M	Yes	
		3.5.1.10.2	Monitor ECLA Data Input Time	M	Yes	
2.5.2	Manage Signal Operations			M	Yes	
2.5.2.1	Manage Signal Configuration			M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.2.1.1	Manage Controller Startup Functions			M	Yes	
		3.5.2.1.1.1.1	Configure Start-Up Flash Mode	O	Yes / No	
		3.5.2.1.1.1.2	Configure Start-Up Flash Time	M	Yes	
		3.5.2.1.1.2	Configure Backup Time	M	Yes	
2.5.2.1.2	Manage Phase Configurations			M	Yes	
		3.5.2.1.2.1.1	Enable/Disable Phase	M	Yes	
		3.5.2.1.2.1.2	Configure Phase Minimum Green Time	M	Yes	
		3.5.2.1.2.1.3	Configure Phase Passage Time	M	Yes	
		3.5.2.1.2.1.4	Configure Two Fixed Phase Maximum Green Times	M	Yes	
		3.5.2.1.2.1.5	Configure Three Fixed Phase Maximum Green Times	O	Yes / No	
		3.5.2.1.2.1.6	Configure Phase Yellow Change Time	M	Yes	
		3.5.2.1.2.1.7	Configure Phase Red Clearance Time	M	Yes	
		3.5.2.1.2.1.8	Configure Phase Red Revert Time	O	Yes / No	
		3.5.2.1.2.1.9	Configure Unit Red Revert Time	Unit:M	Yes / NA	
		3.5.2.1.2.1.10	Configure Phase Added Initial Time	M	Yes	
		3.5.2.1.2.1.11	Configure Phase Maximum Initial Time	M	Yes	
		3.5.2.1.2.1.12	Configure Phase Time Before Reduction	M	Yes	
		3.5.2.1.2.1.13	Configure Phase Time to Reduce	M	Yes	
		3.5.2.1.2.1.14	Configure Phase Cars Before Reduction	O	Yes / No	
		3.5.2.1.2.1.15	Configure Phase Reduce By Time	O	Yes / No	
		3.5.2.1.2.1.16	Configure Phase Minimum Gap Time	M	Yes	
		3.5.2.1.2.1.17	Configure Phase Dynamic Maximum Limit	O	Yes / No	
		3.5.2.1.2.1.18	Configure Phase Dynamic Maximum Step	O	Yes / No	
		3.5.2.1.2.1.19	Configure Phase Start-Up State	M	Yes	
		3.5.2.1.2.1.20	Configure Automatic Flash Entry Phase	O	Yes / No	
		3.5.2.1.2.1.21	Configure Automatic Flash Exit Phase	O	Yes / No	
		3.5.2.1.2.1.22	Configure Call to Non-Actuated 1	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.2.1.23	Configure Call to Non-Actuated 2	O	Yes / No	
		3.5.2.1.2.1.24	Configure Non-Lock Detector Memory	O	Yes / No	
		3.5.2.1.2.1.25	Configure Phase Minimum Vehicle Recall	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.26	Configure Phase Maximum Vehicle Recall	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.27	Configure Phase Soft Vehicle Recall	O	Yes / No	
		3.5.2.1.2.1.28	Configure Dual Phase Entry	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.29	Configure Simultaneous Gap Disable	O	Yes / No	
		3.5.2.1.2.1.30	Configure Guaranteed Passage	O	Yes / No	
		3.5.2.1.2.1.31	Configure Actuated Rest-in-Walk	M	Yes	Should this be mandatory?
		3.5.2.1.2.1.32	Configure Conditional Service Enable	O	Yes / No	
		3.5.2.1.2.1.33	Configure Added Initial Calculation	O	Yes / No	
		3.5.2.1.2.1.34	Configure Phase-to-Ring Association	M	Yes	
		3.5.2.1.2.1.35	Configure Phase Concurrency	M	Yes	
		3.5.2.1.2.1.36	Configure Pedestrian Clearance Time Allowed During Vehicle Clearance	O	Yes / No	
		3.5.2.1.2.1.37	Configure Pedestrian Walk Time	M	Yes	
		3.5.2.1.2.1.38	Configure Pedestrian Clearance Time	M	Yes	
		3.5.2.1.2.1.39	Configure Pedestrian Phase Walk Service Limit	M	Yes	
		3.5.2.1.2.1.40	Configure Pedestrian Phase Don't Walk Revert Time	M	Yes	
		3.5.2.1.2.1.41	Configure Non-Lock Ped Detector Memory	M	Yes	
		3.5.2.1.2.1.42	Configure Pedestrian Phase Recall	M	Yes	
		3.5.2.1.2.1.43	Configure Phase Alternate Pedestrian Clearance Time	O	Yes / No	
		3.5.2.1.2.1.44	Configure Phase Alternate Pedestrian Walk Time	O	Yes / No	
		3.5.2.1.2.1.45	Configure Pedestrian Phase Advanced Walk Time	O	Yes / No	
		3.5.2.1.2.1.46	Configure Pedestrian Phase Delayed Walk Time	O	Yes / No	
		3.5.2.1.2.1.47	Configure Phase Advance Warning Green	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.2.1.48	Configure Phase Advance Warning Red	O	Yes / No	
		3.5.2.1.2.1.49	Configure Flashing Yellow Arrow Associated Vehicle Phase	O	Yes / No	
		3.5.2.1.2.1.50	Configure Flashing Red Arrow Associated Vehicle Phase	O	Yes / No	
		3.5.2.1.2.1.51	Configure Alternate Minimum Green Time during Transition	O	Yes / No	
		3.5.2.1.2.1.52	Configure Alternate Minimum Walk Time during Transition	O	Yes / No	
		3.5.2.1.2.1.53	Configure Alternate Minimum Pedestrian Clearance Time during Transition	O	Yes / No	
		3.5.2.1.2.2(PhaseSet)	Configure Multiple Phase Sets	O	Yes / No	
		3.5.2.1.2.3.1	Determine Maximum Number of Phases	M	Yes	The ASC shall support at least _____ phases.
		3.5.2.1.2.3.2	Determine Maximum Number of Phase Sets	PhaseSet:M	Yes / NA	The ASC shall support at least _____ phase sets.
2.5.2.1.3 (Coord)	Manage Coordination Configurations			O	Yes / No	
		3.5.2.1.3.1	Configure Operational Mode for Coordination	M	Yes	The ASC shall support the following values: ___ automatic ___ pattern ___ manual free ___ manual flash
		3.5.2.1.3.2	Configure Correction Mode for Coordination	M	Yes	The ASC shall support the following values: __ dwell ___ shortway __ add only ___ subtract only
		3.5.2.1.3.3	Configure Maximum Mode for Coordination	M	Yes	The ASC shall support the following values: ___ maxInhibit ___maximum1 ___ maximum2 ___maximum3
		3.5.2.1.3.4	Configure Unit-level Force Mode for Coordination	O	Yes / No	
		3.5.2.1.3.5	Configure Phase-level Force Mode for Coordination	O	Yes / No	
		3.5.2.1.3.6	Configure Pattern Reference Phase	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.3.7	Configure Pattern Reference Point	M	Yes	
		3.5.2.1.3.9	Configure Pattern Synchronization Time	M	Yes	
2.5.2.1.4	Manage Timing Patterns			Coord:M	Yes / NA	
		3.5.2.1.4.1.1	Configure Pattern to Run Free	M	Yes	
		3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	O	Yes / No	
		3.5.2.1.4.1.3	Configure Pattern Cycle Time for Coordination	M	Yes	
		3.5.2.1.4.1.4	Configure Pattern Offset Time	M	Yes	
		3.5.2.1.4.1.5	Configure Pattern Split Association	M	Yes	
		3.5.2.1.4.1.6	Configure Pattern Sequence Association	M	Yes	
		3.5.2.1.4.1.7	Configure Pattern Maximum Mode	O	Yes / No	
		3.5.2.1.4.1.8	Configure Pattern Phase Set	O	Yes / No	
		3.5.2.1.4.1.9	Configure Pattern Overlap Set	O	Yes / No	
		3.5.2.1.4.1.10	Configure Pattern Vehicle Detector Set	O	Yes / No	
		3.5.2.1.4.1.11	Configure Pattern Pedestrian Detector Set	O	Yes / No	
		3.5.2.1.4.1.12	Configure Pattern Special Functions	O	Yes / No	
		3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	M	Yes	The ASC shall support at least ____ timing patterns.
2.5.2.1.5	Manage Splits Configurations			O	Yes / No	
		3.5.2.1.3.8	Configure Omit Phases During Transitions	O	Yes / No	
		3.5.2.1.5.1.1	Configure Phase Split Time	M	Yes	
		3.5.2.1.5.1.2	Configure Phase Split Mode	M	Yes	
		3.5.2.1.5.1.3	Configure Split Coordination Phase	M	Yes	
		3.5.2.1.5.2	Determine Maximum Number of Phase Splits	M	Yes	The ASC shall support at least ____ splits
2.5.2.1.6 (Ring)	Manage Ring Configurations			O	Yes / No	
		3.5.2.1.6.1	Configure Sequence Data	M	Yes	
		3.5.2.1.6.2	Determine Maximum Number of Rings	M	Yes	The ASC shall support at least ____ rings

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.6.3	Determine Maximum Number of Sequences	M	Yes	The ASC shall support at least ____ sequences
2.5.2.1.7 (Channel)	Manage Channel Configurations			O	Yes / No	
		3.5.2.1.7.1.1	Configure Channel Control Source	M	Yes	
		3.5.2.1.7.1.2	Configure Channel Control Type	M	Yes	
		3.5.2.1.7.1.3.1	Configure Channel Flash Yellow	M	Yes	
		3.5.2.1.7.1.3.2	Configure Channel Flash Red	M	Yes	
		3.5.2.1.7.1.3.3	Configure Channel Flash Alternate Half Hertz	O	Yes / No	
		3.5.2.1.7.1.3.4	Configure Channel Flash Alternate First or Second	O	Yes / No	
		3.5.2.1.7.2	Determine Maximum Number of Channels	M	Yes	The ASC shall support at least ____ channels (See appropriate hardware specification such as NEMA TS 2 to determine maximum number of supported channels)
2.5.2.1.8 (Overlap)	Manage Overlap Configurations			O	Yes / No	
		3.5.2.1.8.1.1.1	Configure Overlap Type - Vehicle Normal	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.2	Configure Overlap Type - Vehicle Minus Green and Yellow	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.3	Configure Overlap Type - Pedestrian Normal	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.4	Configure Overlap Type - Flashing Yellow Arrow - 3 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.5	Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.6	Configure Overlap Type - Flashing Red Arrow - 3 Section Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.7	Configure Overlap Type - Flashing Red Arrow - 4 Section Head	O.4 (1..*)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.8.1.1.8	Configure Overlap Type - 2 Section Transit Specific Signal Head	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.1.9	Configure Overlap Type - Minus Green Yellow Alternate	O.4 (1..*)	Yes / No	
		3.5.2.1.8.1.2	Configure Overlap Included Phases	M	Yes	
		3.5.2.1.8.1.3	Configure Overlap Modifier Phases	M	Yes	
		3.5.2.1.8.1.4	Configure Pedestrian Modifier Phases	O	Yes / No	
		3.5.2.1.8.1.5	Configure Overlap Trailing Green	M	Yes	
		3.5.2.1.8.1.6	Configure Overlap Trailing Yellow	M	Yes	
		3.5.2.1.8.1.7	Configure Overlap Trailing Red Clearance	M	Yes	
		3.5.2.1.8.1.8	Configure Overlap Walk	O	Yes / No	
		3.5.2.1.8.1.9	Configure Overlap Pedestrian Clearance	O	Yes / No	
		3.5.2.1.8.2 (OverlapSet)	Configure Multiple Overlap Sets	O	Yes / No	
		3.5.2.1.8.3.1	Determine Maximum Number of Overlaps	M	Yes	The ASC shall support at least _____ overlaps
		3.5.2.1.8.3.2	Determine Maximum Number of Overlap Sets	Overlap Set:M	Yes / NA	The ASC shall support at least _____ overlap sets.
2.5.2.1.9 (Preempt)	Manage Preempt Configurations			O	Yes / No	
		3.5.2.1.9.1.1	Enable/Disable Preempt Inputs	M	Yes	
		3.5.2.1.9.1.2	Configure Preempt Control - Non-Locking Memory	O	Yes / No	
		3.5.2.1.9.1.3	Configure Preempt Control - Override Automatic Flash	O	Yes / No	
		3.5.2.1.9.1.4	Configure Preempt Control - Override Preempt	O	Yes / No	
		3.5.2.1.9.1.5	Configure Preempt Control - Flash Dwell	O	Yes / No	
		3.5.2.1.9.1.6	Configure Preempt Control - All Red Entry	O	Yes / No	
		3.5.2.1.9.1.7	Configure Preempt Link	M	Yes	
		3.5.2.1.9.1.8	Configure Preempt Delay	M	Yes	
		3.5.2.1.9.1.9	Configure Preempt Minimum Duration	M	Yes	
		3.5.2.1.9.1.10.1	Configure Preempt Enter Minimum Green Time	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.9.1.10.2	Configure Preempt Enter Minimum Walk Time	O	Yes / No	
		3.5.2.1.9.1.10.3	Configure Preempt Enter Pedestrian Clearance Time	O	Yes / No	
		3.5.2.1.9.1.10.4	Configure Preempt Enter Yellow Change Time	O	Yes / No	
		3.5.2.1.9.1.10.5	Configure Preempt Enter Red Clearance Time	O	Yes / No	
		3.5.2.1.9.1.11.1	Configure Preempt Track Clearance Green Time	M	Yes	
		3.5.2.1.9.1.11.2	Configure Preempt Track Clearance Yellow Change Time	O	Yes / No	
		3.5.2.1.9.1.11.3	Configure Preempt Track Red Clearance Time	O	Yes / No	
		3.5.2.1.9.1.11.4	Configure Preempt Track Clearance Phases	M	Yes	
		3.5.2.1.9.1.11.5	Configure Preempt Track Clearance Overlaps	O	Yes / No	
		3.5.2.1.9.1.12.1	Configure Preempt Minimum Green Dwell Time	M	Yes	
		3.5.2.1.9.1.12.2	Configure Preempt Dwell Phases	M	Yes	
		3.5.2.1.9.1.12.3	Configure Preempt Dwell Pedestrian Movements	O	Yes / No	
		3.5.2.1.9.1.12.4	Configure Preempt Dwell Overlaps	O	Yes / No	
		3.5.2.1.9.1.12.5	Configure Preempt Cycling Phases	O	Yes / No	
		3.5.2.1.9.1.12.6	Configure Preempt Cycling Pedestrian Movements	O	Yes / No	
		3.5.2.1.9.1.12.7	Configure Preempt Cycling Phases Sequence	O	Yes / No	
		3.5.2.1.9.1.12.8	Configure Preempt Cycling Overlaps	O	Yes	
		3.5.2.1.9.1.13.1	Configure Preempt Exit Phases	O	Yes / No	
		3.5.2.1.9.1.13.2	Configure Preempt Exit Phase Strategy	O	Yes / No	
		3.5.2.1.9.1.13.3	Configure Preempt Exit Priority Levels	C	Yes / No	Mandatory for Conformance if 'Exit to Queue Delay Recovery' is supported as a Preempt Exit Phase (See 3.5.2.1.9.1.13.2)
		3.5.2.1.9.1.14.1	Configure Preempt Maximum Presence Time	M	Yes	
		3.5.2.1.9.1.14.2	Configure Preempt Maximum Presence Action	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.9.1.15	Configure Preempt Gate Description	O	Yes / No	
		3.5.2.1.9.2	Determine Maximum Number of Preempts	M	Yes	The ASC shall support at least _____ preempts
2.5.2.1.10	Manage Timing Pattern Scheduler			M	Yes / No	
		ISO 26048-1§ 8.1.2.3	Configure an action owner	M	Yes	
		ISO 26048-1§ 8.1.2.4	Configure an action group	M	Yes	
		ISO 26048-1§ 8.1.2.5	Configure an action	M	Yes	
		ISO 26048-1§ 8.6	Day plan feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.5.2.1.11	Manage Action Scheduler			M	Yes / No	
		ISO 26048-1§ 8.1.2.3	Configure an action owner	M	Yes	
		ISO 26048-1§ 8.1.2.4	Configure an action group	M	Yes	
		ISO 26048-1§ 8.1.2.5	Configure an action	M	Yes	
		ISO 26048-1§ 8.6	Day plan feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
		3.5.2.1.4.1.1	Configure Pattern to Run Free	patternFunction: M	Yes / NA	
		3.5.2.1.4.1.2	Configure Pattern for Automatic Flash	patternFunction: M	Yes / NA	
		3.5.2.1.4.1.12(patternFunction)	Configure Pattern Special Functions	O.5 (1..*)	Yes / No	
		3.5.2.1.4.1.13	Determine Maximum Number of Timing Patterns	patternFunction: M	Yes	
		3.5.2.1.10.1.1	Configure Timebased Action - Pattern	actionFunction:M	Yes / NA	
		3.5.2.1.10.1.2(actionFunction)	Configure Timebased Action - Special Functions	O.5 (1..*)	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.10.1.3	Determine Maximum Number of Timebased Actions	actionFunction:M	Yes / NA	
2.5.2.1.12	Manage I/O Mapping			O	Yes / No	
		3.5.2.1.11.1.1	Set Active I/O Map	M	Yes	
		3.5.2.1.11.1.2.1	Configure I/O Map Description	M	Yes	
		3.5.2.1.11.1.2.2.1	Configure I/O Map Input Device	M	Yes	
		3.5.2.1.11.1.2.2.2	Configure I/O Map Input Device Pin	M	Yes	
		3.5.2.1.11.1.2.2.3	Configure I/O Map Input Function	M	Yes	
		3.5.2.1.11.1.2.3.1	Configure I/O Map Output Device	M	Yes	
		3.5.2.1.11.1.2.3.2	Configure I/O Map Output Device Pin	M	Yes	
		3.5.2.1.11.1.2.3.3	Configure I/O Map Output Function	M	Yes	
		3.5.2.1.11.2.1	Retrieve Maximum Number of I/O Maps	M	Yes	
		3.5.2.1.11.2.2	Retrieve Maximum Number of I/O Map Inputs	M	Yes	
		3.5.2.1.11.2.3	Retrieve Maximum Number of I/O Map Outputs	M	Yes	
		3.5.2.1.11.2.4	Retrieve I/O Mapping Activate Conditions	M	Yes	The following conditions shall be satisfied before a new I/O map can be activated: ___ Cabinet Door Open ___ in any flash state ___ programmed all red flash ___ in CVM flash ___ ASC restart
		3.5.2.1.11.2.5	Retrieve I/O Mapping Input Functions	M	Yes	
		3.5.2.1.11.2.6	Retrieve I/O Mapping Output Functions	M	Yes	
		3.5.2.1.11.2.7	Retrieve I/O Map Input Device Pin Status	M	Yes	
		3.5.2.1.11.2.8	Retrieve I/O Map Output Device Pin Status	M	Yes	
		3.5.2.1.11.2.9.1	Enumerate I/O Map - FIO Inputs	332:M	Yes / NA	
		3.5.2.1.11.2.9.2	Enumerate I/O Map - FIO Outputs	332:M	Yes / NA	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.1.11.2.9.3	Enumerate I/O Map - TS1 Inputs	TS1, TS2-2:M	Yes / NA	
		3.5.2.1.11.2.9.4	Enumerate I/O Map - TS1 Outputs	TS1, TS2-2:M	Yes / NA	
		3.5.2.1.11.2.9.5	Enumerate I/O Map - TS2 BIU Inputs	TS2-1:M	Yes / NA	
		3.5.2.1.11.2.9.6	Enumerate I/O Map - TS2 BIU Outputs	TS2-1:M	Yes / NA	
		3.5.2.1.11.2.9.7	Enumerate I/O Map - ATC Cabinet SIU Inputs	ATC:M	Yes / NA	
		3.5.2.1.11.2.9.8	Enumerate I/O Map - ATC Cabinet SIU Outputs	ATC:M	Yes / NA	
		3.5.2.1.11.2.9.9	Enumerate I/O Map - Auxiliary Device Inputs	O	Yes / No	
		3.5.2.1.11.2.9.10	Enumerate I/O Map - Auxiliary Device Outputs	O	Yes / No	
2.5.2.1.13	Manage Intra-Cabinet Communications Configuration			O	Yes / No	
		3.5.2.1.12.1.1	Determine Serial Bus 1 Device Present	ATC:M	Yes / NA	The ASC shall support at least ____ Serial Bus 1 Addresses (between 1 and 255).
		3.5.2.1.12.2.1	Determine TS2 Port 1 Device Present	TS2-2:M	Yes / NA	The ASC shall support at least ____ TS2 Port1 Addresses (between 1 and 255).
		3.5.2.1.12.2.2	Enable/Disable TS2 Port 1 Frame 40 Messages	TS2-2:M	Yes / NA	
2.5.2.1.14	Manage Accessible Pedestrian Support			O	Yes / No	
		3.5.2.1.13.1.1	Configure APS Push Button Minimum Press Time	M	Yes	
		3.5.2.1.13.1.2	Configure APS Push Button to Phase Association	M	Yes	
		3.5.2.1.13.1.3	Configure APS Extra Crossing Time	M	Yes	
		3.5.2.1.13.1.4	Configure Pedestrian Detector for Alternate Pedestrian Timing	M	Yes	
2.5.2.2	Monitor Signal Operations Status					
2.5.2.2.1	Determine Controller Health			M	Yes	
		ISO 26048-1§ 8.5.2.9	Monitor when the SNMP capabilities last changed	M	Yes	
		ISO 26048-1§ 8.5.2.10	Monitor controller communications	M	Yes	
		ISO 26048-1§ 8.5.2.11	Monitor controller operational status	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		ISO 26048-1§ 8.5.2.12	Monitor controller up time	M	Yes	
		ISO 26048-1§ 8.5.2.13	Monitor watchdog failure count	M	Yes	
		3.5.2.2.1.1	Monitor External Alarm Input States	M	Yes	The ASC shall support at least ____ Alarm Groups (between 1 and 255).
		3.5.2.2.1.2	Monitor External Alarm Active	M	Yes	
		3.5.2.2.1.3	Monitor Flash Status	M	Yes	
		3.5.2.2.1.4	Monitor Local Override	M	Yes	
		3.5.2.2.1.5	Monitor Coordination Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.6	Monitor Detector Fault	Detector: M	Yes / NA	
		3.5.2.2.1.7	Monitor Stop Time Input Alarm	M	Yes	
		3.5.2.2.1.8	Monitor Cycle Fault Alarm	M	Yes	
		3.5.2.2.1.9	Monitor Coordination Fault	Coord:M	Yes / NA	
		3.5.2.2.1.10	Monitor Coordination Fail Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.11	Monitor Cycle Fail Alarm	Coord:M	Yes / NA	
		3.5.2.2.1.12	Monitor Cabinet IO Link Alarm	M	Yes	
		3.5.2.2.1.13	Monitor SMU Communications Error	O	Yes / No	
		3.5.2.2.1.14	Monitor Preempt Maximum Presence Alarm	Preempt: O	Yes / No / NA	
2.5.2.2.2	Determine Mode of Operation					
2.5.2.2.2.1 (Unit)	Monitor Unit-wide General Operations			O	Yes / No	
		3.5.2.2.2.1	Monitor Unit Control Status	M	Yes	
		3.5.2.2.2.2	Monitor Preempt Active	Preempt: M	Yes / NA	
		3.5.2.2.2.3	Monitor Offset Transitioning	Coord:M	Yes / NA	
		3.5.2.2.2.4	Monitor Priority Call Active	O	Yes / No	
		3.5.2.2.2.5	Monitor Local Free Status	M	Yes	
		3.5.2.2.2.6	Monitor Coordination Active	Coord:M	Yes / NA	
		3.5.2.2.2.7	Monitor ECLA Control Active	O	Yes / No	
2.5.2.2.2.2	Monitor Flashing			M	Yes	
		3.5.2.2.1.3	Monitor Flash Status	M	Yes	
2.5.2.2.2.3	Monitor Current Timing Pattern			Coord:M	Yes / NA	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.2.2.5	Monitor Local Free Status	M	Yes	
		3.5.2.2.2.8.1	Monitor Current Pattern Status	M	Yes	
		3.5.2.2.2.8.2	Monitor Current Pattern Command Source	O	Yes / No	
		3.5.2.2.2.8.3	Monitor Current Pattern Fault Status	O	Yes / No	
2.5.2.2.2.4	Monitor Current Cycle			Coord:M	Yes / NA	
		3.5.2.2.2.9.1	Monitor Coordination Cycle Status	M	Yes	
		3.5.2.2.2.9.2	Monitor Coordination Synchronization Status	M	Yes	
		3.5.2.2.2.9.3	Monitor Current Offset	M	Yes	
2.5.2.2.3	Monitor Phase Status			M	Yes	
		3.5.2.2.3.1	Monitor Active Red Phases	M	Yes	
		3.5.2.2.3.2	Monitor Active Yellow Phases	M	Yes	
		3.5.2.2.3.3	Monitor Active Green Phases	M	Yes	
		3.5.2.2.3.4	Monitor Active Don't Walk Phases	M	Yes	
		3.5.2.2.3.5	Monitor Active Pedestrian Clearance Phases	M	Yes	
		3.5.2.2.3.6	Monitor Active Walk Phases	M	Yes	
		3.5.2.2.3.7	Monitor Active On Phases	M	Yes	
		3.5.2.2.3.8	Monitor Next Phases	M	Yes	
		3.5.2.2.3.9	Monitor Phase Vehicle Calls	M	Yes	
		3.5.2.2.3.10	Monitor Phase Pedestrian Calls	M	Yes	
		3.5.2.2.7.4	Monitor Current Gate Status	O	Yes / No	
2.5.2.2.4	Monitor Ring Status			Ring:M	Yes / NA	
		3.5.2.2.4.1	Monitor Ring Status	M	Yes	
		3.5.2.2.4.2	Monitor Ring Termination Cause	M	Yes	
		3.5.2.2.4.3	Monitor Current Phase On Time	M	Yes	
2.5.2.2.5	Monitor Channel Status			Channel: M	Yes / NA	
		3.5.2.2.5.1	Monitor Active Red Channels	M	Yes	
		3.5.2.2.5.2	Monitor Active Yellow Channels	M	Yes	
		3.5.2.2.5.3	Monitor Active Green Channels	M	Yes	
2.5.2.2.6	Monitor Overlap Status			Overlap: M	Yes / NA	
		3.5.2.2.6.1	Monitor Active Red Overlaps	M	Yes	
		3.5.2.2.6.2	Monitor Active Yellow Overlaps	M	Yes	
		3.5.2.2.6.3	Monitor Active Green Overlaps	M	Yes	
		3.5.2.2.6.4	Monitor Active Flashing Yellow Arrow Overlaps	O	Yes / No	
		3.5.2.2.6.5	Monitor Active Flashing Red Arrow Overlaps	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.2.2.7	Monitor Preempt Status			Preempt: M	Yes / NA	
		3.5.2.2.7.1	Monitor Currently Active Preempt	M	Yes	
		3.5.2.2.7.2	Monitor Current Preempt Inputs	M	Yes	
		3.5.2.2.7.3	Monitor Current Preempt State	M	Yes	
		3.5.2.2.7.4	Monitor Current Gate Status	O	Yes / No	
2.5.2.2.8 (SpecialFunc)	Monitor Special Function Outputs			O	Yes / No	
		3.5.2.2.8.1	Determine Maximum Number of Special Functions	M	Yes	The ASC shall support at least ____ Special Functions (between 1 and 255).
		3.5.2.2.8.2	Monitor Special Function Status	M	Yes	
		3.5.2.2.8.3	Monitor Special Function Control Source	O	Yes / No	
2.5.2.2.9	Monitor Timebase Action Status			M	Yes	
		3.5.2.2.2.8.1	Monitor Current Pattern Status	patternFunction: M	Yes / NA	
		3.5.2.1.10.1.4	Determine Action in Effect	actionFunction:M	Yes / NA	
2.5.2.2.10	Monitor Intra-Cabinet Communications Configuration			O	Yes / No	
		3.5.2.2.9.1	Monitor TS2 Port 1 Status	TS2-2:M	Yes / NA	
		3.5.2.2.9.2	Monitor TS2 Port 1 Fault Frame	TS2-2:M	Yes / NA	
		3.5.2.2.9.3	Monitor Serial Bus 1 Status	ATC:M	Yes / NA	
2.5.2.2.11	Monitor Signal Monitoring Unit			O	Yes / No	
		3.5.2.2.10.1	Monitor Signal Monitoring Unit Channel Voltage	ATC:M	Yes	It's optional for any cabinet, but format is standard on an ATC.
		3.5.2.2.10.2	Monitor Signal Monitoring Unit Channel Current	ATC:M	Yes	
2.5.2.3	Control Signal Operations			M	Yes	
2.5.2.3.1	Control ASC-wide General Operations			M	Yes	
		3.5.2.3.1.1	Enable/Disable Manual Backup	O	Yes / No	
		3.5.2.3.1.2	Control Global Minimum Recall	M	Yes	
		3.5.2.3.1.3	Control Call to Non-Actuated 1	M	Yes	
		3.5.2.3.1.4	Control Call to Non-Actuated 2	M	Yes	
		3.5.2.3.1.5	Control Walk Rest Modifier	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.2.3.1.6	Control Interconnect	O	Yes / No	
2.5.2.3.2	Activate Timing Pattern			Coord:M	Yes / NA	
		3.5.2.3.2.1	Activate System Timing Pattern Remotely	M	Yes	
		3.5.2.3.2.2	Control System Reference Point	M	Yes	
2.5.2.3.3	Phase Requests			O	Yes / No	
		3.5.2.3.3.1	Control Phase Omits	M	Yes	
		3.5.2.3.3.2	Control Pedestrian Phase Omits	M	Yes	
		3.5.2.3.3.3	Control Phase Holds	M	Yes	
		3.5.2.3.3.4	Control Phase Force Offs	O	Yes / No	
		3.5.2.3.3.5	Control Phase Vehicle Calls	M	Yes	
		3.5.2.3.3.6	Control Phase Pedestrian Calls	M	Yes	
2.5.2.3.4	Activate Preempt			Preempt:O	Yes / No / NA	
		3.5.2.3.4	Activate Preempt Remotely	M	Yes	
2.5.2.3.5	Control Ring Operations			Ring:O	Yes / No / NA	
		3.5.2.3.5.1	Control Ring Stop Time	M	Yes	
		3.5.2.3.5.2	Control Ring Force Offs	M	Yes	
		3.5.2.3.5.3	Control Ring Maximum 2 Settings	M	Yes	
		3.5.2.3.5.4	Control Ring Maximum 3 Settings	O	Yes / No	
		3.5.2.3.5.5	Control Ring Maximum Inhibit Settings	M	Yes	
		3.5.2.3.5.6	Control Ring Pedestrian Recycle Settings	M	Yes	
		3.5.2.3.5.7	Control Ring Red Rest Settings	M	Yes	
		3.5.2.3.5.8	Control Ring Red Clearance Omit Settings	M	Yes	
2.5.2.3.6	Activate Special Function Output			SpecialFunc:O	Yes / No / NA	
		3.5.2.3.6	Activate Special Function Remotely	M	Yes	
2.5.2.3.7	Activate Action Plan			O	Yes / No	
		3.5.2.1.10.5	Activate Action Plan Remotely	actionFunction:M	Yes / NA	
		3.5.2.3.2.1	Activate System Timing Pattern Remotely	patternFunction: M	Yes / NA	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.2.3.8	Remote Manual Control			O	Yes / No	
		3.5.2.3.7.1	Enable Remote Manual Control	M	Yes	
		3.5.2.3.7.2	Advance Interval During Remote Manual Control	M	Yes	
		3.5.2.3.7.3	Configure Manual Control Timeout	M	Yes	
		3.5.2.3.7.4	Enable/Disable Automatic Pedestrian Clearance Setting	M	Yes	
2.5.3	Manage Detectors					
2.5.3.1 (Detector)	Manage Detector Configuration			M	Yes	
		3.5.3.1.1.1	Configure Vehicle Travel Mode	O	Yes / No	
		3.5.3.1.1.2	Configure Vehicle Detector Description	O	Yes / No	
		3.5.3.1.1.3	Configure Vehicle Detector Yellow Lock Call Enabled	O	Yes / No	
		3.5.3.1.1.4	Configure Vehicle Detector Red Lock Call Enabled	O	Yes / No	
		3.5.3.1.1.5	Configure Vehicle Detector Passage Enabled	O	Yes / No	
		3.5.3.1.1.6	Configure Vehicle Detector Added Initial Time Enabled	O	Yes / No	
		3.5.3.1.1.7	Configure Vehicle Detector Queue Enabled	O	Yes / No	
		3.5.3.1.1.8	Configure Vehicle Detector Call Enabled	M	Yes	
		3.5.3.1.1.9	Configure Vehicle Detector Call Phase	M	Yes	
		3.5.3.1.1.10	Configure Vehicle Detector Switch Phase	M	Yes	
		3.5.3.1.1.11	Configure Vehicle Detector Delay Time	M	Yes	
		3.5.3.1.1.12	Configure Vehicle Detector Extend Time	M	Yes	
		3.5.3.1.1.13	Configure Vehicle Detector Queue Limit Time	O	Yes / No	
		3.5.3.1.1.14	Configure Vehicle Detector No Activity Fault Time	M	Yes	
		3.5.3.1.1.15	Configure Vehicle Detector Maximum Presence Fault Time	M	Yes	
		3.5.3.1.1.16	Configure Vehicle Detector Erratic Counts	M	Yes	
		3.5.3.1.1.17	Configure Vehicle Detector Fail Time	O	Yes / No	
		3.5.3.1.2 (VehDetectSet)	Configure Multiple Vehicle Detector Sets for Actuation	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.3.1.3.1	Configure Pedestrian Detector Description	O	Yes / No	
		3.5.3.1.3.2	Configure Pedestrian Detector Call Phase	M	Yes	
		3.5.3.1.3.3	Configure Pedestrian Detector No Activity Fault Time	M	Yes	
		3.5.3.1.3.4	Configure Pedestrian Detector Maximum Presence Fault Time	M	Yes	
		3.5.3.1.3.5	Configure Pedestrian Detector Erratic Counts	M	Yes	
		3.5.3.1.3.6	Configure Pedestrian Detector Non-Lock Calls	O	Yes / No	
		3.5.3.1.3.7	Configure Pedestrian Detector for Presence Detection	O	Yes / No	
		3.5.3.1.3.8	Configure Pedestrian Detector for Delayed Walk	O	Yes / No	
		3.5.3.1.3.9	Configure Pedestrian Detector for Advanced Walk	O	Yes / No	
		3.5.3.1.4 (PedDetectSet)	Configure Multiple Pedestrian Detector Sets for Actuations	O	Yes / No	
		3.5.3.1.5.1	Determine Maximum Number of Vehicle Detectors	M	Yes	The ASC shall support at least ____ vehicle detectors (between 1 and 255).
		3.5.3.1.5.2	Determine Maximum Number of Vehicle Detector Sets	VehDetectSet:M	Yes / NA	The ASC shall support at least ____ vehicle detector sets.
		3.5.3.1.5.3	Determine Maximum Number of Pedestrian Detectors	M	Yes	The ASC shall support at least ____ pedestrian detectors (between 1 and 255).
		3.5.3.1.5.4	Determine Maximum Number of Pedestrian Detector Sets	PedDetectSet:M	Yes / NA	The ASC shall support at least ____ pedestrian detector sets.
2.5.3.2	Monitor Detector Status			O	Yes / No	
		3.5.3.2.1	Monitor Active Vehicle Detector Actuations	M	Yes	
		3.5.3.2.2	Monitor Active Pedestrian Detector Actuations	M	Yes	
2.5.3.3	Monitor Detector Health			O	Yes / No	
		3.5.3.3.1.1	Monitor Vehicle Detector Alarm Status	M	Yes	
		3.5.3.3.1.2	Monitor Vehicle Detector Faults from Controller	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.3.3.1.3	Monitor Vehicle Detector Faults from Detector	O	Yes / No	
		3.5.3.3.3.1	Monitor Pedestrian Detector Alarm Status	M	Yes	
		3.5.3.3.3.2	Monitor Pedestrian Detector Faults	M	Yes	
2.5.3.4	Control Detectors			O	Yes / No	
		3.5.3.4.1	Control Vehicle Detector Reset	M	Yes	
		3.5.3.4.2	Control Pedestrian Detector Reset	M	Yes	
		3.5.3.4.3	Control Detector Diagnostic Reset	O	Yes / No	
		3.5.3.4.4	Control Vehicle Detector Actuation	O	Yes / No	
		3.5.3.4.5	Control Pedestrian Detector Actuation	O	Yes / No	
2.5.3.5	Manage Detector Data Collection			O	Yes / No	
		3.5.3.5.3.1	Configure Vehicle Detector Data Sample Period	M	Yes	
		3.5.3.5.3.2	Configure Pedestrian Detector Sample Period	M	Yes	
		3.5.3.5.3.3 (Speed)	Configure Vehicle Speed Detectors	O	Yes / No	
		3.5.3.5.3.4	Configure Single Detector Speed Mode	Speed:M	Yes / NA	
		3.5.3.5.3.5	Configure Paired Detector	Speed:M	Yes / NA	
		3.5.3.5.3.6	Configure Paired Detector Placement	Speed:M	Yes / NA	
		3.5.3.5.3.7	Configure Paired Detector Spacing	Speed:M	Yes / NA	
		3.5.3.5.3.8	Configure Average Vehicle Length	O	Yes / No	
		3.5.3.5.3.9	Configure Vehicle Detection Zone Length	O	Yes / No	
		3.3.3.5.4	Configure Multiple Vehicle Detector Sets for Data Collection	O	Yes / No	
2.5.3.6	Monitor Detector Data from Controller			O	Yes / No	
		3.5.3.5.1.1	Monitor Vehicle Detector Data Sequence	M	Yes	
		3.5.3.5.1.2	Monitor Vehicle Volume Data	O	Yes / No	
		3.5.3.5.1.3	Monitor Vehicle Occupancy Data	O	Yes / No	
		3.5.3.5.1.4	Monitor Vehicle Average Speed	Speed:M	Yes / NA	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.3.5.1.5	Monitor Vehicle Detector Data Sample Time	O	Yes / No	
		3.5.3.5.2.1	Monitor Pedestrian Detector Data Sequence	O	Yes / No	
		3.5.3.5.2.2	Monitor Pedestrian Counts	O	Yes / No	
		3.5.3.5.2.3	Monitor Pedestrian Actuations	O	Yes / No	
		3.5.3.5.2.4	Monitor Pedestrian Services	O	Yes / No	
2.5.4 (CV)	Manage Connected Vehicles Interface			O	Yes / No	
2.5.4.1	Connected Vehicle Interface: Management Station – ASC Interface			M	Yes / No	
2.5.4.1.1	Manage ASC - RSU Interface			M	Yes	
		3.5.4.1.1	Configure ASC Communications Port for RSU	M	Yes	
		3.5.4.1.2	Configure Logical RSU Ports and Address	M	Yes	
		3.5.4.1.3	Configure RSU Interface Polling Period	O	Yes / No	
2.5.4.1.2	Manage ASC - RSU Interface Watchdog			O	Yes / No	
		3.5.4.1.4	Configure RSU Interface Watchdog	M	Yes	
		3.5.4.1.5	Monitor RSU Interface Watchdog Timer	M	Yes	
		3.5.4.1.6	Monitor RSU Interface Watchdog Alarm	M	Yes	
2.5.4.1.3	Manage Signal Phase and Timing Data			M	Yes	
		3.5.4.2.1.1	Enable Signal Phase and Timing Data	M	Yes	
		3.5.4.2.1.2	Retrieve Signal Phase and Timing Generation Time	O	Yes / No	
		3.5.4.2.1.3.1.1	Monitor Movement Minimum End Time	M	Yes	
		3.5.4.2.1.3.1.2	Monitor Movement Maximum End Time	M	Yes	
		3.5.4.2.1.3.1.3	Monitor Movement Likely End Time	O	Yes / No	
		3.5.4.2.1.3.1.4	Monitor Movement Likely End Time Confidence	O	Yes / No	
		3.5.4.2.1.3.1.5	Monitor Movement Next Occurrence	M	Yes	
		3.5.4.2.1.3.1.6	Monitor Movement Start Time	M	Yes	
		3.5.4.2.1.3.1.7	Monitor Next Movement Minimum End Time	M	Yes	
		3.5.4.2.1.3.1.8	Monitor Next Movement Maximum End Time	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.5.4.2.1.3.1.9	Monitor Next Movement Start Time	M	Yes	
		3.5.4.2.1.3.1.10	Determine Maximum Number of Movement Events			
		3.5.4.2.1.3.2.1	Configure Queue Detectors for Movement Assistance	MvtQueue:M	Yes / NA	
		3.5.4.2.1.3.2.2	Configure Pedestrian Detectors for Movement Conflict Assistance	MvtConflict:O.4 (1..*)	Yes / No / NA	
		3.5.4.2.1.3.2.3	Configure Bicycle Detectors for Movement Conflict Assistance	MvtConflict:O.4 (1..*)	Yes / No / NA	
		3.5.4.2.1.3.3.1 (MvtQueue)	Monitor Lane Connection Queue Length	O	Yes / No	
		3.5.4.2.1.3.3.2 (MvtConflict)	Monitor Lane Connection Vulnerable Road User Detection	O	Yes / No	
		3.5.4.2.1.3.4.1 (SpdAdvice)	Configure Advisory Speed Type	O	Yes / No	
		3.5.4.2.1.3.4.2	Configure Advisory Speed	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.4.3	Configure Advisory Speed Zone	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.4.4	Configure Advisory Speed Vehicle Type	SpdAdvice:O	Yes / No / NA	
		3.5.4.2.1.3.5	Monitor Movement State	M	Yes	
		3.5.4.2.1.3.6	Monitor Next Movement State	M	Yes	
		3.5.4.2.1.3.7	Monitor Movement Status	M	Yes	
		3.5.4.2.1.4.1	Configure Concurrent Enabled Lanes	M	Yes	
		3.5.4.2.1.4.2	Configure Enabled Lanes by Time of Day	M	Yes	
		3.5.4.2.1.4.3	Determine Lanes Enabled	M	Yes	
		3.5.4.2.1.4.4	Command Enabled Lanes	M	Yes	
		3.5.4.2.1.5	Enable Signal Phase and Timing Exchange	M	Yes	
		3.5.4.2.1.6	Configure Road Authority Identifier	M	Yes	
		3.5.4.2.1.7.1	Monitor Manual Control Indication	M	Yes	
		3.5.4.2.1.7.2	Monitor Stop Indication	M	Yes	
		3.5.4.2.1.7.3	Monitor Failure Flash Indication	M	Yes	
		3.5.4.2.1.7.4	Monitor Preemption Operation Indication	M	Yes	
		3.5.4.2.1.7.5	Monitor Priority Operation Indication	M	Yes	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.4.2.1.7.6	Monitor Fixed Time Control Indication	M	Yes	
		3.5.4.2.1.7.7	Monitor Non-Fixed Time Control Indication	M	Yes	
		3.5.4.2.1.7.8	Monitor Standby Operation Indication	M	Yes	
		3.5.4.2.1.7.9	Monitor Controller Failure	M	Yes	
		3.5.4.2.1.7.10	Monitor MAP Message Validity	M	Yes	
		3.5.4.2.1.7.11	Monitor SPaT Message Validity	M	Yes	
		3.5.4.2.1.8	Mark SPaT Invalid - Controller	M	Yes	
		3.5.4.2.1.9	Mark SPaT Invalid - Port	O	Yes / No	
		3.5.4.2.1.10	Mark MAP Message Invalid - Controller	M	Yes	
		3.5.4.2.1.11	Mark MAP Message Invalid - Port	O	Yes / No	
		3.5.4.2.1.12.1	Determine Maximum Number of Signal Groups	M	Yes	
		3.5.4.2.1.12.2	Configure Signal Groups Intersection Mapping	M	Yes	
		3.5.4.2.1.12.3	Configure Signal Group Control Source	M	Yes	
		3.5.4.2.1.12.4	Configure Signal Group Indication Types	M	Yes	
		3.5.4.2.1.12.5	Configure Signal Group Protected or Permissive State	M	Yes	
		3.5.4.2.1.12.6	Configure Signal Group Revocable Lanes	M	Yes	
		3.5.4.2.1.12.7	Determine Maximum Number of Signal State Entries	M	Yes	
		3.5.4.2.1.12.8	Configure Customized Signal State Parameters	M	Yes	
		3.5.4.2.1.13	Retrieve Signal Phase and Timing Time Point	M	Yes	
2.5.4.1.4	Manage Assured Green Period			O	Yes / No	
		3.5.4.2.2.1	Enabled Connected Device Connection	M	Yes	
		3.5.4.2.2.2	Configure Vehicle Detector for Connected Vehicle Application	M	Yes	
		3.5.4.2.2.3	Configure Connected Vehicle Detector Assigned Input	M	Yes	
		3.5.4.2.2.4	Configure Connected Vehicle Detector Port Assignment	O	Yes / No	
		3.5.4.2.2.5	Configure Assured Green Period Duration	M	Yes	
		3.5.4.2.2.6	Configure Red Light Violation Warning Parameters	O	Yes / No	
2.5.4.2	Connected Vehicle Interface: ASC – CV Application Process Interface			M	Yes	

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
2.5.4.2.1	Exchange Current and Next Movement Information			M	Yes	
		3.5.4.2.1.1.1	Provide Movement Time Point	M	Yes	
		3.5.4.2.1.1.2	Provide Movement State	M	Yes	
		3.5.4.2.1.1.3	Provide Movement Minimum End Time	M	Yes	
		3.5.4.2.1.1.4	Provide Movement Maximum End Time	M	Yes	
		3.5.4.2.1.1.5	Provide Movement Likely End Time	O	Yes / No	
		3.5.4.2.1.1.6	Provide Movement Likely End Time Confidence	O	Yes / No	
		3.5.4.2.1.1.7	Provide Next Movement State	M	Yes	
		3.5.4.2.1.1.8	Provide Next Movement Minimum End Time	M	Yes	
		3.5.4.2.1.1.9	Provide Next Movement Maximum End Time	M	Yes	
		3.5.4.2.1.1.10	Provide Next Movement Start Time	M	Yes	
		3.5.4.2.1.2.1	Provide Lane Connection Queue Length	O	Yes / No	
		3.5.4.2.1.2.2	Provide Lane Connection Traveler Detection	O	Yes / No	
		3.5.4.2.1.3.1	Provide Advisory Speed Type	O	Yes / No	
		3.5.4.2.1.3.2	Provide Advisory Speed	O	Yes / No	
		3.5.4.2.1.3.3	Provide Advisory Speed Zone	O	Yes / No	
		3.5.4.2.1.3.4	Provide Advisory Speed Vehicle Type	O	Yes / No	
		3.5.4.2.1.4	Provide Road Authority ID	M	Yes	
		3.5.4.2.1.5	Provide Signal Phase and Timing Intersection Status	M	Yes	
		3.5.4.3.1.6	Provide Compressed SPaT Information to External CV Application Process	M	Yes	
		3.5.4.3.2.1	Provide UPER-encoded SPaT Message	M	Yes	
		3.6.5.1	SPaT Maximum Transmission Start Time	M	Yes	The Maximum Transmission Start Time for all SPAT data shall be ___ milliseconds (Default=10).
		3.6.5.2	Movement Time Point Minimum Transmission Rate	M	Yes	The Movement Time Point Minimum Transmission Rate shall be once per ______ milliseconds (Default=100).

Protocol Requirements List (PRL)						
User Need ID	**User Need**	**FR ID**	**Functional Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.6.5.3	SPaT Maximum Transmission Rate	M	Yes	
		3.6.5.4	SPaT Time Accuracy	M	Yes	
2.5.4.2.2	Exchange Next Occurrence of a Movement			M	Yes	
		3.5.4.3.1.1.11	Provide Movement Next Occurrence	M	Yes	
		3.5.4.3.4	Monitor CV Certificate Faults	O	Yes / No	
		3.6.5.1	SPaT Maximum Transmission Start Time	M	Yes	The Maximum Transmission Start Time for all SPAT data that changed shall be ___ milliseconds (Default=10).
		3.6.5.2	Movement Time Point Minimum Transmission Rate	M	Yes	The Movement Time Point Minimum Transmission Rate shall be once per _____ milliseconds (Default=100).
		3.6.5.3	SPaT Maximum Transmission Rate	M	Yes	
		3.6.5.4	SPaT Time Accuracy	M	Yes	
2.5.4.2.3	Exchange Presence of Connected Devices			O	Yes / No	
		3.5.4.3.2.2	Retrieve BSMs	O.12(1..*)	Yes / No	
		3.5.4.3.2.3	Retrieve PSMs	O	Yes / No	
		3.5.4.3.2.4	Retrieve Actuation Report	O.12(1..*)	Yes / No	
		3.5.4.3.2.5	Retrieve Detection Report	O	Yes / No	
2.5.4.2.4	Exchange Roadway Geometrics Information			O	Yes / No	
		3.5.4.3.3.1	Retrieve MAP Plan in Effect	M	Yes	
		3.5.4.3.3.2	Confirm MAP Plan Compatibility	M	Yes	
2.5.4.3	ASC – ECLA Interface			O	Yes / No	
		3.5.4.4.1	Receive Current Phase Minimum End Time from an ECLA	M	Yes	
		3.5.4.4.2	Receive Current Phase Maximum End Time from an ECLA	M	Yes	
		3.5.4.4.3	Receive Current Phase Likely End Time from an ECLA	O	Yes / No	

Protocol Requirements List (PRL)						
User Need ID	User Need	FR ID	Functional Requirement	Conformance	Support	Additional Specifications
		3.5.4.4.4	Receive Current Phase Likely End Time Confidence from an ECLA	O	Yes / No	
		3.5.4.4.5	Receive Next Phase from an ECLA	M	Yes	
		3.5.4.4.6	Receive Compressed ECLA Input Data	M	Yes	
2.5.5	Backward Compatibility Features					
2.6	Security			M	Yes	
2.6.1	Manage Authentication			M	Yes	
2.6.2	Manage Accessibility			M	Yes	
2.6.3	Manage Users			M	Yes	
		ISO 26048-1§ 8.22	View-based access control model (VACM) feature	M	Yes	All requirements that trace to this feature are also inherited. See the FTRT in ISO 26048-1.
2.6.4	Log User Access			O	Yes / No	
		ISO 26048-1§ 8.1.3.1	Validate access upon action activation	M	Yes	
		ISO 26048-1§ 8.1.3.2	Validate access upon action being called	M	Yes	
		3.6.1	Response Time for Requests	M	Yes	The Response Time for all requests shall be ___ milliseconds (5-500: Default=25).
2.6.5	Manage ASC Interface Security			O	Yes / No	
2.6.5.1	Manage Security for the ASC to RSU Interface					
2.6.5.2	Manage Security for Other ASC Interfaces					

). It is recognized that there are various methods and communications protocols that can be used to exchange information across either interface and is highly dependent on the relationship between the ASC Process and the CV Application Process.

For example, one possibility is that the ASC Process and the CV Application Process are integral to and performed by the ASC, with an external V2X Radio for communicating with connected devices. In this situation, the ASC Process - CV Application Process interface does not exist from the perspective of NTCIP 1202 v04 because the functions of the ASC Process and the CV Application Process are considered to be internal to the ASC controller unit.

F.3.3 Notes and Guidance

This section contains notes and guidance for object definitions in NTCIP 1202 that provide SPaT data from the ASC Process to the CV Application Process.

F.3.3.1 Disciplined Time

For the purposes of this discussion, the time from a reliable source will be called **disciplined time** - that is, it "does not accumulate any offset over time." When broadcasting SPaT messages, the CV Application Process has to provide signal timing data using disciplined time.

However, many existing traffic signal controllers use AC line frequency, which is not disciplined time, to determine its internal time. AC line frequency has the benefit that all signal controllers that use the same line frequency, such as along an arterial, remain synchronized for signal timing coordination, but is an issue when providing SPaT data.

To address this issue, the ASC provides ascCurrentTick2 that defines the current time point for the ASC. This time point is provided in units of tenths of a second, with a value of 0 representing the top of the hour, resulting in a range of 0 to 35999. The values of 36009 to are used to support leap seconds. 36111 is used when the time is undefined or unknown. These time points do not need to be synchronized with UTC time or the RSU time.

F.3.3.2 SPaT and MAP Relationship

The ASC is the source of the SPAT data that may be broadcast to travelers, in the form of a SAE J2735 SPAT message, for a signalized intersection. However, this SPAT data has little value to travelers unless the SPAT data can be correlated a desired traveler movement through a signalized intersection. In SAE J2735, this desired movement is characterized by the identity of the lanes the traveler wishes to enter and exit the signalized intersection. The identity and location of the lane is provided in a MAP data message that is broadcast to travelers. Without this MAP data message, a traveler may receive the SPAT data broadcast, but cannot unambiguously determine what data applies to his desired movement through the signalized intersection.

In SAE J2735, the relationship between the SPAT and MAP messages is established by a data element, DE_SignalGroupID. In the SPAT message, DE_SignalGroupID is identified with a movement phase state (e.g., green, yellow, red) and timing data (TimeChangeDetails) for that movement. In the MAP data message, every allowed movement (e.g., identifier of the desired lanes to enter and exit the intersection) is associated with DE_SignalGroupID. Every DE_SignalGroupID in the SPaT data provided by the traffic controller in NTCIP 1202 v03B must also have the DE_SignalGroupID identified in the corresponding MAP message.

F.3.3.3 SPaT Data Overview

Figure 13 shows the relationship between the tables and objects defined by NTCIP 1202 v04 to support providing SPaT data to the CV Application Process to generate a SPaT message. Each box represents a table in NTCIP 1202 v04.

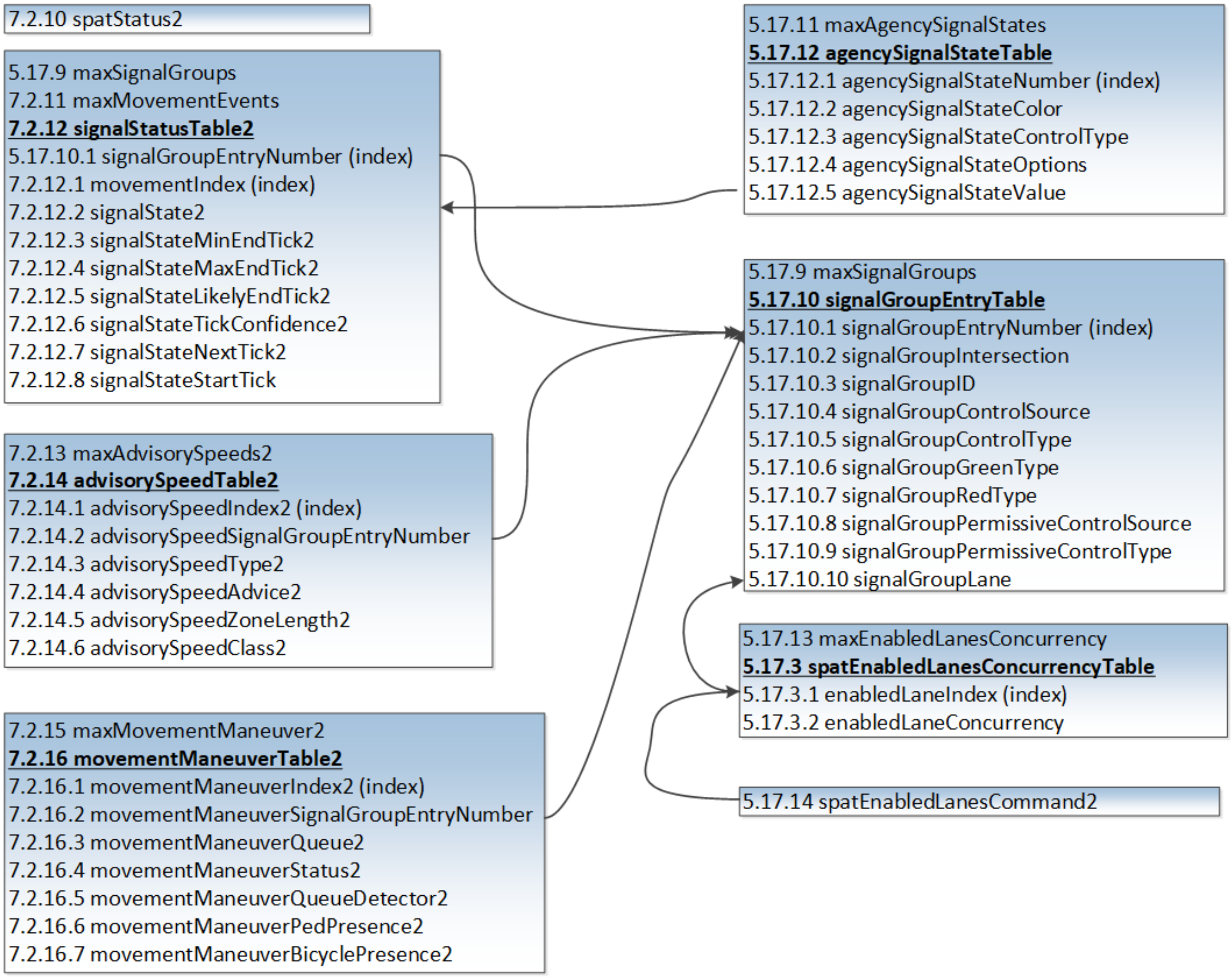

Figure 13 NTCIP 1202 v04 Tables to Provide SPaT Data

The remainder of this section provides an overview of the tables and objects defined in this standard to support the exchange of SPAT data so a CV Application Process can generate and broadcast SPAT messages; and so a management station such as a server at a TMC can monitor the SPAT data being generated by the ASC.

Note that the SPAT data can be exchanged with more than one CV Application Process (or RSU). This scenario may be necessary for ASCs that control signal timing for more than one intersection that may be spaced far apart. Assuming V2X communications such as C-V2X has a broadcast/receive range of approximately 300 meters, when considering the speeds of vehicles approaching a signalized intersection, a broadcast range of 300 meters may be insufficient if the intersections controlled by an ASC are far apart, thus requiring the use of more than one RSU to broadcast SPAT messages. Examples of intersections that also may require more than one RSU are complex traffic circles (e.g., Dupont Circle in Washington, DC) or Texas diamond interchanges.

The SPaT data provided from an ASC to the CV Application Process using NTCIP 1202 is categorized into six parts. The first four parts are mandatory to be provided to the CV Application Process.

1. Intersection Status. Represented by spatStatus2, provides the status of the traffic signal controller. See Annex F.3.3.4 for additional guidance.

2. Movement Phase State. Represented by signalState2 in signalStatusTable2, provides the current and next state of each allowed movement at the intersection. See Annex F.3.3.5 for additional guidance.
3. TimeChangeDetails. Represented by signalStateMinEndTick2, signalStateMaxEndTick2, signalStateNextTick2, and signalStateStartTick objects in signalStatusTable2, provides the predicted start and end of each allowed movement at the intersection. The signalStatusTable2 also includes the signalStateLikelyEndTick2 and signalStateTickConfidence2 objects, but these objects are not mandatory for a SPaT message, as defined in CTI 4501. See Annex F.3.3.6 for additional guidance.
4. Enabled Lanes. Represented by spatEnabledLanesStatus, indicates what revocable lanes are "asserted". See Annex F.3.3.7 for additional guidance.
5. Advisory Speeds. Represented in advisorySpeedTable2, provides advisory speed for a signalGroupID. See Annex F.3.3.8 for additional guidance.
6. Movement Maneuver Assist. Represented in movementManeuverTable2, provides assist information for a movement at a signalized intersection. See Annex F.3.3.9 for additional guidance.

The signalGroupEntryTable and agencySignalStateTable determines the Movement Phase State for a signalGroupID based on the current signal indications. The spatEnabledLanesConcurrencyTable defines the allowable sets of enabled lanes (lanes that can be concurrently enabled) for the intersection.

The signalStatusBlock2 is an oerstring that contains the four mandatory parts required to be provided to the CV Application Process to support the generation of a SPaT message. The oerstring is a "compressed" version that consists of the Intersection Status, Movement Phase States, Enabled Lanes and the mandatory parts of TimeChangeDetails.

It is the intention that the ASC will exchange the details of the signalStatusBlock2 with the CV Application Process ten times per second. The signalStatusBlock2 object is the design from the viewpoint of the ASC – all the "objects" that make up the signalStatusBlock2 object are read-only objects. The equivalent object in the CV Application Process is called rsuSignalStatusBlock, which is a read-write object. Note that the equivalent CV Application Process objects should be defined in NTCIP 1218, but the time of publication the RSU Working Group had not re-convened, so these objects are proposed and are defined in Annex **Error! Reference source not found.**. The specific requirement for this exchange is 3.5.4.3.1.6, Provide Compressed SPaT Information to External CV Application Process, and the design to fulfill this requirement is for the ASC to SET rsuSignalStatusBlock on the CV Application Process. The signalStatusBlock2 can be used by the ASC to provide the SPaT information to a management station, such as a traffic management center.

F.3.3.3.1 Example

The following provides an octet string value for a signalStatusBlock2 object. This example is based on the intersection diagram in Figure 13. The example assumes no right-turn-on-red, so signal group 14 is NOT in effect, but signal group 24 is in effect. Also note that signal group 13 is not used. The example assumes a 3.0 second yellow duration and a 2.0 second red duration.

```
-- SEQUENCE
-- 46 51        ascCurrentTick2 = 18001 tenths of a second from the top of the hour
-- 0C 40        spatStatus2. Bits 6, 10, and 11 are TRUE. Bit 6 indicates Traffic Dependent mode.
-- 01           spatEnabledLanesCount
-- 0E           spatEnabledLanesStatus
-- 0F           signalGroupEntryCount = 15 signalGroupEntryNumbers
-- SEQUENCE OF (SignalStatusData2)
--   SEQUENCE # 1 (signalGroupEntryNumber=1 / movementEventNumber=1)
-- 01           signalGroupEntryNumber
-- 05           signalState2. Indicates stop-And-Remain.
```

-- 46 84 signalStateMinEndTick2. Minimum End Time is 18052 tenths of a second from the top of the hour
-- 46 BE signalStateMaxEndTick2. 18110 tenths of a second from the top of the hour
-- 8D 0F signalStateNextTick2. A value of 36111 is used to represent unknown.
-- 8D 0F signalStateStartTick. The Start Time of the current movement state is not sent.
-- SEQUENCE # 2 (signalGroupEntryNumber=1 / movementEventNumber=2)
-- 01 signalGroupEntryNumber
-- 08 signalState2. Indicates protected-Movement-Allowed
-- 8D 0F signalStateMinEndTick2. A value of 36111 is used to represent unknown.
-- 8D 0F signalStateMaxEndTick2. A value of 36111 is used to represent unknown.
-- 8D 0F signalStateNextTick2. A value of 36111 is used to represent unknown.
-- 8D 0F signalStateStartTick. Next Start Time is equal to the Minimum End Time.
-- SEQUENCE # 3 (signalGroupEntryNumber=2 / movementEventNumber=1)
-- 02 signalGroupEntryNumber
-- 07 signalState2. Indicates permissive-Movement-Allowed.
-- 46 52 signalStateMinEndTick2. Minimum End Time is 18002 tenths of a second from the top of the hour
-- 46 8C signalStateMaxEndTick2. 18060 tenths of a second from the top of the hour
-- 8D 0F signalStateNextTick2. A value of 36111 is used to represent unknown.
-- 8D 0F signalStateStartTick. The Start Time of the current movement state is not sent.
-- SEQUENCE # 4 (signalGroupEntryNumber=2 / movementEventNumber=2)
-- 02 signalGroupEntryNumber
-- 09 signalState2. Indicates permissive-clearance (9)
-- 46 70 signalStateMinEndTick2. Minimum End Time is 18032 tenths of a second from the top of the hour
-- 46 AA signalStateMaxEndTick2. 18090 tenths of a second from the top of the hour
-- 8D 0F signalStateNextTick2. A value of 36111 is used to represent unknown.
-- 46 70 signalStateStartTick. Next Start Time is equal to the Minimum End Time.
…
-- SEQUENCE # 24 (signalGroupEntryNumber=12 / movementEventNumber=2)
-- 0C signalGroupEntryNumber
-- 07 signalState2. Indicates permissive-Movement-Allowed.
-- 47 2E signalStateMinEndTick2. Minimum End Time is 18222 tenths of a second from the top of the hour
-- 47 EA signalStateMaxEndTick2. 18410 tenths of a second from the top of the hour
-- 8D 0F signalStateNextTick2. A value of 36111 is used to represent unknown.
-- 47 2E signalStateStartTick. Next Start Time is equal to the Minimum End Time.
-- SEQUENCE # 25 (signalGroupEntryNumber=15 / movementEventNumber=1)
-- 0F signalGroupEntryNumber
-- 07 signalState2. Indicates permissive-Movement-Allowed.
-- 46 C0 signalStateMinEndTick2. Minimum End Time is 18112 tenths of a second from the top of the hour
-- 47 36 signalStateMaxEndTick2. 18230 tenths of a second from the top of the hour
-- 8D 0F signalStateNextTick2. A value of 36111 is used to represent unknown.
-- 8D 0F signalStateStartTick. The Start Time of the current movement state is not sent.
…
-- SEQUENCE # 29 (signalGroupEntryNumber=24 / movementEventNumber=1)
-- 18 signalGroupEntryNumber
-- 05 signalState2. Indicates stop-And-Remain.
-- 46 F2 signalStateMinEndTick2. Minimum End Time is 18162 tenths of a second from the top of the hour
-- 47 68 signalStateMaxEndTick2. 18280 tenths of a second from the top of the hour
-- 8D 0F signalStateNextTick2. A value of 36111 is used to represent unknown.
-- 8D 0F signalStateStartTick. The Start Time of the current movement state is not sent.
-- SEQUENCE # 30 (signalGroupEntryNumber=24 / movementEventNumber=2)
-- 18 signalGroupEntryNumber

-- 07	signalState2. Indicates permissive-Movement-Allowed.
-- 47 88	signalStateMinEndTick2. Minimum End Time is 18312 tenths of a second from the top of the hour
-- 48 4E	signalStateMaxEndTick2. 18510 tenths of a second from the top of the hour
-- 8D 0F	signalStateNextTick2. A value of 36111 is used to represent unknown.
-- 47 88	signalStateStartTick. Next Start Time is equal to the Minimum End Time.

F.3.3.4 Intersection Status

DE_IntersectionStatusObject is a required data element when broadcasting a SAE J2735 SPaT message. DE_IntersectionStatusObject consists of several bit fields that must be populated – most of the bits are populated by the controller while some bits are populated by the CV Application Process. The traffic controller uses the spatStatus2 object to provide these status bits to the CV Application Process. See Sections 3.6.3.1 and 3.6.3.2 for the performance requirements to send spatStatus2 to the CV Application Process.

This section provides additional guidance for each bit for spatStatus2.

F.3.3.4.1 Manual Control

Represented by Bit 0 in spatStatus2, Manual Control indicates to the receiver of this information that the controller is "not in control", thus the controller generally does not know when the movement phase state will change. Under certain conditions, the controller may know the minimum end time for an interval, such as a yellow interval, in which case the controller will report the minimum end time. An external entity, such as a police officer at the cabinet or an operator at a traffic management center, is controlling the duration of certain intervals. When in manual control, the controller is unaware if the interval may terminate early or the interval may dwell for a longer period of time.

When manual control is "active" and the controller is in a green interval:

- the minimum time of change will be equal to the greater of the current time plus the minimum green time remaining (if in effect), or the assured green end time if the AGP is in use;
- while the controller is in a green dwell interval, the minimum time of change will not be less than 0.1 seconds; and
- the value for maximum time of change will be unknown.

When manual control is "active" and the controller is in a vehicle/pedestrian change or clearance interval that is not timed:

- the minimum time of change will not be less than 0.1 seconds; and
- the value for maximum time of change will be unknown.

When manual control is "active", the controller is in a vehicle/pedestrian change or clearance interval, and the interval durations (such as yellows) are timed, then the controller will provide the known TimeChangeDetails values.

The movement phase states are known by the controller in manual control, and should be reflected in the SPaT output.

F.3.3.4.2 Stop Time

Represented by Bit 1 in spatStatus2, Stop Time indicates to the receiver of this information that all counting/timing by the controller has stopped, thus most TimeChangeDetails are unknown.

When the controller receives a stop time input, the active (frozen) controller state on the affected (stopped) ring shall remain as the basis for SPaT generation. Even if the stop time duration is unknown, the SPaT output (movement phase state) will remain consistent to the current phase timers.

If the form of stop time input allows continuation of countdown timers for clearance intervals or other timers, the TimeChangeDetails will be adjusted accordingly. Otherwise, the values for TimeChangeDetails will be set to unknown.

F.3.3.4.3 Failure Flash

Represented by Bit 2 in spatStatus2, Failure Flash indicates to the receiver of this information that the controller is in a flashing condition and the flashing operation must be terminated by a source external to the controller – that is, the end of the flash condition is indeterminate.

Generally, there are three types of Failure Flash:

- "Monitor Flash" is a flash controlled by the monitor in the following two scenarios. In both cases, the controller sets NTCIP unitFlashStatus to mmu (6) during Monitor Flash:
    - When resuming operation after a power loss or interruption, the monitor keeps the Flash Bus energized for a minimum of 6 seconds before energizing the Signal Bus and transferring control to the controller.
    - When a fault is detected by the monitor, it energizes the Flash Bus until either the fault is cleared by pressing the Reset button on the monitor (latching fault) or, with certain types of faults, when the condition that caused the fault is no longer present (non-latching fault).
- "Fault Monitor Flash" occurs when the controller detects an anomaly in the components within the field cabinet, such as a mismatch between its copy of the permissive channels and the signal monitoring unit's copy of the permissive channels. The way the controller causes Fault Monitor Flash differs by cabinet architecture. For example, the controller sets the Fault Monitor output to FALSE in NEMA TS 2 Type 1 cabinets and sets Message Type 62 Bit 1 to 1 in ITS and ATC cabinets. The controller exits Fault Monitor Flash when the anomaly is no longer present. Contrast Fault Monitor Flash with Fault Flash, which is initiated by the monitor and persists until the monitor is reset. The controller sets NTCIP unitFlashStatus to faultMonitor (5) during Fault Monitor Flash.
- "Local Flash" is a flash controlled by human-operated switches in the cabinet, typically labeled "AUTO/FLASH", that is used by technicians to flash the signals when performing maintenance on the controller (Tech Flash) or by police during unusual traffic conditions or situations (Police Flash). The controller sets NTCIP unitFlashStatus to localManual (4) during Local Flash.

When the intersection is in Failure Flash, the end of Failure Flash is indeterminate because the controller does not know when the monitor is going to be reset (for Monitor Flash or Fault Monitor Flash), when the condition that cause the fault is no longer present, or when the flash switch is going to be moved from "FLASH" to "AUTO". Therefore, valid numeric values for the TimeChangeDetails during Cabinet Flash cannot be supplied by the controller. Thus, the values for TimeChangeDetails are undefined or unknown.

Both Monitor Flash and Local Flash are considered to be Cabinet Flash. Cabinet flash is a flash condition initiated and terminated by a source external to the controller. Fault Monitor Flash is initiated by the controller, but terminated by a source external to the controller.

During Cabinet Flash, the signal indications are determined by cabinet wiring (jumpers or flash program blocks) rather than controller software, so the controller may not have intrinsic knowledge of the signal indications (i.e., the controller may not know the movement phase states).

- If the cabinet architecture allows the controller to monitor the real-time voltage and current measurements, and the controller can determine the colors (Yellow, Red, or Dark) of the signal indications during Cabinet Flash, then the controller can provide the values of the Movement Phase States.
- If the controller is unable to monitor the real-time voltage and current measurements, the controller will provide a movement phase state of unavailable for all movements.

Certain cabinet architectures such as NEMA TS 1 do not define Flash Sense inputs to the controller. Deployers using these cabinets are cautioned that, without special accommodations, controllers running in these cabinets do not know that Cabinet Flash is active and may continue to cycle normally and erroneously provide normal time change details and movement phase states to OBUs/MUs during Cabinet Flash. Thus, controllers that cannot determine if Cabinet Flash is active are PROHIBITED from providing any SPaT data to the RSU.

Note: An intersection can be in either Cabinet Flash or Controller Flash, but not both.

Note: Some transportation agencies, as part of their operational policy or local laws or regulations, may not allow the controller to broadcast that an intersection is in failure flash. As noted in Section 2.7, operational policies, rules and regulations take precedence. If a transportation agency does not permit broadcast of Failure Flash, then all values for TimeChangeDetails shall be set to unknown.

F.3.3.4.4 Preempt Control

Represented by Bit 3 in spatStatus2, Preempt Control indicates to the receiver of this information that the controller is actively servicing a preemption request. The TimeChangeDetails for the intersection may suddenly change as the controller changes from "normal" operations to servicing a preemption request, which may require earlier end times for each subsequent interval as the controller quickly transitions to serve the movement requested.

In cases of an active Preemption, override values may be used to calculate the TimeChangeDetails values – such as preemptMinimumGreen or preemptEnterPedClear. The TimeChangeDetails for the intersection during the preempt clearance phases while the controller is servicing the preemption request are generally known and should be reflected in the SPaT data provided.

When the controller is in the dwell phase of the preemption service:

- the minimum time of change will be equal the current time plus the minimum green time (if in effect) or minimum dwell time remaining;
- once the minimum duration times for the dwell phase are satisfied, the minimum time of change will not be less than 0.1 seconds unless the green termination was commanded, and the field outputs change is imminent; and
- the value for maximum time of change will be the maximum dwell time (if available) or an appropriate value as timed by the controller until the controller determines when it will exit out of the dwell phase

The TimeChangeDetails for the intersection during the exit phases while the controller is servicing the preemption request are generally known and should be reflected in the SPaT data provided.

The movement phase states are known by the controller while servicing the preemption request, and should be reflected in the Movement Phase State.

If the controller is in a controller flash condition while servicing a preemption request, or is in a controller flash condition as a result of a preemption request, the Standby Operation bit (See F.3.3.4.8) will also be active.

F.3.3.4.5 Priority Control

Represented by Bit 4 in spatStatus2, Priority Control indicates to the receiver of this information that the controller is actively servicing a priority request and thus, the TimeChangeDetails for the intersection may suddenly change as the controller changes from "normal" operations to servicing a priority request as the controller transitions to serve the movement requested.

The TimeChangeDetails for the intersection while the controller is servicing the priority request are generally known and should be reflected in the SPaT data provided. The Movement Phase States also are known by the controller while servicing the priority request, and should be reflected in the SPaT data provided.

If the controller supports NTCIP 1211, priority active is enabled when the priorityStrategyRequestStatusInCO for any priority request is a value of requestQueued (2), or activeProcessing (4). The priority bit is not active when the request for service can be fulfilled without changes to the current signal timing because there is no change to the TimeChangeDetails.

F.3.3.4.6 Fixed Time Operation

Represented by Bit 5 in spatStatus2, Fixed Time Operation indicates to the receiver of this information that the controller is operating in fixed time and thus the values in TimeChangeDetails are not dynamically changing cycle by cycle. The TimeChangeDetails for the intersection while the controller is in fixed time operation are generally known and should be reflected in the SPaT data. The movement phase states are known by the controller during fixed time operation, and should be reflected in the SPaT data.

If the controller is in Manual Control, Stop Time or in Standby Operation, then Fixed Time Operation should be disabled.

F.3.3.4.7 Traffic Dependent Operation

Represented by Bit 6 in spatStatus2, Traffic Dependent Operation indicates to the receiver of this information that the controller is operating based on different levels of traffic parameters (requests, duration of gaps or more complex parameters), and not a fixed time mode.

The values for TimeChangeDetails for the intersection while the controller is in traffic dependent operation may change frequently as the traffic parameters (demand) changes and should be reflected in the SPaT data.

The movement phase states are known by the controller during traffic dependent operation, and should be reflected in the SPaT data.

Semi-actuated control, which is a legacy term, is considered traffic dependent operation (non-fixed time).

If the controller is in Manual Control, Stop Time or in StandbyOperation, then Traffic Dependent Operation should be disabled.

The controller cannot be in Fixed Time Operation and Traffic Dependent Operation simultaneously.

F.3.3.4.8 Standby Operation

Represented by Bit 7 in spatStatus2, StandbyOperation indicates that the controller is in a flashing condition but the controller will be aware of when the controller will exit the flashing condition. StandbyOperation indicates to the receiver of this information that the intersection is in a flash condition but the end of the flash condition is determinate at some point. StandbyOperation, also called soft flash, is initiated by the controller through the Signal Bus (load switches or switch packs), thus the signal indications (flashing red, yellow, or not at all/dark) are controlled by the controller, so the movement phase states are known by the controller and is reflected in the TimeChangeDetails.

Generally, there are three types of StandbyOperation:

- "Startup Flash" is a transitory interval timed immediately after the monitor transfers control to the controller. The duration of Startup Flash is controlled by the NTCIP unitStartupFlash object, which may be 0. The controller sets NTCIP unitFlashStatus to startup (7) during Startup Flash. The following TimeChangeDetails are deterministic during Startup Flash:
    - Minimum End Time: use NTCIP 1202 v03A unitStartupFlash.
    - Maximum End Time: use NTCIP 1202 v03A unitStartupFlash.

- "Automatic Flash" is commanded by NTCIP Pattern 255, either manually or on a scheduled basis. It is typically used as an operational strategy during off-peak or overnight times when steady (stop-and-go) operation is not warranted. The controller sets NTCIP unitFlashStatus to automatic (3) during Automatic Flash. The following time change intervals may be deterministic during Scheduled Flash:
    - Minimum End Time: use NTCIP 1202 v03A schedules (TOD schedule look ahead is optional per section)
    - Maximum End Time: use NTCIP 1202 v03A schedules (TOD schedule look ahead is optional per section)
- "Preempt Flash" occurs during a preempt's Dwell state if the preempt's NTCIP preemptControl Bit 3 (Flash Dwell) = 1. The controller sets NTCIP unitFlashStatus to preempt (8) during Preempt Flash. The following TimeChangeDetails may be deterministic during Preempt Flash:
    - Minimum End Time: use NTCIP 1202 v03A preemptDwellGreen
    - Maximum End Time: use NTCIP 1202 v03A preemptMaximumPresence

For Startup Flash and Automatic Flash, the controller enters and exits StandbyOperations in a controlled, deterministic manner and, as such, can provide TimeChangeDetails information for flash entry and flash exit intervals. For other StandbyOperations, the controller must satisfy specific conditions before it exits out of flash conditions. For example, for Preempt Flash, a train must clear the track crossing before the controller starts timing the preempt exit phases. Once the train exits the track crossing, the controller can deterministically provide the time it will exit out of flash and provide numeric values for TimeChangeDetails.

Note that when the controller flash is performed through the Flash Bus due to either controller software behavior (for example, by setting the Fault Monitor output to FALSE) or if the cabinet is specially wired (e.g., connecting a Flash output from the controller to the cabinet's Main Contactor), signal indications are determined by cabinet wiring (jumpers or flash program blocks) rather than controller software and the controller does not have intrinsic knowledge of the signal indications. See the discussion on Cabinet Flash in the Failure Flash section (Section F.3.3.4.3).

F.3.3.4.9 Failure Mode

Represented by Bit 8 in spatStatus2, Failure Mode indicates to the receiver of this information that the controller is not operating properly or has failed. Examples of a controller failure mode includes a process in the controller that is not be operating properly, a memory data error was detected, or an internal voltage failure was detected. Most failures will be fatal and would not result in this bit being SET, but for non-catastrophic failures, this bit can be set and the controller would likely activate cabinet flash.

F.3.3.4.10 Controller Off

Represented by Bit 9 in spatStatus2, this bit in the DE_IntersectionStatusObject representing the controller is OFF is not set by the controller, thus this bit is reserved. This bit is set by the CV Application Process to indicate it has not received valid SPaT data from the controller in the last 300 milliseconds.

F.3.3.4.11 Recent MAP Message Update

Represented by Bit 10 in spatStatus2, Recent MAP Message Update indicates that the MAP message transmitted by the RSU was recently updated. However, in practice in the DE_IntersectionStatusObject is not used by RLVW applications on OBUs. RLVW applications are expected to use the revision counter in the MAP message to determine if there are changes in the MAP messages. Since the controller is generally unaware of what MAP messages are being broadcast, this information is not set by the controller.

By default and consistent with CTI 4501, this bit is always a value of 1.

F.3.3.4.12 Recent Enabled Lanes Update

Represented by Bit 11 in spatStatus2, Recent Enabled Lanes Update indicates that the lane definitions or enabled lanes in the SPaT message transmitted by the RSU has recently been updated. However, in practice this bit is not used by RLVW applications on OBUs. RLVW applications are expected to use the revision counter in the SPaT and MAP messages to determine if there are changes in either message. There is no clear definition of "recently been updated" and although the controller may set this information, CTI 4501 v01.01 states, by default, that this bit is SET to a value of 1.

By default and consistent with CTI 4501, this bit is always a value of 1.

F.3.3.4.13 No MAP Available

Represented by Bit 12 in spatStatus2, No MAP Available indicates that the MAP message being broadcasted is unavailable or invalid. A MAP message can be broadcasted without a SPaT message, but the SPaT message must be sent with a valid MAP message describing the intersection associated with the SPaT message. Since the controller is generally unaware of what MAP messages are being broadcast by the CV Application Process, this information is generally SET by the CV Application Process.

Situations when a MAP may be invalid include:

- Temporary lane closures, when the contents of the MAP message broadcasted by the RSU may not be valid. This applies when contractor is setting up cones/barrels for lane closures, presence of flaggers, etc.
- The IOO decides not to generate MAP message, maybe for operational or maintenance purposes
- The MAP message doesn't have a valid IEEE Std 1609.2-2016 certificate

However, the first two situations above may require that field maintenance personnel also mark the SPaT data generated by the controller be invalid (See), and the controller is generally more accessible by the field maintenance personnel than the RSU (which is sealed for security purposes and may be mounted on a pole). Also, since the SPaT message is a dynamic message (while the MAP message is generally a static message, i.e., changes infrequently), and has a bit to indicate the MAP is invalid, the ASC WG agreed to add a requirement to mark the MAP message invalid via the traffic controller.

F.3.3.4.14 No SPaT Available

Represented by Bit 13 in spatStatus2, No SPaT Available indicates that the SPaT message broadcasted is unavailable or invalid. No SPaT Available indicates to the receiver of this information that the contents of the SAE SPaT message (DE_IntersectionStatusObject except for this bit, TimeChangeDetails, MovementPhaseState) should be ignored and not used by applications.

Situations when a SPaT message may be invalid include:

- The signalized intersection is under test and is not broadcasting a valid SPaT message
- Temporary lane closures, when the contents of the MAP message broadcasted by the RSU may not be valid. This applies when contractor is setting up cones/barrels for lane closures, presence of flaggers, etc.
- The CV Application Process has not received valid SPaT data from the controller within the last 300 milliseconds
- A corresponding valid MAP message is not being broadcasted (the SPaT message must be sent with a valid MAP message describing the intersection associated with the SPaT message)

spatOptions allows a management station to indicate if the SPaT message is valid or invalid. For the latter two bullets above, since the controller is generally unaware of what MAP messages are being broadcast by the CV Application Process, or if the SPaT data provided by the controller is received by the CV Application Process, this bit would be set by the CV Application Process.

F.3.3.5 Movement Phase State

This section provides additional guidance beyond the description in SAE J2735 for providing the movement phase state of each allowed movement at a signalized intersection.

DE_MovementPhaseState is a required data element when broadcasting a SAE J2735 SPaT message. DE_MovementPhaseState describes the movement state (both current and future) for a signalGroupID. The signalGroupID maps a group of common movements, defined as a connection between an ingress lane to an egress lane in the MAP message, at a signalized intersection to a movement state. The controller then provides the movement phase state and the TimeChangeDetails for each signalGroupID.

The controller uses the signalState2 object to provide the movement phase state for each signalGroupID to the CV Application Process. CTI 4501 requires that the controller provide the movement phase state for the current and the next interval. This is accomplished by the movementEventNumber, where movementEventNumber.1 represents the current interval and movementEventNumber.2 represents the next interval. See Sections 3.6.3.1 and 3.6.3.2 for the performance requirements to send signalState2 to the CV Application Process.

SAE J2735 describes signal state data elements and MAP data so that external roadway users can understand the current and future state signalization of the roadway. This status is provided in a simplified format using MovementPhaseState and TimeChangeDetails to represent the roadway indications. This mapping implies an assumption that the traffic controller can map its perceived signal timing into the signal indications (Signal Groups) that are presented to the roadway users.

In real-world applications, this is not always the case. Overlaps, Controller I/O processing, In-Cabinet output panels, and other possible esoteric applications can have impact to override the actual signal outputs that are displayed to the roadway users to control their movements on associated Lane IDs. Given the critical nature of SPaT messages for real-time safely applications, intersections that generate SPaT data must remove any in-cabinet output panels, controller output logic processing, or other post-processed means of output override that would render the SPaT data inaccurate relative to the signal group displays above the roadway. In cases where overlaps are mapped to control the signal outputs, it is the responsibility of the traffic controller to generate SPaT consistent to the overlap and/or phase states that are actively controlling the signal group. This output assignment often occurs on a time of day and/or preemptive basis and cannot easily be performed outside the traffic controller itself. If the traffic controller is both independently controlling phase timing as well as generating the SPaT message, it must consider all internally known details that will affect the controller-level outputs prior to establishing these TimeChangeDetails. A cabinet with external logic cannot send a SPaT.

F.3.3.5.1 signalGroupEntryTable

The signalGroupEntryTable is to be completed by a signal timing engineer to represent the signal phasing of an intersection for each movement at an intersection. Each movement at an intersection (signalGroupIntersection) is identified by a signalGroupID number (signalGroupID). signalGroupControlSource and signalGroupControlType generally define the vehicle phase, pedestrian phase or overlap number associated with the signalGroupID.

signalGroupGreenType defines the type of movement allowed – whether the movement is protected, permissive, or protected-permissive. For vehicle movements, steady green arrows (left, thru or right) are considered protected movements, and all green balls are represented as permissive movements. For pedestrian movements, if there is no conflicts with any movement other than another pedestrian movement during a pedestrian WALK interval, then the signalGroupGreenType is a protected movement, otherwise it is a permissive movement. preMovement informs the traveler that a permitted (protected or permissive) movement is imminent, such as a light rail vehicle, but otherwise is not common in the US.

signalGroupRedType defines the type of movement during red signal indications – whether the traveler is to stop and remain (at the stop bar for vehicles), or can stop then proceed (such as for flashing red indications or right turn on red).

For movements that can be protected or permissive, the signalGroupPermissiveControlSource and signalGroupPermissiveControlType defines the phase or overlap that determines if the movement is protected or permissive. Otherwise, these objects can be ignored.

The signalGroupLane defines the enabled lanes that determines if the signalGroupID for that row is active. If the lane numbers specified are enabled, then the signalGroupID for that row is active. If no lanes are specified, then the signalGroupID for that row is always active.

Figure 14 represents an example intersection depicting the phases (Ph#) and overlaps (OLn), the lane numbers, and the signalGroupID. Lane numbers within a box are revocable lanes (e.g., lanes 4 and 14).

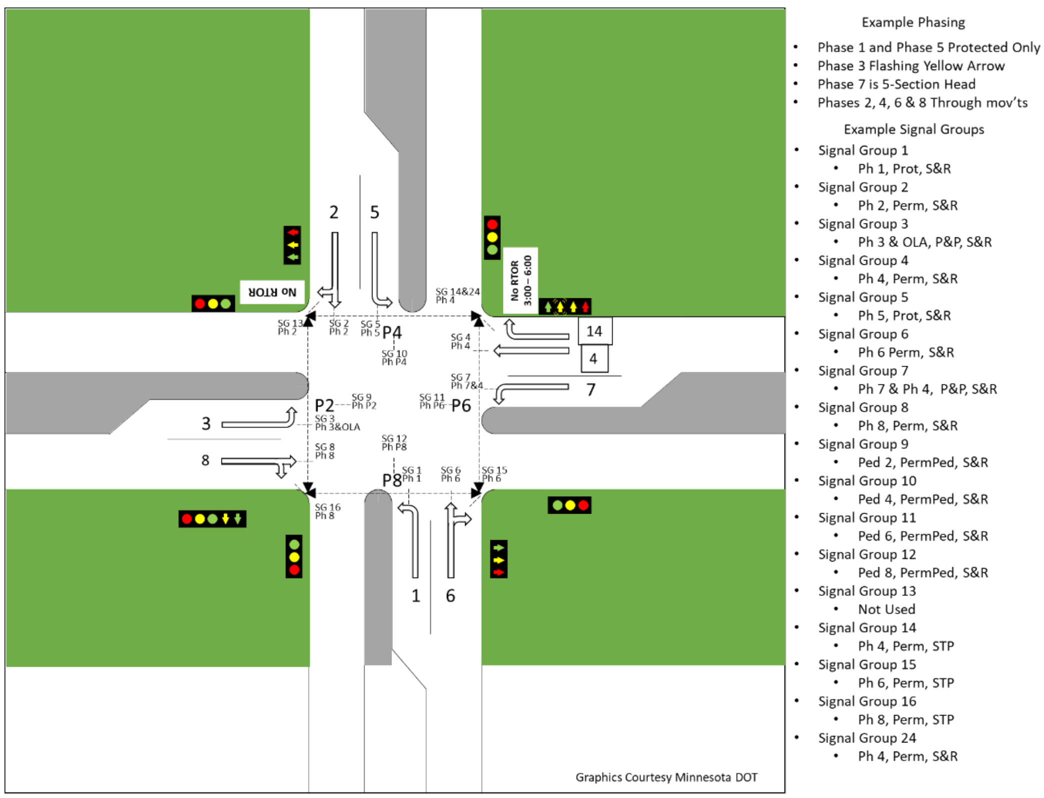

Figure 14 Example Intersection Diagram

Figure 15 represents how the signalGroupEntryTable for the signalized intersection in Figure 14 would be completed. In the example intersection, whether Signal Group 3 is a protected-permissive green is determined by Vehicle Phase 3 plus the 4-section Flashing Yellow Arrow Overlap A controlled left turn movement, while whether Signal Group 7 is a protected-permissive green is determined by Vehicle Phase 7 and Vehicle Phase 4 for the 5-section controlled left turn movement.

Similarly, whether Signal Group 9 and Signal Group 11 are protected-permissive walk movements are determined by Pedestrian Phase 2 and Pedestrian Phase 6 and Vehicle Phase 2 and Vehicle Phase 6 respectively. Note: It is assumed that if the vehicle phase is red when the associated pedestrian phase is walk, then the pedestrian phase is either a leading pedestrian interval or a "Barnes dance" pedestrian interval, not in conflict with any vehicle green or yellow intervals.

signalGroupIDs 1 to 8 represent either through movements or left turn movements for Vehicle Phases 1 to 8 – thus, the signalGroupRedType is Stop And Remain for those vehicle phases when the signal indication for that movement is red. signalGroupIDs 14 to 16 represents the right turn movements for Vehicle Phases 4, 6, and 8, but the signalGroupRedType for these signalGroupIDs is Stop Then Proceed indicating that right turn on red is permitted. However, note that signalGroupID 13 for the right turn movement for Vehicle Phase 2 is unnecessary because right-turn-on-red is prohibited and Stop And Remain is already addressed in signalGroupID 2.

However, the Right Turn on Red by time-of-day for Vehicle Phase 4 is determined by which revocable lane set (1 or 2 – See Table 10) is enabled. A right-turn-on-red scenario is presented in more detail below.

signalGroupEntryNumber	signalGroupIntersection	signalGroupID	signalGroupControlSource	signalGroupControlType	signalGroupGreenType	signalGroupRedType	signalGroupPermissiveControlSource	signalGroupPermissiveControlType	signalGroupLane
1	1	1	1	phaseVehicle (2)	protectedMovementAllowed (4)	StopAndRemain (3)	0	none (2)	0
2	1	2	2	phaseVehicle (2)	permissiveMovementAllowed (3)	StopAndRemain (3)	0	none (2)	0
3	1	3	3	phaseVehicle (2)	protectedPermissive (5)	StopAndRemain (3)	1	overlap (5)	0
4	1	4	4	phaseVehicle (2)	permissiveMovementAllowed (3)	StopAndRemain (3)	0	none (2)	0
5	1	5	5	phaseVehicle (2)	protectedMovementAllowed (4)	StopAndRemain (3)	0	none (2)	0
6	1	6	6	phaseVehicle (2)	permissiveMovementAllowed (3)	StopAndRemain (3)	0	none (2)	0
7	1	7	7	phaseVehicle (2)	protectedPermissive (5)	StopAndRemain (3)	4	phaseVehicle (3)	0
8	1	8	8	phaseVehicle (2)	permissiveMovementAllowed (3)	StopAndRemain (3)	0	none (2)	0
9	1	9	2	phasePedestrian (3)	protectedPermissive (5)	StopAndRemain (3)	2	phaseVehicle (3)	0
10	1	10	4	phasePedestrian (3)	permissiveMovementAllowed (3)	StopAndRemain (3)	0	none (2)	0
11	1	11	6	phasePedestrian (3)	protectedPermissive (5)	StopAndRemain (3)	6	phaseVehicle (3)	0
12	1	12	8	phasePedestrian (3)	permissiveMovementAllowed (3)	StopAndRemain (3)	0	none (2)	0
13	1	13							
14	1	14	4	phaseVehicle (2)	permissiveMovementAllowed (3)	stopThenProceed (2)	0	none (2)	1
15	1	15	6	phaseVehicle (2)	permissiveMovementAllowed (3)	stopThenProceed (2)	0	none (2)	0
16	1	16	8	phaseVehicle (2)	permissiveMovementAllowed (3)	stopThenProceed (2)	0	none (2)	0
17	1	24	4	phaseVehicle (2)	permissiveMovementAllowed (3)	StopAndRemain (3)	0	none (2)	2

Figure 15 Example signalGroupEntryTable

F.3.3.5.2 agencySignalStateTable

The agencySignalStateTable is a lookup table to determine the movement phase state (both current for the next interval) for each signalGroupID defined for the signalized intersections. The agencySignalStateTable has default values configured but is customizable. While the movement phase state typically depends on what color a signal is, the state may also depend on whether the signal is for a vehicle or pedestrian, whether turns on red are permitted, and whether the movement is protected or permissive (which may be driven by another source in the controller's output).

Using the entries in the signalGroupEntryTable, the current and future movementPhaseState for each signalGroupID is determined by:

- **agencySignalStateColor.** The current signal indication color for the signalGroupID defined by the signalGroupControlType/signalGroupControlSource.
- **agencySignalStateControlType.** The control type (vehicle or pedestrian) for the signalGroupID. Vehicle, bicycle, and transit phases and overlaps are considered a vehicle control type. Pedestrian phases and pedestrian overlaps are considered a pedestrian control type.
- **agencySignalStateOptions.** This object contains 3 pieces of information.
    - Bits 0 – 1. The signalGroupRedType for the signalGroupID. It is either stopAndRemain, stopThenProceed or not applicable.
    - Bit 2 is reserved for preMovement green, although it is not used in the United States in practice.
    - Bits 3 – 4. The signalGroupGreenType for the signalGroupID. When the movement is allowed, it is either a permissive movement, a protected movement, could be permissive or protected, or not applicable.
    - Bits 5 – 9. The current signal indication color for the signalGroupPermissiveControlType / signalGroupPermissiveControlSource of the signalGroupID. If all Bits are FALSE, then the signal indication color is not applicable. If more than 1 bit is TRUE, then either signal indication color is applicable (e.g., if Bits 7 and 8 are TRUE, then either Flashing Yellow or Green).
- **agencySignalStateValue.** The valid movement phase state for the signalGroupID. The valid movement phase states are defined in SAE J2735 DE_MovementPhaseState.

The values in Figure 15 are the default values for the agencySignalStateTable. If the controller provides features that are not compatible with the logic in this lookup table the controller vendor must provide lookup logic such that the SPaT data reflects the movement phase state conveyed by the physical signal indications in the field given the vendor supported features. An agency may also add, delete or change row entries as appropriate depending on their agency-specific rules, regulations and standards.

Row	agencySignalStateColor	agencySignalStateControlType	signalGroupGreenType agencySignalStateOptions Bits 2 - 4	signalGroupPermissiveControlColor agencySignalStateOptions Bits 5- 9	signalGroupRedType agencySignalStateOptions Bits 0 - 1	agencySignalStateValue
1	green (4)	vehicle (2)	Bit 4 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	protectedMovementAllowed (8)
2	green (4)	vehicle (2)	Bit 3 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	permissiveMovementAllowed (7)
3	green (4)	vehicle (2)	Bits 3 & 4 = TRUE	Bits 5-9 = TRUE	Bits 0-1 = FALSE	protectedMovementAllowed (8)
4	green (4)	pedestrian (3)	Bit 4 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	protectedMovementAllowed (8)
5	green (4)	pedestrian (3)	Bit 3 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	permissiveMovementAllowed (7)
6	green (4)	pedestrian (3)	Bits 3 & 4 = TRUE	Bit 5 = TRUE	Bits 0-1 = FALSE	protectedMovementAllowed (8)
7	green (4)	pedestrian (3)	Bits 3 & 4 = TRUE	Bit 7 = TRUE	Bits 0-1 = FALSE	permissiveMovementAllowed (7)
8	green (4)	pedestrian (3)	Bits 3 & 4 = TRUE	Bit 6 = TRUE	Bits 0-1 = FALSE	permissiveMovementAllowed (7)
9	yellow (3)	vehicle (2)	Bit 4 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	protectedClearance (10)
10	yellow (3)	vehicle (2)	Bit 3 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	permissiveClearance (9)
11	yellow (3)	vehicle (2)	Bits 3 & 4 = TRUE	Bits 5-9 = TRUE	Bits 0-1 = FALSE	protectedClearance (10)
12	red (2)	vehicle (2)	Bit 4 = TRUE	Bits 5-9 = FALSE	Bit 0 = TRUE	stopThenProceed (4)
13	red (2)	vehicle (2)	Bit 4 = TRUE	Bits 5-9 = FALSE	Bit 1 = TRUE	stopAndRemain (5)
14	red (2)	vehicle (2)	Bit 3 = TRUE	Bits 5-9 = FALSE	Bit 0 = TRUE	stopThenProceed (4)
15	red (2)	vehicle (2)	Bit 3 = TRUE	Bits 5-9 = FALSE	Bit 1 = TRUE	stopAndRemain (5)
16	red (2)	vehicle (2)	Bits 3 & 4 = TRUE	Bits 7-8 = TRUE	Bits 0-1 = FALSE	permissiveMovementAllowed (7)
17	red (2)	vehicle (2)	Bits 3 & 4 = TRUE	Bit 6 = TRUE	Bits 0-1 = FALSE	permissiveClearance (9)
18	red (2)	vehicle (2)	Bits 3 & 4 = TRUE	Bit 5 = TRUE	Bit 0 = TRUE	stopThenProceed (4)
19	red (2)	vehicle (2)	Bits 3 & 4 = TRUE	Bit 5 = TRUE	Bit 1 = TRUE	stopAndRemain (5)
20	red (2)	vehicle (2)	Bits 3 & 4 = TRUE	Bit 9 = TRUE	Bits 0-1 = FALSE	stopThenProceed (4)
21	dark (5)	vehicle (2)	Bits 3 & 4 = TRUE	Bits 7-8 = TRUE	Bits 0-1 = FALSE	permissiveMovementAllowed (7)
22	dark (5)	vehicle (2)	Bits 3 & 4 = TRUE	Bit 6 = TRUE	Bits 0-1 = FALSE	permissiveClearance (9)
23	dark (5)	vehicle (2)	Bits 3 & 4 = TRUE	Bit 5 = TRUE	Bit 0 = TRUE	stopThenProceed (4)
24	dark (5)	vehicle (2)	Bits 3 & 4 = TRUE	Bit 5 = TRUE	Bit 1 = TRUE	stopAndRemain (5)
25	dark (5)	vehicle (2)	Bits 3 & 4 = TRUE	Bit 9 = TRUE	Bits 0-1 = FALSE	stopThenProceed (4)
26	red (2)	pedestrian (3)	Bit 4 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	stopAndRemain (5)
27	red (2)	pedestrian (3)	Bit 3 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	stopAndRemain (5)
28	red (2)	pedestrian (3)	Bits 3 & 4 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	stopAndRemain (5)
29	flashingYellow (7)	vehicle (2)	Bits 3 & 4 = FALSE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	cautionConflictingTraffic (11)
30	flashingRed (6)	vehicle (2)	Bits 3 & 4 = FALSE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	stopThenProceed (4)
31	flashingRed (6)	pedestrian (3)	Bit 4 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	protectedClearance (10)
32	flashingRed (6)	pedestrian (3)	Bit 3 = TRUE	Bits 5-9 = FALSE	Bits 0-1 = FALSE	permissiveClearance (9)
33	flashingRed (6)	pedestrian (3)	Bits 3 & 4 = TRUE	Bit 5 = TRUE	Bits 0-1 = FALSE	protectedClearance (10)
34	flashingRed (6)	pedestrian (3)	Bits 3 & 4 = TRUE	Bit 7 = TRUE	Bits 0-1 = FALSE	permissiveClearance (9)
35	flashingRed (6)	pedestrian (3)	Bits 3 & 4 = TRUE	Bit 6 = TRUE	Bits 0-1 = FALSE	permissiveClearance (9)

Figure 16 Example agencySignalStateTable

Using the example signalGroupEntryTable in Figure 15, below are some sample signalState2 values:

- For signalGroupEntryNumber 1 (Intersection 1, signalGroupID 1), when phaseVehicle 1 is Green, then Row 1 in the agencySignalStateTable applies because the signalGroupGreenType is protected. So the value for signalState2 for signalGroupEntryNumber 1 is protectedMovementAllowed.
- For signalGroupEntryNumber 3 (Intersection 1, signalGroupID 3), when phaseVehicle 3 is Green and Overlap A is On, then Row 3 in the agencySignalStateTable applies because the signalGroupGreenType is protectedPermissive. So the value for signalState2 for signalGroupEntryNumber 3 is protectedMovementAllowed.
- For signalGroupEntryNumber 9 (Intersection 1, signalGroupID 9), when phasePedestrian 2 is Walk and phaseVehicle 2 is Red, then Row 6 in the agencySignalStateTable applies because the signalGroupGreenType is protectedPermissive. So the value for signalState2 for signalGroupEntryNumber 9 is protectedMovementAllowed. This example may be a leading pedestrian interval.
- For signalGroupEntryNumber 24 (Intersection 1, signalGroupID 24), when phaseVehicle 4 is Red and Lane 14 is enabled, then Row 15 in the agencySignalStateTable applies because the signalGroupGreenType is permissive and signalGroupRedType is stopAndRemain. So the value for signalState2 for signalGroupEntryNumber 24 is stopAndRemain. This example may be a right-turn-on-red by time of day.

In addition, the following guidance is reprinted from CTI 4501, Connected Intersections (CI) Implementation Guide.

- **Flashing Yellow Arrow Permissive Movement.** At an intersection that uses a flashing yellow arrow to control a permissive movement as part of a protected-permissive turn, a connected intersection shall use the value of permissive-Movement-Allowed as the current movement state for the signal group when the flashing yellow arrow is active, as defined by DE_MovementPhaseState in SAE J2735_202007.
- **Protected and Permissive Clearance.** A connected intersection shall use the value of protected-clearance or permissive-Clearance as the current movement state of a signal group to correspond with the protected or permissive condition of the allowed movement immediately preceding the current (clearance) interval, as defined by DE_MovementPhaseState in SAE J2735_202007.
- **Conflict Causes Permissive.** When any allowed movement controlled by a signal group includes a maneuver in conflict with any other movement that is in a permitted or clearance state, a connected intersection shall use permissive-Movement-Allowed or permissive-Clearance as the current movement state for the signal group, as defined by DE_MovementPhaseState in SAE J2735_202007. Examples include a green ball with an opposing green ball, and a green ball with a permitted pedestrian movement for a turn.
- **No Conflict Causes Protected.** When an allowed movement controlled by a signal group includes no maneuver in conflict with any other vehicle, pedestrian, or bicycle movement that is in a permitted or clearance state, a connected intersection shall use protected-Movement-Allowed or protected-clearance as the current movement state for the signal group, as defined by DE_MovementPhaseState in SAE J2735_202007.
- **WALK State Enumeration (No Conflict).** When an allowed pedestrian movement has no conflict with a vehicle movement controlled by a signal group that is in a permitted or clearance state, a connected intersection shall use protected-Movement-Allowed as the current movement state for the pedestrian WALK interval, as defined by DE_MovementPhaseState in SAE J2735_202007. Examples of a WALK state with no conflict include leading pedestrian intervals or exclusive pedestrian interval (Barnes Dance).
- **WALK State Enumeration (Potential Conflict).** When an allowed pedestrian movement is in conflict with a vehicle movement controlled by a signal group that is in a permitted or clearance state, a connected intersection shall use permissive-Movement-Allowed as the current movement state for the pedestrian WALK interval, as defined by DE_MovementPhaseState in SAE J2735_202007. Examples of a WALK state with potential conflicts include vehicles turning right on green across the pedestrian crosswalk in a WALK state.
- **Flashing DON'T WALK State Enumeration.** A connected intersection shall use the protected-clearance or permissive-Clearance as the current movement state for the pedestrian Flashing DON'T WALK interval, as defined by DE_MovementPhaseState in SAE J2735_202007, to correspond with the protected or permissive condition of the allowed WALK movement immediately preceding the current (clearance) interval.
- **Steady DON'T WALK State Enumeration.** A connected intersection shall use stop-And-Remain as the current movement state for the pedestrian Steady DON'T WALK interval, as defined by DE_MovementPhaseState in SAE J2735_202007.

The ASC WG agreed that steady green arrows (left, thru or right) are represented in DE_MovementPhaseState as protected-Movement-Allowed; and green balls are represented as permissive-Movement-Allowed.

F.3.3.5.3 Editing agencySignalStateTable

This section provides details on how to edit the agencySignalStateTable. As noted in Annex F.3.3.5.2, the agencySignalStateTable is expected to be a static table with default values but can be customized for each agency or intersection. It is expected that the controller vendor will create and populate the agencySignalStateTable based on the operating agency's operations and maintenance procedures and policies.

The agencySignalStateColor object is used to identify the current or next (interval) color of the primary source that drives the signal, which may be Red, Yellow, Green, Dark, Flashing Red, or Flashing Yellow. The agencySignalStateType object allows the movement to be specified as a vehicle or pedestrian.

The agencySignalStateOptions object allows multiple properties to be configured for the signal group. The object is an integer with each bit in its binary representation representing different properties for the signal group. Combinations of different properties may result in different movement phase states for the signal group. The following properties are configurable.

- The Red indication may be configured as stopThenProceed or stopAndRemain. This is done using bits 0 and 1 of the agencySignalStateOptions object.
- The Green indication may be configured as preMovement (not used in the United States), protectedMovementAllowed or permissiveMovementAllowed. This is done using bits 2 - 4 of the agencySignalStateOptions object. If the movement may be protected or permissive, then both bits 3 and 4 are asserted (TRUE).
- The movement phase state for a signal group may depend on the movement phase state of another source (such as another phase or overlap). Bits 5-9 allow the user to specify the signal indication of that other source to determine the movement phase state of the signal group. Bit 5 is asserted (TRUE) if the signal indication of the other source is red. Bits 7 and 8 are asserted (TRUE) if the signal indication of the other source is either Green or Flashing Yellow. If Bits 5 through 9 are asserted, then the signal indication of the other source can be any, while if none of Bits 5 through 9 are asserted, then the signal indication of the source is not applicable.

Figure 17 shows a four-section signal with a Flashing Yellow Arrow and a green arrow for a left turn from Lane 3. This left turn is a protected-permissive movement, the movement is protected during some portions of the cycle and permissive during other portions of the cycle. The green arrow is controlled by Phase 3 and Overlap A controls the flashing-yellow-arrow, the yellow-arrow, and the red-arrow.

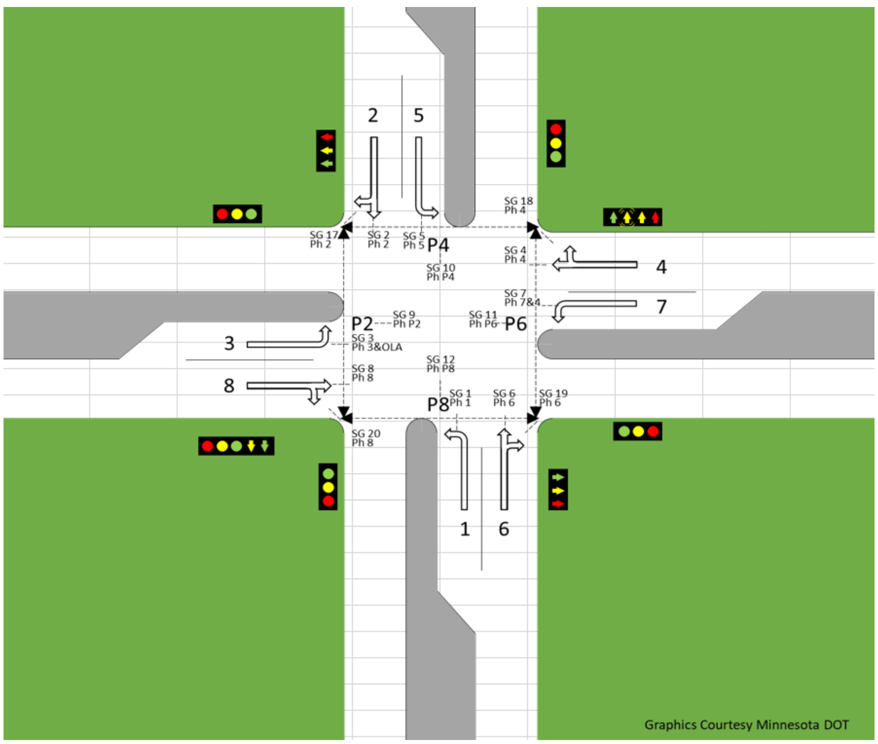

Figure 17 Example Intersection Diagram 2

The row in Table 8 is the details for signalGroupID 3 as copied from the table in Figure 14. The left turn movement is represented as signalGroupID 3, controlled by vehicle phase 3, with a signalGroupGreenType of protectedPermissive, with its permissive controlled by Overlap A (represented as 1), and is of signalGroupRedType Stop And Remain.

Table 8 signalGroupEntryTable Example

signal GroupID	signalGroup ControlSource	signalGroup ControlType	signalGroup GreenType	signalGroup RedType	signalGroupPermissive ControlSource	signalGroup Permissive ControlType
3	3	phaseVehicle(2)	protectedPermissive (5)	stopAndRemain(3)	A	overlap(5)

Table 9 agencySignalStateTable Excerpt

Row	agencySignal StateColor	agencySignal StateControlType	agencySignal StateOptions	agencySignalStateValue
3	green(4)	vehicle(2)	1016 Bits 3 – 9 are TRUE	protectedMovementAllowed (8)
11	yellow(3)	vehicle(2)	1016 Bits 3 – 9 are TRUE	protectedClearance (10)
16	red(2)	vehicle(2)	408 Bits 3, 4, 7 & 8 are TRUE	permissiveMovementAllowed (7)
17	red(2)	vehicle(2)	88 Bits 3, 4 & 6 are True	permissiveClearance (9)
19	red(2)	vehicle(2)	56 Bits 3, 4 & 5 is True	stopAndRemain (5)

When vehicle phase 3 is Green, overlap A will be dark, and the signal indication will be showing a solid Green Arrow. Using the agencySignalStateTable in Figure 16, this state is represented by row 3 because the agencySignalStateColor is Green, the agencySignalControlType is vehicle, and agencySignalStateOption is a value of 1016 because the signalGroupGreenType is protectedPermissive (Bits 3 and 4 are TRUE), and the signalGroupPermissiveControlColor is Any (Bits 5 – 9 are TRUE), and signalGroupRedType is not applicable (Bits 0 – 1 are FALSE). Thus the agencySignalStateValue for Signal Group ID 3 is protectedMovementAllowed (See the first row in Table 9).

When vehicle phase 3 is Yellow, overlap A is Yellow, and the signal indication is showing a solid Yellow Arrow. Using the agencySignalStateTable in Figure 16, this state is represented by row 11 because the agencySignalStateColor is Yellow, the agencySignalControlType is vehicle, and agencySignalStateOption is a value of 1016 because the signalGroupGreenType is protectedPermissive (Bits 3 and 4 are TRUE), and the signalGroupPermissiveControlColor is Any (Bits 5 – 9 are TRUE), and signalGroupRedType is not applicable (Bits 0 – 1 are FALSE). Thus the agencySignalStateValue for Signal Group ID 3 is protectedClearance (See the second row in Table 9).

When vehicle phase 3 is Red, overlap A is flashing-yellow-arrow, and the signal indication is showing a Flashing Yellow Arrow. Using the agencySignalStateTable in Figure 16, this state is represented by row 16 because the agencySignalStateColor is Red, the agencySignalControlType is vehicle, and agencySignalStateOption is a value of 408 because the signalGroupGreenType is protectedPermissive (Bits 3 and 4 are TRUE), and the signalGroupPermissiveControlColor is Green or Flashing Yellow (Bits 7 – 8 are TRUE), and signalGroupRedType is not applicable (Bits 0 – 1 are FALSE). Thus the agencySignalStateValue for Signal Group ID 3 is permissiveMovementAllowed (See the third row in Table 9).

When vehicle phase 3 is Red, overlap A is Yellow, and the signal indication is showing a Solid Yellow Arrow. Using the agencySignalStateTable in Figure 16, this state is represented by row 17 because the agencySignalStateColor is Red, the agencySignalControlType is vehicle, and agencySignalStateOption is a value of 88 because the signalGroupGreenType is protectedPermissive (Bits 3 and 4 are TRUE), and the signalGroupPermissiveControlColor is Yellow (Bit 6 is TRUE), and signalGroupRedType is not applicable (Bits 0 – 1 are FALSE). Thus the agencySignalStateValue for Signal Group ID 3 is permissiveClearance (See the fourth row in Table 9).

When vehicle phase 3 is Red, overlap A is Red, and the signal indication is showing a Solid Red Arrow. Using the agencySignalStateTable in Figure 16, this state is represented by row 19 because the agencySignalStateColor is Red, the agencySignalControlType is vehicle, and agencySignalStateOption is a value of 56 because the signalGroupGreenType is protectedPermissive (Bits 3 and 4 are TRUE), and the signalGroupPermissiveControlColor is Red (Bit 5 is TRUE), and signalGroupRedType is not applicable (Bits 0 – 1 are FALSE). Thus the agencySignalStateValue for Signal Group ID 3 is stopAndRemain (See the fifth row in Table 9).

F.3.3.5.4 Right Turn On Red

This section provides guidance for right-turn-on-red (RTOR). The same guidance applies for left-turn-on-red but for simplicity, only RTOR scenarios are presented.

There have been inconsistencies by implementations on whether to use stop-Then-Proceed (2) or stop-And-Remain (3) for DE_MovementPhaseState in a SAE J2735 SPaT message for a right turn movement at signalized intersections where RTOR is allowed. There is a need that IOOs represent RTOR scenarios when a red ball signal indication is present in a nationally interoperable manner so vehicle applications can properly address RTOR scenarios. NTCIP 1202 v03B adds support for how RTOR in a connected vehicle environment should be handled a Time-of-day scenario. In the future, NTCIP 1202 will support RTOR in a connected vehicle environment with preemption scenarios.

MUTCD contains the following for a CIRCULAR RED signal indication as follows:

> Except when a traffic control device is in place prohibiting a turn on red or a steady RED ARROW signal indication is displayed, vehicular traffic facing a steady CIRCULAR RED signal indication is permitted to enter the intersection to turn right, or to turn left from a one-way street into a one-way street, after stopping. The right to proceed with the turn shall be subject to the rules applicable after making a stop at a STOP sign[1].

The application of DE_MovementPhaseState in SAE J2735 is unclear with respect to RTOR:

- stop-Then-Proceed (2),
  - Often called 'flashing red' in US
  - Driver Action:
  - Stop vehicle at stop line.
  - Do not proceed unless it is safe.
  - Note that the right to proceed either right or left when it is safe may be contained in the lane description to handle what is called a 'right on red'
- stop-And-Remain (3),
  - e.g., called 'red light' in US
  - Driver Action:
  - Stop vehicle at stop line.
  - Do not proceed.
  - Note that the right to proceed either right or left when it is safe may be contained in the lane description to handle what is called a 'right on red'

There are several assumptions in a connected vehicle environment for RTOR.

1. A vehicle application intending to make a right turn at the intersection would process the SPaT message, determine what the DE_MovementPhaseState is for the right turn and what enabled lanes are active.

[1] Manual on Uniform Traffic Control Devices (MUTCD), 2009, Section 4D.04.C.1

2. A vehicle application intending to make a right turn at the intersection would process the MAP message and determine if the lane (that the vehicle is currently in) allows a RTOR movement.
3. A valid MAP message is broadcasted with a valid SPaT message at a connected intersection.

The following are the requirements for RTOR in a connected vehicle environment.

1. The MAP message for a connected intersection shall indicate if RTOR is permitted for a lane.
    a. If RTOR is permitted for the lane, both maneuverRightAllowed AND maneuverRightTurnOnRedAllowed bits are TRUE in DE_AllowedManeuvers for that lane and for that movement to indicate that right turn and RTOR are allowed.
    b. If RTOR is NOT permitted for the lane, the maneuverRightAllowed bit is TRUE but the maneuverRightTurnOnRedAllowed bit is FALSE in DE_AllowedManeuvers for that lane and for that movement to indicate RTOR is not allowed.
    c. If whether RTOR is permitted for a lane at the connected intersection varies, two revocable lanes shall be defined for the same physical lane in the MAP message – one revocable lane indicating RTOR is permitted, and one RTOR indicating RTOR is not permitted.
2. The SPaT message for a connected intersection shall indicate if RTOR is permitted for a lane.
    a. If RTOR is permitted for a lane, the SPaT message shall indicate stop-Then-Proceed for DE_MovementPhaseState for that right turn movement.
    b. If RTOR is not permitted for a lane, the SPaT message shall indicate stop-And-Remain for DE_MovementPhaseState for that right turn movement.
    c. If whether RTOR is permitted for a lane at the connected intersection varies, the SPaT message shall indicate which revocable lane, as defined in the MAP message, is enabled. Only the signal group associated with the right turn for that revocable lane, with a stop-Then-Proceed or stop-And-Remain state as appropriate for the current RTOR effective condition, is included in the SPaT message.
    d. The requirements above shall be fulfilled even if whether RTOR is permitted varies by time-of-day/day-of-week.

To fulfill Requirements 1, 1a, and 1b above, the MAP message shall be used to indicate that whether RTOR is permitted.

In Figure 14, RTOR is allowed in Lanes 6 and 8, RTOR is NOT allowed in Lane 2, and whether RTOR is allowed is dependent on the time-of-day in Lanes 4/14.

The SAE J2735 MAP message uses DE_AllowedManeuvers to indicate when a RTOR scenario exists. The individual bits in the bitstring of DE_AllowedManeuvers indicates what turns are allowed for each lane at an intersection. Using Figure 14 as a reference, Table 10 shows how the MAP message should be used to indicate that RTOR is allowed – Lane indicates the lane number, AllowedManeuvers indicate what maneuvers are allowed for that lane and for that movement, and Description is a textual description of the movement at the intersection.

Note that DE_AllowedManeuvers is provided for each lane and for each movement (lane connection) for the lane. For the lane DE_AllowedManeuvers lists all maneuvers that can be made at the stop line from that lane. For the connection, DE_AllowedManevuers only includes the maneuver allowed in that connection. The signalGroupID is associated with the connection.

Table 10 Example Lane Attributes

Lane	DE_AllowedManeuvers (MAP)	Description
1	bit (1) maneuverLeftAllowed	NB Protected Left
2	bit (0, 2) maneuverStraightAllowed, maneuverRightAllowed	SB Thru SB Permissive Right (RTOR Not Allowed)
3	bit (1) maneuverLeftAllowed	EB Protected Left

Lane	DE_AllowedManeuvers (MAP)	Description
4	bit (0, 2, 5) maneuverStraightAllowed, maneuverRightAllowed, maneuverRightTurnOnRedAllowed	WB Thru WB Permissive Right (RTOR Allowed)
5	bit (1) maneuverLeftAllowed	SB Protected Left
6	bit (0, 2, 5) maneuverStraightAllowed, maneuverRightAllowed, maneuverRightTurnOnRedAllowed	EB Thru EB Permissive Right (RTOR Allowed)
7	bit (1) maneuverLeftAllowed	WB Protected Left
8	bit (0, 2, 5) maneuverStraightAllowed, maneuverRightAllowed, maneuverRightTurnOnRedAllowed	EB Thru EB Permissive Right (RTOR Allowed)
14	bit (0, 2) maneuverStraightAllowed, maneuverRightAllowed	WB Thru WB Permissive Right (RTOR Not Allowed)

Both bits, maneuverRightAllowed AND maneuverRightTurnOnRedAllowed are SET to TRUE for DE_AllowedManeuvers to show that both right turn and RTOR are allowed. maneuverRightAllowed is SET to TRUE and maneuverRightTurnOnRedAllowed is SET to FALSE when a right turn is allowed and RTOR is not allowed.

However, an intersection may allow RTOR only at specific times of the day and days of the week, such as for Lane 4 in Figure 14. In Figure 14, RTOR is not allowed in Lane 4 from 3:00 PM to 6:00 PM every day.

To address this situation, revocable lanes are defined in the MAP message so its contents do not have to updated on a daily basis. Revocable lanes are not considered to be part of the current MAP unless they are in the DF_EnabledLaneList in the SPaT message.

In Figure 14, two revocable lanes are defined for each physical lane where RTOR is allowed, one with RTOR movement allowed (Lane 4), and the other lane with RTOR prohibited (Lane 14). DF_EnabledLaneList in the SPaT message indicates which lane, if any, is currently enabled (in effect). Only one revocable lane defined for each physical lane should be provided in DF_EnabledLaneList at any given time. Only the signalGroupIDs associated with currently active lanes should be broadcast.

All the lanes shown in Figure 14 are vehicle lanes and Table 11 shows the specific properties enabled for each lane. DE_LaneAttributes-Vehicle is used to indicate if the lane is revocable.

Table 11 Lanes with Revocable Indications

Lane	Description	DE_LaneAttributes-Vehicle
1	NB Protected Left	
2	SB Thru and SB Permissive Right	
3	EB Protected Left	
4	WB Thru and WB Permissive Right (RTOR Permitted)	isVehicleRevocableLane (0)
5	SB Protected Left	
6	NB Thru and NB Permissive Right	
7	WB Protected Left	
8	EB Thru and EB Permissive Right	
14	WB Thru and WB Permissive Right (RTOR Prohibited)	isVehicleRevocableLane (0)

To fulfill requirement 2a, the SPaT message shall indicate stop-Then-Proceed for DE_MovementPhaseState when the signal indication for the right turn movement is a circular red ball.

The DE_MovementPhaseState is determined using the signalGroupEntryTable (See Figure 15) and the agencySignalStateTable (See Figure 16). Each signalGroupID for an intersection is assigned a unique signalGroupEntryNumber. The DE_MovementPhaseState (agencySignalStateValue) for that signalGroupID is defined by the current signal indication color (agencySignalStateColor), agencySignalStateControlType, signalGroupGreenType, signalGroupPermissiveControl Color and signalGroupRedType in the agencySignalStateTable.

For example, for signalGroupID 15 in Figure 14 corresponds to signalGroupEntryNumber 15 in Figure 15 (this example has only 1 intersection). SignalGroupID 15 only applies to the right turn movement of the northbound lane. The through movement is signalGroupID 6. signalGroupEntryNumber 15 has a signalGroupControlType of phaseVehicle, a signalGroupGreenType of permissive, and a signalGroupRedType of stopThenProceed. When the signal indication for signalGroupEntryNumber 15 is red, row ID 14 in Figure 16 is applicable so DE_MovementPhaseState is stop-Then-Proceed. At the same time, signalGroupID is 6 is used for the thru movement from the same lane but has signalGroupRedType of stopAndRemain.

NOTE: The signalGroupEntryNumber, NOT signalGroupID, is the index for the signalGroupEntryTable because a controller may support multiple intersections. The signalGroupEntryNumber uniquely identifies an intersectionID and a signalGroupID.

To fulfill requirement 2b, the SPaT message shall indicate stop-And-Remain for DE_MovementPhaseState when the signal indication for the right turn movement is a circular red ball.

To fulfill requirements 2c and 2d above, the SPaT message shall indicate which revocable lane and by extension, which signalGroupID is in effect.

Referencing Figure 14, Table 12 indicates which revocable lanes are enabled during different times of the day. When RTOR is prohibited, Lane 14 in Figure 14 is included in enabled while lane 4 is omitted. When RTOR is permitted, Lane 4 is enabled while 14 is omitted.

Table 12 Example Enabled Lanes by Time Of Day

Time	EnabledLaneList
3:00 – 6:00 PM	14
All Other Times	4

At any point in time either Lane 4 or 14 must be enabled and appear in the spatEnabledLanesStatus object. Lanes marked as revocable but not included in spatEnabledLanesStatus are not active.

The spatEnabledLanesConcurrencyTable contains the set of revocable lanes allowed to exist concurrently. Thus, no set of lanes in this table should contain Lanes 4 and 14; (i.e., these lanes MUST be exclusive). Table 13 shows how enabled lanes are stored in the NTCIP 1202 spatEnabledLanesConcurrencyTable.

Table 13 Example spatEnabledLanesConcurrencyTable

enabledLaneIndex	enabledLaneConcurrency	enabledLaneIntersection
1	4	1
2	14	1

Lanes can be enabled based on the schedule in the timeBaseDayPlanTable which refers to the timebaseAscAction table using the OID of the timebaseAscNumber object (1.3.6.1.4.1.1206.4.2.1.5.3.1.1) in the day. Table 14 and Table 15 shows how to use the schedule to enable/disable lanes for the intersection shown in Figure 14.

Table 14 Example timeBaseDayPlanTable

dayPlanNumber	dayPlanEventNumber	dayPlanHour	dayPlanMinute	dayPlanActionNumberOID
1	1	0	0	1.3.6.1.4.1.1206.4.2.1.5.3.1.1.1
1	2	15	0	1.3.6.1.4.1.1206.4.2.1.5.3.1.1.2
1	3	18	0	1.3.6.1.4.1.1206.4.2.1.5.3.1.1.1

Table 15 Example timeBaseAscActionTable

timebaseAsc ActionNumber	timebaseAsc Pattern	TimebaseAsc AuxiliaryFunction	timebaseAsc SpecialFunction	timebaseAsc EnabledLane
1	1	0	0	1
2	0	0	0	2

At midnight (dayPlanHour 0, dayPlanMinute 0 in Table 14), timebaseAscActionNumber 1 would be activated, which would activate patternNumber 1 in the patternTable and enabledLaneIndex 1 in the spatEnabledLanesConcurrencyTable, which would enable lane 4. At 3:00 PM, timebaseAscActionNumber 2 would be activated, which would activate enabledLaneIndex 2 in the spatEnabledLanesConcurrencyTable, which would enable lane 14 and revoke lane 4. Since the timebaseAscPattern value is 0, no pattern is specified and pattern 1 would continue cycling. At 6:00 PM timebaseAscActionNumber 1 would be activated which would reenable lane 4 and revoke lane 14.

NOTE: NTCIP 1202 v03A included a patternSpatEnabledLanes object in the patternTable to associate revocable lanes with pattern. The patternSpatEnabledLanes object is deprecated in NTCIP 1202 v03B and replaced with the timebaseAscEnabledLane object refering to the spatEnabledLanesconcurrencyTable. This ensures that lanes can still be active as well as enabled or revoked if the controller is unable to run the specified pattern and falls into Free mode.

To fulfill requirements 2d above, the SPaT message shall indicate if RTOR is in effect. If RTOR is in effect, the SPaT message shall indicate stop-Then-Proceed for DE_MovementPhaseState when the signal indication for the right turn movement is a circular red ball. If RTOR is prohibited, the SPaT message shall indicate stop-And-Remain for DE_MovementPhaseState when the signal indication for the right turn movement is a circular red ball.

The signalGroupLane of the signalGroupEntryTable in Figure 15 defines which signalGroupEntryNumber should be used when a set of revocable lanes are enabled, as defined by enabledLaneConcurrency.

For example, signalGroupEntryNumber 14 has a signalGroupID of 14 and a signalGroupLane of 1. This means signalGroupEntryNumber 14 is active when all lanes defined in enabledLaneConcurrency.1 is enabled. enabledLaneConcurrency.1 contains only lane 4 of intersection 1 (See Table 13). Since lane 4 permits RTOR, the signalGroupRedType is stop-And-Proceed. When the signal indication for signalGroupEntryNumber 14 is red, row ID 14 in Figure 16 is applicable so DE_MovementPhaseState is stop-Then-Proceed.

On the other hand, signalGroupEntryNumber 17 (signalGroupID 24) is active when all lanes defined in enabledLaneConcurrency.2 is enabled. enabledLaneConcurrency.2 contains only lane 14 of intersection 1 (See Table 13). Since lane 14 prohibits RTOR, the signalGroupRedType is stopAndRemain. Thus when the signal indication for signalGroupEntryNumber 14 is red, row ID 15 in Figure 16 is applicable so DE_MovementPhaseState is stop-And-Remain.

F.3.3.6 TimeChangeDetails

This section provides additional guidance to provide the TimeChangeDetails for each allowed movement at a signalized intersection.

DF_TimeChangeDetails is a sequence of data elements that is included in a SAE J2735 SPaT message. DF_TimeChangeDetails describes time points (TimeMark) when the current movement state for each signalGroupID may change to the next movement phase state. Only minEndTime is a required data element in a SAE J2735 SPaT message, but CTI 4501 also requires maxEndTime, startTime and nextTime.

The controller uses the signalStateMinEndTick2, signalStateMaxEndTick2, signalStateNextTick2, and signalStateStartTick objects to provide the time points for each signalGroupID to the CV Application Process. CTI 4501 requires that the controller provide the time points for the current and the next interval. This is accomplished by the movementEventNumber, where movementEventNumber.1 represents the current interval and movementEventNumber.2 represents the next interval. See Sections 3.6.3.1 and 3.6.3.2 for the performance requirements to send signalState2 to the CV Application Process.

Portions of the following sections are reprinted from *CTI 4501, Connected Intersections (CI) Implementation Guide*.

F.3.3.6.1 Vendor-specific values

If the vendor can provide more accurate TimeChangeDetails when using the vendor-specific timing values (objects), the vendor is encouraged to do so.

F.3.3.6.2 signalStateMinEndTick2

This time mark defines the earliest point in time when the signal group may terminate its current indication as output by the traffic controller assuming current demand and operational conditions (preemption state, etc.). This time mark must take into consideration all active phase calls, preemption requests, coordination holds, or other demand inputs at the time of SPaT message generation, but does not need to provide the absolute minimum end time for demand inputs that have not yet been received. The signalStateMinEndTick2 should not be decreasing. The minimum end time shall be generated in accordance with the following rules:

For all numeric values:
TimeMark >= Current Time

For Green Indications of movements that are currently running in actuated mode:
TimeMark = Current Time
+ Minimum Green Timer Remaining for controlling phase (if any)
+ Gap (Passage) Timer Remaining if less than Minimum Green Timer
+ Trailing Overlap Green Timer if signal group is currently being controller by an overlap that will time its trailing green based upon current demand.

For Green Indications of movements that are currently in a non-actuated mode:
TimeMark = Current Time
+ Maximum Green or Split Max Timer Remaining for controlling phase (if any)
+ Trailing Overlap Green Timer if signal group is currently being controller by an overlap that will time its trailing green based upon current demand.

For Green Indications of movements that are actively timing a pedestrian phase that must terminate with the controlling movement, the Time Mark established above must be adjusted based upon the pedestrian state.
TimeMark >= Current Time
+ Pedestrian Walk Time Remaining
+ Pedestrian Clearance Time Remaining
+ MUTCD 3 Second Ped Clearance (if applicable)
– Ped Clearance Through Yellow (if applicable)
– Ped Clearance Through Red (if applicable)

For Yellow Indications of movements:

TimeMark = Current Time
+ Yellow Timer for phase or overlap that is currently controlling the signal group.

For Red Indications of movements:
TimeMark = Current Time
+ Expected minimum duration of all movements that have active demand prior to service of the phase or overlap will next control this signal group.

A common situation that has been difficult to characterize with SPaT is the case of a phase resting in green awaiting service on conflicting movements. The time mark for this case can either reflect a minimum time to change of 0.1 second or 0.0 seconds, a likely time that is very indeterminate, and a maximum time that can be for even hours in late night operation. It is recommended that 0.1 second be used as the minimum time to change, rather than 0.0. 0.0 should be reserved for the case where green termination was commanded, and the field outputs change is imminent. A vehicle resting in green may display 0.1 second for an indefinite period with increased value to match the active passage timer if/when additional vehicle actuations on the movement reset the passage timer.

In cases of an active Preemption, the aforementioned intervals may be programmed to run override values. The generation methods above shall remain valid; however, the traffic controller must substitute the preemptively overridden timing values upon controller activation of the preemptive timing (may occur before or after preemption input delay interval).

F.3.3.6.3 signalStateMaxEndTick2

This time mark defines the absolute latest point in time when the signal group may terminate its current indication as output by the traffic controller. This time mark must take into consideration all active phase calls, coordination force-offs, preemption requests, or other demand inputs at the time of SPaT message generation, but does not need to provide the absolute maximum end time for demand inputs that have not yet been received or consider the possibility of the removal of an actively received demand inputs.

This time mark should not be increasing (extending further into the future (incrementing)) because then it really wasn't a Maximum. However, the time mark is allowed to change from a value of Unknown to an integer to allow a vehicle to go through the intersection, and then change back to a value of Unknown.

The maximum end time shall be generated in accordance with the following rules.

For Green Indications of movements that are currently running in actuated mode with a call on a conflicting phase:
TimeMark = Current Time
+ Maximum Green or Maximum Split Timer remaining for controlling phase (whichever is greater, if either apply)
+ Added Extension Timer Remaining (if this feature is enabled and current demand will require added extension)
+ Trailing Overlap Green Timer if signal group is currently being controller by an overlap that will time its trailing green based upon current demand.
+ any dynamic max green additions

Note: Maximum Split Timer is defined as the time to phase force off if running under a coordinated or preempted mode.

Note: In cases where no conflicting demand is present, this Time Mark will be set to the value for unknown.

For Green Indications of movements that are currently in a non-actuated mode:
TimeMark = Current Time
+ Maximum Green or Split Max Timer Remaining for controlling phase (if any)

+ Trailing Overlap Green Timer if signal group is currently being controller by an overlap that will time its trailing green based upon current demand.

For Green Indications of movements that are actively timing a pedestrian phase that must terminate with the controlling movement, the Time Mark established above must be adjusted based upon the pedestrian state.

TimeMark >= Current Time
+ Pedestrian Walk Time Remaining
+ Pedestrian Clearance Time Remaining
+ MUTCD 3 Second Ped Clearance (if applicable)
– Ped Clearance Through Yellow (if applicable)
– Ped Clearance Through Red (if applicable)

For Yellow Indications of movements:

TimeMark = Current Time
+ Yellow Timer for phase or overlap that is currently controlling the signal group.

For Red Indications of movements:

TimeMark = Current Time
+ Expected maximum duration of all movements that have active demand prior to service of the phase or overlap will next control this signal group.

For non-actuated mode (fixed time operations) or when the interval duration of the current movement state is deterministic (e.g., max recall, yellow interval duration, etc.):

Time Mark for the maxEndTime = Time Mark for the minEndTime

F.3.3.6.4 signalStateLikelyEndTick2

This time mark defines the most likely point in time when the signal group will terminate its current indication as output by the traffic controller. This time mark has already been applied in practice along with the associated TimeIntervalConfidence for green indications. The confidence factor requires a cycle-by-cycle and/or historic statistical analysis of signal timing that is beyond the capabilities of NTCIP 1202 v03A standard traffic controllers. Rather than undermine the existent deployments of this data field, this time mark was left as an optional field in NTCIP 1202. This field can be utilized by external systems that are performing a statistical analysis of signal timing.

The predictive nature of signal timing is expected to be characterized and bounded using the minEndTime and maxEndTime for all indications (R, Y, G). A mean value for green duration is not available unless the traffic controller has capability to generate this likely time and interval confidence. In cases where the green duration is deterministic (fixed time, max recall, etc.), the minEndTime will equal the maxEndTime revealing this level of certainty to the end user.

F.3.3.6.5 signalStateTickConfidence2

The likelyTimeConfidence is defined as the statistical confidence that the reported likely end time point of the current movement (e.g., at the end of a protected green or end of a permissive clearance time) at an intersection is accurate. The confidence value is measured as a probability class, as defined by DE_TimeIntervalConfidence in SAE J2735. However, at this time, there is no defined method or formula by this standard or SAE J2735 to determine the statistical confidence. Until a standard method or formula to calculate likelyTimeConfidence is defined, signalStateTickConfidence is unused and should be SET to 0.

F.3.3.6.6 signalNextTick2

This time mark defines the point in time when the movement phase state is NEXT allowed to proceed AFTER a clearance interval. The movement phase state is next allowed to proceed when

DE_MovementPhaseState is equal to permissive-Movement-Allowed or protected-Movement-Allowed. The movement phase state is considered to be in a clearance interval if DE_MovementPhaseState is equal to stop-Then-Proceed, stop-And-Remain, permissive-Clearance, protected-Clearance, or caution-Conflicting-Traffic.

This is a best predicted value based upon awareness of the intersection. Full demand of all phases can be assumed unless the traffic controller is capable of monitoring current phase usage and can more accurately estimate future split/phase utilization. This value should include the time to service full demand for all phases that precede the next beginning of green service of the current signal group.

If the movement is currently in a flashing yellow arrow state, and the next movement state is a protected green arrow, the nextTime is NOT the time mark when the movement state being the protected green arrow. The nextTime is the time mark when the movement state is next permissive-Movement-Allowed or protected-Movement-Allowed AFTER the clearance interval.

In **coordinated operation**, this value establishes the time mark for the next service of a protected-permissive movement based upon the continued in-step coordinated operation and timing commensurate to the regular permissive/split window of the phase. In cases of coordinated transition or transit signal priority, accuracy can be improved if the split adjustment methodologies are factored into this value computation. This improvement is desirable, but not mandatory for the traffic control to apply.

In **free operation**, this value establishes the time mark for the next service of a protected-permissive movement based upon current intersection demand. If the traffic controller can apply data for recent average green timing of each phase (split monitor), it will improve this estimate. However, at a minimum, the traffic controller generates this value assuming minimum green timing of all active calls to preceding phases and assuming pedestrian service for all actively registered pedestrian demand.

F.3.3.6.7 signalStateStartTick

This time mark defines the point in time when the next interval (after the current interval) will begin for the same signal group. To avoid ambiguity in the meaning of the time mark, the time mark is always considered to be a time in the future. Therefore, time mark cannot be used for intervals that have already begun timing.

This time mark will be a numeric value only if the start time of the next movement phase state immediately after the current movement phase state is known.

If the start time is unknown, a value of unknown is used.

If the next movement phase state is unknown, a value of unknown for the start time will be used.

F.3.3.6.8 Special Traffic Control Situations

The following guidance is reprinted from CTI 4501 and is offered to resolve any ambiguity and ensure SPaT messaging can be supported in these situations.

There is a common case here when a flashing yellow arrow (permissive left) will revert directly to a protected left green arrow. These time change details may lead roadway users to assume these time marks are the time when change will occur from a protected or permissive movement into a restricted movement and not realize this change can occur between protected and permissive movements. The TimeChange detail shall indicate the following:

> nextTime = likelyTime for cases when the movement will shift between protected and permissive movements, without a clearance interval present.

nextTime = likelyTime + Clearance Interval for cases when the movement will shift between protected and permissive movements, with an interim clearance interval present.

F.3.3.7 Enabled Lanes

Revocable lanes are defined in SAE J2735 to define the different uses of a physical lane that may vary by time-of-day or during special events. For example, an agency may define a revocable lane each for when a physical lane can be used for travel in a direction, not allowed for travel (e.g., a parking lane), or for travel in the opposite direction. The MAP data message describes the attributes for each lane defined for an intersection, including if a lane is a revocable lane.

When a revocable lane is "asserted" or in use, it is called an enabled lane. The spatEnabledLanesStatus object definition indicates what lanes are currently enabled.

NTCIP 1202 v03B provides two methods to enable revocable lanes. Lanes can also be enabled using an entry in the timebaseAscActionTable. A new object definition, timebaseAscEnabledLane, was added to the timebaseAscActionTable to indicate what set of revocable lanes are enabled, by pointing to entries in the spatEnabledLanesConcurrencyTable2. Each entry in the spatEnabledLanesConcurrencyTable2 defines the set of revocable lanes to be concurrently enabled.

The spatEnabledLanesCommand2 object allows a management station to set which revocable lanes are enabled. This object overrides any enabled lanes established in the timebaseAscActionTable. An octet value of "0xFF" will cancel this command and revert back to the values currently in effect in the timebaseAscActionTable.

In scenarios where a controller generates SPaT data for more than one signalized intersection, the same lane identifier should be assigned to a revocable lane for all signalized intersections that the controller provides SPaT data for. With support for up to 255 lanes per controller, a lane identifier used by all signalized intersections managed by a controller should be possible. NTCIP 1202 does not support using different lane identifiers at signalized intersections for a revocable lane.

For example, in Figure 18, the revocable lane between signalized intersection A and signalized intersection B is assigned the same lane identifier - lane identifier 9 when vehicular traffic is moving westbound (from intersection B to intersection A); and lane identifier 109 when vehicular traffic is moving eastbound (from intersection A to intersection B). In this example, lane 9 is an ingress lane into and lane 109 is an egress lane out of intersection A. Conversely, lane 109 is an ingress lane into and lane 9 is an egress lane out of intersection B. Lane identifiers 9 and 109 should not be assigned to any other lane for any signalized intersection managed by the controller.

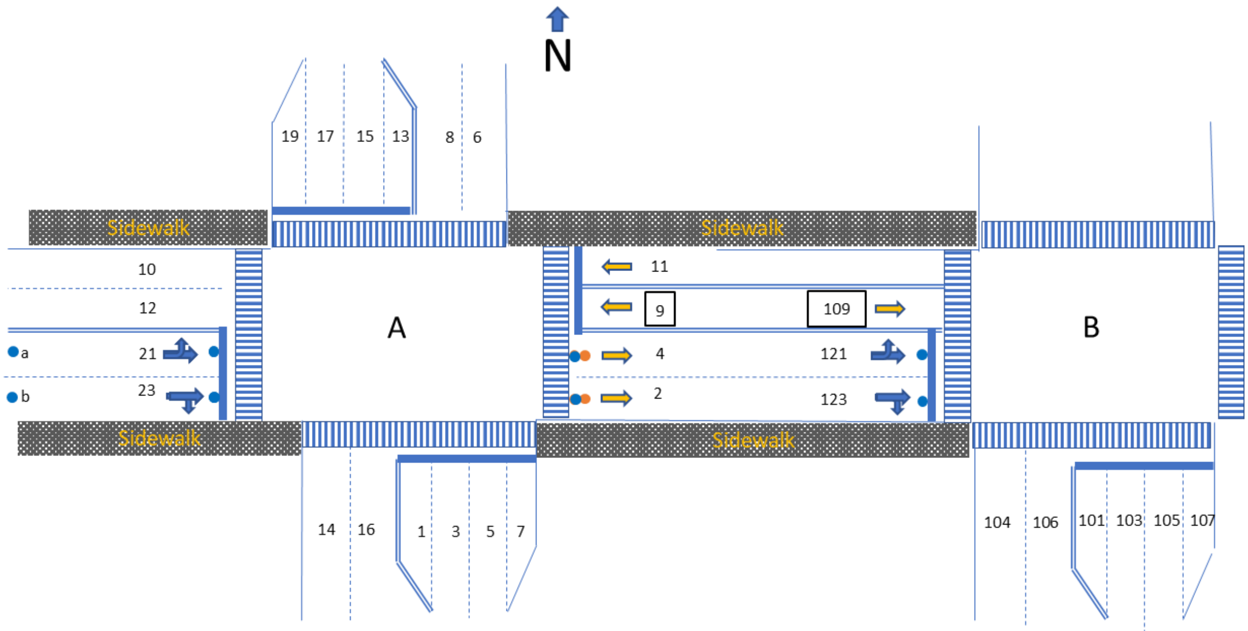

Figure 18 Enabled Lanes Example

F.3.3.8 Advisory Speeds

The advisorySpeedTable2 contains advisory speeds for an approach into the intersection. For a specified signalGroupEntry Number, each row contains an advisory speed type (as defined by DE_AdvisorySpeedType in SAE J2735), the advisory speed, the distance from and to the stop bar that the advisory speed is applicable for, and if applicable, the specific class of user defined for the intersection. Providing advisory speeds in a SPaT message is optional.

F.3.3.9 Movement Maneuvers

The movementManeuverTable2 provides additional information that may assist a traveler wishing to perform a specific maneuver at the intersection. SAE J2735 allows a controller to provide movement data for up to 16 movement maneuvers (e.g., northbound left turn, southbound left turn, etc) for a signalGroupID. For a specified signalGroupEntry Number, each row contains a lane connection identifier (which defines the lane to lane connection in the corresponding MAP message), the existing queue length, in meters, that has been detected for that connection; and a bit that indicates if any pedestrians or bicyclists are detected that conflict with the desired maneuver. Each row also identifies the detector number input(s), in the form of octet strings, that provide the information to determine the queue length, and the potential presence of pedestrians and bicyclists. An octet of 00 indicates no additional detectors in the octet string.

SAE J2735 also supports a data element to provide the distance from the stop bar within which travelers have a high probability of successfully executing the desired movement before the movement ends (e.g., before the phase starts the clearance interval); however, the ASC WG decided not to support that data element at this time, because current traffic controllers are unable to calculate that information.

The movementManeuverId2 object, when set to 0, indicates that row in the movementManeuverTable2 does not contains any valid information (i.e., that row is disabled).

F.3.3.10 Implementation

NTCIP 1202 v03 allows an ASC to provide the information needed to generate a SPAT message. A bit in the spatOptions object is used to enable the ASC to generate and exchange SPAT data. Once the SPAT data is generated, a SNMP manager, such as a management station at a traffic management center, can retrieve that SPAT data.

A management station can enable communications with the CV Application Process using the rsuPortTable. The rsuPortTable contains the configuration and status information for the logical ports used by an ASC to communicate with RSUs. The rsuPortTable uses the logicalNameTranslationTable to identify the IP address of the RSU. A polling period and a watchdog timer to measure inactivity can be configured for each rsuPortEntry.

The spatPortTable establishes which RSU ports should the ASC Process provide SPAT data to, and the status of that provision of SPAT data. The spatPortTable is indexed by the rsuPortIndex used for the rsuPortTable. Each row in the spatPortTable contains a bit in the spatPortOptions object to enable the exchange of SPAT data with that logical RSU port, a bit to indicate if the SPaT data provided (assuming SPaT is enabled) is valid or invalid, and a bit to indicate if the MAP message is valid or invalid. Although the MAP message is broadcasted by the RSU, field maintenenace personnel are more likely have easier access to the traffic controller than the RSU. Since field maintenance personnel are likely to mark the SPaT data invalid during testing or temporary field work, the capability to indicate if the MAP message is valid or invalid is added to the traffic controller since the bit to indicate an invalid MAP message is in the SPaT message (which represents dynamic data, and specifically in the DE_IntersectionStatus object) and so the field maintenance personnel will not need access to the RSU also.

spatPortStatus indicates the status of the SPAT data exchange for that logical RSU port. The spatPortStatus is disabled (2) if Bit 0 of spatPortOptions for the same row is FALSE. spatPortStatus is enabledLanesError (5) if the value of the patternSpatEnabledLanes object does not match one of the entries in spatEnabledLanesConcurrencyTable.

The ASC also has a spatTimestamp object to indicate the time when the SPAT data was last updated by the ASC. Note that the time for this object uses the ASC's local time. However, the SPAT message broadcast by the CV Application Process uses UTC time. It is expected that the CV Application Process will make any time adjustments necessary before the SPAT message is broadcast.

F.3.3.11 Connected Device Detectors [DEPRECATED]

NOTE: This section on Connected Vehicle Detectors and the next section on Connected Vehicle Data, and the object definitions, were written for NTCIP 1202 v03. NTCIP 1202 v03 was developed and published before NTCIP 1218, Object Definitions for Roadside Units (RSUs) was developed and published. Since the detection of the presence of travelers via "connected vehicle" will be determined by the RSU, users should refer to NTCIP 1218 for user needs, requirements, and object definitions for a "connected vehicle" detector. However, there may still be needs and requirements for the ASC to use the detection actuation information from the RSUs so this section and the object definitions are retained for informational purposes until the needs and requirements are developed in a future version of NTCIP 1202.

BSMs and PSMs broadcast by connected devices can provide a rich source of information about travelers around an ASC. Connected vehicles broadcasting BSMs and connected devices on travelers broadcasting PSMs can supplement, or replace, detectors around the ASC. RSUs at an intersection can receive BSMs and PSMs 360 degrees around the RSU, as long as the connected devices is within the receiving range of the RSU, nominally assumed to be 300 meters.

NTCIP 1202 v02 already allows vehicle and pedestrian detectors to be defined as inputs for signal operations and for data collection. NTCIP 1202 v03 extends that capability to connected vehicles by allowing a user to define detection zones for BSMs and PSMs, called a connected device detector. If the CV Application Process detects a BSM or PSM within the detection zone, NTCIP 1202 v03 can use the data from the BSM or PSM as an input. NTCIP 1202 v03 also allows filters to be configured for each connected device detector so only BSMs and PSMs that satisfy the user defined criteria.

To support connected device detectors, the cvDetectorTable and detectionZoneNodePointTable were added to allow NTCIP 1202 v03 to use the BSMs and PSMs as inputs to the ASC.

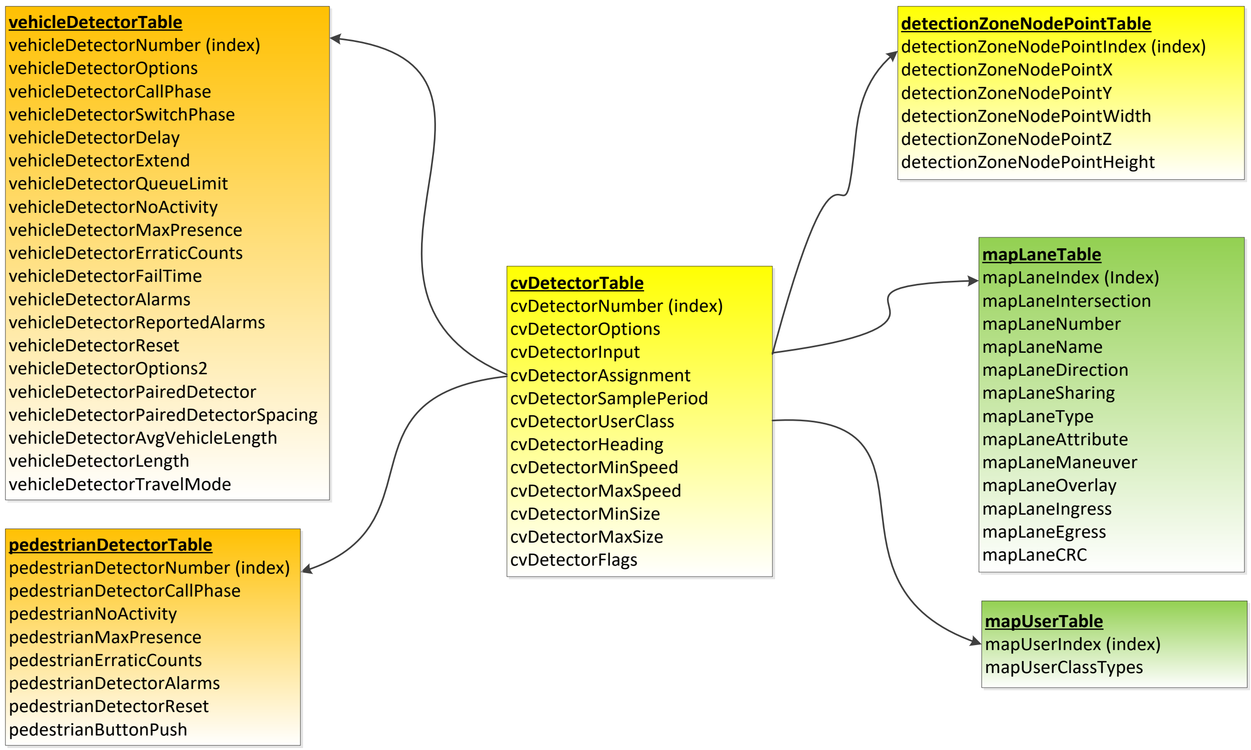

Figure 19 Connected Data Detectors Mapping

The cvDetectorTable defines the connected device detection zones, that is, the detection zones based on BSM and PSM transmissions from connected devices. This table assigns each connected device detection zone to either the mapLaneIndex in the mapLaneTable or to a detectionZoneNodePointIndex in the detectionZoneNodePointTable. Assigning the detection zone to the mapLaneIndex allows the CV Application Process to exchange actuation data or detection reports, for that lane with the ASC. The detectionZoneNodePointTable defines a series of node points that form a sequence of X-Y-Z offsets values, like defining a lane, and including the width and the elevation. The first node point is offset from the intersection's (cvDetectorIntersection) reference point, while all subsequent node points are offset values from the previous node point.

If no mapLaneIndex or detectionZoneNodePointIndex is assigned (i.e., cvDetectorInput = 00 for that connected device detection zone), then that connected device detection zone will process any BSM or PSM received by the CV Application Process, regardless of the geographic location of the connected device. Figure 19 depicts the relationship between these tables. The detector tables in NTCIP 1202 v03 and found on the ASC are in orange, MAP related tables are in green, and tables specific to the collection of connected vehicles data are in yellow.

Each connected device detector can also be assigned to one or more detector inputs on the ASC. This assignment allows the assigned detector input(s) to use the connected device data from that connected device detector for signal operations or for data collection. It also allows the connected device detector to adopt the parameters and settings for that detector input, such as its status, alarms, and options. The detector input is a detector number in either the vehicleDetectorTable, or pedestrianDetectorTable. Note that while a connected device detector may be assigned to more than one detector input, the detector inputs has to be of the same type (i.e., a vehicle type should be input to a vehicle detector).

The cvDetectorTable also defines the criteria to filter BSMs or PSMs within the detection zone to be used as actuations or for data collection. Only BSMs or PSMs that satisfy all the criteria are processed by the CV Application Process and are exchanged with the ASC Process. Filters supported in cvDetectorTable are:

a) cvDetectorUserClass.

   Note: currently, the BSMs currently do not broadcast vehicle type according to SAE J2945/1. PSMs does include a DE_PersonalDeviceUserType to describe the type of non-vehicular traveler, however the relevant user class restrictions are defined by DE_RestrictionAppliesTo limited to equippedBicycle, pedestrians, slowMovingPersons, wheelchairUsers, visualDisabilities, audioDisabilities, and otherUnknownDisabilities.

b) cvDetectorHeading. The direction of travel.
c) cvDetectorMinSpeed and cvDetectorMaxSpeed. The speed of the connected device.
d) cvDetectorMinSize and cvDetectorMaxSize. The size of the connected vehicle.
e) cvDetectorFlags. The status of one or more event flags or the brake status of the connected vehicle. Note this is a OR flag, not an AND flag, if more than one flag or brake status is selected.

F.3.3.12 Connected Device Data

NOTE: The previous section on Connected Vehicle Detectors and this section on Connected Vehicle Data, and the object definitions, were written for NTCIP 1202 v03. NTCIP 1202 v03 was developed and published before NTCIP 1218, Object Definitions for Roadside Units (RSUs) was developed and published. Since the detection of the presence of travelers via "connected vehicle" will be determined by the RSU, users should refer to NTCIP 1218 for user needs, requirements, and object definitions for a "connected vehicle" detector. However, there may still be needs and requirements for the ASC to use the detection actuation information from the RSUs so this section and the object definitions are retained for informational purposes until the needs and requirements are developed in a future version of NTCIP 1202.

There are different methods that a CV Application Process can provide the ASC with the data collected from the BSMs and PSMs within the connected device detection zones (detectors). One simple method is to place a call on the appropriate detector input on the backplane, on the appropriate serial port, or via the ASC's Application Programming Interface (API). Each of these methods are outside the scope of NTCIP 1202 v03.

NTCIP 1202 v03 supports two formats for exchanging BSM and PSM data collected in the connected device detection zones with an ASC:

a) Actuations: Actuation data is exchanged across the ASC Process - CV Application Process when the only input needed by the ASC is if the presence of a traveler (vehicle, pedestrian, or bicyclist) is detected within the detection zone. This may be a vehicle on the side street, or a pedestrian wishing to cross the street.
b) Processed Data: Processed data is used when the processing of the BSMs and PSMs is performed by the CV Application Process. The results of the processing may then be exchanged across the ASC Process - CV Application Process as inputs to the ASC for its signal timing operations.

The two different formats are not mutually exclusive, that is, both formats can be exchanged across the ASC Process and the CV Application Process interface. The formats are enabled in cvDetectorOptions for each connected device detector in cvDetectorTable.

F.3.3.12.1 Actuations

The cvDetectionGroupActuation in the cvDetectionGroupTable provides actuation data for connected devices detected within a connected device detection zone and satisfies the criteria established for that connected device detector in the cvDetectorTable (e.g., heading, speed). Each row in the cvDetectionGroupTable represents a set of 8 connected device detectors. The cvDetectionActuationSamplePeriod defines the frequency the cvDetectionGroupActuations is exchanged between the CV Application Process and the ASC Process. An implementation that supports NTCIP 1103-based traps can configure a trap so this cvDetectionGroupActuation object is transmitted on change (assuming the CV Application Process is the SNMP agent and the ASC Process is the SNMP manager).

F.3.3.12.2 Processed Data

The detectionReportTable reports the data processed by the CV Application Process. This data may be used by an ASC for signal operations or by a management station for archival purposes. Each row in the detectionReportTable represents a snapshot of the status of connected devices within the connected vehicle detection zone. Data in each snapshot may include the number of connected devices in the detection zone, the average speed of the connected devices, the average travel time through the detection zone, the average queue (as a number of vehicles), the average gap between connected vehicles, and the number of connected vehicles in a platoon.

As the connected device data is received and processed by the CV Application Process, each detection report is stored in the detectionReportTable. A sequence counter is maintained to track every time a detection report is successfully stored in a row. The frequency of how often a detection report is generated and exchanged is defined in cvDetectorSamplePeriod in the cvDetectorTable.

§