A Project Document of the Joint Committee on the NTCIP

Draft NTCIP 1202 v04 ASC Test Procedures

NTCIP 1202 Version 04 Annex C

National Transportation Communications for ITS Protocol

Object Definitions for Actuated Signal Controllers (ASC) Interface Test Procedures

Draft v04.07b January 28, 2025

This is a draft document, which is distributed for review and comment purposes only. You may reproduce and distribute this document within your organization, but only for the purposes of and only to the extent necessary to facilitate review and comment to the NTCIP Coordinator.

Please ensure that all copies include this notice. This document contains preliminary information that is subject to change.

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

NOTICES

TRF Distribution Permission [when standard contains a TRF]

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."

Test Procedures [Normative]

C.1 Purpose

This annex defines the detailed, but generic, test procedures for testing an implementation of this standard.

C.1.1 Scope

Annex C defines test procedures that cover communication testing for features defined in NTCIP 1202 v04. Test procedures for functional testing may be developed as part of a future revision of NTCIP 1202 v04.

These test procedures are intended to be used as a portion of the overall set of tests that would be performed during component testing of a management station.

Certain test procedures involve testing the Transaction Feature defined in ISO 26048-1. Official documentation for testing global objects, as indicated in the RTM in Annex A.

C.1.2 Keywords

The following key words are used:

- ASSIGN: The test application shall designate a declared variable or specified object as equal to a specific variable or value.
- DELAY: The test application and user shall not perform any actions for a defined period, which may be measured in time units or by monitoring some event that does not involve any exchange of information over the communications media (e.g., DELAY until the temperature exceeds a threshold). In the latter case, the step should also define exception conditions to allow for possibility that the event never happens.
- ERROR INDEX: The value contained in the 'error-index' field of the last SNMPv3Message received from the DUT. See RFC 3416 for additional details related to the error index.
- EXIT: This keyword indicates that the user and test application should terminate the test case without performing any more steps. The keyword by itself does not have any implications as to whether a given test passes or fails.
- FOR: This keyword causes the user (or application) to begin a looping process that shall increment through a series of values. It is comparable to the "for...next" expression in C.
- GET: The test application shall transmit to the DUT one SNMPv3Message containing a GetRequest-PDU, per the rules of NTCIP 2301. Each statement using this keyword shall unambiguously reference the value for the 'name' field(s) to be included in the request. The GetRequest-PDU shall include all of the names in its 'variable-bindings' field. See RFC 3416 for additional details related to the GetRequest-PDU.

Unless otherwise indicated, the user or test application shall VERIFY the following, in order:

a) The DUT responds with exactly one SNMPv3Message that contains a Response-PDU, per the rules of NTCIP 2301; this is the RESPONSE. The DUT may also transmit one or more SNMPv3Messages, each of which containing either an SNMPv2-Trap-PDU or an InformRequest-PDU.

b) The value contained in the 'msgVersion' field of the RESPONSE equals 3 (snmpv3)

- c) MESSAGE ID IN equals (MESSAGE ID OUT 1)¹
- d) MESSAGE MAX SIZE IN is at least 484

e) The value contained in the 'msgFlags' field of the RESPONSE is one octet with the final two (least significant) bits set

- f) The value contained in the 'msgSecurityModel' field of the RESPONSE is equal to 4 (TSM)
- g) The value contained in the 'msgData' field of the RESPONSE is 'plaintext'
- h) CONTEXT ENGINE ID IN equals CONTEXT ENGINE ID OUT
- i) CONTEXT NAME IN equals CONTEXT NAME OUT

j) REQUEST ID IN equals (REQUEST ID OUT - 1)¹

- k) RESPONSE ERROR equals 0 (noError)
- I) ERROR INDEX equals 0

m) The 'variable-bindings' field contains the same number of VarBind structures as were contained in the GetRequest-PDU

n) The value of each name field in the RESPONSE equals the value of the name field in the GetRequest-PDU that is in the same ordered position.

o) The RESPONSE TYPE of each VarBind Structure in the VarBind List is not equal to noSuchObject

In addition, if MESSAGE MAX SIZE is 0, RECORD MAX MESSAGE SIZE IN as MAX MESSAGE SIZE; otherwise VERIFY that MAX MESSAGE SIZE IN equals MAX MESSAGE SIZE.

 GET-NEXT: The test application shall transmit to the DUT one SNMPv3Message containing a GetNextRequest-PDU, per the rules of NTCIP 2301. Each statement using this keyword shall unambiguously reference the value for the 'name' field(s) to be included in the request. The GetNextRequest-PDU shall include all the names in its 'variable-bindings' field. See RFC 3416 for additional details related to the GetNextRequest-PDU.

Unless otherwise indicated, the user or test application shall VERIFY the following, in order:

a) The DUT responds with exactly one SNMPv3Message that contains a Response-PDU, per the rules of NTCIP 2301; this is the RESPONSE. The DUT may also transmit one or more SNMPv3Messages, each of which containing either an SNMPv2-Trap-PDU or an InformRequest-PDU.

b) The value contained in the 'msgVersion' field of the RESPONSE equals 3 (snmpv3)

c) MESSAGE ID IN equals (MESSAGE ID OUT – 1)¹

d) MESSAGE MAX SIZE IN is at least 484

e) The value contained in the 'msgFlags' field of the RESPONSE is one octet with the final two (least significant) bits set

 f) The value contained in the 'msgSecurityModel' field of the RESPONSE is equal to 4 (TSM)

g) The value contained in the 'msgData' field of the RESPONSE is an 'plaintext'

h) CONTEXT ENGINE ID IN equals CONTEXT ENGINE ID OUT

i) CONTEXT NAME IN equals CONTEXT NAME OUT

j) REQUEST ID IN equals (REQUEST ID OUT – 1)¹

- k) RESPONSE ERROR equals 0 (noError)
- I) ERROR INDEX equals 0

m) The 'variable-bindings' field contains the same number of VarBind structures as were contained in the GetNextRequest-PDU

n) The name field in the RESPONSE either 1) contains the same name as in the GetNextRequest-PDU that is in the same ordered position and its value is 'endOfMibView' or 2) contains a value that is lexicographically greater than the name in the GetNextRequest-PDU that is in the same ordered position.

¹ The MESSAGE ID OUT and REQUEST ID OUT values logically increment after sending a message; when comparing response messages, the "minus 1" reflects the need to compare the value sent in the request, which is prior to its being incremented.

In addition, if MESSAGE MAX SIZE is 0, RECORD MAX MESSAGE SIZE IN as MAX MESSAGE SIZE; otherwise VERIFY that MAX MESSAGE SIZE IN equals MAX MESSAGE SIZE.

- GOTO: This keyword shall cause the user (or application) to immediately jump to the indicated location in the test procedure (e.g., to another Step).
- IF: This keyword shall cause the user (or application) to perform a comparison and take one action if the comparison evaluates to true and another action if the comparison evaluates to false. It is comparable to the "if...else..." expression in C.
- PERFORM: The user or test application shall perform another test case as a part of this test case. Unless otherwise indicated in the "PERFORM" statement, the user (and test application) shall use the variable values defined when the other test case is performed in a stand-alone fashion.
- PRE-CONDITION: The PRE-CONDITION keyword shall be used as a predicate to the text of a test step to indicate that the text provides a textual description of any pre-conditions for the test case. Pre-conditions are conditions that must be met prior to running a test case. Only one pre-condition shall exist in a test case and it shall always be the first step listed, if present.
- RECORD: The user (or test application) shall record the information indicated by the test step as a part of the test results. This information may be referenced by a later step of the test case (or by a later step of a calling step case).
- RESPONSE ERROR: The value contained in the 'error-status' field of the last SNMPv3Message received from the DUT. See RFC 3416 for additional details related to the error status.
- SET: The test application shall transmit to the DUT one SNMPv3Message containing a SetRequest-PDU, per the rules of NTCIP 2301. Each statement using this keyword shall unambiguously reference the order and value for the 'name' field(s) to be included in the request. The statement shall also indicate the value of the 'value' field associated with each 'name' field. Unless otherwise indicated, the value will be encoded according to the SYNTAX of the associated object. The SetRequest-PDU shall include all of the names and values, with their indicated associations in its 'variable-bindings' field. See RFC 3412 for additional details related to the SetRequest-PDU.

Unless otherwise indicated, the user or test application shall VERIFY that:

a) The DUT responds with exactly one SNMP Message that contains a GetResponse-PDU, per the rules of NTCIP 2301; this is the RESPONSE. The DUT may also respond with one or more SNMPv3Messages, each of which containing either an SNMPv2-Trap-PDU or an InformRequest-PDU

- b) The value contained in the 'msgVersion' field of the RESPONSE equals 3 (snmpv3)
- c) MESSAGE ID IN equals (MESSAGE ID OUT 1)¹
- d) MESSAGE MAX SIZE IN is at least 484

e) The vaue contained in the 'msgFlags' field of the RESPONSE is one octet with the final two (least significant) bits set

f) The value contained in the 'msgSecurityModel' field of the RESPONSE is equal to 4 (TSM)

g) The value contained in the 'msgData' field of the RESPONSE is an 'encryptedPDU'

h) CONTEXT ENGINE ID IN equals CONTEXT ENGINE ID OUT

i) CONTEXT NAME IN equals CONTEXT NAME OUT

j) REQUEST ID IN equals (REQUEST ID OUT – 1)¹

k) RESPONSE ERROR equals 0 (noError)

I) ERROR INDEX equals 0

m) The 'variable-bindings' field contains the same number of VarBind structures as contained in the SetRequest-PDU

n) The value of each name field in the RESPONSE equals the value of the name field in the SetRequest-PDU that is in the same ordered position.

o) The value of each value field in the RESPONSE equals the value of the value field in the SetRequest-PDU that is in the same ordered position.

In addition, if MESSAGE MAX SIZE is 0, RECORD MAX MESSAGE SIZE IN as MAX MESSAGE SIZE; otherwise VERIFY that MAX MESSAGE SIZE IN equals MAX MESSAGE SIZE.

• VERIFY: The user or test application shall evaluate the expression that follows this keyword. Each statement using this keyword shall contain an unambiguous expression that will always evaluate to either true or false without subjective or qualitative judgments by the tester.

If the result is true the following will take place:

- a) The verification step shall pass, and
- b) The test shall continue to the next step, unless otherwise indicated in the test case.

Otherwise, if the result is false the following will take place:

- a) The verification step shall fail.
- b) The test case shall fail.
- c) The test case shall EXIT, unless otherwise indicated in the test case.

NOTE—While criteria are often stated in exact terms (e.g., "The response shall be '3'" or, "The sign shall display 'TEST," etc.), it may also be the case that criteria may be stated as ranges or thresholds (e.g., "The response shall be between '2' and '16' inclusive;" or, "The response shall be '3' or greater;" etc.). Each approach is valid and should be considered in the construction of a test case.

C.1.3 Rules for Executing Test Procedures

The test procedures contained in this annex are designed to be used for component testing a device for conformance to the NTCIP 1204 v04 interface standard. To component test a device for conformance to the NTCIP 1204 v04 interface standard, the user shall follow the steps as written, filling in the pass/fail information in the 'Results' column.

A given test procedure may entail multiple steps that may require multiple interactions between the user and the management station to fulfill the complete test procedure. For example, a single test procedure may transfer the definition of a message to the device and then retrieve the contents of the message to ensure that the values were updated; this might require two user interface operations.

C.2 Testing Requirements

C.2.1 Field Device Test Environment

All Test Cases covered by this Testing Requirements documentation require the Device Under Test (DUT) to be connected to a test application as depicted in Figure C-1. A data analyzer may also be used to capture the data exchanged between the two components. The test environment should be designed to minimize any complicating factors that may result in anomalies unrelated to the specific test case. Failure to isolate such variables in the test environment may result in false results to the test. For example, the device may be conformant with the standard, but communication delays could result in timeouts and be misinterpreted as failures.



Figure C-1: Field Device Test Environment

The following pre-conditions apply to all test cases unless otherwise defined:

- a) All components should be turned on and be provided sufficient time to start up prior to starting any test case
- b) The test software should be connected to the central port of the DUT and the DUT should be set for central control
- c) The test software, data analyzer, and DUT should all be configured to use a common set of communication settings, including data rates, lower layer protocols, community names, etc.
- d) The DUT should be exposed to a medium amount of ambient light so that tests can increase or decrease the amount of light as needed
- e) The DUT should have definitions for all font sets that it supports
- f) The DUT should have a valid illumination brightness curve defined with a positive slope.

C.2.2 Requirements to Test Case Traceability Table (RTCTT) Table

The Requirements to Test Case Traceability Table defines the traceability between the requirements in Section 3 and the Test Cases presented in this Annex. This table defines the minimal test procedure(s) that shall be completed to confirm that an implementation fulfills a requirement and still conform to this standard.

To confirm that an implementation fulfills a requirement, the DUT shall successfully pass all test cases that trace to that requirement.

| Requirement ID | Functional Requirement | Test Case ID | Test Case Title |
|----------------|---|-----------------|------------------------------------|
| 3.4 | Architectural Requirements | | |
| 3.4.1 | Support Basic Communications Requirements | | |
| 3.4.1.1 | Retrieve Data | | |
| | | C.3.3.51 | Determine Maximum Number of Phases |

| 3.4.1.2 | Deliver Data | | |
|-------------|--|---------|---|
| | | C.3.2.2 | Configure Startup Flash Time |
| 3.4.1.3 | Explore Data | | |
| | | C.3.1.3 | Explore Data |
| 3.4.1.4 | Monitor SNMP Requirements | | |
| 3.4.1.4.1 | Monitor SNMP Information | | |
| | | C.3.1.4 | Monitor SNMP Information |
| 3.4.2 | Manage Data Blocks Requirements | | |
| 3.4.2.1 | Store Pre-Defined Compressed Data Blocks | | |
| 3.4.3 | Support Logged Data Requirements | | |
| 3.4.4 | Support Database Management | | |
| 3.4.5 | Support Condition-based Exception Reporting Requirements | | |
| 3.5 | Data Exchange and Operational Environment Requirements | | |
| 3.5.1 | ASC Configuration Management Requirements | | |
| 3.5.1.1 | Manage ASC Information Requirements | | |
| 3.5.1.1.1 | Configure ASC Location - Antenna Offset | | |
| | | C.3.2.4 | Configure ASC Antenna Offset |
| 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 | | N/A |
| 3.5.1.2.2 | Retrieve Communications Requirements | | |
| 3.5.1.2.2.1 | Determine Number of ASC Communications Ports | | |
| | | C.3.9.1 | Determine Number of ASC Communications Ports |
| 3.5.1.3 | Manage Cabinet Environment Requirements | | |
| 3.5.1.4 | Monitor Power Source Requirements | | |

| 3.5.1.5 | Manage Operational Performance Data Requirements | | |
|---------------|--|----------|--------------------------------------|
| 3.5.1.6 | Manage Auxiliary External Inputs/Outputs Requirements | | |
| 3.5.1.7 | Manage Database Operations | | |
| 3.5.1.7.1 | Determine Configuration Identifier Parameter Content | | |
| 3.5.1.8 | Manage Interface with External Detectors Requirements | | |
| 3.5.1.9 | Manage ASC Clock Requirements | | |
| 3.5.1.10 | Manage External Control Local Application State Requirements | | |
| 3.5.1.10.1 | Manage ECLA Interface Requirements | | |
| 3.5.1.10.1.1 | Enable ECLA Communications | | |
| | | C.3.9.18 | Enable/Disable ECLA Communications |
| 3.5.1.10.1.2 | Disable ECLA Communications | | |
| | | C.3.9.18 | Enable/Disable ECLA Communications |
| | | C.3.12.1 | Enable/Disable Manual Backup |
| 3.5.1.10.2 | Monitor ECLA Interface Data Input Time | | N/A |
| 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 | | |
| 3.5.2.1.1.1.1 | Configure Start-Up Flash Mode | | |
| | | C.3.2.1 | Configure Startup All-Red Flash Mode |
| 3.5.2.1.1.1.2 | Configure Start-Up Flash Time | | |
| | | C.3.2.2 | Configure Startup Flash Time |
| 3.5.2.1.1.2 | Configure Backup Time | | |
| | | C.3.2.3 | Configure Backup Time |
| 3.5.2.1.2 | Manage Phase Configuration Requirements | | |
| 3.5.2.1.2.1 | Configure Phases Requirements | | |

| 3.5.2.1.2.1.1 | Enable/Disable Phase | | |
|----------------|--|----------|--|
| | | C.3.3.1 | Enable/Disable Phase |
| 3.5.2.1.2.1.2 | Configure Phase Minimum Green Time | | |
| | | C.3.3.2 | Configure Phase Minimum Green Time |
| 3.5.2.1.2.1.3 | Configure Phase Passage Time | | |
| | | C.3.3.3 | Configure Phase Passage Time |
| 3.5.2.1.2.1.4 | Configure Two Phase Maximum Green Times | | |
| | | C.3.3.4 | Configure Two Phase Maximum Green Times |
| 3.5.2.1.2.1.5 | Configure Three Phase Maximum Green Times | | |
| | | C.3.3.4 | Configure Two Phase Maximum Green Times |
| | | C.3.3.5 | Configure Three Phase Maximum Green Times |
| 3.5.2.1.2.1.6 | Configure Phase Yellow Change Time | | |
| | | C.3.3.6 | Configure Phase Yellow Change Time |
| 3.5.2.1.2.1.7 | Configure Phase Red Clearance Time | | |
| | | C.3.3.7 | Configure Phase Red Clearance Time |
| 3.5.2.1.2.1.8 | Configure Phase Red Revert Time | | |
| | | C.3.3.8 | Configure Phase Red Revert Time |
| 3.5.2.1.2.1.9 | Configure Unit Red Revert Time | | |
| | | C.3.3.9 | Configure Unit Red Revert Time |
| 3.5.2.1.2.1.10 | Configure Phase Added Initial Time | | |
| | | C.3.3.10 | Configure Phase Added Initial Time |
| 3.5.2.1.2.1.11 | Configure Phase Maximum Initial Time | | |
| | | C.3.3.11 | Configure Phase Maximum Initial Time |
| 3.5.2.1.2.1.12 | Configure Phase Time Before Reduction | <u> </u> | |
| | | C.3.3.12 | Configure Phase Time Before Reduction |
| 3.5.2.1.2.1.13 | Configure Phase Time to Reduce | | |
| | | C.3.3.13 | Configure Phase Time to Reduce |
| 3.5.2.1.2.1.14 | Configure Phase Cars Before Reduction | | |
| | | C 3.3.14 | Configure Cars Before Reduction |

| 3.5.2.1.2.1.15 | Configure Phase Reduce By Time | | |
|----------------|---|----------|---------------------------------------|
| | | C.3.3.15 | Configure Phase Reduce By Time |
| 3.5.2.1.2.1.16 | Configure Phase Minimum Gap Time | | |
| | | C.3.3.16 | Configure Phase Minimum Gap Time |
| 3.5.2.1.2.1.17 | Configure Phase Dynamic Maximum Limit | | |
| | | C.3.3.17 | Configure Phase Dynamic Maximum Limit |
| 3.5.2.1.2.1.18 | Configure Phase Dynamic Maximum Step | | |
| | | C.3.3.18 | Configure Phase Dynamic Maximum Step |
| 3.5.2.1.2.1.19 | Configure Phase Startup State | | |
| | | C.3.3.19 | Configure Phase Startup State |
| 3.5.2.1.2.1.20 | Configure Automatic Flash Entry Phase | | |
| | | C.3.3.20 | Configure Automatic Flash Entry Phase |
| 3.5.2.1.2.1.21 | Configure Automatic Flash Exit Phase | | |
| | | C.3.3.21 | Configure Automatic Flash Exit Phase |
| 3.5.2.1.2.1.22 | Configure Call to Non- Actuated 1 | | |
| | | C.3.3.22 | Configure Call to Non-Actuated 1 |
| 3.5.2.1.2.1.23 | Configure Call to Non- Actuated 2 | | |
| | | C.3.3.23 | Configure Call to Non-Actuated 2 |
| 3.5.2.1.2.1.24 | Configure Non-Lock Detector Memory | | |
| | | C.3.3.24 | Configure Non-Lock Detector Memory |
| 3.5.2.1.2.1.25 | Configure Phase Minimum Vehicle Recall | | |
| | | C.3.3.25 | Configure Minimum Recall |
| 3.5.2.1.2.1.26 | Configure Phase Maximum Vehicle Recall | | |
| | | C.3.3.26 | Configure Maximum Vehicle Recall |
| 3.5.2.1.2.1.27 | Configure Phase Soft Vehicle Recall | | |
| | | C.3.3.27 | Configure Soft Recall |
| 3.5.2.1.2.1.28 | Configure Dual Phase Entry | | |
| | | C.3.3.28 | Configure Dual Phase Entry |
| 3.5.2.1.2.1.29 | Configure Simultaneous Gap Disable | | |
| | | C.3.3.29 | Configure Simultaneous Gap Disable |

| 3.5.2.1.2.1.30 | Configure Guaranteed Passage | | |
|----------------|--|----------|---|
| | | C.3.3.30 | Configure Guaranteed Passage |
| 3.5.2.1.2.1.31 | Configure Actuated Rest-in- Walk | | |
| | | C.3.3.31 | Configure Actuated Rest-in-Walk |
| 3.5.2.1.2.1.32 | Configure Conditional Service Enable | | |
| | | C.3.3.32 | Configure Conditional Service Enable |
| 3.5.2.1.2.1.33 | Configure Added Initial Calculation | | |
| | | C.3.3.33 | Configure Added Initial Calculation |
| 3.5.2.1.2.1.34 | Configure Phase-to-Ring Association | | |
| | | C.3.3.34 | Configure Phase-to-Ring Association |
| 3.5.2.1.2.1.35 | Configure Phase Concurrency | | |
| | | C.3.3.35 | Configure Phase Concurrency |
| 3.5.2.1.2.1.36 | Configure Pedestrian Clearance Time Allowed During Vehicle Clearance | | |
| | | C.3.3.36 | Configure Pedestrian Clearance Time Allowed During Vehicle Clearance |
| 3.5.2.1.2.1.37 | Configure Pedestrian Walk Time | | |
| | | C.3.3.37 | Configure Pedestrian Walk Time |
| 3.5.2.1.2.1.38 | Configure Pedestrian Clearance Time | | |
| | | C.3.3.38 | Configure Pedestrian Clearance Time |
| 3.5.2.1.2.1.39 | Configure Pedestrian Phase Walk Service Limit | | |
| | | C.3.3.39 | Configure Pedestrian Clearance Time |
| 3.5.2.1.2.1.40 | Configure Pedestrian Phase Don't Walk Revert Time | | |
| | | C.3.3.40 | Configure Pedestrian Phase Don't Walk Revert Time |
| 3.5.2.1.2.1.41 | Configure Non-Lock Ped Detector Memory | | |
| | | C.3.3.24 | Configure Non-Lock Detector Memory |
| 3.5.2.1.2.1.42 | Configure Pedestrian Phase Recall | | |
| | | C.3.3.41 | Configure Pedestrian Phase Recall |
| 3.5.2.1.2.1.43 | Configure Phase Alternate Pedestrian Clearance Time | | |
| | | C.3.3.42 | Configure Phase Alternate Pedestrian Clearance Time |

| 3.5.2.1.2.1.44 | Configure Alternate Pedestrian Walk Time | | |
|----------------|---|----------|--|
| | | C.3.3.43 | Configure Phase Alternate Pedestrian Walk Time |
| 3.5.2.1.2.1.45 | Configure Pedestrian Phase Advanced Walk Time | | |
| | | C.3.3.44 | Configure Pedestrian Phase Advanced Walk Time |
| 3.5.2.1.2.1.46 | Configure Pedestrian Phase Delayed Walk Time | | |
| | | C.3.3.45 | Configure Pedestrian Phase Delayed Walk Time |
| 3.5.2.1.2.1.47 | Configure Phase Advance Warning Green | | |
| | | C.3.3.46 | Configure Phase Advance Warning Green |
| 3.5.2.1.2.1.48 | Configure Phase Advance Warning Red | | |
| | | C.3.3.47 | Configure Phase Advance Warning Red |
| 3.5.2.1.2.1.49 | Configure Flashing Yellow Arrow Associated Vehicle Phase | | |
| | | C.3.7.11 | Configure Overlap Included Phases |
| | | C.3.7.12 | Configure Overlap Modifier Phases |
| 3.5.2.1.2.1.50 | Configure Flashing Red Arrow Associated Vehicle Phase | | |
| | | C.3.7.11 | Configure Overlap Included Phases |
| | | C.3.7.12 | Configure Overlap Modifier Phases |
| 3.5.2.1.2.1.51 | Configure Alternate Minimum Vehicle Green Time during Transition | | |
| | | C.3.3.48 | Configure Alternate Minimum Vehicle Green Time during Transition |
| 3.5.2.1.2.1.52 | Configure Alternate Minimum Pedestrian Walk Time during Transition | | |
| | | C.3.3.49 | Configure Alternate Minimum Pedestrian Walk Time during Transition |
| 3.5.2.1.2.1.53 | Configure Alternate Minimum Pedestrian Clearance Time during Transition | | |
| | | C.3.3.50 | Configure Alternate Minimum Pedestrian Clearance Time during Transition |
| 3.5.2.1.2.2 | Configure Multiple Phase Sets | | N/A |
| 3.5.2.1.2.3 | Retrieve Phase Configuration Requirements | | |

| 3.5.2.1.2.3.1 | Determine Maximum Number of Phases | | |
|---------------|--|----------|--|
| | | C.3.3.51 | Determine Maximum Number of Phases |
| 3.5.2.1.2.3.2 | Determine Maximum Number of Phase Sets | | |
| | | C.3.3.52 | Determine Maximum Number of Phase Sets |
| 3.5.2.1.3 | Manage Coordination Configuration Requirements | | |
| 3.5.2.1.3.1 | Configuration Operational Mode for Coordination | | |
| | | C.3.4.1 | Configure Operational Mode for Coordination - Automatic |
| 3.5.2.1.3.2 | Configuration Correction Mode for Coordination | | |
| | | C.3.4.2 | Configure Correction Mode for Coordination |
| 3.5.2.1.3.3 | Configure Maximum Mode for Coordination | | |
| | | C.3.4.3 | Configure Correction Mode for Coordination |
| 3.5.2.1.3.4 | Configure Unit-Level Force Mode for Coordination | | |
| | | C.3.4.4 | Configure Unit-level Force Mode for Coordination |
| 3.5.2.1.3.5 | Configure Phase-Level Force Mode for Coordination | | |
| | | C.3.4.5 | Configure Phase-Level Force Mode for Coordination |
| 3.5.2.1.3.6 | Configure Pattern Reference Phase | | |
| | | C.3.4.6 | Configure Pattern Reference Phase |
| 3.5.2.1.3.7 | Configure Pattern Reference Point | | |
| | | C.3.4.7 | Configure Pattern Reference Point |
| 3.5.2.1.3.8 | Configure Omit Phases During Transitions | | |
| | | C.3.4.8 | Configure Omit Phases During Transitions |
| 3.5.2.1.3.9 | Configure Pattern Synchronization Time | | |
| | | C.3.4.9 | Configure Pattern Synchronization Time |
| 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 | | |
| | | C.3.4.10 | Configure Pattern to Run Free |

| 3.5.2.1.4.1.2 | Configure Pattern to Run Automatic Flash | | |
|----------------|--|----------|--|
| | | C.3.4.11 | Configure Pattern to Run Automatic Flash |
| 3.5.2.1.4.1.3 | Configure Pattern Cycle Time for Coordination | | |
| | | C.3.4.12 | Configure Pattern Cycle Time for Coordination |
| 3.5.2.1.4.1.4 | Configure Pattern Offset Time | | |
| | | C.3.4.13 | Configure Pattern Offset Time |
| 3.5.2.1.4.1.5 | Configure Pattern Split Association | | |
| | | C.3.4.14 | Configure Pattern Split Association |
| 3.5.2.1.4.1.6 | Configure Pattern Sequence Association | | |
| | | C.3.4.15 | Configure Pattern Sequence Association |
| 3.5.2.1.4.1.7 | Configure Pattern Maximum Mode | | |
| | | C.3.4.16 | Configure Pattern Maximum Mode |
| 3.5.2.1.4.1.8 | Configure Pattern Phase Set | | |
| | | C.3.4.17 | Configure Pattern Phase Set |
| 3.5.2.1.4.1.9 | Configure Pattern Overlap Set | | |
| | | C.3.4.18 | Configure Pattern Overlap Set |
| 3.5.2.1.4.1.10 | Configure Pattern Vehicle Detector Set | | |
| | | C.3.4.19 | Configure Pattern Vehicle Detector Set |
| 3.5.2.1.4.1.11 | Configure Pattern Pedestrian Detector Set | | |
| | | C.3.4.20 | Configure Pattern Pedestrian Detector Set |
| 3.5.2.1.4.1.12 | Configure Pattern Special Functions | | |
| | | C.3.4.21 | Configure Pattern Special Functions |
| 3.5.2.1.4.1.13 | Determine Maximum Number of Timing Patterns | | |
| | | C.3.4.22 | Determine Maximum Number of Timing Patterns |
| 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 | | |
| | | C.3.4.23 | Configure Phase Split Time |
| 3.5.2.1.5.1.2 | Configure Phase Split Mode | | |
| | | C.3.4.24 | Configure Phase Split Mode |

| 3.5.2.1.5.1.3 | Configure Split Coordination Phase | | |
|-----------------|--|----------|--|
| | | C.3.4.25 | Configure Split Coordinated Phase |
| 3.5.2.1.5.2 | Determine Maximum Number of Phase Splits | | |
| | | C.3.4.26 | Determine Maximum Number of Phase Splits |
| 3.5.2.1.6 | Manage Ring Configuration Requirements | | |
| 3.5.2.1.6.1 | Configure Sequence Data | | |
| | | C.3.5.1 | Configure Sequence Data |
| 3.5.2.1.6.2 | Determine Maximum Number of Rings | | |
| | | C.3.5.2 | Determine Maximum Number of Rings |
| 3.5.2.1.6.3 | Determine Maximum Number of Sequences | | |
| | | C.3.5.2 | Determine Maximum Number of Sequences |
| 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 | | |
| | | C.3.6.1 | Configure Channel Control Source |
| 3.5.2.1.7.1.2 | Configure Channel Control Type | | |
| | | C.3.6.2 | Configure Channel Control Type |
| 3.5.2.1.7.1.3 | Configure Channel Flash Requirements | | |
| 3.5.2.1.7.1.3.1 | Configure Channel Flash Yellow | | |
| | | C.3.6.3 | Configure Channel Flash Yellow |
| 3.5.2.1.7.1.3.2 | Configure Channel Flash Red | | |
| | | C.3.6.4 | Configure Channel Flash Red |
| 3.5.2.1.7.1.3.3 | Configure Channel Flash Alternate Half Hertz | | |
| | | C.3.6.5 | Configure Channel Flash Alternate Half Hertz |
| 3.5.2.1.7.1.3.4 | Configure Channel Flash Alternate First or Second | | |
| | | C.3.6.6 | Configure Channel Flash Alternate First or Second |
| 3.5.2.1.7.2 | Determine Maximum Number of Channels | | |
| | | C.3.6.7 | Determine Maximum Number of Channels Supported |

| 3.5.2.1.8 | Manage Overlap Configuration Requirements | | |
|-----------------|---|---------|--|
| 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 | | |
| | | C.3.7.1 | Configure Overlap Type - Normal |
| 3.5.2.1.8.1.1.2 | Configure Overlap Type - Vehicle Minus Green and Yellow | | |
| | | C.3.7.2 | Configure Overlap Type - Vehicle Minus Green and Yellow |
| 3.5.2.1.8.1.1.3 | Configure Overlap Type - Pedestrian Normal | | |
| | | C.3.7.3 | Configure Overlap Type - Pedestrian Normal |
| 3.5.2.1.8.1.1.4 | Configure Overlap Type - Flashing Yellow Arrow - 3 Section Head | | |
| | | C.3.7.4 | Configure Overlap Type - Flashing Yellow Arrow - 3 Section Head |
| 3.5.2.1.8.1.1.5 | Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head | | |
| | | C.3.7.5 | Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head |
| 3.5.2.1.8.1.1.6 | Configure Overlap Type - Flashing Red Arrow - 3 Section Head | | |
| | | C.3.7.6 | Configure Overlap Type - Flashing Red Arrow - 3 Section Head |
| 3.5.2.1.8.1.1.7 | Configure Overlap Type - Flashing Red Arrow - 4 Section Head | | |
| | | C.3.7.7 | Configure Overlap Type - Flashing Red Arrow - 4 Section Head |
| 3.5.2.1.8.1.1.8 | Configure Overlap Type - 2 Section Transit Specific Signal Head | | |
| | | C.3.7.8 | Configure Overlap Type - 2 Section Transit Specific Signal Head |
| 3.5.2.1.8.1.1.9 | Configure Overlap Type - Minus Green Yellow Alternate | | |
| | | C.3.7.9 | Configure Overlap Type - Minus Green Yellow Alternate |

| 3.5.2.1.8.1.2 | Configure Overlap Included Phases | | |
|---------------|---|----------|---|
| | | C.3.7.10 | Configure Overlap Included Phases |
| 3.5.2.1.8.1.3 | Configure Overlap Modifier Phases | | |
| | | C.3.7.11 | Configure Overlap Modifier Phases |
| 3.5.2.1.8.1.4 | Configure Pedestrian Modifier Phases | | |
| | | C.3.7.12 | Configure Pedestrian Modifier Phases |
| 3.5.2.1.8.1.5 | Configure Overlap Trailing Green | | |
| | | C.3.7.13 | Configure Overlap Trailing Green |
| 3.5.2.1.8.1.6 | Configure Overlap Trailing Yellow | | |
| | | C.3.7.14 | Configure Overlap Trailing Yellow |
| 3.5.2.1.8.1.7 | Configure Overlap Trailing Red Clearance | | |
| | | C.3.7.15 | Configure Overlap Trailing Red Clearance |
| 3.5.2.1.8.1.8 | Configure Overlap Walk | | |
| | | C.3.7.16 | Configure Overlap Walk |
| 3.5.2.1.8.1.9 | Configure Overlap Pedestrian Clearance | | |
| | | C.3.7.17 | Configure Overlap Pedestrian Clearance |
| 3.5.2.1.8.2 | Configure Multiple Overlap Sets | | N/A |
| 3.5.2.1.8.3 | Retrieve Overlaps Requirements | | |
| 3.5.2.1.8.3.1 | Determine Maximum Number of Overlaps | | |
| | | C.3.7.20 | Determine Maximum Number of Overlaps |
| 3.5.2.1.8.3.2 | Determine Maximum Number of Overlap Sets | | |
| | | C.3.7.19 | Determine Maximum Number of Overlap Sets |
| 3.5.2.1.9 | Manage Preempt Configuration Requirements | | |
| 3.5.2.1.9.1 | Configure Preempts Requirements | | |
| 3.5.2.1.9.1.1 | Enable/Disable Preempt Inputs | | |
| | | C.3.8.1 | Enable/Disable Preempt Inputs |
| 3.5.2.1.9.1.2 | Configure Preempt Control - Non-Locking Memory | | |
| | | C.3.8.2 | Configure Preempt Control - Non-Locking Memory |

| 3.5.2.1.9.1.3 | Configure Preempt Control - Override Automatic Flash | | |
|------------------|---|----------|--|
| | | C.3.8.3 | Configure Preempt Control - Preempt Override Flash |
| 3.5.2.1.9.1.4 | Configure Preempt Control - Override Preempt | | |
| | | C.3.8.4 | Configure Preempt Control - Preempt Override Priority |
| 3.5.2.1.9.1.5 | Configure Preempt Control - Flash Dwell | | |
| | | C.3.8.5 | Configure Preempt Control - Flash Dwell |
| 3.5.2.1.9.1.6 | Configure Preempt Control - All Red Entry | | |
| | | C.3.8.6 | Configure Preempt Control – All Red Entry |
| 3.5.2.1.9.1.7 | Configure Preempt Link | | |
| | | C.3.8.7 | Configure Preempt Link |
| 3.5.2.1.9.1.8 | Configure Preempt Delay | | |
| | | C.3.8.8 | Configure Preempt Delay |
| 3.5.2.1.9.1.9 | Configure Preempt Minimum Duration | | |
| | | C.3.8.9 | Configure Preempt Minimum Duration |
| 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 | | |
| | | C.3.8.10 | Configure Preempt Minimum Green Time |
| 3.5.2.1.9.1.10.2 | Configure Preempt Enter Minimum Walk Time | | |
| | | C.3.8.11 | Configure Preempt Minimum Walk Time |
| 3.5.2.1.9.1.10.3 | Configure Preempt Enter Pedestrian Clearance Time | | |
| | | C.3.8.12 | Configure Preempt Enter Pedestrian Clearance Time |
| 3.5.2.1.9.1.10.4 | Configure Preempt Enter Yellow Change Time | | |
| | | C.3.8.13 | Configure Preempt Enter Yellow Change Time |
| 3.5.2.1.9.1.10.5 | Configure Preempt Enter Red Clearance Time | | |
| | | C.3.8.14 | Configure Preempt Enter Red Clearance Time |
| 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 | | |

| | | C.3.8.15 | Configure Preempt Track Clearance Green Time |
|------------------|--|----------|---|
| 3.5.2.1.9.1.11.2 | Configure Preempt Track Clearance Yellow Change Time | | |
| | | C.3.8.16 | Configure Preempt Track Clearance Yellow Change Time |
| 3.5.2.1.9.1.11.3 | Configure Preempt Track Red Clearance Time | | |
| | | C.3.8.17 | Configure Preempt Track Red Clearance Time |
| 3.5.2.1.9.1.11.4 | Configure Preempt Track Clearance Phases | | |
| | | C.3.8.18 | Configure Preempt Track Clearance Phases |
| 3.5.2.1.9.1.11.5 | Configure Preempt Track Clearance Overlaps | | |
| | | C.3.8.19 | Configure Preempt Track Clearance Overlaps |
| 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 | | |
| | | C.3.8.20 | Configure Preempt Minimum Dwell Time |
| 3.5.2.1.9.1.12.2 | Configure Preempt Dwell Phases | | |
| | | C.3.8.21 | Configure Preempt Dwell Phases |
| 3.5.2.1.9.1.12.3 | Configure Preempt Dwell Pedestrian Movements | | |
| | | C.3.8.22 | Configure Preempt Dwell Pedestrian Movements |
| 3.5.2.1.9.1.12.4 | Configure Preempt Dwell Overlaps | | |
| | | C.3.8.23 | Configure Preempt Dwell Overlaps |
| 3.5.2.1.9.1.12.5 | Configure Preempt Cycling Phases | | |
| | | C.3.8.24 | Configure Preempt Cycling Phases |
| 3.5.2.1.9.1.12.6 | Configure Preempt Cycling Pedestrian Movements | | |
| | | C.3.8.25 | Configure Preempt Cycling Pedestrian Movements |
| 3.5.2.1.9.1.12.7 | Configure Preempt Cycling Phases Sequence | | |
| | | C.3.8.26 | Configure Preempt Cycling Phases Sequence |
| 3.5.2.1.9.1.12.8 | Configure Preempt Cycling Overlaps | | |
| | | C.3.8.27 | Configure Preempt Cycling Overlaps |

| 3.5.2.1.9.1.13 | Configure Preempt Exit Requirements | | |
|------------------|--|----------|---|
| 3.5.2.1.9.1.13.1 | Configure Preempt Exit Phases | | |
| | | C.3.8.28 | Configure Preempt Exit Phases |
| 3.5.2.1.9.1.13.2 | Configure Preempt Exit Phase Strategy | | |
| | | C.3.8.29 | Configure Preempt Exit Phase Strategy |
| 3.5.2.1.9.1.13.3 | Configure Preempt Exit Priority Levels | | |
| | | C.3.8.30 | Configure Preempt Exit Priority Levels |
| 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 | | |
| | | C.3.8.31 | Configure Preempt Maximum Presence Time |
| 3.5.2.1.9.1.14.2 | Configure Preempt Maximum Presence Action | | |
| | | C.3.8.32 | Configure Preempt Maximum Presence Action |
| 3.5.2.1.9.1.15 | Configure Preempt Gate Description | | |
| | | C.3.8.33 | Configure Preempt Gate Description |
| 3.5.2.1.9.2 | Determine Maximum Number of Preempts | | |
| | | C.3.8.34 | Determine Maximum Number of Preempts |
| 3.5.2.1.10 | Manage ASC Scheduler Requirements | | |
| 3.5.2.1.10.1 | Configure ASC Timebased Action Requirements | | |
| 3.5.2.1.10.1.1 | Configure Timebased Action - Pattern | | |
| | | C.3.4.27 | Configure Timebased Action - Pattern |
| 3.5.2.1.10.1.2 | Configure Timebased Action - Special Functions | | |
| | | C.3.4.28 | Configure Timebased Action – Special Functions |
| 3.5.2.1.10.1.3 | Determine Maximum Number of Timebased Actions | | |
| | | C.3.4.29 | Determine Maximum Number of Timebased Actions |
| 3.5.2.1.10.1.4 | Determine Action In Effect | | |
| | | C.3.4.30 | Determine Action In Effect |

| 3.5.2.1.10.1.5 | Activate Action Plan Remotely | | |
|--------------------|---|----------|---|
| | | C.3.4.31 | Activate Action Plan Remotely |
| 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 | | N/A |
| 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 | | N/A |
| 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 | | N/A |
| 3.5.2.1.11.1.2.2.2 | Configure I/O Map Input Device Pin | | N/A |
| 3.5.2.1.11.1.2.2.3 | Configure I/O Map Input Function | | N/A |
| 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 | | N/A |
| 3.5.2.1.11.1.2.3.2 | Configure I/O Map Output Device Pin | | N/A |
| 3.5.2.1.11.1.2.3.3 | Configure I/O Map Output Function | | N/A |
| 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 | | |
| | | C.3.9.2 | Retrieve Maximum Number of I/O Maps |
| 3.5.2.1.11.2.2 | Retrieve Maximum Number of I/O Map Inputs | | |
| | | C.3.9.3 | Retrieve Maximum Number of I/O Map Inputs |
| 3.5.2.1.11.2.3 | Retrieve Maximum Number of I/O Map Outputs | | |
| | | C.3.9.4 | Retrieve Maximum Number of I/O Map Outputs |
| 3.5.2.1.11.2.4 | Retrieve I/O Mapping Activate Conditions | | |
| | | C.3.9.5 | Retrieve I/O Mapping Activate Conditions |
| 3.5.2.1.11.2.5 | Retrieve I/O Mapping Input Functions | | |
| | | C.3.9.6 | Retrieve I/O Mapping Input Functions |

| 3.5.2.1.11.2.6 | Retrieve I/O Mapping Output Functions | | |
|-------------------|--|----------|---|
| | | C.3.9.7 | Retrieve I/O Mapping Output Functions |
| 3.5.2.1.11.2.7 | Retrieve I/O Map Input Device Pin Status | | |
| | | C.3.9.8 | Retrieve I/O Map Input Device Pin Status |
| 3.5.2.1.11.2.8 | Retrieve I/O Map Output Device Pin Status | | |
| | | C.3.9.9 | Retrieve I/O Map Output Device Pin Status |
| 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 | | N/A |
| 3.5.2.1.11.2.9.2 | Enumerate I/O Map - FIO Outputs | | N/A |
| 3.5.2.1.11.2.9.3 | Enumerate I/O Map - TS1 Inputs | | N/A |
| 3.5.2.1.11.2.9.4 | Enumerate I/O Map - TS1 Outputs | | N/A |
| 3.5.2.1.11.2.9.5 | Enumerate I/O Map - TS2 BIU Inputs | | N/A |
| 3.5.2.1.11.2.9.6 | Enumerate I/O Map - TS2 BIU Outputs | | N/A |
| 3.5.2.1.11.2.9.7 | Enumerate I/O Map - ATC Cabinet SIU Inputs | | N/A |
| 3.5.2.1.11.2.9.8 | Enumerate I/O Map - ATC Cabinet SIU Outputs | | N/A |
| 3.5.2.1.11.2.9.9 | Enumerate I/O Map - Auxiliary Device Inputs | | N/A |
| 3.5.2.1.11.2.9.10 | Enumerate I/O Map - Auxiliary Device Outputs | | N/A |
| 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 | | |
| | | C.3.9.10 | Determine Serial Bus 1 Device Present |
| 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 | | |
| | | C.3.9.11 | Determine TS2 Port 1 Device Present |

| 3.5.2.1.12.2.2 | Enable/Disable TS2 Port 1 Frame 40 Messages | | |
|----------------|---|-----------|--|
| | | C.3.9.12 | Determine TS2 Port 1 Frame 40 Enable |
| | | | |
| 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 | | |
| | | | |
| 3.5.2.1.13.1.2 | Configure APS Push Button to Phase Association | | |
| | | C.3.13.19 | Configure Pedestrian Detector Call Phase |
| 3.5.2.1.13.1.3 | Configure APS Extra Crossing Time | | |
| | | C.3.3.42 | Configure Alternate Pedestrian Clearance Time |
| | | C.3.3.43 | Configure Alternate Pedestrian Walk Time |
| 3.5.2.1.13.1.4 | Configure Pedestrian Detector for Alternate Pedestrian Timing | | |
| | | C.3.13.24 | Configure Pedestrian Detector for Alternate Pedestrian Timing |
| 3.5.2.1.14 | Manage Block Object Requirements | | |
| 3.5.2.1.14.1 | Configure Block Object Requirements | | |
| 3.5.2.1.14.1.1 | Configure Block Object Get Control - Phase Data | | |
| | | C.3.10.1 | Configure Phase Data Block |
| 3.5.2.1.14.1.2 | Configure Block Object Get Control - Vehicle Detector Data | | |
| | | C.3.10.2 | Configure Vehicle Detector Block |
| 3.5.2.1.14.1.3 | Configure Block Object Get Control - Pedestrian Detector Data | | |
| | | C.3.10.3 | Configure Pedestrian Detector Block |
| 3.5.2.1.14.1.4 | Configure Block Object Get Control - Pattern Data | | |
| | | C.3.10.4 | Configure Pattern Data Block |
| 3.5.2.1.14.1.5 | Configure Block Object Get Control - Split Data | | |
| | | C.3.10.5 | Configure Split Data Block |

| 3.5.2.1.14.1.6 | Configure Block Object Get Control - Overlap Data | | |
|----------------|--|----------|---|
| | | C.3.10.6 | Configure Overlap Data Block |
| 3.5.2.1.14.1.7 | Configure Block Object Get Control - Preempt Data | | |
| | | C.3.10.7 | Configure Preempt Data Block |
| 3.5.2.1.14.1.8 | Configure Block Object Get Control - Sequence Data | | |
| | | C.3.10.8 | Configure Sequence Data Block |
| 3.5.2.1.14.1.9 | Configure Block Object Get Control - Channel Data | | |
| | | C.3.10.9 | Configure Channel Data Block |
| 3.5.2.1.14.2 | Monitor Block Error Status - Error-causing Data Element | | |
| | | C.3.10.1 | Monitor Block Error Status - Value Set Validity Error |
| 3.5.2.2 | Monitor Signal Operations Requirements | | |
| 3.5.2.2.1 | Determine Controller Health Requirements | | |
| 3.5.2.2.1.1 | Monitor External Alarm Input States | | |
| | | C.3.11.1 | Monitor External Alarm Input States |
| 3.5.2.2.1.2 | Monitor External Alarm Active | | |
| | | C.3.11.2 | Monitor Unit Level Alarms - Short |
| 3.5.2.2.1.3 | Monitor Flash Status | | |
| | | C.3.11.2 | Monitor Unit Level Alarms - Short |
| | | C.3.11.4 | Monitor Flash Status |
| 3.5.2.2.1.4 | Monitor Local Override | | |
| | _ | C.3.11.2 | Monitor Unit Level Alarms - Short |
| 3.5.2.2.1.5 | Monitor Coordination Alarm | | |
| | | C.3.11.2 | Monitor Unit Level Alarms - Short |
| 3.5.2.2.1.6 | Monitor Detector Fault | | |
| | | C.3.11.2 | Monitor Unit Level Alarms - Short |
| 3.5.2.2.1.7 | Monitor Stop Time Active | 0.0.44.0 | |
| 0.5.0.0.0 | | C.3.11.2 | Monitor Unit Level Alarms - Short |
| 3.5.2.2.1.8 | IVIONITOR Cycle Fault Alarm | 0.0.44.0 | Meniter Linit Lovel Alerra |
| 252240 | Monitor Coordination Fault | 0.3.11.3 | |
| 3.3.2.2.1.9 | | 0.2.44.2 | Monitor Linit Lovel Alerres |
| | Monitor Coordination Fail | 0.3.11.3 | |
| 3.5.2.2.1.10 | Alarm | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |

| 3.5.2.2.1.11 | Monitor Cycle Fail Alarm | | |
|---------------|--|----------|--|
| | | C.3.11.3 | Monitor Unit Level Alarms |
| 3.5.2.2.1.12 | Monitor Cabinet IO Link Alarm | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |
| 3.5.2.2.1.13 | Monitor SMU Communications Error | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |
| 3.5.2.2.1.14 | Monitor Preempt Maximum Presence Alarm | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |
| 3.5.2.2.2 | Retrieve Current Operation Requirements | | |
| 3.5.2.2.2.1 | Monitor Unit Control Status | | |
| | | C.3.11.5 | Monitor Unit Control Status |
| 3.5.2.2.2.2 | Monitor Preempt Active | | |
| | | C.3.11.2 | Monitor Unit Level Alarms - Short |
| 3.5.2.2.3 | Monitor Offset Transitioning | | |
| | | C.3.11.2 | Monitor Unit Level Alarms - Short |
| 3.5.2.2.2.4 | Monitor Priority Call Active | | |
| | | C.3.11.2 | Monitor Unit Level Alarms - Short |
| 3.5.2.2.2.5 | Monitor Local Free Status | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |
| | | C.3.11.6 | Monitor Local Free Status |
| 3.5.2.2.2.6 | Monitor Coordination Active | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |
| 3.5.2.2.7 | Monitor ECLA Active | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |
| 3.5.2.2.2.8 | Monitor Current Timing Pattern Requirements | | |
| 3.5.2.2.2.8.1 | Monitor Current Pattern Status | | |
| | | C.3.11.7 | Monitor Current Pattern Status |
| 3.5.2.2.2.8.2 | Monitor Current Pattern Command Source | | |
| | | C.3.11.8 | Monitor Current Pattern Command Source |
| 3.5.2.2.2.8.3 | Monitor Current Pattern Fault Status | | |
| | | C.3.11.9 | Monitor Current Pattern Fault Status |
| 3.5.2.2.2.9 | Monitor Current Timing Pattern Requirements | | |
| 3.5.2.2.2.9.1 | Monitor Coordination Cycle Status | | |

| | | C.3.11.10 | Monitor Coordination Cycle Status |
|---------------|--|-----------|---|
| 3.5.2.2.2.9.2 | Monitor Coordination Synchronization Status | | |
| | | C.3.11.11 | Monitor Coordination Synchronization Status |
| 3.5.2.2.9.3 | Monitor Current Offset | | |
| | | C.3.11.12 | Monitor Current Offset |
| 3.5.2.2.3 | Retrieve Current Signal Indication Requirements | | |
| 3.5.2.2.3.1 | Monitor Active Red Phases | | |
| | | C.3.11.13 | Monitor Active Red Phases |
| 3.5.2.2.3.2 | Monitor Active Yellow Phases | | |
| | | C.3.11.14 | Monitor Active Yellow Phases |
| 3.5.2.2.3.3 | Monitor Active Green Phases | | |
| | | C.3.11.15 | Monitor Active Green Phases |
| 3.5.2.2.3.4 | Monitor Active Don't Walk Phases | | |
| | | C.3.11.16 | Monitor Active Don't Walk Phases |
| 3.5.2.2.3.5 | Monitor Active Pedestrian Clearance Phases | | |
| | | C.3.11.17 | Monitor Active Pedestrian Clearance Phases |
| 3.5.2.2.3.6 | Monitor Active Walk Phases | | |
| | | C.3.11.18 | Monitor Active Walk Phases |
| 3.5.2.2.3.7 | Monitor Active On Phases | | |
| | | C.3.11.19 | Monitor Active On Phases |
| 3.5.2.2.3.8 | Monitor Next Phases | | |
| | | C.3.11.20 | Monitor Next Phases |
| 3.5.2.2.3.9 | Monitor Phase Vehicle Calls | | |
| | | C.3.11.21 | Monitor Phase Vehicle Calls |
| 3.5.2.2.3.10 | Monitor Phase Pedestrian Calls | | |
| | | C.3.11.22 | Monitor Phase Pedestrian Calls |
| 3.5.2.2.4 | Retrieve Current Ring Requirements | | |
| 3.5.2.2.4.1 | Monitor Ring Status | | |
| | | C.3.11.23 | Monitor Ring Status |
| 3.5.2.2.4.2 | Monitor Ring Termination Cause | | |
| | | C.3.11.23 | Monitor Ring Status |
| 3.5.2.2.4.3 | Monitor Current Phase On Time | | |
| | | C.3.11.24 | Monitor Current Phase On Time |

| 35225 | Retrieve Current Channel | | |
|-------------|---|-----------|--|
| 3.3.2.2.3 | Status Requirements | | |
| 3.5.2.2.5.1 | Monitor Active Red Channels | C.3.11.25 | |
| | | | Monitor Active Red Channels |
| 3.5.2.2.5.2 | Monitor Active Yellow Channels | C.3.11.26 | |
| | | | Monitor Active Yellow Channels |
| 3.5.2.2.5.3 | Monitor Active Green Channels | C.3.11.27 | |
| | | | Monitor Active Green Channels |
| 3.5.2.2.6 | Retrieve Current Overlap Status Requirements | | |
| 3.5.2.2.6.1 | Monitor Active Red Overlaps | | |
| | | C.3.11.28 | Monitor Active Red Overlaps |
| 3.5.2.2.6.2 | Monitor Active Yellow Overlaps | | |
| | | C.3.11.29 | Monitor Active Yellow Overlaps |
| 3.5.2.2.6.3 | Monitor Active Green Overlaps | | |
| | | C.3.11.30 | Monitor Active Green Overlaps |
| 3.5.2.2.6.4 | Monitor Active Overlap Flashing Yellow Arrows | | |
| | | C.3.11.31 | Monitor Active Overlap Flashing Yellow Arrows |
| 3.5.2.2.6.5 | Monitor Active Overlap Flashing Red Arrows | | |
| | | C.3.11.32 | Monitor Active Overlap Flashing Red Arrows |
| 3.5.2.2.7 | Retrieve Current Preempt Status Requirements | | |
| 3.5.2.2.7.1 | Monitor Currently Active Preempt | | |
| | | C.3.11.33 | Monitor Currently Active Preempt |
| 3.5.2.2.7.2 | Monitor Current Preempt Inputs | | |
| | | C.3.11.34 | Monitor Current Preempt Inputs |
| 3.5.2.2.7.3 | Monitor Current Preempt State | | |
| | | C.3.11.35 | Monitor Current Preempt State |
| 3.5.2.2.7.4 | Monitor Current Gate Status | | |
| | | C.3.11.36 | Monitor Current Gate Status |
| 3.5.2.2.8 | Retrieve Special Function Outputs Requirements | | |
| 3.5.2.2.8.1 | Determine Maximum Number of Special Functions | | |
| | | C.3.11.37 | Determine Maximum Number of Special Functions |

| 3.5.2.2.8.2 | Monitor Special Function Status | | |
|--------------|---|-----------|---|
| | | C.3.11.38 | Monitor Special Function Status |
| 3.5.2.2.8.3 | Monitor Special Function Control Source | | |
| | | C.3.11.39 | Monitor Special Function Control Source |
| 3.5.2.2.9 | Monitor Intra-Cabinet Communications Requirements | | |
| 3.5.2.2.9.1 | Monitor TS2 Port 1 Status | | |
| | | C.3.9.13 | Monitor TS2 Port 1 Status |
| 3.5.2.2.9.2 | Monitor TS2 Port 1 Fault Frame | | |
| | | C.3.9.14 | Monitor TS2 Port 1 Fault Frame |
| 3.5.2.2.9.3 | Monitor ATC Serial Bus 1 Status | | |
| | | C.3.9.15 | Monitor ATC Serial Bus 1 Status |
| 3.5.2.2.10 | Monitor Signal Monitoring Unit Requirements | | |
| 3.5.2.2.10.1 | Monitor Signal Monitoring Unit Channel Voltage | | |
| | | C.3.9.16 | Monitor Signal Monitoring Unit Channel Voltage |
| 3.5.2.2.10.2 | Monitor Signal Monitoring Unit Channel Current | | |
| | | C.3.9.17 | Monitor Signal Monitoring Unit Channel Current |
| 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 | | |
| | | C.3.12.1 | Enable/Disable Manual Backup |
| 3.5.2.3.1.2 | Control Global Minimum Recall | | |
| | | C.3.12.2 | Control Global Minimum Recall |
| 3.5.2.3.1.3 | Control Call to Non-Actuated | | |
| | | C.3.12.3 | Control Call to Non-Actuated 1 |
| 3.5.2.3.1.4 | Control Call to Non-Actuated 2 | | |
| | | C.3.12.4 | Control Call to Non-Actuated 2 |
| 3.5.2.3.1.5 | Control Walk Rest Modifier | | |
| | | C.3.12.5 | Control Walk Rest Modifier |
| 3.5.2.3.1.6 | Control Interconnect | | |
| | | C.3.12.6 | Control Interconnect |

| 3.5.2.3.2 | Command Timing Plan Requirements | | |
|-------------|---|-----------|--|
| 3.5.2.3.2.1 | Activate System Timing Pattern Remotely | | |
| | | C.3.12.7 | Activate System Timing Pattern Remotely |
| 3.5.2.3.2.2 | Control System Reference Point | | |
| | | C.3.12.8 | Control System Reference Point |
| 3.5.2.3.3 | Control Phases Requirements | | |
| 3.5.2.3.3.1 | Control Phase Omits | | |
| | | C.3.12.9 | Control Phase Omits |
| 3.5.2.3.3.2 | Control Pedestrian Phase Omits | | |
| | | C.3.12.10 | Control Pedestrian Phase Omits |
| 3.5.2.3.3.3 | Control Phase Holds | | |
| | | C.3.12.11 | Control Phase Holds |
| 3.5.2.3.3.4 | Control Phase Force Offs | | |
| | | C.3.12.12 | Control Phase Force Offs |
| 3.5.2.3.3.5 | Control Phase Vehicle Calls | | |
| | | C.3.12.13 | Control Phase Vehicle Calls |
| 3.5.2.3.3.6 | Control Phase Pedestrian Calls | | |
| | | C.3.12.14 | Control Phase Pedestrian Calls |
| 3.5.2.3.4 | Activate Preempt Remotely | | |
| | | C.3.12.15 | Activate Preempt Remotely |
| 3.5.2.3.5 | Control Ring Requirements | | |
| 3.5.2.3.5.1 | Control Ring Stop Time | | |
| | | C.3.12.16 | Control Ring Stop Time |
| 3.5.2.3.5.2 | Control Ring Force Offs | | |
| | | C.3.12.17 | Control Ring Force Offs |
| 3.5.2.3.5.3 | Control Ring Maximum 2 Time Settings | | |
| | | C.3.12.18 | Control Ring Maximum 2 Time Settings |
| 3.5.2.3.5.4 | Control Ring Maximum 3 Time Settings | | |
| | | C.3.12.19 | Control Ring Maximum 3 Time Settings |
| 3.5.2.3.5.5 | Control Ring Maximum Inhibit Settings | | |
| | | C.3.12.20 | Control Ring Maximum Inhibit Settings |
| 3.5.2.3.5.6 | Control Ring Pedestrian Recycle Settings | | |
| | - | C.3.12.21 | Control Ring Pedestrian Recycle Settings |

| 3.5.2.3.5.7 | Control Ring Red Rest Settings | | |
|-------------|---|-----------|--|
| | | C.3.12.22 | Control Ring Red Rest Settings |
| 3.5.2.3.5.8 | Control Ring Red Clearance Omit Settings | | |
| | | C.3.12.23 | Control Ring Red Clearance Omit Settings |
| 3.5.2.3.6 | Activate Special Function Remotely | | |
| | | C.3.12.24 | Activate Special Function Remotely |
| 3.5.2.3.7 | Remote Manual Control Requirements | | |
| 3.5.2.3.7.1 | Enable Remote Manual Control | | |
| | | C.3.12.25 | Remote Manual Control |
| 3.5.2.3.7.2 | Advance Interval During Remote Manual Control | | |
| | | C.3.12.25 | Remote Manual Control |
| 3.5.2.3.7.3 | Configure Manual Control Timeout | | |
| | | C.3.12.25 | Remote Manual Control |
| 3.5.2.3.7.4 | Enable/Disable Automatic Pedestrian Clearance Setting | | |
| | | C.3.12.26 | Enable/Disable Automatic Pedestrian Clearance Setting |
| 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 | | |
| | | C.3.13.1 | Configure Vehicle Detector Travel Mode |
| 3.5.3.1.1.2 | Configure Vehicle Detector Description | | |
| | | C.3.13.2 | Configure Vehicle Detector Description |
| 3.5.3.1.1.3 | Configure Vehicle Detector Yellow Lock Call Enabled | | |
| | | C.3.13.3 | Configure Vehicle Detector Yellow Lock Call Enabled |
| 3.5.3.1.1.4 | Configure Vehicle Detector Red Lock Call Enabled | | |
| | | C.3.13.4 | Configure Vehicle Detector Red Lock Call Enabled |
| 3.5.3.1.1.5 | Configure Vehicle Detector Passage Enabled | | |

| | | C.3.13.5 | Configure Vehicle Detector Passage Enabled |
|--------------|---|-----------|--|
| 3.5.3.1.1.6 | Configure Vehicle Detector Added Initial Time Enabled | | |
| | | C.3.13.6 | Configure Vehicle Detector Added Initial Time Enabled |
| 3.5.3.1.1.7 | Configure Vehicle Detector Queue Enabled | | |
| | | C.3.13.7 | Configure Vehicle Detector Queue Enabled |
| 3.5.3.1.1.8 | Configure Vehicle Detector Call Enabled | | |
| | | C.3.13.8 | Configure Vehicle Detector Call Enabled |
| 3.5.3.1.1.9 | Configure Vehicle Detector Call Phase | | |
| | | C.3.13.9 | Configure Vehicle Detector Call Phase |
| 3.5.3.1.1.10 | Configure Vehicle Detector Switch Phase | | |
| | | C.3.13.10 | Configure Vehicle Detector Switch Phase |
| 3.5.3.1.1.11 | Configure Vehicle Detector Delay Time | | |
| | | C.3.13.11 | Configure Vehicle Detector Delay Time |
| 3.5.3.1.1.12 | Configure Vehicle Detector Extend Time | | |
| | | C.3.13.12 | Configure Vehicle Detector Extend Time |
| 3.5.3.1.1.13 | Configure Vehicle Detector Queue Limit Time | | |
| | | C.3.13.13 | Configure Vehicle Detector Queue Limit Time |
| 3.5.3.1.1.14 | Configure Vehicle Detector No Activity Fault Time | | |
| | | C.3.13.14 | Configure Vehicle Detector No Activity Fault Time |
| 3.5.3.1.1.15 | Configure Vehicle Detector Maximum Presence Time | | |
| | | C.3.13.15 | Configure Vehicle Detector Maximum Presence Time |
| 3.5.3.1.1.16 | Configure Vehicle Detector Erratic Counts | | |
| | | C.3.13.16 | Configure Vehicle Detector Erratic Counts |
| 3.5.3.1.1.17 | Configure Vehicle Detector Fail Time | | |
| | | C.3.13.17 | Configure Vehicle Detector Fail Time |
| 3.5.3.1.2 | Configure Multiple Vehicle Detector Sets for Actuation | | N/A |
| 3.5.3.1.3 | Configure Pedestrian Detector Requirements | | |

| 3.5.3.1.3.1 | Configure Pedestrian Detector Description | | |
|-------------|---|-----------|--|
| | | C.3.13.18 | Configure Pedestrian Detector Description |
| 3.5.3.1.3.2 | Configure Pedestrian Detector Call Phase | | |
| | | C.3.13.19 | Configure Pedestrian Detector Call Phase |
| 3.5.3.1.3.3 | Configure Pedestrian Detector No Activity Fault Time | | |
| | | C.3.13.20 | Configure Pedestrian Detector No Activity Fault Time |
| 3.5.3.1.3.4 | Configure Pedestrian Detector Maximum Presence Fault Time | | |
| | | C.3.13.21 | Configure Pedestrian Detector Maximum Presence Fault Time |
| 3.5.3.1.3.5 | Configure Pedestrian Detector Erratic Counts | | |
| | | C.3.13.22 | Configure Pedestrian Detector Erratic Counts |
| 3.5.3.1.3.6 | Configure Pedestrian Detector Non-Lock Calls | | |
| | | C.3.13.23 | Configure Pedestrian Detector Non-Lock Calls |
| 3.5.3.1.3.7 | Configure Pedestrian Detector for Presence Detection | | |
| | | C.3.13.25 | Configure Pedestrian Detector for Presence Detection |
| 3.5.3.1.3.8 | Configure Pedestrian Detector for Delayed Walk | | |
| | | C.3.13.26 | Configure Pedestrian Detector for Delayed Walk |
| 3.5.3.1.3.9 | Configure Pedestrian Detector for Advanced Walk | | |
| | | C.3.13.27 | Configure Pedestrian Detector for Advanced Walk |
| 3.5.3.1.4 | Configure Multiple Pedestrian Detector Sets for Actuation | | N/A |
| 3.5.3.1.5 | Retrieve Detector Configuration Requirements | | |
| 3.5.3.1.5.1 | Determine Maximum Number of Vehicle Detectors | | |
| | | C.3.13.28 | Determine Maximum Number of Vehicle Detectors |
| 3.5.3.1.5.2 | Determine Maximum Number of Vehicle Detector Sets | | |

| | | C.3.13.29 | Determine Maximum Number of Vehicle Detector Sets |
|-------------|--|-----------|---|
| 3.5.3.1.5.3 | Determine Maximum Number of Pedestrian Detectors | | |
| | | C.3.13.30 | Determine Maximum Number of Pedestrian Detectors |
| 3.5.3.1.5.4 | Determine Maximum Number of Pedestrian Detector Sets | | |
| | | C.3.13.31 | Determine Maximum Number of Pedestrian Detector Sets |
| 3.5.3.2 | Retrieve Detector Status Requirements | | |
| 3.5.3.2.1 | Monitor Active Vehicle Detector Actuations | | |
| | | C.3.13.32 | Monitor Active Vehicle Detector Actuations |
| 3.5.3.2.2 | Monitor Active Pedestrian Detector Actuations | | |
| | | C.3.13.33 | Monitor Active Pedestrian Detector Actuations |
| 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 | | |
| | | C.3.13.34 | Monitor Vehicle Detector Alarm Status |
| 3.5.3.3.1.2 | Monitor Vehicle Detector Faults from Controller | | |
| | | C.3.13.35 | Monitor Vehicle Detector Faults from Controller |
| 3.5.3.3.1.3 | Monitor Vehicle Detector Faults from Detector | | |
| | | C.3.13.36 | Monitor Vehicle Detector Faults from Detector |
| 3.5.3.3.2 | Retrieve Pedestrian Detector Health Requirements | | |
| 3.5.3.3.2.1 | Monitor Pedestrian Detector Alarm Status | | |
| | | C.3.13.37 | Monitor Pedestrian Detector Alarm Status |
| 3.5.3.3.2.2 | Monitor Pedestrian Detector Faults | | |
| | | C.3.13.38 | Monitor Pedestrian Detector Faults |
| 3.5.3.4 | Control Detector Requirements | | |
| 3.5.3.4.1 | Control Vehicle Detector Reset | | |

| | | C.3.13.39 | Control Vehicle Detector Reset |
|-------------|--|-----------|--|
| 3.5.3.4.2 | Control Pedestrian Detector Reset | | |
| | | C.3.13.40 | Control Pedestrian Detector Reset |
| 3.5.3.4.3 | Control Detector Diagnostic Reset | | |
| | | C.3.13.41 | Control Detector Diagnostic Reset |
| 3.5.3.4.4 | Control Vehicle Detector Actuation | | |
| | | C.3.13.42 | Control Vehicle Detector Actuation |
| 3.5.3.4.5 | Control Pedestrian Detector Actuation | | |
| | | C.3.13.43 | Control Pedestrian Detector Actuation |
| 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 | | |
| | | C.3.14.1 | Monitor Vehicle Detector Data Sequence |
| 3.5.3.5.1.2 | Monitor Vehicle Volume Data | | |
| | | C.3.14.2 | Monitor Vehicle Volume Data |
| 3.5.3.5.1.3 | Monitor Vehicle Occupancy Data | | |
| | | C.3.14.3 | Monitor Vehicle Occupancy Data |
| 3.5.3.5.1.4 | Monitor Vehicle Average Speed | | |
| | | C.3.14.4 | Monitor Vehicle Average Speed |
| 3.5.3.5.1.5 | Monitor Vehicle Detector Data Sample Time | | |
| | | C.3.14.5 | Monitor Vehicle Detector Data Sample Time |
| 3.5.3.5.1.6 | Monitor Vehicle Detector Data Sample Duration | | |
| | | C.3.14.6 | Monitor Vehicle Detector Data Sample Duration |
| 3.5.3.5.2 | Monitor Pedestrian Detector Data Requirements | | |
| 3.5.3.5.2.1 | Monitor Pedestrian Detector Data Sequence | | |
| | | C.3.14.7 | Monitor Pedestrian Detector Data Sequence |
| 3.5.3.5.2.2 | Monitor Pedestrian Counts | | |
| | | C.3.14.8 | Monitor Pedestrian Counts |
| 3.5.3.5.2.3 | Monitor Pedestrian Actuations | | |
| | | C.3.14.9 | Monitor Pedestrian Actuations |

| 3.5.3.5.2.4 | Monitor Pedestrian Services | 1 | |
|-------------|--|-----------|---|
| | | C.3.14.10 | Monitor Pedestrian Services |
| 3.5.3.5.2.5 | Monitor Pedestrian Detector Data Sample Time | | |
| | | C.3.14.11 | Monitor Pedestrian Detector Data Sample Time |
| 3.5.3.5.2.6 | Monitor Pedestrian Detector Data Sample Duration | | |
| | | C.3.14.12 | Monitor Pedestrian Detector Data Sample Duration |
| 3.5.3.5.3 | Configure Detector Data Collection Requirements | | |
| 3.5.3.5.3.1 | Configure Vehicle Detector Data Sample Period | | |
| | | C.3.14.13 | Configure Vehicle Detector Data Sample Period |
| 3.5.3.5.3.2 | Configure Pedestrian Detector Data Sample Period | | |
| | | C.3.14.14 | Configure Pedestrian Detector Data Sample Period |
| 3.5.3.5.3.3 | Configure Vehicle Speed Detectors | | |
| | | C.3.14.15 | Configure Vehicle Speed Detectors |
| 3.5.3.5.3.4 | Configure Single Detector Speed Mode | | |
| | | C.3.14.16 | Configure Single Detector Speed Mode |
| 3.5.3.5.3.5 | Configure Paired Detector | | |
| | | C.3.14.17 | Configure Paired Detector |
| 3.5.3.5.3.6 | Configure Paired Detector Placement | | |
| | | C.3.14.18 | Configure Paired Detector Placement |
| 3.5.3.5.3.7 | Configure Paired Detector Spacing | | |
| | | C.3.14.19 | Configure Paired Detector Spacing |
| 3.5.3.5.3.8 | Configure Average Vehicle Length | | |
| | | C.3.14.20 | Configure Average Vehicle Length |
| 3.5.3.5.3.9 | Configure Vehicle Detection Zone Length | | |
| | | C.3.14.21 | Configure Vehicle Detection Zone Length |
| 3.3.3.5.4 | Configure Multiple Vehicle Detector Sets for Data Collection | | N/A |
| 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 | | |
| | | C.3.15.1 | Configure ASC Communications Port for RSU |
| 3.5.4.1.2 | Configure Logical RSU Ports and Address | | |
| | | C.3.15.2 | Configure Logical RSU Ports and Address |
| 3.5.4.1.3 | Configure RSU Interface Polling Period | | |
| | | C.3.15.3 | Configure RSU Interface Polling Period |
| 3.5.4.1.4 | Configure RSU Interface Watchdog | | |
| | | C.3.15.4 | Configure RSU Interface Watchdog |
| 3.5.4.1.5 | Monitor RSU Interface Watchdog Timer | | |
| | | C.3.15.5 | Monitor RSU Interface Watchdog Timer |
| 3.5.4.1.6 | Monitor RSU Interface Watchdog Alarm | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |
| 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 | C.3.15.6 | |
| | | | Enable Signal Phase and Timing Data |
| 3.5.4.2.1.2 | Retrieve Signal Phase and Timing Generation Time | | |
| | | C.3.15.7 | Retrieve Signal Phase and Timing Generation Time |
| 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 | | |
| | | C.3.15.8 | Monitor CV Movement Minimum End Time |
| 3.5.4.2.1.3.1.2 | Monitor Movement Maximum End Time | | |
| | | C.3.15.9 | Monitor CV Movement Maximum End Time |
| 3.5.4.2.1.3.1.3 | Monitor Movement Likely End Time | | |
| | | C.3.15.10 | Monitor CV Movement Expected End Time |

| 3.5.4.2.1.3.1.4 | Monitor Movement Likely End Time Confidence | | |
|------------------|---|-----------|---|
| | | C.3.15.11 | Monitor CV Movement Likely End Time Confidence |
| 3.5.4.2.1.3.1.5 | Monitor Movement Next Occurrence | | |
| | | C.3.15.12 | Monitor CV Movement Next Occurrence |
| 3.5.4.2.1.3.1.6 | Monitor Movement Start Time | | N/A |
| 3.5.4.2.1.3.1.7 | Monitor Next Movement Minimum End Time | | |
| | | C.3.15.13 | Monitor CV Next Movement Minimum End Time |
| 3.5.4.2.1.3.1.8 | Monitor Next Movement Maximum End Time | | |
| | | C.3.15.14 | Monitor CV Next Movement Maximum End Time |
| 3.5.4.2.1.3.1.9 | Monitor Next Movement Start Time | | |
| | | C.3.15.15 | Monitor Next CV Movement Start Time |
| 3.5.4.2.1.3.1.10 | Determine Maximum Number of Movement Events | | |
| | | C.3.15.16 | Determine Maximum Number of CV Movement Events |
| 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 | | |
| | | C.3.15.17 | Configure Queue Detectors for CV Movement Assistance |
| 3.5.4.2.1.3.2.2 | Configure Pedestrian Detectors for Movement Conflict Assistance | | |
| | | C.3.15.18 | Configure Pedestrian Detectors for CV Movement Conflict Assistance |
| 3.5.4.2.1.3.2.3 | Configure Bicycle Detectors for Movement Conflict Assistance | | |
| | | C.3.15.19 | Configure Bicycle Detectors for CV Movement Conflict Assistance |
| 3.5.4.2.1.3.3 | Retrieve Movement Assistance Requirements | | |
| 3.5.4.2.1.3.3.1 | Monitor Lane Connection Queue Length | | |
| | | C.3.15.20 | Monitor CV Lane Connection Queue Length |
| 3.5.4.2.1.3.3.2 | Monitor Lane Connection Vulnerable Road User Detection | | |

| | | C.3.15.21 | Monitor CV Lane Connection Vulnerable Road User Detection |
|-----------------|---|-----------|--|
| 3.5.4.2.1.3.4 | Manage Advisory Speed Requirements | | |
| 3.5.4.2.1.3.4.1 | Configure Advisory Speed Type | | |
| | | C.3.15.22 | Configure Advisory Speed Type |
| 3.5.4.2.1.3.4.2 | Configure Advisory Speed | | |
| | | C.3.15.23 | Configure Advisory Speed |
| 3.5.4.2.1.3.4.3 | Configure Advisory Speed Zone | | |
| | | C.3.15.24 | Configure Advisory Speed Zone |
| 3.5.4.2.1.3.4.4 | Configure Advisory Speed Vehicle Type | | |
| | | C.3.15.25 | Configure Advisory Speed Vehicle Type |
| 3.5.4.2.1.3.5 | Monitor Movement State | | |
| | | C.3.15.26 | Monitor CV Movement States |
| 3.5.4.2.1.3.6 | Monitor Next Movement State | | |
| | | C.3.15.26 | Monitor CV Movement States |
| 3.5.4.2.1.3.7 | Monitor Movement Status | | |
| | | C.3.15.27 | Monitor CV Movement Status |
| 3.5.4.2.1.4 | Manage Enabled Lane Requirements | | |
| 3.5.4.2.1.4.1 | Configure Concurrent Enabled Lanes | | |
| | | C.3.15.28 | Configure Concurrent Enabled Lanes |
| 3.5.4.2.1.4.2 | Configure Enabled Lanes by Time of Day | | |
| | | C.3.15.29 | Configure Enabled Lanes by Time of Day |
| 3.5.4.2.1.4.3 | Determine Lanes Enabled | | |
| | | C.3.15.30 | Determine Lanes Enabled |
| 3.5.4.2.1.4.4 | Command Enabled Lanes | | N/A |
| 3.5.4.2.1.5 | Enable Signal Phase and Timing Exchange | | |
| | | C.3.15.31 | Enable Signal Phase and Timing Data Exchange by Port |
| 3.5.4.2.1.6 | Configure Road Authority Identifier | | |
| | | C.3.15.32 | Configure Road Authority Identifier |
| 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 | | |

| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
|----------------|--|-----------|--|
| 3.5.4.2.1.7.2 | Monitor Stop Indication | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.3 | Monitor Failure Flash Indication | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.4 | Monitor Preemption Operation Indication | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.5 | Monitor Priority Operation Indication | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.6 | Monitor Fixed Time Control Indication | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.7 | Monitor Non-Fixed Time Control Indication | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.8 | Monitor Standby Operation Indication | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.9 | Monitor Controller Failure | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.10 | Monitor MAP Message Validity | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.7.11 | Monitor SPaT Data Validity | | |
| | | C.3.15.33 | Retrieve Signal Phase and Timing Data Intersection Status |
| 3.5.4.2.1.8 | Mark SPaT Invalid - Controller | | |
| | | C.3.15.34 | Mark SPaT Invalid - Controller |
| 3.5.4.2.1.9 | Mark SPaT Invalid - Port | | |
| | | C.3.15.35 | Mark SPaT Invalid - Port |
| 3.5.4.2.1.10 | Mark MAP Message Invalid - Controller | | |
| | | C.3.15.36 | Mark MAP Message Invalid - Controller |

| 3.5.4.2.1.11 | Mark MAP Message Invalid - Port | | |
|----------------|---|-----------|---|
| | | C.3.15.37 | Mark MAP Message Invalid - Port |
| 3.5.4.2.1.12 | Manage Signal Group Requirements | | |
| 3.5.4.2.1.12.1 | Determine Maximum Number of Signal Groups | | |
| | | C.3.15.38 | Determine Maximum Number of Signal Groups |
| 3.5.4.2.1.12.2 | Configure Signal Group Intersection Mapping | | |
| | | C.3.15.39 | Configure Signal Group Intersection Mapping |
| 3.5.4.2.1.12.3 | Configure Signal Group Control Source | | |
| | | C.3.15.40 | Configure Signal Group Control Source |
| 3.5.4.2.1.12.4 | Configure Signal Group Indication Types | | |
| | | C.3.15.41 | Configure Signal Group Indication Types |
| 3.5.4.2.1.12.5 | Configure Signal Group Protected or Permissive State | | |
| | | C.3.15.42 | Configure Signal Group Protected or Permissive State |
| 3.5.4.2.1.12.6 | Configure Signal Group Revocable Lanes | | |
| | | C.3.15.43 | Configure Signal Group Revocable Lanes |
| 3.5.4.2.1.12.7 | Determine Maximum Number of Signal State Entries | | |
| | | C.3.15.44 | Determine Maximum Number of Signal State Entries |
| 3.5.4.2.1.12.8 | Configure Customized Signal State Parameters | | |
| | | C.3.15.45 | Configure Customized Signal State Parameters |
| 3.5.4.2.1.13 | Retrieve Signal Phase and Timing Time Point | | |
| | | C.3.15.46 | Retrieve Signal Phase and Timing Time Point |
| 3.5.4.2.2 | Manage Connected Vehicle Detector Requirements | | |
| 3.5.4.2.2.1 | Enabled Connected Device Connection | | N/A |
| 3.5.4.2.2.2 | Configure Vehicle Detector for Connected Vehicle Applications | | N/A |
| 3.5.4.2.2.3 | Configure Connected Vehicle Detector Input Assignment | | N/A |

| 3.5.4.2.2.4 | Configure Connected Vehicle Detector Port Assignment | N/A |
|----------------|--|-----|
| 3.5.4.2.2.5 | Configure Assured Green Period Duration | N/A |
| 3.5.4.2.2.6 | Configure Red Light Violation Warning Application Parameters | N/A |
| 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 | N/A |
| 3.5.4.3.1.1.2 | Provide Movement State | N/A |
| 3.5.4.3.1.1.3 | Provide Movement Minimum End Time | N/A |
| 3.5.4.3.1.1.4 | Provide Movement Maximum End Time | N/A |
| 3.5.4.3.1.1.5 | Provide Movement Likely End Time | N/A |
| 3.5.4.3.1.1.5 | Provide Movement Likely End Time | N/A |
| 3.5.4.3.1.1.6 | Provide Movement Likely End Time Confidence | N/A |
| 3.5.4.3.1.1.7 | Provide Next Movement State | N/A |
| 3.5.4.3.1.1.8 | Provide Next Movement Minimum End Time | N/A |
| 3.5.4.3.1.1.9 | Provide Next Movement Maximum End Time | N/A |
| 3.5.4.3.1.1.10 | Provide Next Movement Start Time | N/A |
| 3.5.4.3.1.2 | Provide Movement Assistance Requirements | |
| 3.5.4.3.1.2.1 | Provide Lane Connection Queue Length | N/A |
| 3.5.4.3.1.2.2 | Provide Lane Connection Vulnerable Road User Detection | N/A |
| 3.5.4.3.1.3 | Provide Advisory Speed Requirements | |
| 3.5.4.3.1.3.1 | Provide Advisory Speed Type | N/A |
| 3.5.4.3.1.3.2 | Provide Advisory Speed | N/A |

| 3.5.4.3.1.3.3 | Provide Advisory Speed Zone | | N/A |
|---------------|--|-----------|-----------------------------|
| 3.5.4.3.1.3.4 | Provide Advisory Speed Vehicle Type | | N/A |
| 3.5.4.3.1.4 | Provide Road Authority ID | | N/A |
| 3.5.4.3.1.5 | Provide Signal Phase and Timing Intersection Status | | N/A |
| 3.5.4.3.1.6 | Provide Compressed SPaT Information to External CV Application Process | | N/A |
| 3.5.4.3.2 | ASC - Internal CV Application Process Requirements | | |
| 3.5.4.3.2.1 | Provide UPER-encoded SPaT Message | | N/A |
| 3.5.4.3.2.2 | Retrieve BSMs | | N/A |
| 3.5.4.3.2.3 | Retrieve PSMs | | N/A |
| 3.5.4.3.2.4 | Retrieve Actuation Report | | N/A |
| 3.5.4.3.2.5 | Retrieve Detection Report | | N/A |
| 3.5.4.3.3 | Exchange Roadway Geometrics Information Requirements | | |
| 3.5.4.3.3.1 | Retrieve MAP Plan in Effect | | |
| | | C.3.15.47 | Retrieve MAP Plan in Effect |
| 3.5.4.3.3.2 | Confirm MAP Plan Compatibility | | N/A |
| 3.5.4.3.4 | Monitor CV Certificate Faults | | |
| | | C.3.11.3 | Monitor Unit Level Alarms |
| 3.5.4.4 | Manage ASC - ECLA Interface Requirements | | |
| 3.5.4.4.1 | Receive Current Phase Minimum End Time from an ECLA | | N/A |
| 3.5.4.4.2 | Receive Current Phase Maximum End Time from an ECLA | | N/A |
| 3.5.4.4.3 | Receive Current Phase Likely End Time from an ECLA | | N/A |
| 3.5.4.4.4 | Receive Current Phase Likely End Time Confidence from an ECLA | | N/A |
| 3.5.4.4.5 | Receive Next Phase from an ECLA | | N/A |
| 3.5.4.4.6 | Receive Compressed ECLA Input Data | | N/A |
| 3.5.5 | Backward Compatibility Requirements | | |

| 3.6 | Supplemental Non- communications Requirements | |
|---------|---|-----|
| 3.6.1 | Response Time for Requests | N/A |
| 3.6.2 | Condition-based Maximum Start Time | N/A |
| 3.6.3 | Signal Phase and Timing Data Performance Requirements | |
| 3.6.3.1 | SPaT Maximum Transmission Start Time | N/A |
| 3.6.3.2 | Movement Time Point Minimum Transmission Rate | N/A |
| 3.6.3.3 | SPaT Maximum Transmission Rate | N/A |
| 3.6.3.4 | SPaT Time Accuracy | N/A |

C.3 Test Procedures

- C.3.1 System Information
- C.3.1.1 Create Database Transaction

| Test Procedure: | | Create Database Transaction | |
|---|------------------------------------|--|-----------|
| Description: | | This test case has the ASC perform consistency checks. | |
| Requirement(| s): | | |
| Variable(s): | Variable(s): | | |
| Pass/Fail Criteria:The device under test shall pass every verification step test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure Results | |
| 1 | GET the follow | ing objects: fdTransactionMode. | Pass/Fail |
| 2 | IF fdTransactio | nMode IS_NOT_EQUAL_TO 1. | |
| 2.1 | NOTE | '1 = normal, 4 = done'. | |
| 2.2 | GOTO step 1. | | |
| 3 | ASSIGN fdTransactionMode EQUALS 2. | | |
| 3.1 | NOTE | '2 = transaction'. | |
| 4 | SET the followi | ng objects: fdTransactionMode. | Pass/Fail |
| 5 | GET the follow | ing objects: fdTransactionMode. | Pass/Fail |
| 6 | IF fdTransactio | nMode IS_NOT_EQUAL_TO 2. | |

| 6.1 | GOTO step 5. | | |
|------------------|------------------------|-----------------|-----------|
| - | Test Procedure Results | 6 | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.1.2 Verify Database Status and Verify Database Error

| Test Procedure: | | C.3.1.2 Verify Database Status and Verify Database Er | ror |
|------------------------|---|---|-----------|
| Description: | | This test case has a management station monitor the results of a consistency check. | |
| Requirement(| (s): | | |
| Variable(s): | | | |
| Pass/Fail Crit | Pass/Fail Criteria:The device under test shall pass every verification step in the test case to pass the test case. If step 8.3 is not executed, the test case that called this test case is considered failed. | | |
| Test Step Number | Test Procedure | | Results |
| 1 | ASSIGN fdTransactionMode EQUALS 3. | | |
| 1.1 | NOTE '3 = verify'. | | |
| 2 | SET the following objects: fdTransactionMode. | | Pass/Fail |
| 3 | DELAY 500 millisecond(s). | | |
| 4 | GET the following objects: fdTransactionMode. | | Pass/Fail |
| 5 | 5 VERIFY fdTransactionMode IS_EQUAL_TO 4. | | Pass/Fail |
| 5.1 | NOTE '4 = done'. | | |
| 6 | GET the following objects: fdTransactionStatus. | | Pass/Fail |
| 7 | IF fdTransactio | onStatus IS_EQUAL_TO 3. | |
| 7.1 | GOTO | step '6'. | |
| 7.2 | NOTE | '3 = verifying'. | |
| 8 | IF fdTransactio | nStatus IS_EQUAL_TO 4. | |
| 8.1 | ASSIG | N fdTransactionMode EQUALS 1. | |
| 8.2 | SET th | e following objects: fdTransactionMode. | Pass/Fail |
| 8.3 | EXIT | EXIT | |
| 8.4 | NOTE | 'For dbVerifyStatus, 4 = doneWithNoError'. | |
| 8.5 | NOTE | 'For fdTransactionMode, 1 = normal'. | |
| 10 | IF fdTransactio | nStatus IS_EQUAL_TO 5. | |
| 10.1 | GET th | GET the following objects: fdTransactionError. Pass/Fail | |
| 10.2 | NOTE | '5 = doneWithError'. | |
| Test Procedure Results | | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.1.3 Explore Data

| Test Procedure: | Explore Data | | | |
|---------------------|--|---|---|------------|
| Description: | | This test case verifies that station to configure which overlaps. | This test case verifies that the ASC allows a management station to configure which Flashing Arrows consists of overlaps. | |
| Requiremen | t(s): | | | |
| Variable(s): | | Last_Object_OID RESPONSE ERROR Continue | OBJECT IDENTIFIEI Int Boolean | र |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedu | Test Procedure | | Results |
| 1 | ASSIGN Last | ASSIGN Last_Object_OID EQUALs ZeroDotZero | | |
| 2 | GET-NEXT the following variables: Last_Object_OID. | | | |
| 3 | IF RESPONSE ERROR IS_NOT_EQUAL_TO noError(0). | | | |
| 3.1 | VERIFY RESPONSE ERROR IS_EQUAL_TO NoSuchObject. Pass/Fai | | Pass/Fail | |
| 4 | USER-ACTION 'Determine whether the OID of the returned object is lexicographically larger than the OID contained in the request.' | | | |
| 4.1 | RECO | RECORD this information as Continue. | | |
| 5 | IF RESPONSE | IF RESPONSE ERROR IS_EQUAL_TO noError(0). | | |
| 5.1 | VERIF | Y Continue IS_EQUAL_TO 1. | | Pass/Fail |
| 5.2 | GOTO | step 7. | | |
| 6 | USER-ACTION the OID contain | N 'Verify that the OID of the retr ned in the request.' | ieved object is identical to | |
| 7 | USER-ACTION | USER-ACTION 'Determine the OID value of the retrieved object.' | | |
| 7.1 | RECO | RECORD this information as Last_Object_OID. | | |
| 8 | IF RESPONSE | IF RESPONSE ERROR IS_EQUAL_TO noError(0). | | |
| 8.1 | IF Con | IF Continue IS_EQUAL_TO 1. | | |
| 8.1.1 | GOTO step 2. | | | |
| Test Procedu | ure Results | | | - |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | - | |

C.3.1.4 Monitor SNMP Information

| Test Procedure: | | Monitor SNMP Information | | |
|---------------------|--|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to retrieve basic instrumentation and control information of the SNMP device. | | |
| Requirement(s): | | • 3.4.1.4.1 Monitor SN | IMP Information | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure Results | | Results | |
| 1 | GET the following objects: snmpInPkts, snmpInBadVersions, snmpInBadCommunityNames, snmpInBadCommunityUses, snmpInASNParseErrs, snmpSilentDrops, snmpProxyDrops | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.2 Unit Settings

C.3.2.1 Configure Start-Up Flash Mode

| Test Procedure: | | Configure Startup Flash Mo | de | |
|---------------------|--|---|---|-------------|
| Description: | | This test case verifies that the station to configure how the | he ASC allows a managem e ASC flashes during start | ient up. |
| Requirement(| s): | • 3.5.2.1.1.1.1 Configu | ire Startup Flash Mode | |
| Variable(s): | | OriginalFlashMode TestFlashMode | unitStartUpFlashMoo unitStartUpFlashMoo | de de |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure Results | | Results | |
| 1 | GET the following objects: unitStartUpFlashMode. | | Pass/Fail | |
| 1.1 | RECORD this information as OriginalFlashMode. | | | |
| 2 | ASSIGN TestFlashMode EQUALS RANDOM (1 TO 3). | | | |
| 2.1 | IF TestFlashMode IS_EQUAL_TO OriginalFlashMode. | | | |
| 2.1.1 | GOTO step 2. | | | |
| 3 | ASSIGN unitStartUpFlashMode EQUALS TestFlashMode. | | | |
| 4 | SET the following objects: unitStartUpFlashMode. Pass/Fa | | Pass/Fail | |
| 5 | GET the following objects: unitStartUpFlashMode. Pass/Fail | | Pass/Fail | |
| 6 | VERIFY unitStartUpFlashMode IS_EQUAL_TO TestFlashMode. Pass/Fail | | Pass/Fail | |
| 7 | ASSIGN unitStartUpFlashMode EQUALS OriginalFlashMode. | | | |
| 8 | SET the following objects: unitStartUpFlashMode. Pass/Fail | | Pass/Fail | |
| 9 | GET the following objects: unitStartUpFlashMode. Pass/Fail | | Pass/Fail | |
| 10 | 10 VERIFY unitStartUpFlashMode IS_EQUAL_TO OriginalFlashMode. Pass/F | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |

C.3.2.2 Configure Start-Up Flash Mode

| Test Procedure: | Configure Startup Flash Time | | |
|---------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to configure the startup flash time. | | |
| Requirement(s): | 3.4.1.2 Deliver Data 3.5.2.1.1.1.2 Configure Startup Flash Time | | |
| Variable(s): | OriginalStartUpFlash unitStartUpFlash TestStartUpFlash unitStartUpFlash | | |
| Pass/Fail Criteria: | The device under test shall pass every verification step in this test case to pass the test case. | | |

| Test Step Number | Test Procedure | | |
|------------------------|---|------------------------|-----------|
| 1 | GET the following objects: unitStartUpFlash. | | Pass/Fail |
| 1.1 | RECORD this information as Original | StartUpFlash. | |
| 2 | ASSIGN TestStartUpFlash EQUALS RANDO | M (0 TO 255). | |
| 2.1 | IF TestStartUpFlash IS_EQUAL_TO | OriginalStartUpFlash . | |
| 2.1.1 | GOTO step 2. | | |
| 3 | ASSIGN unitStartUpFlash EQUALS TestStart | tUpFlash. | |
| 4 | SET the following objects: unitStartUpFlash. | | |
| 5 | GET the following objects: unitStartUpFlash. | | Pass/Fail |
| 6 | VERIFY unitStartUpFlash IS_EQUAL_TO TestStartUpFlash. | | |
| 7 | ASSIGN unitStartUpFlash EQUALS OriginalStartUpFlash. | | |
| 8 | SET the following objects: unitStartUpFlash. Pass | | |
| 9 | GET the following objects: unitStartUpFlash. | | |
| 10 | VERIFY unitStartUpFlash IS_EQUAL_TO OriginalStartUpFlash. | | |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.2.3 Configure Backup Time

| Test Procedure: | | Configure Backup Time | |
|---------------------|--|---|-------------------------|
| Description: | | This test case verifies that the ASC allows station to configure the backup time. | a management |
| Requirement(s): | | • 3.5.2.1.1.2 Configure Backup Time | 9 |
| Variable(s): | | OriginalBackUpTime unitBack TestBackUpTime unitBack | cupTime cupTime |
| Pass/Fail Criteria: | | The device under test shall pass every ve test case to pass the test case. | rification step in this |
| Test Step Number | Test Procedure | | Results |
| 1 | GET the following objects: unitBackupTime. | | Pass/Fail |
| 1.1 | RECO | RD this information as OriginalBackUpTime. | |
| 2 | ASSIGN TestBackUpTime EQUALS RANDOM (0 TO 16777216). | | 216). |
| 2.1 | IF TestBackUpTime IS_EQUAL_TO OriginalBackUpTime. | | Time. |
| 2.1.1 | GOTO step 2. | | |
| 3 | ASSIGN unitBackupTime EQUALS TestBackUpTime. | | |
| 4 | SET the following objects: unitBackupTime. Pass/Fai | | |
| 5 | GET the following objects: unitBackupTime. Pass/Fail | | |
| 6 | VERIFY unitBa | ckupTime IS_EQUAL_TO TestBackUpTime. | Pass/Fail |
| 7 | ASSIGN unitBa | ackupTime EQUALS OriginalBackUpTime. | |

| 8 | SET the following objects: unitBackupTime. | Pass/Fail | | |
|-----------------------|---|-----------|--|--|
| 9 | GET the following objects: unitBackupTime. | Pass/Fail | | |
| 10 | VERIFY unitBackupTime IS_EQUAL_TO OriginalBackUpTime. Pass/Fa | | | |
| Test Procedur | re Results | | | |
| Tested By: | Date Tested: | Pass/Fail | | |
| Test Procedure Notes: | | | | |

C.3.2.4 Configure ASC Antenna Offset

| Test Procedure: | | Configure ASC Antenna Of | fset | |
|-------------------------|---|--|---|---------------------------|
| Description: | | This test case verifies that station to configure an offs external GNSS positioning | the ASC allows a managen et between the antenna of device and the base of the | nent a mounted ASC. |
| Requirement(| s): | • 3.5.1.1.1 Configure | ASC Location – Antenna O | ffset |
| Variable(s): | | OriginalElevationOffset ascElevationOffset TestElevationOffset ascElevationOffset | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test case | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure Result | | Results | |
| 1 | GET the following objects: ascElevationOffset. Pass/Fa | | Pass/Fail | |
| 1.1 | RECORD this information as OriginalElevationOffset. | | | |
| 2 | ASSIGN TestElevationOffset EQUALS RANDOM (0 TO 31). | | | |
| 2.1 | IF TestElevationOffset IS_EQUAL_TO OriginalElevationOffset | | | |
| 2.1.1 | GOTO Step 2 | | | |
| 3 | ASSIGN ascElevationOffset EQUALS TestElevationOffset. | | | |
| 4 | SET the followi | ing objects: ascElevationOffset | | Pass/Fail |
| 5 | GET the follow | ing objects: ascElevationOffset | | Pass/Fail |
| 6 | VERIFY TestE | levation IS_EQUAL_TO TestEl | evationOffset. | Pass/Fail |
| 7 | ASSIGN ascElevationOffset EQUALS OriginalElevationOffset. | | | |
| 8 | SET the following objects: ascElevationOffset. Pas | | Pass/Fail | |
| 9 | GET the following objects: ascElevationOffset. | | Pass/Fail | |
| 10 |) VERIFY TestElevation IS_EQUAL_TO OriginalElevationOffset. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: Date Tested: | | Pass/Fail | | |
| Test Procedure N | lotes: | | | |

C.3.3 Phases

C.3.3.1 Enable/Disable Phase

| Test Procedure: | | Enable/Disable Phase | | |
|---------------------|--|--|--------------------------------------|------------|
| Description: | | This test case verifies that t station to enable/disable ph | the ASC shall allow a mana nases. | agement |
| Requirement(| s): | • 3.5.2.1.2.1.1 Enable/ | Disable Phase | |
| Variable(s): | | Table_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptionsMaxRowsmaxPhases | | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loade | ed'. | |
| 2 | GET the following objects: maxPhases. Pass/Fail | | Pass/Fail | |
| 2.1 | RECORD this information as MaxRows. | | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 4 | GET the following objects: phaseOptions.Table_Row. Pass/Fail | | | |
| 4.1 | RECORD this information as OriginalPhaseOptions. | | | |
| 5 | ASSIGN TestP | haseOptions EQUALS Original | IPhaseOptions XOR 1. | |
| 6 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 7 | ASSIGN phaseOptions.Table_Row EQUALS TestPhaseOptions. | | | |
| 8 | SET the following objects: phaseOptions.Table_Row. Pass/Fail | | | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 10 | GET the following objects: phaseOptions.Table Row. Pass/Fail | | Pass/Fail | |
| 11 | VERIFY phaseOptions. Table_Row IS_EQUAL_TO TestPhaseOptions. Pass/Fail | | Pass/Fail | |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS OriginalPhaseOptions. | | | |
| 14 | SET the following objects: phaseOptions.Table_Row. Pass/Fail | | | |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 16 | GET the following objects: phaseOptions.Table_Row. Pass/Fail | | | |
| 17 | 17 VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.3.2 Configure Phase Minimum Green Time

| Test Procedure: | | Configure Phase Minimum Green Time | | |
|---------------------|--|--|--|--------------|
| Description: | | This test case verifies that t station to configure the min | he ASC allows a managen imum green time for a pha | nent ase. |
| Requirement(| s): | • 3.5.2.1.2.1.2 Configu | re Phase Minimum Green | Time |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseMinimumGreenphaseMinimumGreenTestPhaseMinimumGreenphaseMinimumGreen | | n n |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxPhases. Pass/Fai | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: phaseMinimumGreen.Table_Row. Pass/Fail | | | |
| 3.1 | RECORD this information as OriginalPhaseMinimumGreen. | | | |
| 4 | ASSIGN TestPhaseMinimumGreen EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestPhaseMinimumGreen IS_EQUAL_TO OriginalPhaseMinimumGreen. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN phase | MinimumGreen EQUALS Test | PhaseMinimumGreen. | |
| 6 | SET the followi | ng objects: phaseMinimumGre | en.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: phaseMinimumGre | en.Table_Row. | Pass/Fail |
| 8 | VERIFY phaseMinimumGreen.Table_Row IS_EQUAL_TO Pass/Fail | | Pass/Fail | |
| 9 | ASSIGN phaseMinimumGreen.Table_Row EQUALS OriginalPhaseMinimumGreen. | | | |
| 10 | SET the following objects: phaseMinimumGreen.Table_Row. Pass/Fail | | | |
| 11 | GET the following objects: phaseMinimumGreen.Table_Row. Pass/Fail | | | |
| 12 | VERIFY phaseMinimumGreen.Table_Row IS_EQUAL_TO OriginalPhaseMinimumGreen. | | | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.3.3 Configure Phase Passage Time

Configure Phase Passage Time

| Test Procedure: | | | | |
|---------------------|---|--|--|---------------------------------|
| Description: | | This test case verifies that the station to configure the past time the phase's GREEN in vehicle actuation is removed. | the ASC allows a managem sage timer for the phase. T dication will be extended w d. | ient This is the vhen any |
| Requirement(| s): | • 3.5.2.1.2.1.3 Configu | ıre Phase Passage Time | |
| Variable(s): | | Table_Row MaxRows OriginalPhasePassage TestPhasePassage | Int maxPhases phasePassage phasePassage | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure Res | | | Results |
| 1 | GET the following objects: maxPhases | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: phasePassage.Table_Row. | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalPhasePassage. | | | |
| 4 | ASSIGN TestPhasePassage EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestPhasePassage IS_EQUAL_TO OriginalPhasePassage. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN phasePassage.Table_Row EQUALS TestPhasePassage. | | | |
| 6 | SET the following objects: phasePassage.Table_Row. Pass/Fail | | | Pass/Fail |
| 7 | GET the following objects: phasePassage.Table_Row. Pass/Fail | | | Pass/Fail |
| 8 | VERIFY phasePassage.Table_Row IS_EQUAL_TO TestPhasePassage. Pass/Fail | | | |
| 9 | ASSIGN phasePassage.Table_Row EQUALS OriginalPhasePassage. | | | |
| 10 | SET the following objects: phasePassage.Table_Row. Pass/Fail | | | |
| 11 | GET the following objects: phasePassage.Table_Row. Pass/Fail | | | |
| 12 | VERIFY phasePassage.Table_Row IS_EQUAL_TO OriginalPhasePassage. | | | |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.3.4 Configure Two Phase Maximum Green Times

| Test Procedure: | Configure Two Phase Maximum Green Times |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to configure the maximum length of time the phase display a GREEN indication. |

| Requirement(s): | | • 3.5.2.1.2.1.4 Configure Ty Times | wo Phase Maximum G | reen |
|---------------------|--|--|--|-----------|
| Variable(s): | | MaxRows Table_Row TestPhaseMaximum1 OriginalPhaseMaximum1 TestPhaseMaximum2 OriginalPhaseMaximum2 | maxPhases Int phaseMaximum1 phaseMaximum1 phaseMaximum2 phaseMaximum1 | |
| Pass/Fail Crite | eria: | The device under test shall pass test case to pass the test case. | every verification ste | p in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO Ma | axRows). | |
| 3 | GET the follow | ing objects: phaseMaximum1.Table | _Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalPhas | eMaximum1. | |
| 4 | GET the follow | ing objects: phaseMaximum2.Table | _Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalPhas | eMaximum2. | |
| 5 | ASSIGN TestP | haseMaximum1 EQUALS RANDON | /I (0 TO 999). | |
| 5.1 | IF TestPhaseMaximum1 IS_EQUAL_TO OriginalPhaseMaximum1. | | | |
| 5.1.1 | GOTO step 5. | | | |
| 6 | ASSIGN TestPhaseMaximum2 EQUALS RANDOM (0 TO 999). | | | |
| 6.1 | IF TestPhaseMaximum2 IS_EQUAL_TO OriginalPhaseMaximum2. | | | |
| 6.1.1 | | GOTO step 6. | | |
| 7 | ASSIGN phase | Maximum1.Table_Row EQUALS Te | estPhaseMaximum1. | |
| 8 | ASSIGN phaseMaximum2.Table_Row EQUALS TestPhaseMaximum2. | | | |
| 9 | SET the following objects: phaseMaximum1.Table_Row, phaseMaximum2.Table_Row. | | Pass/Fail | |
| 10 | GET the following objects: phaseMaximum1.Table_Row, phaseMaximum2.Table_Row. | | Pass/Fail | |
| 11 | VERIFY phaseMaximum1.Table_Row IS_EQUAL_TO TestPhaseMaximum1. | | Pass/Fail | |
| 12 | VERIFY phaseMaximum2.Table_Row IS_EQUAL_TO TestPhaseMaximum2. | | Pass/Fail | |
| 13 | ASSIGN phaseMaximum1.Table_Row EQUALS OriginalPhaseMaximum1. | | | |
| 14 | ASSIGN phaseMaximum2.Table_Row EQUALS OriginalPhaseMaximum2. | | | |
| 15 | SET the following objects: phaseMaximum1.Table_Row, phaseMaximum2.Table_Row. | | _Row, | Pass/Fail |
| 16 | GET the following objects: phaseMaximum1.Table_Row, phaseMaximum2.Table_Row. | | Pass/Fail | |
| 17 | VERIFY phaseMaximum1.Table_Row IS_EQUAL_TO OriginalPhaseMaximum1. | | Pass/Fail | |
| 18 | VERIFY phase OriginalPhase | Maximum2.Table_Row IS_EQUAL_ /laximum2. | ТО | Pass/Fail |

| Test Procedure Results | | |
|------------------------|-----------------|-----------|
| Tested By: | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | |

C.3.3.5 Configure Three Phase Maximum Green Times

| Test Procedure: | | Configure Three Phase Maximum Green Times | |
|---------------------|--|--|--------------------|
| Description: | | This test case verifies that the ASC allows a managen station to configure three maximum times a phase main Green. | nent ay be held |
| Requirement | (s): | • 3.5.2.1.2.1.5 Configure Three Phase Maximum Times | Green |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseMaximum3phaseMaximum3TestPhaseMaximum3phaseMaximum3 | |
| Pass/Fail Crit | teria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | Test Procedure | |
| 1 | PERFORM the Green Times'. | PERFORM the Test Procedure 'C.3.3.4 Configure Two Phase Maximum Green Times'. | |
| 2 | GET the follow | ing objects: maxPhases. | Pass/Fail |
| 2.1 | RECORD this information as MaxRows. | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | |
| 4 | GET the following objects: phaseMaximum3.Table_Row. | | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalPhaseMaximum3. | |
| 5 | ASSIGN TestT | hreshold EQUALS RANDOM (0 TO 999). | |
| 5.1 | IF Te OriginalPhasel | stPhaseMaximum3 IS_EQUAL_TO Maximum3. | |
| 5.1.1 | | GOTO step 4. | |
| 6 | ASSIGN phase | Maximum3.Table_Row EQUALS TestPhaseMaximum3. | |
| 7 | SET the follow | ing objects: phaseMaximum3.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: phaseMaximum3.Table_Row. | Pass/Fail |
| 9 | VERIFY phaseMaximum3.Table_Row IS_EQUAL_TO TestPhaseMaximum3. | | Pass/Fail |
| 10 | ASSIGN phase OriginalPhase | ASSIGN phaseMaximum3.Table_Row EQUALS OriginalPhaseMaximum3. | |
| 11 | SET the follow | ing objects: phaseMaximum3.Table_Row. | Pass/Fail |
| 12 | GET the follow | ing objects: phaseMaximum3.Table_Row. | Pass/Fail |
| 13 | VERIFY phase OriginalPhase | Maximum3.Table_Row IS_EQUAL_TO Maximum3. | Pass/Fail |
| Test Procedu | re Results | | |

| Tested By: | Date Tested: | Pass/Fail | | |
|-----------------------|-----------------|-----------|--|--|
| Test Procedure Notes: | | | | |

C.3.3.6 Configure Phase Yellow Change Time

| Test Procedure: | | Configure Phase Yellow Change Time | | |
|---------------------|--|--|---------------------|--|
| Description: | | This test case verifies that the ASC allows a manager station to configure the yellow change interval time for | nent or a phase. | |
| Requirement | (s): | • 3.5.2.1.2.1.6 Configure Phase Yellow Change | Time | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseYellowChangephaseYellowChangeTestYellowPhaseChangephaseYellowChange | ; ; | |
| Pass/Fail Criteria: | | The device under test shall pass every verification sto test case to pass the test case. | ep in this | |
| Test Step Number | Test Procedu | Test Procedure | | |
| 1 | GET the follow | GET the following objects: maxPhases. | | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: phaseYellowChange.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPhaseYellowChange. | | | |
| 4 | ASSIGN TestY | ASSIGN TestYellowPhaseChange EQUALS RANDOM (0 TO 255). | | |
| 4.1 | IF TestYellowPhaseChange IS_EQUAL_TO OriginalPhaseYellowChange. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phase TestYellowPha | eYellowChange.Table_Row EQUALS aseChange. | | |
| 6 | SET the following objects: phaseYellowChange.Table_Row. Pa | | Pass/Fail | |
| 7 | GET the following objects: phaseYellowChange.Table_Row. Pas | | Pass/Fail | |
| 8 | VERIFY phaseYellowChange.Table_Row IS_EQUAL_TO TestYellowPhaseChange. | | Pass/Fail | |
| 9 | ASSIGN phaseYellowChange.Table_Row EQUALS OriginalPhaseYellowChange. | | | |
| 10 | SET the following objects: phaseYellowChange.Table_Row. | | Pass/Fail | |
| 11 | GET the following objects: phaseYellowChange.Table_Row. Pa | | Pass/Fail | |
| 12 | VERIFY phaseYellowChange.Table_Row IS_EQUAL_TO OriginalPhaseYellowChange. | | Pass/Fail | |
| Test Procedu | re Results | | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.3.7 Configure Phase Red Clearance Time

| Test Procedure: | | Configure Phase Red Clearance Time | | |
|---------------------|--|--|--|-------------------------------|
| Description: | | This test case verifies that t station to configure the red | the ASC allows a managen clearance interval time for | nent [.] a phase. |
| Requirement(s): | | • 3.5.2.1.2.1.7 Configu | ire Red Clearance Time | |
| Variable(s): | | MaxRowsmaxPhasesOriginalPhaseRedClearphaseRedClearTestPhaseRedClearphaseRedClearTable RowInt | | |
| Pass/Fail Crit | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedu | Test Procedure | | |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: phaseRedClear.Table_Row. | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalPhaseRedClear. | | | |
| 4 | ASSIGN TestPhaseRedClear EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF Tes | IF TestPhaseRedClear IS_EQUAL_TO OriginalPhaseRedClear | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phase | RedClear.Table_Row EQUAL | S TestPhaseRedClear. | |
| 6 | SET the follow | ing objects: phaseRedClear.Ta | ble_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: phaseRedClear.Ta | ble_Row. | Pass/Fail |
| 8 | VERIFY phase TestPhaseRed | RedClear.Table_Row IS_EQU Clear. | AL_TO | Pass/Fail |
| 9 | ASSIGN phase | RedClear.Table_Row EQUAL | S OriginalPhaseRedClear. | |
| 10 | SET the follow | ing objects: phaseRedClear.Ta | ble_Row. | Pass/Fail |
| 11 | GET the following objects: phaseRedClear.Table_Row. Pass | | | Pass/Fail |
| 12 | VERIFY phaseRedClear.Table_Row IS_EQUAL_TO OriginalPhaseRedClear. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | Date Tested: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.3.8 Configure Phase Red Revert Time

| Test Procedure: | | Configure Phase Red Revert Time | | |
|-------------------------|--|---|--|----------------------------|
| Description: | | This test case verifies that t station to configure the min RED indication before it ma | the ASC allows a managen nimum time a phase must o y display a GREEN indicat | nent display a tion. |
| Requirement(| s): | • 3.5.2.1.2.1.8 Configu | ıre Phase Red Revert Time | • |
| Variable(s): | | MaxRows OriginalPhaseRedRevert TestPhaseRedRevert Table_Row | maxPhases phaseRedRevert phaseRedRevert Int | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: phaseRedRevert.Table_Row. | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalPhaseRedRevert. | | | |
| 4 | ASSIGN TestPhaseRedRevert EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestPhaseRedRevert IS_EQUAL_TO OriginalPhaseRedRevert. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN phase | eRedRevert.Table_Row EQUA | LS TestPhaseRedRevert. | |
| 6 | SET the follow | ing objects: phaseRedRevert.T | able_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: phaseRedRevert.T | able_Row. | Pass/Fail |
| 8 | VERIFY phaseRedRevert.Table_Row IS_EQUAL_TO TestPhaseRedRevert. | | Pass/Fail | |
| 9 | ASSIGN phaseRedRevert.Table_Row EQUALS OriginalPhaseRedRevert. | | | |
| 10 | SET the following objects: phaseRedRevert.Table_Row. Pass/Fail | | | Pass/Fail |
| 11 | GET the follow | ing objects: phaseRedRevert.T | able_Row. | Pass/Fail |
| 12 | VERIFY phaseRedRevert.Table_Row IS_EQUAL_TO OriginalPhaseRedRevert. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: Date Tested: | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.3.9 Configure Unit Red Revert Time

Configure Unit Red Revert Time

| Test Procedure: | | | | |
|------------------------|---|---|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure the minimum red revert time for all phases. | | |
| Requirement(s): | | • 3.5.2.1.2.1.9 Configure Unit Red Revert Time | | |
| Variable(s): | | OriginalRedRevert unitRedRevert TestRedRevert unitRedRevert | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the following objects: unitRedRevert. | | | Pass/Fail |
| 1.1 | RECORD this information as OriginalRedRevert. | | RedRevert. | |
| 2 | ASSIGN TestRedRevert EQUALS RANDOM (0 TO | | (0 TO 255). | |
| 2.1 | IF Test | tRedRevert IS_EQUAL_TO Ori | ginalRedRevert. | |
| 2.1.1 | | GOTO step 2. | | |
| 3 | ASSIGN unitRe | edRevert EQUALS TestRedRe | vert. | |
| 4 | SET the followi | ing objects: unitRedRevert. | | Pass/Fail |
| 5 | GET the follow | ing objects: unitRedRevert. | | Pass/Fail |
| 6 | VERIFY unitRe | edRevert IS_EQUAL_TO TestR | ledRevert. | Pass/Fail |
| 7 | ASSIGN unitRedRevert EQUALS OriginalRedRevert. | | | |
| 8 | SET the following objects: unitRedRevert. Pass/Fa | | | Pass/Fail |
| 9 | GET the following objects: unitRedRevert. | | Pass/Fail | |
| 10 | VERIFY unitRedRevert IS_EQUAL_TO OriginalRedRevert. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | Date Pass/Fa Tested: | | | Pass/Fail |
| Test Procedure Notes: | | | | |

C.3.3.10 Configure Phase Added Initial Time

| Test Procedure: | Configure Phase Added Initial | Configure Phase Added Initial Time | | |
|--------------------|--|---|--|--|
| Description: | This test case verifies that the station to configure the time b period will be increased from received during the associated | This test case verifies that the ASC allows a management station to configure the time by which the variable initial time period will be increased from zero with each vehicle actuation received during the associated vehicle clearance intervals. | | |
| Requirement(s): | • 3.5.2.1.2.1.10 Configur | • 3.5.2.1.2.1.10 Configure Phase Added Initial Time | | |
| Variable(s): | MaxRows Table_Row OriginalPhaseAddedInitial TestPhaseAddedInitial | maxPhases Int phaseAddedInitial phaseAddedInitial | | |

| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
|------------------------|--|---|--------------------|-----------|
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | vs. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: phaseAddedInitial. | Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | PhaseAddedInitial. | |
| 4 | ASSIGN TestP | haseAddedInitial EQUALS RA | NDOM (0 TO 255). | |
| 4.1 | IF Test OriginalPhaseA | PhaseAddedInitial IS_EQUAL_ AddedInitial. | _TO | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phaseAddedInitial.Table_Row EQUALS TestPhaseAddedInitial. | | | |
| 6 | SET the following objects: phaseAddedInitial.Table_Row. | | | Pass/Fail |
| 7 | GET the following objects: phaseAddedInitial.Table_Row. Pass/F | | | Pass/Fail |
| 8 | VERIFY phaseAddedInitial.Table_Row IS_EQUAL_TO TestPhaseAddedInitial. | | | Pass/Fail |
| 9 | ASSIGN phaseAddedInitial.Table_Row EQUALS OriginalPhaseAddedInitial. | | | |
| 10 | SET the following objects: phaseAddedInitial.Table_Row. P | | Pass/Fail | |
| 11 | GET the following objects: phaseAddedInitial.Table_Row. Pass/ | | Pass/Fail | |
| 12 | VERIFY phaseAddedInitial.Table_Row IS_EQUAL_TO OriginalPhaseAddedInitial. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | Tested By: Date Tested: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.3.11 Configure Phase Maximum Initial Time

| Test Procedure: | Configure Phase Maximum Initial Time | | |
|---------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to configure the maximum a variable green time period of a phase can be increased. | | |
| Requirement(s): | • 3.5.2.1.2.1.11 Configure Phase Maximum Initial Time | | |
| Variable(s): | MaxRows Table_Row OriginalPhaseMaximumInitial TestPhaseMaximumInitial | maxPhases Int phaseMaximumInitial phaseMaximumInitial | |
| Pass/Fail Criteria: | The device under test shall pass every verification step in this test case to pass the test case. | | |

| Test Step Number | Test Procedure | | Results |
|---------------------|--|----------------------|-----------|
| 1 | GET the following objects: maxPhases. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRov | VS. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: phaseMaximumInit | ial.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | PhaseMaximumInitial. | |
| 4 | ASSIGN TestPhaseMaximumInitial EQUALS | RANDOM (0 TO 255). | |
| 4.1 | IF TestPhaseMaximumInitial IS_EQU OriginalPhaseMaximumInitial. | AL_TO | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN phaseMaximumInitial.Table_Row EQUALS TestPhaseMaximumInitial. | | |
| 6 | SET the following objects: phaseMaximumInitial.Table Row. | | Pass/Fail |
| 7 | GET the following objects: phaseMaximumInit | ial.Table_Row. | Pass/Fail |
| 8 | VERIFY phaseMaximumInitial.Table_Row IS_EQUAL_TO TestPhaseMaximumInitial. | | Pass/Fail |
| 9 | ASSIGN phaseMaximumInitial.Table_Row EQUALS OriginalPhaseMaximumInitial. | | |
| 10 | SET the following objects: phaseMaximumInit | ial.Table_Row. | Pass/Fail |
| 11 | GET the following objects: phaseMaximumInit | ial.Table_Row. | Pass/Fail |
| 12 | VERIFY phaseMaximumInitial.Table_Row IS_EQUAL_TO OriginalPhaseMaximumInitial. | | Pass/Fail |
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.3.12 Configure Phase Time Before Reduction

| Test Procedure: | | Configure Phase Time Before Reduction | | |
|---------------------|--------------|--|--|--------------------|
| Description: | | This test case verifies that the ASC allows a management station to configure the time before the allowable gap between the Passage Time and Minimum Gap setting is reduced. | | |
| Requirement(s): | | • 3.5.2.1.2.1.12 Configure I | Phase Time Before Re | duction |
| Variable(s): | | MaxRows Table_Row OriginalTimeBeforeReduction TestTimeBeforeReduction | maxPhases Int phaseTimeBeforeRe phaseTimeBeforeRe | duction duction |
| Pass/Fail Criteria: | | The device under test shall pass test case to pass the test case. | every verification ste | p in this |
| Test Step Number | Test Procedu | re | | Results |

| 1 | GET the following objects: maxPhases. | | Pass/Fail |
|------------------|---|--|-----------|
| 1.1 | RECORD this information as MaxRov | VS. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: phaseTimeBeforeF | Reduction.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | TimeBeforeReduction. | |
| 4 | ASSIGN TestTimeBeforeReduction EQUALS | RANDOM (0 TO 255). | |
| 4.1 | IF TestTimeBeforeReduction IS_EQU OriginalTimeBeforeReduction. | IAL_TO | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN phaseTimeBeforeReduction.Table_Row EQUALS TestTimeBeforeReduction. | | |
| 6 | SET the following objects: phaseTimeBeforeReduction.Table_Row. Pass/ | | |
| 7 | GET the following objects: phaseTimeBeforeReduction.Table_Row. Pa | | |
| 8 | VERIFY phaseTimeBeforeReduction.Table_Row IS_EQUAL_TO TestTimeBeforeReduction. | | |
| 9 | ASSIGN phaseTimeBeforeReduction.Table_Row EQUALS OriginalTimeBeforeReduction. | | |
| 10 | SET the following objects: phaseTimeBeforeReduction.Table_Row. Pass/Fail | | Pass/Fail |
| 11 | GET the following objects: phaseTimeBeforeF | GET the following objects: phaseTimeBeforeReduction.Table_Row. Pass/Fail | |
| 12 | VERIFY phaseTimeBeforeReduction.Table_Row IS_EQUAL_TO OriginalTimeBeforeReduction. | | |
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.3.13 Configure Phase Time to Reduce

| Test Procedure: | | Configure Phase Time to Reduce | |
|---------------------|----------------|---|-----------|
| Description: | | This test case verifies that the ASC allows a management station to configure the rate of reduction of the allowable gap between the Passage Time and Minimum Gap setting. | |
| Requirement(| s): | • 3.5.2.1.2.1.13 Configure Phase Time to Reduce | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalTimeToReducephaseTimeToReduceTestTimeToReducephaseTimeToReduce | e e |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | p in this |
| Test Step Number | Test Procedu | e | Results |
| 1 | GET the follow | ing objects: maxPhases. | Pass/Fail |

| 1.1 | RECORD this information as MaxRov | /S. | |
|------------------|---|-------------------------|-----------|
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: phaseTimeToRedu | ice.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | TimeToReduce. | |
| 4 | ASSIGN TestTimeToReduce EQUALS RAND | OM (0 TO 255). | |
| 4.1 | IF TestTimeToReduce IS_EQUAL_T | O OriginalTimeToReduce. | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN phaseTimeToReduce.Table_Row E0 TestTimeToReduce. | QUALS | |
| 6 | SET the following objects: phaseTimeToReduce.Table_Row. | | Pass/Fail |
| 7 | GET the following objects: phaseTimeToReduce.Table_Row. | | Pass/Fail |
| 8 | VERIFY phaseTimeToReduce.Table_Row IS_EQUAL_TO TestTimeToReduce. | | Pass/Fail |
| 9 | ASSIGN phaseTimeToReduce.Table_Row EQUALS OriginalTimeToReduce. | | |
| 10 | SET the following objects: phaseTimeToReduce.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: phaseTimeToReduce.Table_Row. | | Pass/Fail |
| 12 | VERIFY phaseTimeToReduce.Table_Row IS_EQUAL_TO OriginalTimeToReduce. | | Pass/Fail |
| Test Procedure | e Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.3.14 Configure Phase Cars Before Reduction

| Test Procedure: | | Configure Phase Cars Before Re | eduction | |
|---------------------|---------------------------------------|--|--|--------------------|
| Description: | | This test case verifies that the ASC allows a management station to configure the number of actuations on conflicting phases before the reduction of the allowable gap from the Passage Time shall begin. | | |
| Requirement(s): | | • 3.5.2.1.2.1.14 Configure I | Phase Cars Before Re | duction |
| Variable(s): | | MaxRows Table_Row OriginalCarsBeforeReduction TestCarsBeforeReduction | maxPhases Int phaseCarsBeforeRe phaseCarsBeforeRe | duction duction |
| Pass/Fail Criteria: | | The device under test shall pass test case to pass the test case. | every verification ste | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxPhases. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table | Row EQUALS RANDOM (1 TO Ma | axRows). | |

| 3 | GET the following objects: phaseCarsBeforeF | Reduction.Table_Row. | Pass/Fail |
|------------------|---|----------------------|-----------|
| 3.1 | RECORD this information as Original | CarsBeforeReduction. | |
| 4 | ASSIGN Table_Row EQUALS RANDOM (0 T | O 255). | |
| 4.1 | IF TestCarsBeforeReduction IS_EQU OriginalCarsBeforeReduction. | AL_TO | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN phaseCarsBeforeReduction.Table_F TestCarsBeforeReduction. | Row EQUALS | |
| 6 | SET the following objects: phaseCarsBeforeR | Reduction.Table_Row. | Pass/Fail |
| 7 | GET the following objects: phaseCarsBeforeReduction.Table_Row. | | Pass/Fail |
| 8 | VERIFY phaseCarsBeforeReduction.Table_Row IS_EQUAL_TO TestCarsBeforeReduction. | | Pass/Fail |
| 9 | ASSIGN phaseCarsBeforeReduction.Table_Row EQUALS OriginalCarsBeforeReduction. | | |
| 10 | SET the following objects: phaseCarsBeforeReduction.Table_Row. Pass/Fail | | Pass/Fail |
| 11 | GET the following objects: phaseCarsBeforeReduction.Table_Row. Pass/F | | Pass/Fail |
| 12 | VERIFY phaseCarsBeforeReduction.Table_Row IS_EQUAL_TO OriginalCarsBeforeReduction. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.3.15 Configure Phase Reduce By Time

| Test Procedure: | | Configure Phase Reduce By Time | |
|---------------------|---|---|---------------------|
| Description: | | This test case verifies that the ASC allows a managen station to configure the rate of reduction for volume or reduction. | nent lensity gap |
| Requirement(| s): | • 3.5.2.1.2.1.15 Configure Phase Reduce By Time | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseReduceByphaseReduceByTestPhaseReduceByphaseReduceBy | |
| Pass/Fail Crite | iteria: The device under test shall pass every verification step in this test case to pass the test case. | | ep in this |
| Test Step Number | | Test Procedure | Results |
| 1 | GET the following objects: maxPhases. Pass/Fa | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the follow | ing objects: phaseReduceBy.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalPhaseReduceBy. | |

| 4 | ASSIGN TestPhaseReduceBy EQUALS RAN | DOM (0 TO 255). | |
|----------------|--|-----------------------|-----------|
| 4.1 | IF TestPhaseReduceBy IS_EQUAL_TO OriginalPhaseReduceBy. | | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN phaseReduceBy.Table_Row EQUAL | _S TestPhaseReduceBy. | |
| 6 | SET the following objects: phaseReduceBy.Ta | able_Row. | Pass/Fail |
| 7 | GET the following objects: phaseReduceBy.T | able_Row. | Pass/Fail |
| 8 | VERIFY phaseReduceBy.Table_Row IS_EQUAL_TO TestPhaseReduceBy. | | Pass/Fail |
| 9 | ASSIGN phaseReduceBy.Table_Row EQUALS OriginalPhaseReduceBy. | | |
| 10 | SET the following objects: phaseReduceBy.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: phaseReduceBy.Table_Row. | | Pass/Fail |
| 12 | VERIFY phaseReduceBy.Table_Row IS_EQUAL_TO OriginalPhaseReduceBy. | | Pass/Fail |
| | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.16 Configure Phase Minimum Gap Time

| Test Procedure: | | Configure Phase Minimum Gap Time | | |
|---|--|--|-----------|--|
| Description: | | This test case verifies that the ASC allows a management station to configure the minimum time the gap between vehicles reported by detector actuations must be before a phase can be terminated. | | |
| Requirement | (s): | • 3.5.2.1.2.1.16 Configure Phase Minimum Gap Time | | |
| Variable(s): | ble(s): MaxRows Table_Row OriginalPhaseMinimumGap TestPhaseMinimumGap phaseMinimumGap | | | |
| Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this | | |
| Test Step Number | Test Procedu | e | Results | |
| 1 | GET the follow | ing objects: maxPhases. | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: phaseMinimumGap.Table_Row. Pass/Fail | | | |
| 3.1 | RECO | RD this information as OriginalPhaseMinimumGap. | | |
| 4 | ASSIGN TestP | haseMinimumGap EQUALS RANDOM (0 TO 255). | | |

| 4.1 | IF TestPhaseMinimumGap IS_EQUAL_TO OriginalPhaseMinimumGap. | | |
|----------------|--|-----------------|-----------|
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN phaseMinimumGap.Table_Row EQUALS TestPhaseMinimumGap. | | |
| 6 | SET the following objects: phaseMinimumGap | o.Table_Row. | Pass/Fail |
| 7 | GET the following objects: phaseMinimumGa | p.Table_Row. | Pass/Fail |
| 8 | VERIFY phaseMinimumGap.Table_Row IS_EQUAL_TO TestPhaseMinimumGap. | | Pass/Fail |
| 9 | ASSIGN phaseMinimumGap.Table_Row EQUALS OriginalPhaseMinimumGap. | | |
| 10 | SET the following objects: phaseMinimumGap.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: phaseMinimumGap.Table_Row. | | Pass/Fail |
| 12 | VERIFY phaseMinimumGap.Table_Row IS_EQUAL_TO OriginalPhaseMinimumGap. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.3.17 Configure Phase Dynamic Maximum Limit

| Test Procedure: | | Configure Phase Dynamic Maximum Limit | |
|---------------------|--|--|--------------|
| Description: | | This test case verifies that the ASC allows a managen station to configure an upper or lower time limit of the Indication. | ent GREEN |
| Requirement(| s): | • 3.5.2.1.2.1.17 Configure Phase Dynamic Maxin | num Limit |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseDynamicMaxLiphaseDynamicMaxLmitTestPhaseDynamicMaxLimit | imit imit |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | e. | Results |
| 1 | GET the follow | ing objects: maxPhases. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: phaseDynamicMaxLimit.Table_Row. | | Pass/Fail |
| 3.1 | RECORD this information as OriginalPhaseDynamicMaxLimit. | | |
| 4 | ASSIGN TestP | haseDynamicMaxLimit EQUALS RANDOM (0 TO 255). | |
| 4.1 | IF Test OriginalPhase[| tPhaseDynamicMaxLimit IS_EQUAL_TO | |

| 4.1.1 | GOTO step 4. | | |
|---------------|--|------------------|-----------|
| 5 | ASSIGN phaseDynamicMaxLimit.Table_Row TestPhaseDynamicMaxLimit. | EQUALS | |
| 6 | SET the following objects: phaseDynamicMax | Limit.Table_Row. | Pass/Fail |
| 7 | GET the following objects: phaseDynamicMax | Limit.Table_Row. | Pass/Fail |
| 8 | VERIFY phaseDynamicMaxLimit.Table_Row TestPhaseDynamicMaxLimit. | IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN phaseDynamicMaxLimit.Table_Row OriginalPhaseDynamicMaxLimit. | EQUALS | |
| 10 | SET the following objects: phaseDynamicMax | Limit.Table_Row. | Pass/Fail |
| 11 | GET the following objects: phaseDynamicMax | Limit.Table_Row. | Pass/Fail |
| 12 | VERIFY phaseDynamicMaxLimit.Table_Row OriginalPhaseDynamicMaxLimit. | IS_EQUAL_TO | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |

C.3.3.18 Configure Phase Dynamic Maximum Limit

| Test Procedure: | | Configure Phase Dynamic Maximum Step | |
|---------------------|--------------------------|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure how much the allowable maximum time a phase's GREEN Indication may be incremented or decremented before reaching its limit. | |
| Requirement(| s): | • 3.5.2.1.2.1.18 Configure Phase Dynamic Maxin | num Step |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseDynamicMaxStphaseDynamicMaxStepTestPhaseDynamicMaxStepphaseDynamicMaxStep | tep tep |
| Pass/Fail Crite | eria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | e | Results |
| 1 | GET the follow | ing objects: maxPhases. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 3 | GET the follow | ing objects: phaseDynamicMaxStep.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalPhaseDynamicMaxStep. | |
| 4 | ASSIGN TestF | haseDynamicMaxStep EQUALS RANDOM (0 TO 255). | |
| 4.1 | IF Tes OriginalPhaseI | PhaseDynamicMaxStep IS_EQUAL_TO DynamicMaxStep. | |
| 4.1.1 | | GOTO step 4. | |

| 5 | ASSIGN phaseDynamicMaxStep.Table_Row TestPhaseDynamicMaxStep. | EQUALS | |
|----------------|--|----------------------------|-----------|
| 6 | SET the following objects: phaseDynamicMax | Step.Table_Row. | Pass/Fail |
| 7 | GET the following objects: phaseDynamicMax | <pre>Step.Table_Row.</pre> | Pass/Fail |
| 8 | VERIFY phaseDynamicMaxStep.Table_Row TestPhaseDynamicMaxStep. | IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN phaseDynamicMaxStep.Table_Row OriginalPhaseDynamicMaxStep. | EQUALS | |
| 10 | SET the following objects: phaseDynamicMax | Step.Table_Row. | Pass/Fail |
| 11 | GET the following objects: phaseDynamicMax | <pre>Step.Table_Row.</pre> | Pass/Fail |
| 12 | VERIFY phaseDynamicMaxStep.Table_Row OriginalPhaseDynamicMaxStep. | IS_EQUAL_TO | Pass/Fail |
| Test Procedu | re Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.19 Configure Phase Startup State

| Test Procedure: | | Configure Phase Startup State | |
|---------------------|----------------|---|------------|
| Description: | | This test case verifies that the ASC allows a manager station to set a startup state for a phase. | nent |
| Requirement(| s): | • 3.5.2.1.2.1.19 Configure Phase Startup State | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseStartupphaseStartupTestPhaseStartupphaseStartup | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification statest case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded'. | |
| 2 | GET the follow | ing objects: maxPhases. | Pass/Fail |
| 2.1 | RECO | RD this information as MaxRows. | |
| 3 | ASSIGN Table | Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: phaseStartup.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalPhaseStartup. | |
| 5 | ASSIGN TestF | PhaseStartup EQUALS 2. | |
| 5.1 | NOTE | 2 = phaseNotOn. | |
| 6 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | eStartup.Table_Row EQUALS TestPhaseStartup. | |
| 8 | SET the follow | ing objects: phaseStartup.Table Row. | Pass/Fail |

| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
|----------------|--|---------------------------|-----------|
| 10 | GET the following objects: phaseStartup.Tabl | e_Row. | Pass/Fail |
| 11 | VERIFY phaseStartup.Table_Row IS_EQUAL | TO TestPhaseStartup. | Pass/Fail |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | te Database Transaction'. | |
| 13 | ASSIGN phaseStartup.Table_Row EQUALS | OriginalPhaseStartup. | |
| 14 | SET the following objects: phaseStartup.Table | e_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseStartup.Tabl | e_Row. | Pass/Fail |
| 17 | VERIFY phaseStartup.Table_Row IS_EQUAL_TO OriginalPhaseStartup. | | Pass/Fail |
| Test Procedu | re Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.20 Configure Automatic Flash Entry Phase

| Test Procedure: | | Configure Automatic Flash Entry Phase | |
|---|---|--|---------------------------------------|
| Description: | | This test case verifies that the ASC allows a manager station to configure which phases must be serviced b going into Automatic Flash. | nent oefore |
| Requirement(| s): | • 3.5.2.1.2.1.20 Configure Automatic Flash Entry | / Phase |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification sto test case to pass the test case. | ep in this |
| | | | |
| Test Step Number | Test Procedur | 'e | Results |
| Test Step Number 1 | Test Procedur | ' e ON 'A valid timing plan is loaded'. | Results |
| Test Step Number 1 2 | Test Procedur PRE-CONDITI GET the follow | ' e ON 'A valid timing plan is loaded'. ing objects: maxPhases. | Results Pass/Fail |
| Test Step Number 1 2 2.1 | Test Procedur PRE-CONDITI GET the follow RECO | ' e ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. | Results Pass/Fail |
| Test Step Number122.13 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table | 'e ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). | Results Pass/Fail |
| Test Step Number122.134 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow | 'e ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. | Results Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 | Test Procedur PRE-CONDITION GET the follow RECO ASSIGN Table GET the follow RECO | e ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. haseOptions EQUALS OriginalPhaseOptions XOR 2. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 6 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the | 'e ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. haseOptions EQUALS OriginalPhaseOptions XOR 2. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 6 7 | Test Procedur PRE-CONDITH GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the ASSIGN phase | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. ThaseOptions EQUALS OriginalPhaseOptions XOR 2. Test Procedure 'C.3.1.1 Create Database Transaction'. eOptions.Table_Row EQUALS TestPhaseOptions. | Results Pass/Fail Pass/Fail Pass/Fail |

| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
|----------------|--|---------------------------|-----------|
| 10 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
| 11 | VERIFY phaseOptions.Table_Row IS_EQUA | L_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | te Database Transaction'. | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 14 | SET the following objects: phaseOptions.Tabl | le_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedur | re Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.21 Configure Automatic Flash Exit Phase

| Test Procedure: | | Configure Automatic Flash Exit Phase | |
|---|---|---|---------------------------------------|
| Description: | | This test case verifies that the ASC allows a manager station to configure which phases are serviced when Flash ends. | nent Automatic |
| Requirement(| s): | • 3.5.2.1.2.1.21 Configure Automatic Flash Exit | Phase |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification sto | ep in this |
| | | lest case to pass the lest case. | |
| Test Step Number | Test Procedur | re | Results |
| Test Step Number 1 | Test Procedur | re ON 'A valid timing plan is loaded'. | Results |
| Test Step Number 1 2 | Test Procedur PRE-CONDITI GET the follow | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. | Results Pass/Fail |
| Test Step Number 1 2 2.1 | Test Procedur PRE-CONDITI GET the follow RECO | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. | Results Pass/Fail |
| Test Step Number122.13 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). | Results Pass/Fail |
| Test Step Number122.134 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. haseOptions EQUALS OriginalPhaseOptions XOR 4. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 6 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. haseOptions EQUALS OriginalPhaseOptions XOR 4. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 6 7 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the ASSIGN phase | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. haseOptions EQUALS OriginalPhaseOptions XOR 4. Test Procedure 'C.3.1.1 Create Database Transaction'. eOptions.Table_Row EQUALS TestPhaseOptions. | Results Pass/Fail Pass/Fail Pass/Fail |

| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
|---------------|--|--------------------------|-----------|
| 10 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
| 11 | VERIFY phaseOptions.Table_Row IS_EQUA | L_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
| 16 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUA OriginalPhaseOptions. | L_TO | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |

C.3.3.22 Configure Call to Non-Actuated 1

| Test Procedure: | | Configure Call to Non-Actuated 1 | |
|--|---|--|---|
| Description: | | This test case verifies that the ASC allows a manag station to configure a phase to respond to the Call t Actuated 1 input. | ement to Non- |
| Requirement(| s): | • 3.5.2.1.2.1.22 Configure Call to Non-Actuate | d 1 |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification s test case to pass the test case. | step in this |
| | | | |
| Test Step Number | Test Procedu | re | Results |
| Test Step Number 1 | Test Procedui PRE-CONDITI | re ON 'A valid timing plan is loaded'. | Results |
| Test Step Number 1 2 | Test Procedur PRE-CONDITI GET the follow | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. | Results Pass/Fail |
| Test Step Number 1 2 2.1 | Test Procedur PRE-CONDITI GET the follow RECO | r e ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. | Results Pass/Fail |
| Test Step Number 1 2 2.1 3 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). | Results Pass/Fail |
| Test StepNumber122.134 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. | Results Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN Test | RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 8. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 6 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the | Pre ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. a Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 8. a Test Procedure 'C.3.1.1 Create Database Transaction'. | Results Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 6 7 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the ASSIGN phase | ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. Comparison as MaxRows. Compari | Results Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 4 4.1 5 6 7 8 | Test Procedur PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the ASSIGN phase SET the follow | ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. Prove EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 8. PhaseOptions EQUALS OriginalPhaseOptions XOR 8. Test Procedure 'C.3.1.1 Create Database Transaction'. PhaseOptions.Table_Row EQUALS TestPhaseOptions. ing objects: phaseOptions.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 10 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
|----------------|--|--------------------------|-----------|
| 11 | VERIFY phaseOptions.Table_Row IS_EQUA | L_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
| 16 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUA OriginalPhaseOptions. | L_TO | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.23 Configure Call to Non-Actuated 2

| Test Procedure: | | Configure Call to Non-Actuated 2 | |
|--|--|--|---|
| Description: | | This test case verifies that the ASC allows a managestation to configure a phase to respond to the Call Actuated 2 input. | jement to Non- |
| Requirement | (s): | • 3.5.2.1.2.1.23 Configure Call to Non-Actuate | ed 2 |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Crit | teria: | The device under test shall pass every verification step in the test case to pass the test case. | |
| Test Sten | | | |
| Number | Test Procedu | re | Results |
| Number 1 | PRE-CONDITI | re ON 'A valid timing plan is loaded'. | Results |
| Number 1 2 | Test ProcedurePRE-CONDITIGET the follow | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. | Results Pass/Fail |
| Number 1 2 2.1 | Test Procedure PRE-CONDITI GET the follow RECO | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. RD this information as MaxRows. | Results Pass/Fail |
| Number 1 2 2.1 3 | Test Procedure PRE-CONDITI GET the follow RECO ASSIGN Table | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. RD this information as MaxRows. e_Row EQUALS RANDOM (1 TO MaxRows). | Results Pass/Fail |
| Number 1 2 2.1 3 4 | Test Procedure PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. RD this information as MaxRows. e_Row EQUALS RANDOM (1 TO MaxRows). ring objects: phaseOptions.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 | Test ProcedurePRE-CONDITIGET the followRECOASSIGN TableGET the followRECO | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. RD this information as MaxRows. Row EQUALS RANDOM (1 TO MaxRows). ring objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. | Results Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 | Test ProcedurePRE-CONDITIGET the followRECOASSIGN TableGET the followRECOASSIGN TestF | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. RD this information as MaxRows. e_Row EQUALS RANDOM (1 TO MaxRows). ring objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 16. | Results Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 6 | Test Procedure PRE-CONDITI GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN Table GET the follow RECO PRESIGN Table PERFORM the | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. RD this information as MaxRows. a_Row EQUALS RANDOM (1 TO MaxRows). ring objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 16. a Test Procedure 'C.3.1.1 Create Database Transaction'. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 6 7 | Test ProcedurePRE-CONDITIGET the followRECOASSIGN TableGET the followGET the followRECOASSIGN TestFPERFORM theASSIGN phase | re ON 'A valid timing plan is loaded'. ing objects: maxPhases. RD this information as MaxRows. c_Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 16. c Test Procedure 'C.3.1.1 Create Database Transaction'. eOptions.Table_Row EQUALS TestPhaseOptions. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Results |
| Number 1 2 2.1 3 4 4.1 5 6 7 8 | Test ProcedurePRE-CONDITIGET the followRECOASSIGN TableGET the followRECOASSIGN TestFPERFORM theASSIGN phaseSET the follow | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. RD this information as MaxRows. e_Row EQUALS RANDOM (1 TO MaxRows). ring objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 16. e Test Procedure 'C.3.1.1 Create Database Transaction'. eOptions.Table_Row EQUALS TestPhaseOptions. ing objects: phaseOptions.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 6 7 8 9 | Test ProcedurePRE-CONDITIGET the followRECOASSIGN TableGET the followGET the followRECOASSIGN TestFPERFORM theASSIGN phaseSET the followPERFORM theVerify Database | re ON 'A valid timing plan is loaded'. ring objects: maxPhases. RD this information as MaxRows. a Row EQUALS RANDOM (1 TO MaxRows). ring objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 16. a Test Procedure 'C.3.1.1 Create Database Transaction'. eOptions.Table_Row EQUALS TestPhaseOptions. ing objects: phaseOptions.Table_Row. a Test Procedure 'C.3.1.2 Verify Database Status and se Error'. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Tested By: | | Date Tested: | Pass/Fail |
|--------------|--|--------------------------|-----------|
| Test Procedu | re Results | | |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 11 | VERIFY phaseOptions.Table_Row IS_EQUA | L_TO TestPhaseOptions. | Pass/Fail |

C.3.3.24 Configure Non-Lock Detector Memory

| Test Procedure: | Test Procedure: Configure Non-Lock Detector Memory | | |
|-----------------------------------|---|---|---|
| Description: | | This test case verifies that the ASC allows a manager station to cause a call to the phase to lock at the begi the phase's yellow change interval. This allows calls even if actuations are removed before the phase is se again. | nent Inning of to remain erviced |
| Requirement(| s): | 3.5.2.1.2.1.24 Configure Non-Lock Detector Me 3.5.2.1.2.1.41 Configure Non-Lock Ped Detected | emory or Memory |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedure | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded'. | |
| 2 | GET the follow | ing objects: maxPhases. | Pass/Fail |
| 2.1 | RECORD this information as MaxRows. | | |
| | RECO. | RD this information as maxRows. | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 3 4 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 3 4 4.1 | ASSIGN Table GET the follow RECO | RD this information as maxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. | Pass/Fail |
| 3 4 4.1 5 | ASSIGN Table GET the follow RECO ASSIGN TestP | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 32. | Pass/Fail |
| 3 4 4.1 5 6 | ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the | Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 32. Test Procedure 'C.3.1.1 Create Database Transaction'. | Pass/Fail |
| 3 4 4.1 5 6 7 | ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the ASSIGN phase | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. haseOptions EQUALS OriginalPhaseOptions XOR 32. Test Procedure 'C.3.1.1 Create Database Transaction'. eOptions.Table_Row EQUALS TestPhaseOptions. | Pass/Fail |
| 3 4 4.1 5 6 7 8 | ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the ASSIGN phase SET the follow | Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS OriginalPhaseOptions XOR 32. Test Procedure 'C.3.1.1 Create Database Transaction'. eOptions.Table_Row EQUALS TestPhaseOptions. ing objects: phaseOptions.Table_Row. | Pass/Fail Pass/Fail Pass/Fail |

| 10 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
|----------------|--|--------------------------|-----------|
| 11 | VERIFY phaseOptions.Table_Row IS_EQUA | L_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.25 Configure Phase Minimum Vehicle Recall

| Test Procedure: | ocedure: Configure Phase Minimum Recall | | | | |
|---------------------|---|---|--|---|------------------------------|
| Description: | | This test case verif station to cause a r phase to be served time. | ies that the A ecurring call for at least th | SC allows a manage for vehicle service e ne phase's minimum | ment kists for a Green |
| Requirement | (s): | • 3.5.2.1.2.1.2 | 5 Configure F | Phase Minimum Vehi | cle Recall |
| Variable(s): | | MaxRows Table_Row OriginalPhaseOpt TestPhaseOptions | ions | maxPhases Int phaseOptions phaseOptions | |
| Pass/Fail Crit | teria: | The device under te test case to pass the | est shall pass le test case. | every verification st | ep in this |
| Test Step Number | Test Procedu | | | | Results |
| 1 | PRE-CONDITI | N 'A valid timing plar | n is loaded'. | | |
| 2 | GET the follow | ig objects: maxPhase | es. | | Pass/Fail |
| 2.1 | RECO | D this information as | MaxRows. | | |
| 3 | ASSIGN Table | Row EQUALS RANE | DOM (1 TO Ma | xRows). | |
| 4 | GET the follow | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail | |
| 4.1 | RECO | D this information as | OriginalPhase | eOptions. | |
| 5 | ASSIGN TestPhaseOptions EQUALS OriginalPhaseOptions XOR 64. | | | | |
| 6 | PERFORM the | Fest Procedure C.3. | 1.1 Create Dat | abase Transaction'. | |
| 7 | ASSIGN phase | Options.Table_Row E | QUALS TestF | haseOptions. | |
| 8 | SET the follow | g objects: phaseOpti | ons.Table_Ro | W. | Pass/Fail |
| 9 | PERFORM the Verify Databas | Fest Procedure 'C.3.' Error'. | 1.2 Verify Data | base Status and | |

| 10 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
|----------------|--|--------------------------|-----------|
| 11 | VERIFY phaseOptions.Table_Row IS_EQUA | L_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.26 Configure Phase Maximum Vehicle Recall

| Test Procedure: | edure: Configure Phase Maximum Vehicle Recall | | |
|--|--|---|---|
| Description: | | This test case verifies that the ASC allows a manager station to cause a recurring call for a phase to be ser- maximum time that may be allocated to the phase. | nent ved for the |
| Requirement | (s): | • 3.5.2.1.2.1.26 Configure Phase Maximum Vehi | cle Recall |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Crit | eria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Tost Stop | | | |
| Number | Test Procedu | re | Results |
| Number 1 | Test Procedur | re e Test Procedure "A valid timing plan is loaded". | Results |
| Number 1 2 | Test Procedure PERFORM the GET the follow | re e Test Procedure "A valid timing plan is loaded". ing objects: maxPhases. | Results Pass/Fail |
| Number122.1 | Test Procedur PERFORM the GET the follow RECO | re e Test Procedure "A valid timing plan is loaded". ing objects: maxPhases. RD this information as MaxRows. | Results Pass/Fail |
| Number 1 2 2.1 3 | Test Procedure PERFORM the GET the follow RECO ASSIGN Table | re e Test Procedure "A valid timing plan is loaded". ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). | Results Pass/Fail |
| Number 1 2 2.1 3 4 | Test Procedure PERFORM the GET the follow RECO ASSIGN Table GET the follow | re = Test Procedure "A valid timing plan is loaded". ing objects: maxPhases. RD this information as MaxRows. = Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 | Test Procedur PERFORM the GET the follow RECO ASSIGN Table GET the follow RECO | re = Test Procedure "A valid timing plan is loaded". ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. | Results Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 | Test Procedur PERFORM the GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestF | re Test Procedure "A valid timing plan is loaded". ing objects: maxPhases. RD this information as MaxRows. Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS 'OriginalPhaseOptions XOR 128'. | Results Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 6 | Test Procedure PERFORM the GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN Table PERFORM the | re Test Procedure "A valid timing plan is loaded". Ting objects: maxPhases. RD this information as MaxRows. Prove EQUALS RANDOM (1 TO MaxRows). Ting objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS 'OriginalPhaseOptions XOR 128'. Test Procedure 'C.3.1.1 Create Database Transaction' | Results Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 6 7 | Test Procedur PERFORM the GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the ASSIGN phase | re a Test Procedure "A valid timing plan is loaded". ing objects: maxPhases. RD this information as MaxRows. <u>CROW EQUALS RANDOM (1 TO MaxRows).</u> ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS 'OriginalPhaseOptions XOR 128'. a Test Procedure 'C.3.1.1 Create Database Transaction' eOptions.Table_Row EQUALS TestPhaseOptions. | Results Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 6 7 8 | Test Procedur PERFORM the GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP PERFORM the ASSIGN phase SET the follow | re Test Procedure "A valid timing plan is loaded". Ting objects: maxPhases. RD this information as MaxRows. Prove EQUALS RANDOM (1 TO MaxRows). Ting objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS 'OriginalPhaseOptions XOR 128'. Test Procedure 'C.3.1.1 Create Database Transaction' Procedure 'C.3.1.1 Create Database Transaction' PhaseOptions.Table_Row EQUALS TestPhaseOptions. Ting objects: phaseOptions.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Number 1 2 2.1 3 4 4.1 5 6 7 8 9 | Test Procedur PERFORM the GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestF PERFORM the SET the follow PERFORM the Verify Databas | re a Test Procedure "A valid timing plan is loaded". ing objects: maxPhases. RD this information as MaxRows. a Row EQUALS RANDOM (1 TO MaxRows). ing objects: phaseOptions.Table_Row. RD this information as OriginalPhaseOptions. PhaseOptions EQUALS 'OriginalPhaseOptions XOR 128'. a Test Procedure 'C.3.1.1 Create Database Transaction' eOptions.Table_Row EQUALS TestPhaseOptions. ing objects: phaseOptions.Table_Row. a Test Procedure 'C.3.1.2 Verify Database Status and a Error'. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 11 | VERIFY phaseOptions.Table_Row IS_EQUA | TO TestPhaseOptions. | Pass/Fail |
|----------------|--|--------------------------|-----------|
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL OriginalPhaseOptions. | Pass/Fail | |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.27 Configure Phase Soft Vehicle Recall

| Test Procedure: | Configure Phase Soft Vehicle Recall | | |
|---------------------|--|--|-----------------------------|
| Description: | | This test case verifies that the ASC allows a manager station to cause a call is to be placed on a phase who conflicting phases are in resting in green or red, and no serviceable conflicting calls. | ment en all there are |
| Requirement | (s): | • 3.5.2.1.2.1.27 Configure Phase Soft Vehicle R | ecall |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Crit | eria: | The device under test shall pass every verification st test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded'. | |
| 2 | GET the follow | ing objects: maxPhases. | Pass/Fail |
| 2.1 | RECO | RD this information as MaxRows. | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 4.1 | RECORD this information as OriginalPhaseOptions. | | |
| 5 | ASSIGN TestPhaseOptions EQUALS OriginalPhaseOptions XOR 512. | | |
| 6 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | eOptions.Table_Row EQUALS TestPhaseOptions. | |
| 8 | SET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 9 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 4.0 | | ing objects: phaseOptions Table, Pow | Dace/Eail |

| 11 | VERIFY phaseOptions.Table_Row IS_EQUA | L_TO TestPhaseOptions. | Pass/Fail |
|----------------|--|--------------------------|-----------|
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedu | re Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.28 Configure Dual Phase Entry

| Test Procedure: | Test Procedure: Configure Dual Phase Entry | | |
|---------------------|---|---|---------------------|
| Description: | | This test case verifies that the ASC allows a manager station to configure a phase to be active upon entry in concurrency group when no calls exist in its own ring | nent nto a J. |
| Requirement(| s): | • 3.5.2.1.2.1.28 Configure Dual Phase Entry | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded'. | |
| 2 | GET the follow | ing objects: maxPhases. | Pass/Fail |
| 2.1 | RECO | RD this information as MaxRows. | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 4.1 | RECORD this information as MaxRows. | | |
| 5 | ASSIGN TestPhaseOptions EQUALS OriginalPhaseOptions XOR 1024. | | |
| 6 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | Options.Table_Row EQUALS TestPhaseOptions. | |
| 8 | SET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 9 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 10 | GET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 11 | VERIFY phase | Options.Table_Row IS_EQUAL_TO TestPhaseOptions. | Pass/Fail |

| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
|----------------|--|--------------------------|-----------|
| 13 | ASSIGN phaseOptions.Table_Row EQUALS | OriginalPhaseOptions. | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.3.29 Configure Simultaneous Gap Disable

| Test Procedure: | | Configure Simultaneous Gap Disable | | |
|---------------------|--|--|------------|--|
| Description: | | This test case verifies that the ASC allows a management station to configure if a gapped-out phase is allowed to revert to the extensible portion of the phase. | | |
| Requirement(| (s): | 3.5.2.1.2.1.29 Configure Simultaneous Gap Dis | sable | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | | |
| Pass/Fail Crit | eria: | The device under test shall pass every verification sto test case to pass the test case. | ep in this | |
| Test Step Number | Test Procedu | - .e | Results | |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded'. | | |
| 2 | GET the follow | ing objects: maxPhases. | Pass/Fail | |
| 2.1 | RECORD this information as MaxRows. | | | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | | |
| 4 | GET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail | |
| 4.1 | RECO | RD this information as MaxRows. | | |
| 5 | ASSIGN TestPhaseOptions EQUALS OriginalPhaseOptions XOR 2048. | | | |
| 6 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 7 | ASSIGN phase | Options.Table_Row EQUALS TestPhaseOptions. | | |
| 8 | SET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 10 | GET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail | |
| 11 | VERIFY phase | Options.Table_Row IS_EQUAL_TO TestPhaseOptions. | Pass/Fail | |
| 12 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction' | | |

| Test Procedure Notes: | | | |
|------------------------|--|-----------------|-----------|
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure Results | | | |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 14 | SET the following objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS OriginalPhaseOptions. | | |

C.3.3.30 Configure Guaranteed Passage

| Test Procedure: | | Configure Guaranteed Passage | |
|---------------------|--|--|------------|
| Description: | | This test case verifies that the ASC allows a managen station to enable/disable volume density mode. | nent |
| Requirement(| s): | • 3.5.2.1.2.1.30 Configure Guaranteed Passage | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITION 'A valid timing plan is loaded'. | | |
| 2 | GET the following objects: maxPhases. Pass/Fa | | Pass/Fail |
| 2.1 | RECORD this information as MaxRows. | | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalPhaseOptions. | |
| 5 | ASSIGN TestF | haseOptions EQUALS OriginalPhaseOptions XOR 4096. | |
| 6 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | eOptions.Table_Row EQUALS TestPhaseOptions. | |
| 8 | SET the following objects: phaseOptions.Table_Row. Pass/Fail | | Pass/Fail |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 10 | GET the following objects: phaseOptions.Table_Row. Pass/Fail | | Pass/Fail |
| 11 | VERIFY phase | Options.Table_Row IS_EQUAL_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 13 | ASSIGN phase | Options.Table Row EQUALS OriginalPhaseOptions. | |

| 14 | SET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
|----------------|--|-----------------|-----------|
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Tab | le_Row. | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.3.31 Configure Actuated Rest-in-Walk

| Test Procedure: | | Configure Actuated Rest-in-Walk | |
|---------------------|--|--|-------------------------|
| Description: | | This test case verifies that the ASC allows a managen station to set a phase to rest in Walk if there is no ser conflicting call at the end of the Walk time or if Maxim Vehicle Recall is enabled. | nent viceable ium |
| Requirement(| s): | • 3.5.2.1.2.1.31 Configure Actuated Rest-in-Wall | ¢ |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedur | re | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded'. | |
| 2 | GET the following objects: maxPhases. | | Pass/Fail |
| 2.1 | RECORD this information as MaxRows. | | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalPhaseOptions. | |
| 5 | ASSIGN TestP | haseOptions EQUALS OriginalPhaseOptions XOR 8192. | |
| 6 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | eOptions.Table_Row EQUALS TestPhaseOptions. | |
| 8 | SET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 10 | GET the following objects: phaseOptions.Table_Row. Pa | | Pass/Fail |
| 11 | VERIFY phase | Options.Table_Row IS_EQUAL_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 13 | ASSIGN phase | eOptions.Table_Row EQUALS OriginalPhaseOptions. | |

| 14 | SET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
|------------------------|--|-----------------|-----------|
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Tab | e_Row. | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.3.32 Configure Conditional Service Enable

| Test Procedure: | | Configure Conditional Service Enable | |
|---------------------|--|--|------------|
| Description: | | This test case verifies that the ASC allows a manager station to enable/disable conditional service. | ment |
| Requiremer | nt(s): | • 3.5.2.1.2.1.32 Configure Conditional Service E | Enable |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Criteria: | | The device under test shall pass every verification st test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITION 'A valid timing plan is loaded'. | | |
| 2 | GET the follow | GET the following objects: maxPhases. | |
| 2.1 | RECO | RECORD this information as MaxRows. | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as MaxRows. | |
| 5 | ASSIGN TestF 16384. | PhaseOptions EQUALS OriginalPhaseOptions XOR | |
| 6 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | ASSIGN phaseOptions.Table_Row EQUALS TestPhaseOptions. | |
| 8 | SET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 9 | PERFORM the Verify Databas | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | |
| 10 | GET the follow | GET the following objects: phaseOptions.Table_Row. Pass/Fail | |
| 11 | VERIFY phase | Options.Table_Row IS_EQUAL_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 13 | ASSIGN phase | Options.Table_Row EQUALS OriginalPhaseOptions. | |

| 14 | SET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
|------------------------|--|-----------------|-----------|
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.3.33 Configure Added Initial Calculation

| Test Procedure: | | Configure Added Initial Calculation | |
|---|--|---|---------------------|
| Description: | | This test case verifies that the ASC allows a manager station to set what detector values to use for the calc the variable portion of the green time. | ment culation of |
| Requiremen | nt(s): | • 3.5.2.1.2.1.33 Configure Added Initial Calculat | lion |
| Variable(s): MaxRows Table_Row OriginalPha TestPhase | | MaxRowsmaxPhasesTable_RowIntOriginalPhaseOptionsphaseOptionsTestPhaseOptionsphaseOptions | |
| Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITION 'A valid timing plan is loaded'. | | |
| 2 | GET the following objects: maxPhases. | | Pass/Fail |
| 2.1 | RECO | RECORD this information as MaxRows. | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalPhaseOptions. | |
| 5 | ASSIGN TestF 32768. | PhaseOptions EQUALS OriginalPhaseOptions XOR | |
| 6 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | ASSIGN phaseOptions.Table_Row EQUALS TestPhaseOptions. | |
| 8 | SET the follow | ing objects: phaseOptions.Table_Row. | Pass/Fail |
| 9 | PERFORM the Verify Databas | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | |
| 10 | GET the follow | GET the following objects: phaseOptions.Table_Row. Pass/Fail | |
| 11 | VERIFY phase | Options.Table_Row IS_EQUAL_TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 13 | ASSIGN phase | eOptions.Table_Row EQUALS OriginalPhaseOptions. | |

| 14 | SET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
|------------------------|--|-----------------|-----------|
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseOptions.Table_Row. | | Pass/Fail |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.3.34 Configure Phase-to-Ring Association

| Test Procedure: | | Configure Phase-to-Ring Association | |
|---|--|--|-----------|
| Description: | | This test case verifies that the ASC allows a manager station to set which ring a phase is in for a sequence | ment |
| Requiremer | nt(s): | • 3.5.2.1.2.1.34 Configure Phase-to-Ring Assoc | iation |
| Variable(s): | MaxRowsmaxPhasesVariable(s):Table_RowIntOriginalPhaseRingphaseRingTestPhaseRingphaseRing | | |
| Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITION 'A valid timing plan is loaded'. | | |
| 2 | GET the follow | GET the following objects: maxPhases. | |
| 2.1 | RECO | RECORD this information as MaxRows. | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: phaseRing.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalPhaseRing. | |
| 5 | USER-ACTION | N 'Select a valid value for phaseRing.Table_Row.' | |
| 5.1 | RECO | RD this information as TestPhaseRing. | |
| 6 | PERFORM the | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | eRing.Table_Row EQUALS TestPhaseRing. | |
| 8 | SET the follow | SET the following objects: phaseRing.Table_Row. Pass/Fail | |
| 9 | PERFORM the Verify Databas | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | |
| 10 | GET the follow | GET the following objects: phaseRing.Table_Row. Pass/Fai | |
| 11 | VERIFY phase | Ring.Table_Row IS_EQUAL_TO TestPhaseRing. | Pass/Fail |
| 12 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 13 | ASSIGN phase | eRing.Table_Row EQUALS OriginalPhaseRing. | |

| 14 | SET the following objects: phaseRing.Table_F | Row. | Pass/Fail |
|------------------------|--|-----------------|-----------|
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: phaseRing.Table_Row. | | Pass/Fail |
| 17 | VERIFY phaseRing.Table_Row IS_EQUAL_TO OriginalPhaseRing. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |

C.3.3.35 Configure Phase Concurrency

| Test Procedure: | | Configure Phase Concurrency | |
|---------------------|--|---|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure what other phases that may run concurrently with a phase. | |
| Requirement | (s): | • 3.5.2.1.2.1.35 Configure Phase Concurrency | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalConcurrencyphaseConcurrencyTestConcurrencyphaseConcurrency | |
| Pass/Fail Criteria: | | The device under test shall pass every verification sto test case to pass the test case. | ep in this |
| Test Step Number | Test Procedure | | Results |
| 1 | PRE-CONDITION 'A valid timing plan is loaded'. | | |
| 2 | GET the following objects: maxPhases. | | Pass/Fail |
| 2.1 | RECORD this information as MaxRows. | | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | GET the following objects: phaseConcurrency.Table_Row. | |
| 4.1 | RECO | RECORD this information as OriginalConcurrency. | |
| 5 | USER-ACTION Each octet mus | N 'Select a valid value for phaseConcurrency.Table_Row. st not be greater than MaxRows .' | |
| 5.1 | RECO | RD this information as TestConcurrency. | |
| 6 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 7 | ASSIGN phase | eConcurrency.Table_Row EQUALS TestConcurrency. | |
| 8 | SET the follow | ing objects: phaseConcurrency.Table_Row. | Pass/Fail |
| 9 | PERFORM the Verify Databas | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | |
| 10 | GET the follow | ing objects: phaseConcurrency.Table_Row. | Pass/Fail |
| 11 | VERIFY phase TestConcurren | VERIFY phaseConcurrency.Table_Row IS_EQUAL_TO TestConcurrency. | |
| 12 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 13 | ASSIGN phase | ASSIGN phaseConcurrency.Table_Row EQUALS OriginalConcurrency. | |

| 14 | SET the following objects: phaseConcurrency.Table_Row. | | Pass/Fail |
|----------------|--|-----------------|-----------|
| 15 | PERFORM the Test Procedure "C.3.1.2 Verify Database Status and Verify Database Error". | | |
| 16 | GET the following objects: phaseConcurrency | r.Table_Row. | Pass/Fail |
| 17 | VERIFY phaseConcurrency.Table_Row IS_EQUAL_TO OriginalConcurrency. | | |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.3.36 Configure Pedestrian Clearance Time Allowed During Vehicle Clearance

| Test Procedure: | | Configure Pedestrian Clearance Time Allowed During Vehicle Clearance | | |
|--|---|---|---|-----------------------------------|
| Description: | | This test case verifies that the ASC allows a management station to configure how long a pedestrian clearance interval may extend into the yellow change and red clearance intervals. | | nent interval intervals. |
| Requirement(| s): | 3.5.2.1.2.1.36 Configure Allowed During Vehicle | Pedestrian Clearance Clearance | Time |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalTimephasePedClearDuringVehicleTestTimeClearphasePedClearDuringVehicleClearClearClear | | ıgVehicle ıgVehicle |
| Pass/Fail Criteria: | | The device under test shall past test case to pass the test case. | s every verification ste | ep in this |
| - | Test Procedure | | | |
| Test Step Number | Test Procedur | re | | Results |
| Test Step Number 1 | Test Procedur | re ing objects: maxPhases. | | Results Pass/Fail |
| Test Step Number 1 1.1 | Test Procedur GET the follow RECO | re ing objects: maxPhases. RD this information as MaxRows. | | Results Pass/Fail |
| Test Step Number 1 1.1 2 | Test Procedur GET the follow RECO ASSIGN Table | re ing objects: maxPhases. RD this information as MaxRows. _Row EQUALS RANDOM (1 to Ma | xRows). | Results Pass/Fail |
| Test StepNumber11.123 | Test Procedur GET the follow RECO ASSIGN Table GET the follow phasePedClea | re RD this information as MaxRows. RD this information as MaxRows. Row EQUALS RANDOM (1 to Ma ving objects: rDuringVehicleClear.Table_Row. | xRows). | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow phasePedClea RECO | re re RD this information as MaxRows. <u>Row EQUALS RANDOM (1 to Ma</u> ving objects: rDuringVehicleClear.Table_Row. RD this information as OriginalTime | xRows). | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 | Test Procedur GET the follow RECO ASSIGN Table GET the follow phasePedClea RECO ASSIGN TestT | re RD this information as MaxRows. RD this information as MaxRows. Row EQUALS RANDOM (1 to Ma ving objects: rDuringVehicleClear.Table_Row. RD this information as OriginalTime Time EQUALS RANDOM (0 TO 255 | xRows). | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow phasePedClea RECO ASSIGN TestT IF Te | re ring objects: maxPhases. RD this information as MaxRows. Row EQUALS RANDOM (1 to Ma ving objects: rDuringVehicleClear.Table_Row. RD this information as OriginalTime rime EQUALS RANDOM (0 TO 255 stTime IS_EQUAL_TO OriginalTime | xRows). e. j. e. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow phasePedClea RECO ASSIGN TestT IF Test | re re RD this information as MaxRows. Prove EQUALS RANDOM (1 to Ma ving objects: rDuringVehicleClear.Table_Row. RD this information as OriginalTime ime EQUALS RANDOM (0 TO 255 stTime IS_EQUAL_TO OriginalTime GOTO step 4. | xRows). e.). e. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 | Test Procedur GET the follow RECO ASSIGN Table GET the follow phasePedClea RECO ASSIGN TestT IF Test ASSIGN phase TestTime. | re ring objects: maxPhases. RD this information as MaxRows. Row EQUALS RANDOM (1 to Ma ving objects: rDuringVehicleClear.Table_Row. RD this information as OriginalTime ime EQUALS RANDOM (0 TO 255 stTime IS_EQUAL_TO OriginalTime GOTO step 4. PedClearDuringVehicleClear.Table | xRows). 9. 9. 9. 9. 9. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 | Test Procedur GET the follow RECO ASSIGN Table GET the follow phasePedClea RECO ASSIGN TestT IF Test ASSIGN phase TestTime. SET the follow phasePedClea | re ring objects: maxPhases. RD this information as MaxRows. Row EQUALS RANDOM (1 to Ma ving objects: rDuringVehicleClear.Table_Row. RD this information as OriginalTime ime EQUALS RANDOM (0 TO 255 stTime IS_EQUAL_TO OriginalTime GOTO step 4. PedClearDuringVehicleClear.Table ing objects: rDuringVehicleClear.Table_Row. | xRows). e.). e. e. Row EQUALS | Results Pass/Fail Pass/Fail |

| 8 | VERIFY phasePedClearDuringVehicleClear.Table_Row IS_EQUAL_TO TestTime. | | | |
|----------------|--|--|-----------|--|
| 9 | ASSIGN phasePedClearDuringVehicleClear. OriginalTime. | ASSIGN phasePedClearDuringVehicleClear.Table_Row EQUALS OriginalTime. | | |
| 10 | SET the following objects: phasePedClearDuringVehicleClear.Table_Ro | SET the following objects: phasePedClearDuringVehicleClear.Table Row. | | |
| 11 | GET the following objects: phasePedClearDuringVehicleClear.Table Row. | | | |
| 12 | VERIFY phasePedClearDuringVehicleClear.Table_Row IS_EQUAL_TO OriginalTime. | | | |
| Test Procedu | re Results | | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure | Notes: | | | |

C.3.3.37 Configure Pedestrian Walk Time

| Test Procedure: | | Configure Pedestrian Walk Time | | |
|--|--|--|---|---|
| Description: | | This test case verifies that the ASC allows a management station to configure the pedestrian walk time for a phase. | | |
| Requirement | (s): | • 3.5.2.1.2.1.37 Configure Phase Walk Time | | |
| Variable(s): | ble(s): Table_Row OriginalPhaseWalk TestPhaseWalk MaxRows MaxPhases | | | |
| Pass/Fail Criteria: The device under test shall pass every verificat test case to pass the test case. | | pass every verification ste ase. | ep in this | |
| Test Step | Test Procedure | | | |
| Number | Test Procedur | re | | Results |
| Number 1 | GET the follow | re ing objects: maxPhases. | | Results Pass/Fail |
| Number 1 1.1 | GET the follow RECO | re ing objects: maxPhases. RD this information as MaxRov | WS. | Results Pass/Fail |
| Number 1 1.1 2 | GET the follow RECO ASSIGN Table | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 | ws. TO MaxRows). | Results Pass/Fail |
| Number 1 1.1 2 3 | GET the follow RECO ASSIGN Table GET the follow | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ | ws. TO MaxRows). Row. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 | GET the follow RECO ASSIGN Table GET the follow RECO | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ RD this information as Original | ws. O MaxRows). Row. PhaseWalk. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 | Test ProcedurGET the followASSIGN TableGET the followRECOASSIGN TestP | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ RD this information as Original PhaseWalk EQUALS RANDOM | ws. TO MaxRows). Row. PhaseWalk. I (0 TO 255). | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ RD this information as Original PhaseWalk EQUALS RANDOM tPhaseWalk IS_EQUAL_TO O | ws. TO MaxRows). Row. PhaseWalk. I (0 TO 255). riginalPhaseWalk. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 4.1.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ RD this information as Original haseWalk EQUALS RANDOM tPhaseWalk IS_EQUAL_TO O GOTO step 4. | ws. TO MaxRows). Row. PhaseWalk. I (0 TO 255). riginalPhaseWalk. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test ASSIGN phase | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ RD this information as Original PhaseWalk EQUALS RANDOM tPhaseWalk IS_EQUAL_TO O GOTO step 4. Walk.Table_Row EQUALS Te | ws. TO MaxRows). Row. PhaseWalk. I (0 TO 255). riginalPhaseWalk. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test ASSIGN phase SET the follow | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ RD this information as Original PhaseWalk EQUALS RANDOM tPhaseWalk IS_EQUAL_TO O GOTO step 4. Walk.Table_Row EQUALS Technology ing objects: phaseWalk.Table | vs. O MaxRows). Row. PhaseWalk. I (0 TO 255). riginalPhaseWalk. estPhaseWalk. Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 7 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test ASSIGN phase SET the follow GET the follow | re ing objects: maxPhases. RD this information as MaxRov _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ RD this information as Original PhaseWalk EQUALS RANDOM tPhaseWalk IS_EQUAL_TO O GOTO step 4. Walk.Table_Row EQUALS Technology ing objects: phaseWalk.Table_ ing objects: phaseWalk.Table_ | ws. TO MaxRows). Row. PhaseWalk. I (0 TO 255). riginalPhaseWalk. estPhaseWalk. Row. Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 7 8 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test ASSIGN phase SET the follow GET the follow VERIFY phase | re ing objects: maxPhases. RD this information as MaxRou _Row EQUALS RANDOM (1 1 ing objects: phaseWalk.Table_ RD this information as Original PhaseWalk EQUALS RANDOM tPhaseWalk EQUALS RANDOM tPhaseWalk IS_EQUAL_TO O GOTO step 4. Walk.Table_Row EQUALS Te ing objects: phaseWalk.Table_ ing objects: phaseWalk.Table_ Walk.Table_Row IS_EQUAL_ | ws. TO MaxRows). Row. PhaseWalk. I (0 TO 255). riginalPhaseWalk. estPhaseWalk. Row. Row. TO TestPhaseWalk. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 10 | SET the following objects: phaseWalk.Table_Row. | | | |
|------------------------|---|-----------------------|-----------|--|
| 11 | GET the following objects: phaseWalk.Table_Row. | | | |
| 12 | VERIFY phaseWalk.Table_Row IS_EQUAL_ | TO OriginalPhaseWalk. | Pass/Fail | |
| Test Procedure Results | | | | |
| | | | | |
| Tested By: | | Date Tested: | Pass/Fail | |

C.3.3.38 Configure Pedestrian Clearance Time

| Test Procedure: | Configure Pedestrian Clearance Time | | |
|---------------------|---|--|------------|
| Description: | | This test case verifies that the ASC allows a managen station the pedestrian clearance time for a phase. | nent |
| Requirement(| s): | • 3.5.2.1.2.1.38 Configure Pedestrian Clearance | Time |
| Variable(s): | | MaxRowsmaxPhasesOriginalPhasePedClearphasePedestrianCleTestPhasePedClearphasePedestrianCleTable_RowInt | ar ar |
| Pass/Fail Crit | eria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedure | | Results |
| 1 | GET the following objects: maxPhases. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: phasePedestrianClear.Table_Row. | | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalPhasePedClear. | |
| 4 | ASSIGN TestF | hasePedClear EQUALS RANDOM (0 TO 255). | |
| 4.1 | IF TestPhasePedClear IS_EQUAL_TO OriginalPhasePedClear. | | |
| 4.1.1 | | GOTO step 4. | |
| 5 | ASSIGN phase TestPhasePed | PedestrianClear.Table_Row EQUALS Clear. | |
| 6 | SET the following objects: phasePedestrianClear.Table_Row. | | Pass/Fail |
| 7 | GET the following objects: phasePedestrianClear.Table_Row. Pa | | Pass/Fail |
| 8 | VERIFY phasePedestrianClear.Table_Row IS_EQUAL_TO TestPhasePedClear. | | Pass/Fail |
| 9 | ASSIGN phasePedestrianClear.Table_Row EQUALS OriginalPhasePedClear. | | |
| 10 | SET the follow | ing objects: phasePedestrianClear.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: phasePedestrianClear.Table_Row. | Pass/Fail |
| 12 | VERIFY phase OriginalPhase | PedestrianClear.Table_Row IS_EQUAL_TO PedClear. | Pass/Fail |

| Test Procedure Results | | |
|------------------------|-----------------|-----------|
| Tested By: | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | |

C.3.3.39 Configure Pedestrian Phase Walk Service Limit

| Test Procedure: | Configure Pedestrian Phase Walk Service Limit | | | |
|---------------------|---|---|---|--------------|
| Description: | | This test case verifies that the station to configure how maindication to be shown again | the ASC allows a managen any times the pedestrian W in within the same cycle. | nent /alk |
| Requirement(| s): | • 3.5.2.1.2.1.39 Config Limit | jure Pedestrian Phase Wal | k Service |
| Variable(s): | | MaxRows Table_Row OriginalLimit TestPedLimit | maxPhases Int phasePedServiceLin phasePedServiceLin | nit nit |
| Pass/Fail Crit | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | ep in this |
| Test Step Number | Test Procedu | .e | | Results |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | VS. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | o MaxRows). | |
| 3 | GET the following objects: phasePedServiceLimit.Table_Row. | | Limit.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as OriginalLimit | | Limit | |
| 4 | ASSIGN TestL | imit EQUALS RANDOM (0 TO | 255). | |
| 4.1 | IF Te | stLimit IS_EQUAL_TO Original | Limit. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phase | PedServiceLimit.Table_Row E | QUALS TestLimit. | |
| 6 | SET the follow | ing objects: phasePedServiceL | imit.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: phasePedServiceL | .imit.Table_Row. | Pass/Fail |
| 8 | VERIFY phase | PedServiceLimit.Table_Row IS | S_EQUAL_TO TestLimit. | Pass/Fail |
| 9 | ASSIGN phase | PedServiceLimit.Table_Row E | QUALS OriginalLimit. | |
| 10 | SET the follow | SET the following objects: phasePedServiceLimit.Table_Row. | | Pass/Fail |
| 11 | GET the follow | ing objects: phasePedServiceL | .imit.Table_Row. | Pass/Fail |
| 12 | VERIFY phasePedServiceLimit.Table_Row IS_EQUAL_TO OriginalLimit. | | Pass/Fail | |
| Test Procedur | e Results | | | • |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

| C.3.3.40 | Configure Pedestrian Phase Don't Walk Revert Tin | ne |
|----------|--|----|
|----------|--|----|

| Test Procedure: Configure Pedestrian Phase Don't Walk Revert Til | | e Don't Walk Revert Time | | | |
|---|---|---|--|--------------------|--|
| Description: | | This test case verifies that t station to configure the min indication. | the ASC allows a manager nimum time for a Don't Wa | nent Ik | |
| Requirement(| s): | • 3.5.2.1.2.1.40 Config Revert Time | jure Pedestrian Phase Dor | n't Walk | |
| Variable(s): | | MaxRows Table_Row OriginalDontWalkRevert TestPedDontWalkRevert | maxPhases Int phaseDontWalkRev phaseDontWalkRev | kRevert kRevert | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this | |
| Test Step Number | Test Procedur | - | | Results | |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail | |
| 1.1 | RECO | RD this information as MaxRov | VS. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | | | |
| 3 | GET the following objects: phaseDontWalkRevert.Table_Row. | | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalDontWalkRevert. | | | | |
| 4 | ASSIGN TestD | ontWalkRevert EQUALS RAN | DOM (0 TO 255). | | |
| 4.1 | IF Te OriginalDontW | stDontWalkRevert IS_EQUAL_ alkRevert | ТО | | |
| 4.1.1 | onginalbont | GOTO step 4 | | | |
| 5 | ASSIGN phase TestDontWalk | DontWalkRevert.Table_Row E | QUALS | | |
| 6 | SET the followi | ng objects: phaseDontWalkRe | vert.Table_Row. | Pass/Fail | |
| 7 | GET the follow | ing objects: phaseDontWalkRe | vert.Table_Row. | Pass/Fail | |
| 8 | VERIFY phase TestDontWalk | DontWalkRevert.Table_Row IS Revert. | S_EQUAL_TO | Pass/Fail | |
| 9 | ASSIGN phase OriginalDontW | DontWalkRevert.Table_Row E alkRevert. | QUALS | | |
| 10 | SET the following objects: phaseDontWalkRevert.Table_Row. Pass | | | Pass/Fail | |
| 11 | GET the follow | ing objects: phaseDontWalkRe | vert.Table_Row. | Pass/Fail | |
| 12 | VERIFY phaseDontWalkRevert.Table_Row IS_EQUAL_TO OriginalDontWalkRevert. | | | Pass/Fail | |
| Test Procedur | e Results | | | | |
| Tested By: | Date Tested: | | Pass/Fail | | |
| Test Procedure | lotes: | | | | |

C.3.3.41 Configure Pedestrian Phase Recall

Configure Pedestrian Phase Recall

| Test Procedure: | | | | |
|---------------------|--|---|--|-----------|
| Description: | | This test case verifies that the ASC allows a management station to cause 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. | | |
| Requirement(| s): | • 3.5.2.1.2.1.42 Config | jure Pedestrian Phase Rec | all |
| Variable(s): | | MaxRows Table_Row OriginalPhaseOptions TestPhaseOptions | maxPhases Int phaseOptions phaseOptions | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loade | ed'. | |
| 2 | GET the following objects: maxPhases. | | | Pass/Fail |
| 2.1 | RECORD this information as MaxRows. | | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 4 | GET the following objects: phaseOptions.Table_Row. | | le_Row. | Pass/Fail |
| 4.1 | RECORD this information as OriginalPhaseOptions. | | | |
| 5 | ASSIGN TestPhaseOptions EQUALS OriginalPhaseOptions XOR 256. | | PhaseOptions XOR 256. | |
| 6 | PERFORM the | Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 7 | ASSIGN phase | Options.Table_Row EQUALS | TestPhaseOptions. | |
| 8 | SET the followi | ng objects: phaseOptions.Tabl | e_Row. | Pass/Fail |
| 9 | PERFORM the Verify Databas | Test Procedure 'C.3.1.2 Verify e Error'. | Database Status and | |
| 10 | GET the follow | ing objects: phaseOptions.Tabl | le_Row. | Pass/Fail |
| 11 | VERIFY phase | Options.Table_Row IS_EQUAL | TO TestPhaseOptions. | Pass/Fail |
| 12 | PERFORM the | Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 13 | ASSIGN phaseOptions.Table_Row EQUALS OriginalPhaseOptions. | | | |
| 14 | SET the following objects: phaseOptions.Table_Row. Pas | | | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 16 | GET the following objects: phaseOptions.Table_Row. Pass | | Pass/Fail | |
| 17 | VERIFY phaseOptions.Table_Row IS_EQUAL_TO OriginalPhaseOptions. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | ed By: Date Tested: | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

| C.3.3.42 | Configure Phase | Alternate Pedestrian | Clearance Time |
|----------|------------------------|-----------------------------|-----------------------|
|----------|------------------------|-----------------------------|-----------------------|

| Test Procedure: | | Configure Phase Alternate Pedestrian Clearance Time | | |
|-------------------------------|---|---|--|--------------------|
| Description: | | This test case verifies that t station to configure an alter phase. | he ASC allows a managen nate pedestrian clearance | ient time for a |
| Requirement(| s): | 3.5.2.1.2.1.43 Configure Alternate Pedestrian Clearance Time 3.5.2.1.13.1.3 Configure APS Extra Crossing Time | | |
| Variable(s): | | MaxRows maxPhases Table_Row Int OriginalPedAlternateClearance phasePedAlternateClearance TestPedAlternateClearance phasePedAlternateClearance | | arance arance |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste se. | p in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | | |
| 3 | GET the following objects: phasePedAlternateClearance.Table_Row. Pass | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPedAlternateClearance. | | | |
| 4 | ASSIGN TESTPEDALTERNATE CLEARANCE EQUALS RANDOM (UTO 255). | | | |
| 4.1 | OriginalPedAlternateClearance | | | |
| 411 | GOTO step 4. | | | |
| | ASSIGN phasePedAlternateClearance Table Row FOUALS | | | |
| 5 | TestPedAlterna | ateClearance. | | |
| 6 | SET the followi | ng objects: phasePedAlternate | Clearance.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: phasePedAlternate | Clearance.Table_Row. | Pass/Fail |
| 8 | VERIFY phase TestPedAlterna | PedAlternateClearance.Table_ ateClearance. | Row IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN phasePedAlternateClearance.Table_Row EQUALS OriginalPedAlternateClearance. | | | |
| 10 | SET the following objects: phasePedAlternateClearance.Table_Row. Pass/F | | Pass/Fail | |
| 11 | GET the follow | ing objects: phasePedAlternate | Clearance.Table_Row. | Pass/Fail |
| 12 | VERIFY phasePedAlternateClearance.Table_Row IS_EQUAL_TO OriginalPedAlternateClearance. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: Date Tested: Pass/ | | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.3.43 Configure Phase Alternate Pedestrian Walk Time

1

Configure Phase Alternate Pedestrian Walk Time

| Test Procedure: | | | | |
|------------------------------------|---|---|--|-----------|
| | | | | |
| Description: | | This test case verifies that t station to configure an alter phase. | ne ASC allows a managen nate pedestrian walk time | for a |
| Requirement(| s): | 3.5.2.1.2.1.44 Configure Alternate Pedestrian Walk Time 3.5.2.1.13.1.3 Configure APS Extra Crossing Time | | |
| Variable(s): | | MaxRows Table_Row OriginalPedAlternateWalk TestPedAlternateWalk | maxPhases Int phasePedAlternateWalk phasePedAlternateWalk | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the following objects: maxPhases. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | /S. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | o MaxRows). | |
| 3 | GET the following objects: phasePedAlternateWalk.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPedAlternateWalk. | | | |
| 4 | ASSIGN TestPedAlternateWalk EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestPedAlternateWalk IS_EQUAL_TO OriginalPedAlternateWalk. | | ТО | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN phasePedAlternateWalk.Table_Row EQUALS TestPedAlternateWalk. | | | |
| 6 | SET the followi | ng objects: phasePedAlternate | Walk.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: phasePedAlternate | Walk.Table_Row. | Pass/Fail |
| 8 | VERIFY phasePedAlternateWalk.Table_Row IS_EQUAL_TO TestPedAlternateWalk. | | IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN phasePedAlternateWalk.Table_Row EQUALS OriginalPedAlternateWalk. | | | |
| 10 | SET the following objects: phasePedAlternateWalk.Table_Row. | | Pass/Fail | |
| 11 | GET the following objects: phasePedAlternateWalk.Table_Row. | | Walk.Table_Row. | Pass/Fail |
| 12 VERIFY phase OriginalPedAlte | | PedAlternateWalk.Table_Row ernateWalk. | IS_EQUAL_TO | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |

C.3.3.44 Configure Pedestrian Phase Advanced Walk Time

| Description: | Description:This test case verifies that the ASC allows a manageDescription:station to configure the pedestrian WALK indicationdisplayed before the start of the GREEN indication | | he ASC allows a managen lestrian WALK indication to f the GREEN indication | nent o be |
|-------------------------|---|--------------------------------------|--|----------------------|
| Requirement(| • 3.5.2.1.2.1.45 Configure Pedestrian Phase Adva Walk Time | | anced | |
| Variable(s): | e(s): MaxRows Table_Row OriginalAdvan TestAdvanceT | | maxPhases Int phasePedAdvanceD phasePedAdvanceD | elayTime elayTime |
| Pass/Fail Crite | il Criteria: The device under test shall pass every verification step in test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRow | /S. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | | |
| 3 | GET the following objects: phasePedAdvanceWalkTime.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalAdvanceTime. | | | |
| 4 | ASSIGN TestAdvanceTimeTime EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestAdvanceTime IS_EQUAL_TO OriginalAdvanceTime. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN phasePedAdvanceWalkTime.Table_Row EQUALS TestAdvanceTime. | | | |
| 6 | SET the following objects: phasePedAdvanceWalkTime.Table_Row. | | Pass/Fail | |
| 7 | GET the follow | ing objects: phasePedAdvance | WalkTime.Table_Row. | Pass/Fail |
| 8 | VERIFY phasePedAdvanceWalkTime.Table_Row IS_EQUAL_TO TestAdvanceTime. | | Pass/Fail | |
| 9 | ASSIGN phase OriginalAdvance | PedAdvanceWalkTime.Table_ ceTime. | Row EQUALS | |
| 10 | SET the follow | ing objects: phasePedAdvance | WalkTime.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: phasePedAdvance | WalkTime.Table_Row. | Pass/Fail |
| 12 | VERIFY phasePedAdvanceWalkTime.Table_Row IS_EQUAL_TO OriginalAdvanceTime. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: Date Pass/Fa | | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.3.45 Configure Pedestrian Phae Delayed Walk Time

| Test Procedure: | Configure Pedestrian Phase Delayed Walk Time |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to configure the pedestrian WALK indication to be displayed after the start of the GREEN indication |

| Requirement(s): | | • 3.5.2.1.2.1.46 Config Time | jure Pedestrian Phase Dela | ayed Walk |
|---------------------|--|--|--|-----------|
| Variable(s): | | MaxRows Table_Row OriginalDelayTime TestDelayTime | maxPhases Int phasePedDelayTime phasePedDelayTime | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 to | o MaxRows). | |
| 3 | GET the following objects: phasePedDelayTime.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalDelayTime. | | | |
| 4 | ASSIGN TestDelayTime EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestDelayTime IS_EQUAL_TO OriginalDelayTime. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phase | PedDelayTime.Table_Row EC | QUALS TestDelayTime. | |
| 6 | SET the following objects: phasePedDelayTime.Table_Row. Pa | | Pass/Fail | |
| 7 | GET the following objects: phasePedDelayTime.Table_Row. | | Pass/Fail | |
| 8 | VERIFY phasePedDelayTime.Table_Row IS_EQUAL_TO TestDelayTime. | | Pass/Fail | |
| 9 | ASSIGN phase | PedDelayTime.Table_Row EC | UALS OriginalDelayTime. | |
| 10 | SET the followi | ng objects: phasePedDelayTin | ne.Table_Row. | Pass/Fail |
| 11 | GET the following objects: phasePedDelayTime.Table_Row. | | Pass/Fail | |
| 12 | VERIFY phasePedDelayTime.Table_Row IS_EQUAL_TO OriginalDelayTime. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.3.46 Configure Advance Warning Green

| Test Procedure: | Configure Advance Green Warning | |
|--------------------|--|--|
| Description: | This test case verifies that the ASC allows a management station to configure the time for an advanced warning signal before a GREEN indication. | |
| Requirement(s): | • 3.5.2.1.2.1.47 Configure Advance Green Warning | |
| Variable(s): | MaxRows Table_Row OriginalTime | maxPhases Int phaseAdvWarnGrnStartTime |

| | | TestTime | phaseAdvWarnGrnS | tartTime |
|-------------------------|---|---|-----------------------|-----------|
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 to | o MaxRows). | |
| 3 | GET the follow | ving objects: phaseAdvWarnGr | nStartTime.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Time. | |
| 4 | ASSIGN TestT | ime EQUALS RANDOM (0 TO | 255). | |
| 4.1 | NOTE 'The value should not exceed the total amount of the clearance time of the phase that is being terminated prior to the start of this phase'. | | | |
| 4.1 | IF TestTime IS_EQUAL_TO OriginalTime. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN phaseAdvWarnGrnStartTime.Table_Row EQUALS TestTime. | | | |
| 6 | SET the following objects: phaseAdvWarnGrnStartTime.Table_Row. F | | Pass/Fail | |
| 7 | GET the follow | ing objects: phaseAdvWarnGrr | StartTime.Table_Row. | Pass/Fail |
| 8 | VERIFY phaseAdvWarnGrnStartTime.Table_Row IS_EQUAL_TO TestTime. | | Pass/Fail | |
| 9 | ASSIGN phaseAdvWarnGrnStartTime.Table_Row EQUALS OriginalTime. | | | |
| 10 | SET the following objects: phaseAdvWarnGrnStartTime.Table Row. | | Pass/Fail | |
| 11 | GET the following objects: phaseAdvWarnGrnStartTime.Table Row. Pa | | Pass/Fail | |
| 12 | VERIFY phaseAdvWarnGrnStartTime.Table_Row IS_EQUAL_TO OriginalTime. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: Date Tested: | | Pass/Fail | | |
| Test Procedure | Notes: | | | |

C.3.3.47 Configure Advance Warning Red

| Test Procedure: | Configure Advance Rec | Configure Advance Red Warning | | |
|--------------------|--|--|--|--|
| Description: | This test case verifies t station to configure the before a RED indication | This test case verifies that the ASC allows a management station to configure the time for an advanced warning signal before a RED indication. | | |
| Requirement(s): | • 3.5.2.1.2.1.48 Co | • 3.5.2.1.2.1.48 Configure Advance Red Warning | | |
| Variable(s): | MaxRows Table_Row OriginalTime TestTime | maxPhases Int phaseAdvWarnRedStartTime phaseAdvWarnRedStartTime | | |

| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification ste test case to pass the test case. | | p in this | |
|---------------------|--|---------------------------------|------------------------|-----------|
| Test Step Number | Test Procedur | Test Procedure | | Results |
| 1 | GET the followi | ing objects: maxPhases. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | vs. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 to | o MaxRows). | |
| 3 | GET the follow | ving objects: phaseAdvWarnRe | edStartTime.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Time. | |
| 4 | ASSIGN TestT | ime EQUALS RANDOM (0 TO | 255). | |
| 4.1 | IF Tes | stTime IS_EQUAL_TO Origina | ITime. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phaseAdvWarnRedStartTime.Table_Row EQUALS TestTime. | | | |
| 6 | SET the following objects: phaseAdvWarnRedStartTime.Table_Row. Pass/F | | Pass/Fail | |
| 7 | GET the followi | ing objects: phaseAdvWarnRed | dStartTime.Table_Row. | Pass/Fail |
| 8 | VERIFY phaseAdvWarnRedStartTime.Table_Row IS_EQUAL_TO TestTime. Pas | | Pass/Fail | |
| 9 | ASSIGN phaseAdvWarnRedStartTime.Table_Row EQUALS OriginalTime. | | | |
| 10 | SET the following objects: phaseAdvWarnRedStartTime.Table_Row. | | Pass/Fail | |
| 11 | GET the following objects: phaseAdvWarnRedStartTime.Table_Row. | | Pass/Fail | |
| 12 | VERIFY phaseAdvWarnRedStartTime.Table_Row IS_EQUAL_TO OriginalTime. | | Pass/Fail | |
| Test Procedur | est Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |

C.3.3.48 Configure Alternate Minimum Green During Transition

| Test Procedure: | | Configure Alternate Minimum Green During Transition | | |
|--|------------------------------------|--|----------|--|
| Description: | | This test case verifies that the ASC allows a management station to configure an alternate minimum green time to be used during transition between cycles. | | |
| Requirement(| s): | 3.5.2.1.2.1.51 Configure Alternate Minimum Vehicle Green Time during Transition | | |
| Variable(s): Variable(s): MaxRows Table_Rov OriginalTi TestTime | | MaxRowsmaxPhasesTable_RowIntOriginalTimephaseAltMinTimeTraTestTimephaseAltMinTimeTra | Insition | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Step Number Test Procedure | | Results | |

| 1 | GET the following objects: maxPhases. | | Pass/Fail |
|----------------|---|---------------------|-----------|
| 1.1 | RECORD this information as MaxRov | /S. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | |
| 3 | GET the following objects: phaseAltMinTimeTransition.Table_Row. | | |
| 3.1 | RECORD this information as OriginalTime. | | |
| 4 | ASSIGN TestTime EQUALS RANDOM (0 TO | 255). | |
| 4.1 | IF TestTime IS_EQUAL_TO Origina | Time. | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN phaseAltMinTimeTransition.Table_R | ow EQUALS TestTime. | |
| 6 | SET the following objects: phaseAltMinTimeTransition.Table Row. | | |
| 7 | GET the following objects: phaseAltMinTimeTransition.Table_Row. | | Pass/Fail |
| 8 | VERIFY phaseAltMinTimeTransition.Table_Row IS_EQUAL_TO TestTime. | | Pass/Fail |
| 9 | ASSIGN phaseAltMinTimeTransition.Table_Row EQUALS OriginalTime. | | |
| 10 | SET the following objects: phaseAltMinTimeTransition.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: phaseAltMinTimeTransition.Table_Row. | | Pass/Fail |
| 12 | VERIFY phaseAltMinTimeTransition.Table_Row IS_EQUAL_TO OriginalTime. | | |
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |

C.3.3.49 Configure Alternate Pedestrian Walk Time During Transition

| Test Procedure: | | Configure Alternate Pedestrian Walk Time During Transition | | sition |
|--|-------------------------------------|---|---|-----------|
| Description: | | This test case verifies that the ASC allows a management station to configure an alternate minimum walk time to be used during transition between cycles. | | |
| Requirement | (s): | 3.5.2.1.2.1.52 Configure Alternate Minimum Pedestrian Walk Time during Transition | | destrian |
| Variable(s): | | MaxRowsmaxPTable_RowIntOriginalTimephaseTestTimephase | maxPhases v Int ne phaseWalkDuringTransition phaseWalkDuringTransition | |
| Pass/Fail Criteria:The device under test shall pass every verification ste test case to pass the test case. | | verification ste | p in this | |
| Test Step Number | Test Procedu | - .e | | Results |
| 1 | GET the follow | GET the following objects: maxPhases. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | |
| 3 | GET the follow | ving objects: phaseWalkDuringTransition.Ta | able_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalTime. | | |

| 4 | ASSIGN TestTime EQUALS RANDOM (0 TO | 255). | |
|----------------|---|----------------------|-----------|
| 4.1 | IF TestTime IS_EQUAL_TO OriginalTime. | | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN phaseWalkDuringTransition.Table_F | Row EQUALS TestTime. | |
| 6 | SET the following objects: phaseWalkDuringT | ransition.Table_Row. | Pass/Fail |
| 7 | GET the following objects: phaseWalkDuringT | ransition.Table_Row. | Pass/Fail |
| 8 | VERIFY phaseWalkDuringTransition.Table_Row IS_EQUAL_TO TestTime. | | |
| 9 | ASSIGN phaseWalkDuringTransition.Table_Row EQUALS OriginalTime. | | |
| 10 | SET the following objects: phaseWalkDuringTransition.Table_Row. | | |
| 11 | GET the following objects: phaseWalkDuringTransition.Table_Row. | | |
| 12 | VERIFY phaseWalkDuringTransition.Table_Row IS_EQUAL_TO OriginalTime. | | Pass/Fail |
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |

C.3.3.50 Configure Alternate Pedestrian Clearance Time During Transition

| Test Procedure: | | Configure Alternate Pedestrian Clearance Time Durin Transition | ng | |
|---------------------|-------------------------------------|---|--------------------------|--|
| Description: | | This test case verifies that the ASC allows a management station to configure an alternate minimum pedestrian clearance time to be used during transition between cycles. | | |
| Requirement | t(s): | 3.5.2.1.2.1.53 Configure Alternate Minimum Pedestrian Clearance Time during Transition | | |
| Variable(s): | | MaxRowsmaxPhasesTable_RowIntOriginalTimephasePedClearDuriTestTimeonphasePedClearDuriononon | ngTransiti ngTransiti | |
| Pass/Fail Criteria: | | The device under test shall pass every verification st test case to pass the test case. | ep in this | |
| Test Step Number | Test Procedu | re | Results | |
| 1 | GET the follow | ing objects: maxPhases. | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | |
| 3 | GET the follow | GET the following objects: phasPedClearDuringTransition.Table_Row. | | |
| 3.1 | RECO | RECORD this information as OriginalTime. | | |
| 4 | ASSIGN TestT | ime EQUALS RANDOM (0 TO 255). | | |
| 4.1 | IF Te | stTime IS EQUAL TO OriginalTime. | | |

| 4.1.1 | GOTO step 4. | | |
|----------------|---|---------------------------|-----------|
| 5 | ASSIGN phasePedClearDuringTransition.Table_Row EQUALS TestTime. | | |
| 6 | SET the following objects: phasePedClearDu | ringTransition.Table_Row. | Pass/Fail |
| 7 | GET the following objects: phasePedClearDu | ringTransition.Table_Row. | Pass/Fail |
| 8 | VERIFY phasePedClearDuringTransition.Tab TestTime. | le_Row IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN phasePedClearDuringTransition.Table_Row EQUALS OriginalTime. | | |
| 10 | SET the following objects: phasePedClearDuringTransition.Table_Row. | | |
| 11 | GET the following objects: phasePedClearDuringTransition.Table_Row. | | Pass/Fail |
| 12 | VERIFY phasePedClearDuringTransition.Table_Row IS_EQUAL_TO OriginalTime. | | |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | - | |

C.3.3.51 Determine Maximum Number of Phases

| Test Procedure: | | Determine Maximum Number of Phases | |
|---------------------|--------------------------------|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of phases supported by ASC and verifies that the ASC supports the minimum number of phases required by the user. | |
| Requirement(| s): | 3.4.1.1 Retrieve Data 3.5.2.1.2.2.1 Determine Maximum Number of Phases | |
| Variable(s): | | UserMinPhases maxPhases | |
| Pass/Fail Criteria: | | The device under test shall pass every verification statest case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | re | Results |
| 1 | USER-ACTION in FR ID 3.5.2. | V Determine the number of phases required as specified 1.2.3.1 of the PRL. | |
| 1.1 | RECO | RD this information as UserMinPhases. | |
| 2 | GET the follow | ing objects: maxPhases. | Pass/Fail |
| 3 | VERIFY maxP | hases IS_NOT_LESS_THAN UserMinPhases. | Pass/Fail |
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

| C.3.3.52 | Determine Maximum | Number of Phase Sets |
|----------|-------------------|-----------------------------|
|----------|-------------------|-----------------------------|

| Test Procedure: | | Determine Maximum Number of Phases Sets | | |
|---------------------|---|--|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of phase sets supported by ASC and verifies that the ASC supports the minimum number of phase sets required by the user. | | |
| Requirement(| s): | 3.5.2.1.2.2.2 Determine Maximum Number of Phase Sets | | hase Sets |
| Variable(s): | | UserMinPhaseSets | maxPhasesSets | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | USER-ACTION specified in FR | V 'Determine the number of phates of the PRL.' | ases sets required as | |
| 1.1 | RECO | RD this information as UserMir | nPhaseSets. | |
| 2 | GET the follow | ing objects: maxPhasesSets. | | Pass/Fail |
| 3 | VERIFY maxPhases IS NOT LESS THAN UserMinPhaseSets. | | JserMinPhaseSets. | Pass/Fail |
| Test Procedur | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.4 Coordination

C.3.4.1 Configure Operational Mode for Coordination

| Test Procedure: | | Configure Operational Mod | e for Coordination | |
|---------------------|--|---|--|---------------|
| Description: | | This test case verifies that t station to the mode for ope | the ASC allows a managem rational mode for coordina | nent tion. |
| Requirement(| s): | • 3.5.2.1.3.1 Configure | e Operational Mode for Co | ordination |
| Variable(s): | | OriginalMode TestMode | coordOperationalMo coordOperationalMo | ode ode |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: coordOperationalMode. | | lode. | Pass/Fail |
| 1.1 | RECORD this information as OriginalMode. | | | |
| 2 | ASSIGN TestMode EQUALS RANDOM (0 TO 255). | | | |
| 2.1 | IF Test | IF TestMode IS_EQUAL_TO OriginalMode. | | |
| 2.1.1 | | GOTO step 2. | | |
| 3 | ASSIGN coord | OperationalMode EQUALS Te | stMode. | |
| 4 | SET the followi | ng objects: coordOperationalM | ode. | Pass/Fail |
| 5 | GET the follow | ing objects: coordOperationalM | lode. | Pass/Fail |
| 6 | VERIFY coord | OperationalMode IS_EQUAL_1 | O TestMode. | Pass/Fail |
| 7 | ASSIGN coord | OperationalMode EQUALS Ori | ginalMode. | |
| 8 | SET the following objects: coordOperationalMode. | | ode. | Pass/Fail |
| 9 | GET the following objects: coordOperationalMode. | | lode. | Pass/Fail |
| 10 | VERIFY coord | ERIFY coordOperationalMode IS_EQUAL_TO OriginalMode. Pase | | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.2 Configure Correction Mode for Coordination

| Test Procedure: | Configure Correction Mode for | Coordination | |
|--------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to set the coordination correction mode. | | |
| Requirement(s): | • 3.5.2.1.3.2 Configure Correction Mode for Coordination | | |
| Variable(s): | OriginalMode TestMode | coordCorrectionMode coordCorrectionMode | |

| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | ep in this |
|------------------------|--|---|-----------------|------------|
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: coordCorrectionMo | ode. | Pass/Fail |
| 1.1 | RECO | RD this information as Original | Mode. | |
| 2 | ASSIGN TestM | lode EQUALS RANDOM (2 TC | D 5). | |
| 2.1 | IF Test | Mode IS_EQUAL_TO Original | Mode. | |
| 2.1.1 | | GOTO step 2. | | |
| 3 | ASSIGN coord | ASSIGN coordCorrectionMode EQUALS TestMode. | | |
| 4 | SET the following objects: coordCorrectionMode. Pass/Fai | | | Pass/Fail |
| 5 | GET the following objects: coordCorrectionMode. Pass/Fai | | Pass/Fail | |
| 6 | VERIFY coordCorrectionMode IS_EQUAL_TO TestMode. Pass/Fail | | | Pass/Fail |
| 7 | ASSIGN coord | ASSIGN coordCorrectionMode EQUALS OriginalMode. | | |
| 8 | SET the following objects: coordCorrectionMode. Pass/Fa | | Pass/Fail | |
| 9 | GET the follow | ing objects: coordCorrectionMo | ode. | Pass/Fail |
| 10 | VERIFY coordCorrectionMode IS_EQUAL_TO OriginalMode. Pass/Fail | | | Pass/Fail |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.4.3 Configure Maximum Mode for Coordination

| Test Procedure: | | Configure Maximum Mode for Coordination | | |
|---------------------|--|---|------------------|--|
| Description: | | This test case verifies that the ASC allow a mar to set the coordination maximum. | agement station | |
| Requirement(s): | | • 3.5.2.1.3.3 Configure Maximum Mode for Coordination | | |
| Variable(s): | | OriginalMode coordMaximu TestMode coordMaximu | mMode mMode | |
| Pass/Fail Criteria: | | The device under test shall pass every verificat test case to pass the test case. | ion step in this | |
| Test Step Number | Test Procedu | e | Results | |
| 1 | GET the follow | ing objects: coordMaximumMode. | Pass/Fail | |
| 1.1 | RECORD this information as OriginalMode. | | | |
| 2 | ASSIGN TestMode EQUALS RANDOM (2 TO 5). | | | |
| 2.1 | IF TestMode IS_EQUAL_TO OriginalMode. | | | |
| 2.1.1 | | GOTO step 2. | | |
| 3 | ASSIGN coord | MaximumMode EQUALS TestMode. | | |
| 4 | SET the following objects: coordMaximumMode. | | Pass/Fail | |
| 5 | GET the follow | ing objects: coordMaximumMode. | Pass/Fail | |
| 6 | VERIFY coord | MaximumMode IS_EQUAL_TO TestMode. | Pass/Fail | |
| 7 | ASSIGN coord | MaximumMode EQUALS OriginalMode. | | |
| 8 | SET the follow | ng objects: coordMaximumMode. | Pass/Fail | |
| 9 | GET the follow | ing objects: coordMaximumMode. | Pass/Fail | |

| 10 | VERIFY coordMaximumMode IS_EQUAL_TC |) OriginalMode. | Pass/Fail | | |
|------------------------|-------------------------------------|-----------------|-----------|--|--|
| Test Procedure Results | | | | | |
| Tested By: | | Date Tested: | Pass/Fail | | |
| Test Procedure N | lotes: | | | | |

C.3.4.4 Configure Unit-Level Force Mode for Coordination

| Test | | Configure Unit-Level Force Mode for Coordination | | |
|---------------------|--|--|--|-----------|
| Description: | | This test case verifies that the ASC allows a management station to set a default setting for forcing of phases during coordination. | | |
| Requirement(| s): | 3.5.2.1.3.4 Configure Unit-level Force Mode for Coordination | | |
| Variable(s): | | OriginalMode TestMode | unitCoordForceMod unitCoordForceMod | e e |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | GET the following objects: unitCoordForceMode. | | Pass/Fail |
| 1.1 | RECORD this information as OriginalMode. | | Mode. | |
| 2 | IF OriginalMode IS_EQUAL_TO 2. | | | |
| 2.1 | ASSIGN TestMode EQUALS 3. | | | |
| 2.2 | GOTO step 4 | | | |
| 3 | ASSIGN TestM | 1ode EQUALS 2. | | |
| 4 | ASSIGN coord | ForceMode EQUALS TestMod | е. | |
| 5 | SET the followi | ing objects: unitCoordForceMo | de. | Pass/Fail |
| 6 | GET the follow | ing objects: unitCoordForceMo | de. | Pass/Fail |
| 7 | VERIFY coord | ForceMode IS_EQUAL_TO Ter | stMode. | Pass/Fail |
| 8 | ASSIGN coord | ForceMode EQUALS Original | /lode. | |
| 9 | SET the following objects: unitCoordForceMode. Pass/Fa | | Pass/Fail | |
| 10 | GET the following objects: unitCoordForceMode. Pas | | Pass/Fail | |
| 11 | VERIFY coordForceMode IS_EQUAL_TO OriginalMode. Pa | | Pass/Fail | |
| Test Procedur | e Results | | | _ |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Jotes: | | | |

C.3.4.5 Configure Phase-Level Force Mode for Coordination

| Test Procedure: | Configure Phase-Level Force Mode for Coordination |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to set the setting for forcing of phases during coordination that can override the default setting and vary by pattern. |

| Requirement(s): | | 3.5.2.1.3.5 Configue Coordination | re Phase-level Force Mode | for |
|---------------------|--|--|--|------------|
| Variable(s): | | OriginalMode TestMode MaxSplits MaxPhases Selected_Phase Selected_Split | splitCoordForceMoo splitCoordForceMoo maxSplits maxPhases Int Int | le le |
| Pass/Fail Crite | eria: | The device under test sha test case to pass the test | II pass every verification ste case. | ep in this |
| Test Step Number | Test Procedur | re | | Results |
| 1 | GET the follow | ing objects: maxSplits. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxSr | olits. | |
| 2 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 2.1 | RECO | RD this information as MaxPl | nases. | |
| 3 | ASSIGN Selec | ted_Split EQUALS RANDOM | (1 TO MaxSplits). | |
| 4 | ASSIGN Selec | ted_Phase EQUALS RANDC | M (1 TO MaxPhases). | |
| 5 | GET the following objects: splitCoordForceMode.Selected Split.Selected Phase. | | Pass/Fail | |
| 5.1 | RECORD this information as OriginalForceMode. | | | |
| 6 | ASSIGN TestMode EQUALS RANDOM (2 TO 4). | | | |
| 6.1 | IF Test | tMode IS_EQUAL_TO Origin | alMode. | |
| 6.1.1 | | GOTO step 6. | | |
| 7 | ASSIGN splitC TestMode. | oordForceMode.Selected_Sp | lit.Selected_Phase EQUALS | |
| 8 | SET the followi splitCoordForce | ing objects: eMode.Selected_Split.Select | ed_Phase. | Pass/Fail |
| 9 | GET the follow splitCoordForce | ing objects: eMode.Selected_Split.Select | ed_Phase. | Pass/Fail |
| 10 | VERIFY splitCo IS_EQUAL_TO | oordForceMode.Selected_Sp) TestMode. | lit.Selected_Phase | Pass/Fail |
| 11 | ASSIGN splitCoordForceMode.Selected_Split.Selected_Phase EQUALS OriginalMode. | | | |
| 12 | SET the followi splitCoordForce | SET the following objects: splitCoordForceMode.Selected_Split.Selected_Phase. | | Pass/Fail |
| 13 | GET the follow splitCoordForce | the following objects: coordForceMode.Selected_Split.Selected_Phase. | | Pass/Fail |
| 14 | VERIFY splitCoordForceMode.Selected_Split.Selected_Phase IS_EQUAL_TO OriginalMode. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.6 Configure Pattern Reference Phase

Configure Pattern Reference Phase

| Test Procedure: | | | | |
|---------------------|--|--|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure the phase that is to be active when the pattern cycle starts and ends. | | |
| Requirement(| s): | • 3.5.2.1.3.6 Configure | • 3.5.2.1.3.6 Configure Pattern Reference Phase | |
| Variable(s): | | MaxRows Table_Row OriginalPhase TestPhase | maxPatterns Int patternReferencePha patternReferencePha | ise Ise |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste se. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxPatterns. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | | |
| 3 | GET the following objects: patternReferencePhase.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPhase. | | | |
| 4 | ASSIGN TestPhase EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestPhase IS_EQUAL_TO OriginalPhase. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN patter | nReferencePhase.Table_Row | EQUALS TestPhase. | |
| 6 | SET the followi | ng objects: patternReferencePl | nase.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: patternReferenceP | hase.Table_Row. | Pass/Fail |
| 8 | VERIFY patternReferencePhase.Table_Row IS_EQUAL_TO TestPhase. | | Pass/Fail | |
| 9 | ASSIGN patter | nReferencePhase.Table_Row | EQUALS OriginalPhase. | |
| 10 | SET the followi | ng objects: patternReferencePl | nase.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: patternReferenceP | hase.Table_Row. | Pass/Fail |
| 12 | VERIFY patternReferencePhase.Table_Row IS_EQUAL_TO OriginalPhase. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |

C.3.4.7 Configure Pattern Reference Point

| Test Procedure: | Configure Pattern Reference Point |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to select an interval within the reference phase to specify the point when the pattern cycle starts and ends. |

| Requirement(s): | | • 3.5.2.1.3.7 Configure | e Pattern Reference Point | |
|---------------------|--|---|--|-----------|
| Variable(s): | | MaxRows Table_Row OriginalPoint TestPoint | maxPatterns Int patternReferencePoint patternReferencePoint | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxPatterns. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | | |
| 3 | GET the following objects: patternReferencePoint.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPoint. | | | |
| 4 | ASSIGN TestPoint EQUALS RANDOM (2 TO 5). | | | |
| 4.1 | IF TestPoint IS_EQUAL_TO OriginalPoint. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN patternReferencePhase.Table_Row EQUALS TestPoint. | | | |
| 6 | SET the following objects: patternReferencePoint.Table_Row. | | Pass/Fail | |
| 7 | GET the following objects: patternReferencePoint.Table_Row. | | Pass/Fail | |
| 8 | VERIFY patternReferencePoint.Table_Row IS_EQUAL_TO TestPoint. | | Pass/Fail | |
| 9 | ASSIGN patternReferencePoint.Table_Row EQUALS OriginalPoint. | | | |
| 10 | SET the following objects: patternReferencePoint.Table_Row. Pass/Fa | | | Pass/Fail |
| 11 | GET the following objects: patternReferencePoint.Table_Row. | | Pass/Fail | |
| 12 | VERIFY patternReferencePoint.Table_Row IS_EQUAL_TO OriginalPoint. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | Date Tested: | | | Pass/Fail |

C.3.4.8 **Configure Omit Phases During Transitions**

| Test Procedure: | Configure Omit Phases Dur | Configure Omit Phases During Transitions | | |
|--------------------|---|--|--|--|
| Description: | This test case verifies that t station to configure phases transition between cycles. | This test case verifies that the ASC allows a management station to configure phases that may be omitted during a transition between cycles. | | |
| Requirement(s): | • 3.5.2.1.3.8 Configure | • 3.5.2.1.3.8 Configure Omit Phases During Transitions | | |
| Variable(s): | OriginalSplitOptions TestSplitOptions MaxSplits MaxPhases | splitOptions splitOptions maxSplits maxPhases | | |

| | | Selected_Phase Selected_Split | Int Int | |
|---------------------|--|---|-------------------------------------|-----------|
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxSplits. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxSpli | ts. | |
| 2 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 2.1 | RECO | RD this information as MaxPha | ses. | |
| 3 | ASSIGN Selec | ted_Split EQUALS RANDOM (| 1 TO MaxSplits). | |
| 4 | ASSIGN Selec | ted_Phase EQUALS RANDON | l (1 TO MaxPhases). | |
| 5 | GET the follow | ing objects: splitOptions.Select | ed_Split.Selected_Phase. | Pass/Fail |
| 5.1 | RECORD this information as OriginalSplitOptions. | | | |
| 6 | ASSIGN TestSplitTime EQUALS OriginalSplitOptions XOR 1 | | | |
| 8 | ASSIGN splitOptions.Selected_Split.Selected_Phase EQUALS TestSplitOptions. | | | |
| 9 | SET the following objects: splitOptions.Selected_Split.Selected_Phase. | | Pass/Fail | |
| 10 | GET the follow | ing objects: splitOptions.Select | ed_Split.Selected_Phase. | Pass/Fail |
| 11 | VERIFY splitOptions.Selected_Split.Selected_Phase IS_EQUAL_TO TestSplitOptions. | | Pass/Fail | |
| 12 | ASSIGN splitOptions.Selected_Split.Selected_Phase EQUALS OriginalSplitOptions. | | | |
| 13 | SET the followi | ng objects: splitOptions.Select | ed_Split.Selected_Phase. | Pass/Fail |
| 14 | GET the follow | ing objects: splitOptions.Select | ed_Split.Selected_Phase. | Pass/Fail |
| 15 | VERIFY splitOptions.Selected_Split.Selected_Phase IS_EQUAL_TO OriginalSplitOptions. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.9 Configure Pattern Synchronization Time

| Test Procedure: | Configure Pattern Synchro | Configure Pattern Synchronization Time | |
|---------------------|---|---|--|
| Description: | This test case verifies that station to configure the tim when the master cycle beg | This test case verifies that the ASC allows a management station to configure the time pattern synchronization time for when the master cycle begins. | |
| Requirement(s): | • 3.5.2.1.3.9 Configur | 3.5.2.1.3.9 Configure Pattern Synchronization Time | |
| Variable(s): | OriginalPatternSync TestPatternSync | timebaseAscPatternSync timebaseAscPatternSync | |
| Pass/Fail Criteria: | The device under test shall pass every verification step in this test case to pass the test case. | | |

| Test Step Number | Test Procedure | | Results | |
|------------------------|---|--------------------------|-----------|--|
| 1 | GET the following objects: timebaseAscPatter | rnSync. | Pass/Fail | |
| 1.1 | RECORD this information as Original | PatternSync. | | |
| 2 | ASSIGN TestPatternSync EQUALS RANDOM | И (0 ТО 65535). | | |
| 2.1 | IF TestPatternSync IS_EQUAL_TO C | DriginalPatternSync. | | |
| 2.1.1 | GOTO step 2. | | | |
| 3 | ASSIGN timebaseAscPatternSync EQUALS | TestPatternSync. | | |
| 4 | SET the following objects: timebaseAscPatter | nSync. | Pass/Fail | |
| 5 | GET the following objects: timebaseAscPatternSync. | | Pass/Fail | |
| 6 | VERIFY timebaseAscPatternSync IS_EQUAL | Pass/Fail | | |
| 7 | ASSIGN timebaseAscPatternSync EQUALS OriginalPatternSync. | | | |
| 8 | SET the following objects: timebaseAscPatter | nSync. | Pass/Fail | |
| 9 | GET the following objects: timebaseAscPatte | rnSync. | Pass/Fail | |
| 10 | VERIFY timebaseAscPatternSync IS_EQUAL | _TO OriginalPatternSync. | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure | lotes: | | | |

C.3.4.10 Configure Pattern to Run Free

| Test Procedure: | | Configure Pattern to Run Free | |
|---------------------|---|---|------------|
| Description: | | This test case verifies that an ASC allows a management station to configure a pattern for the Free mode when it is called. | |
| Requirement(| s): | • 3.5.2.1.4.1.1 Configure Pattern to Run Free | |
| Variable(s): | | MaxRowsmaxPatternsTable_RowIntOriginalCycleTimepatternCycleTimeTestCycleTimepatternCycleTime | |
| Pass/Fail Criteria: | | The device under test shall pass every verification sto test case to pass the test case. | ∋p in this |
| Test Step Number | Test Procedur | e | Results |
| 1 | GET the follow | ing objects: maxPatterns. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 3 | GET the follow | ing objects: patternCycleTime.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as OriginalCycleTime. | | |
| 4 | ASSIGN TestCycleTime EQUALS 0. | | |
| 5 | ASSIGN patternCycleTime.Table_Row EQUALS TestCycleTime. | | |
| 6 | SET the follow | ng objects: patternCycleTime.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: patternCycleTime.Table_Row. | Pass/Fail |
| 8 | VERIFY patternCycleTime.Table_Row IS_EQUAL_TO TestCycleTime. Pass/Fai | | |
|------------------------|---|-----------------|-----------|
| 9 | ASSIGN patternCycleTime.Table_Row EQUALS OriginalCycleTime. | | |
| 10 | SET the following objects: patternCycleTime.Table_Row. | | |
| 11 | GET the following objects: patternCycleTime.Table_Row. | | Pass/Fail |
| 12 | VERIFY patternCycleTime.Table_Row IS_EQUAL_TO OriginalCycleTime. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: Date Tested | | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | | |

C.3.4.11 Configure Pattern to Run Automatic Flash

| Test Procedure: | | Configure Pattern to Run Automatic Flash | | |
|---------------------|---|---|--------------------------|--|
| Description: | | This test case verifies that an ASC allows a managem to configure a pattern for Automatic Flash to be active is called. | ent station e when it | |
| Requirement(s): | | • 3.5.2.1.4.1.2 Configure Pattern to Run Automa | tic Flash | |
| Variable(s): | | MaxRowsmaxPatternsTable_RowIntOriginalSequencepatternSequenceNumberTestSequencepatternSequenceNumber | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | ep in this | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | GET the following objects: maxPatterns. | | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: patternSequenceNumber.Table_Row. | | Pass/Fail | |
| 3.1 | RECO | RECORD this information as OriginalSequence. | | |
| 4 | ASSIGN TestS | ASSIGN TestSequence EQUALS 0. | | |
| 5 | ASSIGN patter | nSequenceNumber.Table_Row EQUALS TestSequence. | | |
| 6 | SET the follow | ing objects: patternSequenceNumber.Table_Row. | Pass/Fail | |
| 7 | GET the follow | ing objects: patternSequenceNumber.Table_Row. | Pass/Fail | |
| 8 | VERIFY patter TestSequence | VERIFY patternSequenceNumber.Table_Row IS_EQUAL_TO TestSequence. | | |
| 9 | ASSIGN patter OriginalSequer | ASSIGN patternSequenceNumber.Table_Row EQUALS OriginalSequence. | | |
| 10 | SET the follow | SET the following objects: patternSequenceNumber.Table_Row. Pass/Fai | | |
| 11 | GET the follow | ing objects: patternSequenceNumber.Table_Row. | Pass/Fail | |
| 12 | VERIFY patter OriginalSequer | nSequenceNumber.Table_Row IS_EQUAL_TO | Pass/Fail | |
| Test Procedu | re Results | | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.4.12 Configure Pattern Cycle Time for Coordination

| Test Procedure: | | Configure Pattern Cycle Time for Coordination | | |
|---------------------|---|--|--|-----------------------|
| Description: | | This test case verifies that a to configure the length of the pattern. | an ASC allows a managem ne cycle for a coordinated | ent station timing |
| Requirement | (s): | • 3.5.2.1.4.1.3 Configu Coordination | ure Pattern Cycle Time for | |
| Variable(s): | | MaxRows Table_Row OriginalCycleTime TestCycleTime | maxPatterns Int patternCycleTime patternCycleTime | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the following objects: maxPatterns. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the follow | GET the following objects: patternCycleTime.Table_Row. | | Pass/Fail |
| 3.1 | RECORD this information as OriginalCycleTime. | | | |
| 4 | ASSIGN TestCycleTime EQUALS RANDOM (1 TO 999). | | | |
| 4.1 | IF TestCycleTime IS_EQUAL_TO OriginalCycleTime. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN patter | nCycleTime.Table_Row EQUA | LS TestCycleTime. | |
| 6 | SET the follow | ing objects: patternCycleTime. ⁻ | Table_Row. | Pass/Fail |
| 7 | GET the follow | ring objects: patternCycleTime. | Table_Row. | Pass/Fail |
| 8 | VERIFY patter | nCycleTime.Table_Row IS_EQ | UAL_TO TestCycleTime. | Pass/Fail |
| 9 | ASSIGN patter | ASSIGN patternCycleTime.Table_Row EQUALS OriginalCycleTime. | | |
| 10 | SET the following objects: patternCycleTime.Table_Row. Pass/F | | Pass/Fail | |
| 11 | GET the following objects: patternCycleTime.Table_Row. Pass/F | | Pass/Fail | |
| 12 | VERIFY patternCycleTime.Table_Row IS_EQUAL_TO OriginalCycleTime. | | Pass/Fail | |
| Test Procedu | re Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.4.13 Configure Pattern Offset Time

Configure Pattern Offset Time

| Test Procedure: | | | | |
|---------------------|--|--|--|------------------|
| Description: | | This test case verifies that t station to configure the patt cycle and the master cycle. | he ASC allows a managen tern offset time between th | nent le local |
| Requirement(| s): | • 3.5.2.1.4.1.4 Configu | re Pattern Offset Time | |
| Variable(s): | | MaxRows Table_Row OriginalOffsetTime TestOffsetTime | maxPatterns Int patternOffsetTime patternOffsetTime | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxPatterns. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | /S. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the follow | ing objects: patternOffsetTime. | Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as OriginalOffsetTime. | | | |
| 4 | ASSIGN TestOffsetTime EQUALS RANDOM (0 TO 998). | | | |
| 4.1 | IF TestOffsetTime IS_EQUAL_TO OriginalOffsetTime. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN patternOffsetTime.Table_Row EQUALS TestOffsetTime. | | | |
| 6 | SET the followi | ng objects: patternOffsetTime. | Table_Row. | Pass/Fail |
| 7 | GET the following objects: patternOffsetTime.Table_Row. Pas | | Pass/Fail | |
| 8 | VERIFY patternOffsetTime.Table_Row IS_EQUAL_TO TestOffsetTime. Pass/Fail | | Pass/Fail | |
| 9 | ASSIGN patternOffsetTime.Table_Row EQUALS OriginalOffsetTime. | | | |
| 10 | SET the following objects: patternOffsetTime.Table_Row. Pass/Fail | | Pass/Fail | |
| 11 | GET the following objects: patternOffsetTime.Table_Row. | | Pass/Fail | |
| 12 | VERIFY patternOffsetTime.Table_Row IS_EQUAL_TO OriginalOffsetTime. | | Pass/Fail | |
| Test Procedure | e Results | | | - |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.14 Configure Pattern Split Association

| Test Procedure: | Configure Pattern Split Association |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to configure the split group to be used by a pattern. |
| Requirement(s): | • 3.5.2.1.4.1.5 Configure Pattern Split Association |

| Variable(s): | | MaxRows Table_Row OriginalSplit | maxPatterns Int patternSplitNumber | |
|---------------------|--|---|--|------------|
| | | TestSplit MaxSplits | patternSplitNumber maxSplits | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxPatterns. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | GET the follow | ing objects: maxSplits. | | Pass/Fail |
| 2.1 | RECORD this information as MaxSplits. | | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 4 | GET the following objects: patternSplitNumber.Table_Row. | | Pass/Fail | |
| 4.1 | RECORD this information as OriginalSplit. | | | |
| 5 | ASSIGN TestS | plit EQUALS RANDOM (1 TO | MaxSplits). | |
| 5.1 | IF Test | Split IS_EQUAL_TO OriginalS | plit. | |
| 5.1.1 | GOTO step 5. | | | |
| 6 | ASSIGN patter | nSplitNumber.Table_Row EQL | JALS TestSplit. | |
| 7 | SET the followi | ng objects: patternSplitNumber | r.Table_Row. | Pass/Fail |
| 8 | GET the following objects: patternSplitNumber.Table_Row. | | Pass/Fail | |
| 9 | VERIFY patternSplitNumber.Table_Row IS_EQUAL_TO TestSplit. | | Pass/Fail | |
| 10 | ASSIGN patter | nSplitNumber.Table_Row EQL | JALS OriginalSplit. | |
| 11 | SET the following objects: patternSplitNumber.Table_Row. | | Pass/Fail | |
| 12 | GET the following objects: patternSplitNumber.Table_Row. | | Pass/Fail | |
| 13 | VERIFY patternSplitNumber.Table_Row IS_EQUAL_TO OriginalSplit. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.15 Configure Pattern Sequence Association

| Test Procedure: | Configure Pattern Sequen | Configure Pattern Sequence Association | | |
|--------------------|--|---|--|--|
| Description: | This test case verifies that station to configure the se | This test case verifies that the ASC allows a management station to configure the sequence associated with the pattern. | | |
| Requirement(s): | • 3.5.2.1.4.1.6 Config | • 3.5.2.1.4.1.6 Configure Pattern Sequence Association | | |
| | MaxRows | maxPatterns | | |
| Variable(s): | I able_KOW | INT nottornSoquenceNumber | | |
| | TestSequence | patternSequenceNumber | | |

| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
|-----------------------|---|---|-------------------------------------|------------|
| Test Step Number | Test Procedur | - .e | | Results |
| 1 | GET the follow | ing objects: maxPatterns. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: patternSequenceN | lumber.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Sequence. | |
| 4 | ASSIGN TestS | equence EQUALS RANDOM (| (1 TO 255). | |
| 4.1 | IF Test | tSequence IS_EQUAL_TO Orig | ginalSequence. | |
| 4.1.1 | | GOTO step 5. | | |
| 5 | ASSIGN patter | ASSIGN patternSequenceNumber.Table_Row EQUALS TestSequence. | | |
| 6 | SET the follow | SET the following objects: patternSequenceNumber.Table_Row. | | |
| 7 | GET the follow | ing objects: patternSequenceN | lumber.Table_Row. | Pass/Fail |
| 8 | VERIFY patter TestSequence | VERIFY patternSequenceNumber.Table_Row IS_EQUAL_TO TestSequence. | | |
| 9 | ASSIGN patter OriginalSequer | ASSIGN patternSequenceNumber.Table_Row EQUALS OriginalSequence. | | |
| 10 | SET the follow | SET the following objects: patternSequenceNumber.Table Row. | | Pass/Fail |
| 11 | GET the following objects: patternSequenceNumber.Table_Row. | | Pass/Fail | |
| 12 | VERIFY patternSequenceNumber.Table_Row IS_EQUAL_TO OriginalSequence. | | Pass/Fail | |
| Test Procedu | re Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.4.16 Configure Pattern Maximum Mode

| Test Procedure: Configure Pattern Maximum Mode | | | |
|---|----------------|--|-----------|
| Description: | | This test case verifies that the ASC allows a management station to set a maximum mode for a pattern overriding a default setting. | |
| Requirement(| s): | • 3.5.2.1.4.1.7 Configure Pattern Maximum Mode | |
| Variable(s): | | MaxRowsmaxPatternsTable_RowIntOriginalModepatternMaximumModeTestModepatternMaximumMode | de de |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | p in this |
| Test Step Number | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPatterns. | Pass/Fail |

| 1.1 | RECORD this information as MaxRows. | | |
|------------------------|--|-----------------|-----------|
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: patternMaximumM | ode.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | Mode. | |
| 4 | ASSIGN TestMode EQUALS RANDOM (2 TC | D 6). | |
| 4.1 | IF TestMode IS_EQUAL_TO Original | Mode. | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN patternMaximumMode.Table_Row E | QUALS TestMode. | |
| 6 | SET the following objects: patternMaximumMode.Table_Row. Pa | | |
| 7 | GET the following objects: patternMaximumMode.Table_Row. Pass | | |
| 8 | VERIFY patternMaximumMode.Table_Row IS_EQUAL_TO TestMode. | | |
| 9 | ASSIGN patternMaximumMode.Table_Row EQUALS OriginalMode. | | |
| 10 | SET the following objects: patternMaximumMode.Table_Row. | | |
| 11 | GET the following objects: patternMaximumM | ode.Table_Row. | Pass/Fail |
| 12 | VERIFY patternMaximumMode.Table_Row IS_EQUAL_TO OriginalMode. | | |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | | |

C.3.4.17 Configure Pattern Phase Set

| Test Procedure: | | Configure Pattern Phase Set | |
|---------------------|--|--|--------------------|
| Description: | | This test case verifies that the ASC allows a managen station to configure the phase set associated with the | nent e pattern. |
| Requirement(s): | | • 3.5.2.1.4.1.8 Configure Pattern Phase Set | |
| Variable(s): | | MaxRowsmaxPatternsTable_RowIntOriginalSetpatternPhaseSetTestSetpatternPhaseSetMaxPhaseSetsInt | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | e. | Results |
| 1 | PRE-CONDITION 'The user shall know the maximum number of phase sets the ASC supports'. | | |
| 1.1 | RECORD this information as MaxPhaseSets. | | |
| 2 | GET the following objects: maxPatterns. Pass/I | | Pass/Fail |
| 2.1 | RECORD this information as MaxRows. | | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: patternPhaseSet.Table_Row. | Pass/Fail |

| 4.1 | RECORD this information as Original | Set. | |
|----------------|--|-----------------|-----------|
| 5 | ASSIGN TestSet EQUALS RANDOM (1 TO M | /laxPhaseSets). | |
| 5.1 | IF TestSet IS_EQUAL_TO OriginalSe | et. | |
| 5.1.1 | GOTO step 5. | | |
| 6 | ASSIGN patternPhaseSet.Table_Row EQUA | LS TestSet. | |
| 7 | SET the following objects: patternPhaseSet.T | able_Row. | Pass/Fail |
| 8 | GET the following objects: patternPhaseSet.Table_Row. | | |
| 9 | VERIFY patternPhaseSet.Table_Row IS_EQ | UAL_TO TestSet. | Pass/Fail |
| 10 | ASSIGN patternPhaseSet.Table_Row EQUA | LS OriginalSet. | |
| 11 | SET the following objects: patternPhaseSet.T | able_Row. | Pass/Fail |
| 12 | GET the following objects: patternPhaseSet.Table_Row. Pa | | |
| 13 | VERIFY patternPhaseSet.Table_Row IS_EQUAL_TO OriginalSet. Pass/F | | |
| Test Procedur | e Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.4.18 Configure Pattern Overlap Set

| Test Procedure: | | Configure Pattern Overlap Set | |
|---|---|--|---------------------|
| Description: This test case verifies that the ASC allows a man station to configure the overlap set associated v | | This test case verifies that the ASC allows a managen station to configure the overlap set associated with the station to configure the overlap set associated with the station to configure the sta | nent ne pattern. |
| Requirement(| s): | • 3.5.2.1.4.1.9 Configure Pattern Overlap Set | |
| Variable(s): | | MaxRowsmaxPatternsTable_RowIntOriginalSetpatternOverlapSetTestSetpatternOverlapSetMaxOverlapSetsInt | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITION 'The user shall know the maximum number of overlap sets the ASC supports'. | | |
| 1.1 | RECO | RD this information as MaxOverlapSets. | |
| 2 | GET the follow | ing objects: maxPatterns. | Pass/Fail |
| 2.1 | RECO | RD this information as MaxRows. | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 4 | GET the follow | ing objects: patternOverlapSet.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalSet. | |
| 5 | ASSIGN TestS | et EQUALS RANDOM (1 TO MaxOverlapSets). | |
| 5.1 | IF Tes | tSet IS_EQUAL_TO OriginalSet. | |

| 5.1.1 | GOTO step 5. | | |
|----------------|---|-----------------|-----------|
| 6 | ASSIGN patternOverlapSet.Table_Row EQUALS TestSet. | | |
| 7 | SET the following objects: patternOverlapSet | .Table_Row. | Pass/Fail |
| 8 | GET the following objects: patternOverlapSet | .Table_Row. | Pass/Fail |
| 9 | VERIFY patternOverlapSet.Table_Row IS_EQUAL_TO TestSet. | | |
| 10 | ASSIGN patternOverlapSet.Table_Row EQUALS OriginalSet. | | |
| 11 | SET the following objects: patternOverlapSet.Table_Row. | | |
| 12 | GET the following objects: patternOverlapSet.Table_Row. Pass/Fail | | |
| 13 | VERIFY patternOverlapSet.Table_Row IS_EQUAL_TO OriginalSet. | | |
| Test Procedu | re Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | - | |

C.3.4.19 Configure Pattern Vehicle Detector Set

| Test Procedure: | | Configure Pattern Vehicle Detector Set | |
|---|--|--|---------------------------------------|
| Description: | | This test case verifies that the ASC allows a management station to configure the vehicle detector set associated with pattern. | |
| Requirement(| s): | • 3.5.2.1.4.1.10 Configure Pattern Vehicle Detect | tor Set |
| Variable(s): | | MaxRowsmaxPatternsTable_RowIntOriginalSetpatternVehicleDetecTestSetpatternVehicleDetecMaxVehicleDetectorSetsInt | torSet torSet |
| Pass/Fail Crite | eria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| | | | |
| Test Step Number | Test Procedur | re | Results |
| Test Step Number 1 | Test Procedur PRE-CONDITI detector sets th | 'e ON 'The user shall know the maximum number of vehicle ne ASC supports'. | Results |
| Test Step Number 1 1.1 | Test Procedur PRE-CONDITI detector sets th RECO | 'e ON 'The user shall know the maximum number of vehicle ne ASC supports'. RD this information as MaxVehicleDetectorSets. | Results |
| Test Step Number 1 1.1 2 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow | 'e ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. | Results Pass/Fail |
| Test Step Number111.122.1 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow RECO | 'e ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. RD this information as MaxRows. | Results Pass/Fail |
| Test Step Number 1 1.1 2 2.1 3 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow RECO ASSIGN Table | re ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). | Results Pass/Fail |
| Test Step Number 1 1.1 2 2.1 3 4 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow RECO ASSIGN Table GET the follow | re ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. RD this information as MaxRows. RO this information as MaxRows. Row EQUALS RANDOM (1 TO MaxRows). ing objects: patternVehicleDetectorSet.Table_Row. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 2.1 3 4 4.1 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow RECO ASSIGN Table GET the follow RECO | re ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. RD this information as MaxRows. Row EQUALS RANDOM (1 TO MaxRows). ing objects: patternVehicleDetectorSet.Table_Row. RD this information as OriginalSet. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 2.1 3 4 4.1 5 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestS | re ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: patternVehicleDetectorSet.Table_Row. RD this information as OriginalSet. set EQUALS RANDOM (1 TO MaxVehicleDetectorSets). | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 2.1 3 4 4.1 5 5.1 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestS IF Test | re ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. RD this information as MaxRows. ROW EQUALS RANDOM (1 TO MaxRows). ing objects: patternVehicleDetectorSet.Table_Row. RD this information as OriginalSet. set EQUALS RANDOM (1 TO MaxVehicleDetectorSets). tset IS_EQUAL_TO OriginalSet. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 2.1 3 4 4.1 5 5.1 5.1.1 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestS IF Test | re ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. RD this information as MaxRows. Row EQUALS RANDOM (1 TO MaxRows). ing objects: patternVehicleDetectorSet.Table_Row. RD this information as OriginalSet. Set EQUALS RANDOM (1 TO MaxVehicleDetectorSets). tSet IS_EQUAL_TO OriginalSet. GOTO step 5. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 2.1 3 4 4.1 5 5.1 5.1.1 6 | Test Procedur PRE-CONDITI detector sets th RECO GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestS IF Test ASSIGN patter | re ON 'The user shall know the maximum number of vehicle he ASC supports'. RD this information as MaxVehicleDetectorSets. ing objects: maxPatterns. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: patternVehicleDetectorSet.Table_Row. RD this information as OriginalSet. Set EQUALS RANDOM (1 TO MaxVehicleDetectorSets). tSet IS_EQUAL_TO OriginalSet. GOTO step 5. nVehicleDetectorSet.Table_Row EQUALS TestSet. | Results Pass/Fail Pass/Fail Pass/Fail |

| 8 | GET the following objects: patternVehicleDete | ectorSet.Table_Row. | Pass/Fail |
|----------------|--|-----------------------|-----------|
| 9 | VERIFY patternVehicleDetectorSet.Table_Row IS_EQUAL_TO TestSet. | | |
| 10 | ASSIGN patternVehicleDetectorSet.Table_Rc | w EQUALS OriginalSet. | |
| 11 | SET the following objects: patternVehicleDetectorSet.Table_Row. | | |
| 12 | GET the following objects: patternVehicleDetectorSet.Table_Row. | | Pass/Fail |
| 13 | VERIFY patternVehicleDetectorSet.Table_Row IS_EQUAL_TO OriginalSet. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.4.20 Configure Pattern Pedestrian Detector Set

| Test Procedure: | | Configure Pattern Pedestrian Detector Set | | |
|---------------------|---|---|---|------------------------|
| Description: | | This test case verifies that the As station to configure the pedestria the pattern. | SC allows a managen an detector set assoc | nent iated with |
| Requirement | (s): | • 3.5.2.1.4.1.11 Configure P | attern Pedestrian De | tector Set |
| Variable(s): | | MaxRows Table_Row OriginalSet TestSet MaxPedestrianDetectorSets | maxPatterns Int patternPedestrianDe patternPedestrianDe Int | tectorSet tectorSet |
| Pass/Fail Crit | eria: | The device under test shall pass test case to pass the test case. | every verification ste | ep in this |
| Test Step Number | Test Procedu | re | | Results |
| 1 | PRE-CONDITI pedestrian det | ON 'The user shall know the maximu ector sets the ASC supports'. | m number of | |
| 1.1 | RECO | RD this information as MaxPedestria | nDetectorSets. | |
| 2 | GET the follow | ing objects: maxPatterns. | | Pass/Fail |
| 2.1 | RECO | RD this information as MaxRows. | | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 TO Ma | xRows). | |
| 4 | GET the follow | ing objects: patternPedestrianDetect | orSet.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalSet. | | |
| 5 | ASSIGN TestSet EQUALS RANDOM (1 TO MaxPedestrianDetectorSets). | | | |
| 5.1 | IF Tes | tSet IS_EQUAL_TO OriginalSet. | | |
| 5.1.1 | | GOTO step 5. | | |
| 6 | ASSIGN patter | mPedestrianDetectorSet.Table_Row | EQUALS TestSet. | |
| 7 | SET the follow | ing objects: patternPedestrianDetecte | orSet.Table_Row. | Pass/Fail |
| 8 | GET the follow | ring objects: patternPedestrianDetect | orSet.Table_Row. | Pass/Fail |

| 9 | VERIFY patternPedestrianDetectorSet.Table_ TestSet. | _Row IS_EQUAL_TO | Pass/Fail |
|----------------|---|------------------|-----------|
| 10 | ASSIGN patternPedestrianDetectorSet.Table OriginalSet. | _Row EQUALS | |
| 11 | SET the following objects: patternPedestrianDetectorSet.Table_Row. Pass/F | | Pass/Fail |
| 12 | GET the following objects: patternPedestrianDetectorSet.Table_Row. | | Pass/Fail |
| 13 | VERIFY patternPedestrianDetectorSet.Table_Row IS_EQUAL_TO OriginalSet. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.4.21 Configure Pattern Special Functions

| Test Procedure: | | Configure Pattern Special Functions | | |
|------------------------------------|--|--|--|------------------|
| Description: | | This test case verifies that the ASC allows a management station to configure special function outputs to be active when a pattern is active. | | ient ive when |
| Requirement(| s): | 3.5.2.1.4.1.12 Configure Pattern Special Functions | | ons |
| Variable(s): | | MaxRows Table_Row OriginalFunctions TestFunctions MaxSpecialFunctions MaxValue | maxPatterns Int patternSpecialFunct patternSpecialFunct Int Int | ions ions |
| Pass/Fail Criteria: | | The device under test shall pass test case to pass the test case. | every verification ste | p in this |
| Test Step Number | Test Procedu | re | | Results |
| 1 | PRE-CONDITI functions the A | ON 'The user shall know the maximu SC supports'. | um number of special | |
| 1.1 | RECO | RD this information as MaxSpecialF | unctions. | |
| 2 | ASSIGN MaxV | alue EQUALS (2 ^ MaxSpecialFunc | tions – 1) | |
| 3 | GET the follow | ing objects: maxPatterns. | | Pass/Fail |
| 3.1 | RECO | RD this information as MaxRows. | | |
| 4 | | | | |
| | ASSIGN Table | _Row EQUALS RANDOM (1 TO Ma | xRows). | |
| 5 | ASSIGN Table GET the follow | _Row EQUALS RANDOM (1 TO Ma ing objects: patternSpecialFunctions | xRows). .Table_Row. | Pass/Fail |
| 5 5.1 | ASSIGN Table GET the follow RECO | _Row EQUALS RANDOM (1 TO Ma ing objects: patternSpecialFunctions RD this information as OriginalFunct | xRows). .Table_Row. ions. | Pass/Fail |
| 5 5.1 6 | ASSIGN Table GET the follow RECO If MaxValue IS | _Row EQUALS RANDOM (1 TO Ma ing objects: patternSpecialFunctions RD this information as OriginalFunct _GREATER_THAN 4294967295 | xRows). .Table_Row. ions. | Pass/Fail |
| 5 5.1 6 6.1 | ASSIGN Table GET the follow RECO If MaxValue IS ASSIG | _Row EQUALS RANDOM (1 TO Ma ing objects: patternSpecialFunctions RD this information as OriginalFunct _GREATER_THAN 4294967295 iN TestFunctions EQUALS RANDOM | xRows). .Table_Row. ions. / (1 TO 4294967295) | Pass/Fail |
| 5 5.1 6 6.1 6.1.1 | ASSIGN Table GET the follow RECO If MaxValue IS ASSIG | Row EQUALS RANDOM (1 TO Ma ing objects: patternSpecialFunctions RD this information as OriginalFunct GREATER_THAN 4294967295 N TestFunctions EQUALS RANDOM GOTO step 8. | xRows). .Table_Row. ions. / (1 TO 4294967295) | Pass/Fail |
| 5 5.1 6 6.1 6.1.1 7 | ASSIGN Table GET the follow RECO If MaxValue IS ASSIGN ASSIGN TestF | _Row EQUALS RANDOM (1 TO Ma ing objects: patternSpecialFunctions RD this information as OriginalFunct _GREATER_THAN 4294967295 IN TestFunctions EQUALS RANDOM GOTO step 8. functions EQUALS RANDOM (1 TO | xRows). .Table_Row. ions. // (1 TO 4294967295) MaxValue) | Pass/Fail |

| 9 | SET the following objects: patternSpecialFund | ctions.Table_Row. | Pass/Fail |
|----------------|--|-------------------|-----------|
| 10 | GET the following objects: patternSpecialFund | ctions.Table_Row. | Pass/Fail |
| 11 | VERIFY patternSpecialFunctions.Table_Row IS_EQUAL_TO TestFunctions. | | |
| 12 | ASSIGN patternSpecialFunctions.Table_Row EQUALS OriginalFunctions. | | |
| 13 | SET the following objects: patternSpecialFunctions.Table_Row. | | Pass/Fail |
| 14 | GET the following objects: patternSpecialFunctions.Table_Row. | | Pass/Fail |
| 16 | VERIFY patternSpecialFunctions.Table_Row IS_EQUAL_TO OriginalFunctions. | | Pass/Fail |
| Test Procedur | e Results | | _ |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.4.22 Determine Maximum Number of Timing Patterns

| Test Procedure: | Determine Maximum Number Timing Patterns | | | |
|---------------------|---|---|-----------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of patterns that can be configured in the ASC and verifies that the ASC supports the minimum number of patterns required by the user. | | |
| Requirement(| • 3.5.2.1.4.1.13 Determine Maximum Number of Timing Patterns | | Гiming | |
| Variable(s): | | UserMinPatterns maxPatterns | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedur | e | | Results |
| 1 | USER-ACTION in FR ID 3.5.2. | I 'Determine the number of pat 1.4.1.13 of the PRL.' | terns required as specified | |
| 1.1 | RECO | RD this information as UserMir | Patterns. | |
| 2 | GET the follow | ing objects: maxPatterns. | | Pass/Fail |
| 3 | VERIFY maxPa | atterns IS_NOT_LESS_THAN | UserMinPatterns. | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.23 Configure Phase Split Time

| Test Procedure: | Configure Split Time |
|--------------------|----------------------|
| Test Procedure: | Configure Split Time |

| Description: This test case verifies that the ASC allows a station to configure the split times. | | he ASC allows a managen t times. | nent | |
|--|---|--|--|------------|
| Requirement(s | s): | • 3.5.2.1.5.1.1 Configure Phase Split Time | | |
| Variable(s): | | OriginalSplitTime TestSplitTime MaxSplits MaxPhases Selected_Phase Selected_Split | splitTime splitTime maxSplits maxPhases Int Int | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | - .e | | Results |
| 1 | GET the follow | ing objects: maxSplits. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxSpli | ts. | |
| 2 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 2.1 | RECORD this information as MaxPhases. | | | |
| 3 | ASSIGN Selected_Split EQUALS RANDOM (1 TO MaxSplits). | | | |
| 4 | ASSIGN Selected_Phase EQUALS RANDOM (1 TO MaxPhases). | | | |
| 5 | GET the following objects: splitTime.Selected_Split.Selected_Phase. | | Pass/Fail | |
| 5.1 | RECORD this information as OriginalSplitTime. | | | |
| 6 | ASSIGN TestSplitTime EQUALS RANDOM (0 TO 999). | | | |
| 6.1 | IF Test | SplitTime IS_EQUAL_TO Orig | inalSplitTime. | |
| 6.1.1 | | GOTO step 6. | | |
| 7 | ASSIGN Selec | ted_Split.Selected_Phase EQL | IALS TestSplitTime. | |
| 8 | SET the followi | ng objects: splitTime.Selected_ | Split.Selected_Phase. | Pass/Fail |
| 9 | GET the follow | ing objects: splitTime.Selected | Split.Selected_Phase. | Pass/Fail |
| 10 | VERIFY splitTin TestSplitTime. | me.Selected_Split.Selected_Pt | nase IS_EQUAL_TO | Pass/Fail |
| 11 | ASSIGN splitTi OriginalSplitTin | me.Selected_Split.Selected_Pl ne. | hase EQUALS | |
| 12 | SET the followi | ng objects: splitTime.Selected | Split.Selected_Phase. | Pass/Fail |
| 13 | GET the follow | ing objects: splitTime.Selected | Split.Selected_Phase. | Pass/Fail |
| 14 | VERIFY splitTin OriginalSplitTin | splitTime.Selected_Split.Selected_Phase IS_EQUAL_TO Pa | | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.24 Configure Phase Split Mode

| Test Procedure: | Configure Phase Split Mode |
|--------------------|----------------------------|
| | |

| Description: | | This test case verifies that the ASC allows a management station to configure the recall modes specific to the split group. | | |
|---------------------|--|---|--|------------|
| Requirement(s): | | • 3.5.2.1.5.1.2 Configure Phase Split Mode | | |
| Variable(s): | | OriginalSplitMode TestSplitMode MaxSplits MaxPhases Selected_Split Selected_Phase | splitMode splitMode maxSplits maxPhases Int Int | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification sta ase. | ep in this |
| Test Step Number | Test Procedur | - .e | | Results |
| 1 | GET the follow | ing objects: maxSplits. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxSpli | ts. | |
| 2 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 2.1 | RECO | RD this information as MaxPha | ses. | |
| 3 | ASSIGN Selected_Split EQUALS RANDOM (1 TO MaxSplits). | | | |
| 4 | ASSIGN Selected_Phase EQUALS RANDOM (1 TO MaxPhases). | | | |
| 5 | GET the follow | ing objects: splitMode.Selected | Split.Selected_Phase. | Pass/Fail |
| 5.1 | RECO | RD this information as Original | SplitMode. | |
| 6 | ASSIGN TestS | plitTime EQUALS RANDOM (2 | 2 TO 8). | |
| 6.1 | IF TestSplitTime IS_EQUAL_TO OriginalSplitTime. | | | |
| 6.1.1 | GOTO step 6. | | | |
| 7 | ASSIGN splitMode.Table_Row EQUALS TestSplitMode. | | | |
| 8 | SET the followi | ng objects: splitMode.Selected | _Split.Selected_Phase. | Pass/Fail |
| 9 | GET the follow | ing objects: splitMode.Selected | _Split.Selected_Phase. | Pass/Fail |
| 10 | VERIFY splitMode.Selected_Split.Selected_Phase IS_EQUAL_TO TestSplitMode. | | | Pass/Fail |
| 11 | ASSIGN splitMode.Selected_Split.Selected_Phase EQUALS OriginalSplitMode. | | | |
| 12 | SET the following objects: splitMode.Selected_Split.Selected_Phase. | | | Pass/Fail |
| 13 | GET the follow | ing objects: splitMode.Selected | _Split.Selected_Phase. | Pass/Fail |
| 14 | VERIFY splitMode.Selected_Split.Selected_Phase IS_EQUAL_TO OriginalSplitMode. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.25 Configure Split Coordinated Phase

| Test Procedure: | Configure Split Coordination Phase |
|--------------------|------------------------------------|
| | |

| Description: | | This test case verifies that the ASC allows a management station to toggle a phase as a coordinated phase. | | |
|---------------------|--|--|------------------------------------|------------|
| Requirement(s): | | • 3.5.2.1.5.1.3 Configure Split Coordination Phase | | |
| Variable(s): | | OriginalSplitCoordsplitCoordPhaseTestSplitCoordsplitCoordPhaseMaxSplitsmaxSplitsMaxPhasesmaxPhasesSelected_SplitIntSelected PhaseInt | | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification st ase. | ep in this |
| Test Step Number | Test Procedu | e | | Results |
| 1 | GET the follow | ing objects: maxSplits. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxSpli | ts. | |
| 2 | GET the follow | ing objects: maxPhases. | | Pass/Fail |
| 2.1 | RECO | RD this information as MaxPha | ses. | |
| 3 | ASSIGN Selec | ted_Split EQUALS RANDOM (| 1 TO MaxSplits). | |
| 4 | ASSIGN Selec | ted_Phase EQUALS RANDON | l (1 TO MaxPhases). | |
| 5 | GET the following objects: splitCoordPhase.Selected_Split.Selected_Phase. | | Pass/Fail | |
| 5.1 | RECO | RD this information as Original | SplitCoord. | |
| 6 | IF OriginalSplit | Coord IS_EQUAL_TO 0. | | |
| 6.1 | ASSIG | N TestSplitCoord EQUALS 1. | | |
| 6.2 | GOTO | step 8. | | |
| 7 | ASSIGN TestS | ASSIGN TestSplitCoord EQUALS 0. | | |
| 8 | ASSIGN Selec | ASSIGN Selected_Split.Selected_Phase EQUALS TestSplitCoord. | | |
| 9 | SET the following objects: splitCoordPhase.Selected Split.Selected Phase. | | | Pass/Fail |
| 11 | GET the follow splitCoordPhase | ing objects: se.Selected_Split.Selected_Pha | ase. | Pass/Fail |
| 12 | VERIFY splitCo | oordPhase.Selected_Split.Sele) TestSplitCoord. | cted_Phase | Pass/Fail |
| 13 | ASSIGN splitC OriginalSplitCo | oordPhase.Selected_Split.Sele | cted_Phase EQUALS | |
| 14 | SET the following objects: splitCoordPhase.Selected Split.Selected Phase. | | | Pass/Fail |
| 15 | GET the follow splitCoordPhase | <pre>> following objects: prdPhase.Selected Split.Selected Phase.</pre> Pass/Fail | | |
| 16 | VERIFY splitCo IS_EQUAL_TO | litCoordPhase.Selected_Split.Selected_Phase Pass/Fail | | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

| C.3.4.26 | Determine Maximum | Number | of Phase | Splits |
|----------|-------------------|--------|----------|--------|
|----------|-------------------|--------|----------|--------|

| Test Procedure: | | Determine Maximum Number of Phase Splits | | |
|------------------------|---|--|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of phase splits supported by ASC and verifies that the ASC supports the minimum number of phase split groups required by the user. | | |
| Requirement(| s): | 3.5.2.1.5.2 Determine Maximum Number of Phase Splits | | |
| Variable(s): | | UserMinSplits maxSplits | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedur | re Results | | |
| 1 | USER-ACTION 'Determine the number of splits required as specified in FR ID 3.5.2.1.5.2.1 of the PRL.' | | | |
| 1.1 | RECO | RD this information as UserMir | Splits. | |
| 2 | GET the follow | ing objects: maxSplits. | | Pass/Fail |
| 3 | VERIFY maxSplits IS_NOT_LESS_THAN UserMinSplits. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.27 Configure Timebased Action - Pattern

| Test Procedure: | | Configure Timebased Action - Pattern | | |
|---------------------|--------------|---|--|-----------------|
| Description: | | This test case verifies that the A station to associate a pattern wi Base Action Table. | ASC allows a managem ith an action in the 120 | nent 12 Time |
| Requirement(s | s): | • 3.5.2.1.10.1.1 Configure Timebased Action - Pattern | | |
| Variable(s): | | MaxRows Table_Row MaxPatterns OriginalPattern TestPattern | maxTimebaseAscAc Int Int timebaseAscPattern timebaseAscPattern | tions |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedu | lure Results | | |

| 1 | PRE-CONDITION 'The user shall know the m | aximum number of | | |
|-----------------------|---|------------------|-----------|--|
| 1.1 | RECORD this information as MaxPat | terns. | | |
| 2 | GET the following objects: maxTimebaseAsc/ | Actions. | Pass/Fail | |
| 2.1 | RECORD this information as MaxRov | VS. | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 T | O MaxRows). | | |
| 4 | GET the following objects: timebaseAscPatter | rn. | Pass/Fail | |
| 4.1 | RECORD this information as Original | Pattern. | | |
| 5 | ASSIGN TestPattern EQUALS RANDOM (1 1 | O MaxPatterns). | | |
| 5.1 | IF TestPattern IS_EQUAL_TO Origin | alPattern. | | |
| 5.1.1 | GOTO step 5. | | | |
| 6 | ASSIGN timebaseAscPattern.Table_Row EQUALS TestPattern. | | | |
| 7 | SET the following objects: timebaseAscPattern.Table_Row. | | | |
| 8 | GET the following objects: timebaseAscPattern.Table_Row. | | | |
| 9 | VERIFY timebaseAscPattern.Table_Row IS_EQUAL_TO TestPattern. | | | |
| 10 | ASSIGN timebaseAscPattern.Table_Row EQUALS OriginalPattern. | | | |
| 11 | SET the following objects: timebaseAscPatter | n.Table_Row. | Pass/Fail | |
| 12 | GET the following objects: timebaseAscPattern.Table_Row. | | | |
| 13 | VERIFY timebaseAscPattern.Table_Row IS_EQUAL_TO OriginalPattern. | | | |
| Test Procedur | Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure Notes: | | | | |

C.3.4.28 Configure Timebased Action – Special Functions

| Test Procedure: | | Configure Timebased Action – Special Functions | | |
|---------------------|--|--|-------------------------------|--|
| Description: | | This test case verifies that the ASC allows a management station to configure special function outputs to be active when a timebased action is active. | | |
| Requirement(| s): | 3.5.2.1.10.1.2 Configure Timebased Action – Special Functions | | |
| Variable(s): | MaxRows maxtimebaseAscAction Table_Row Int OriginalFunctions timebaseAscSpecialFunctions TestFunctions timebaseAscSpecialFunctions MaxSpecialFunctions Int MaxSpecialFunctions Int MaxSpecialFunctions Int | | tions Function Function | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | PRE-CONDITION 'The user shall know the maximum number of special functions the ASC supports'. | | | |
| 1.1 | RECORD this information as MaxSpecialFunctions. | | | |

| Test Procedure | Notes: | | | |
|----------------|---|------------------------|-----------|--|
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedu | e Results | | | |
| 16 | VERIFY timebaseAscSpecialFunction.Table_Row IS_EQUAL_TO OriginalFunctions. | | | |
| 14 | GET the following objects: timebaseAscSpecialFunction.Table_Row. | | | |
| 13 | SET the following objects: timebaseAscSpecia | alFunction.Table_Row. | Pass/Fail | |
| 12 | ASSIGN timebaseAscSpecialFunction.Table_Row EQUALS OriginalFunctions. | | | |
| 11 | VERIFY timebaseAscSpecialFunction.Table_Row IS_EQUAL_TO TestFunctions. | | | |
| 10 | GET the following objects: timebaseAscSpecialFunction.Table_Row. | | | |
| 9 | SET the following objects: timebaseAscSpecialFunction.Table_Row. | | | |
| 8 | ASSIGN timebaseAscSpecialFunction.Table_ TestFunctions. | Row EQUALS | | |
| 7 | ASSIGN TestFunctions EQUALS RANDOM (| 1 TO MaxValue) | | |
| 6.1.1 | GOTO step 8. | · · · · · | | |
| 6.1 | ASSIGN TestFunctions EQUALS RAN | NDOM (1 TO 4294967295) | | |
| 6 | If MaxValue IS_GREATER_THAN 429496729 | 95 | | |
| 5.1 | RECORD this information as Original | Functions. | | |
| 5 | GET the following objects: timebaseAscSpecia | alFunction.Table Row. | Pass/Fail | |
| 4 | ASSIGN Table Row EQUALS RANDOM (1 T | O MaxRows). | | |
| 3.1 | RECORD this information as MaxRows. | | | |
| 3 | GET the following objects: maxTimebaseAscA | Actions. | Pass/Fail | |
| 2 | ASSIGN MaxValue EQUALS (2 ^ MaxSpecial | Functions – 1) | | |

C.3.4.29 Determine Maximum Number of Timebased Actions

| Test Procedure: | | Determine Maximum Number of Timebased Actions | | |
|--|---|---|---------|--|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of time-based action events supported by ASC and verifies that the ASC supports the minimum number of action events required by the user. | | |
| Requirement(| s): | 3.5.2.1.10.1.3 Determine Maximum Number of Timebased Actions | | |
| Variable(s): UserMinTimebaseAscActions maxTime | | UserMinTimebaseAscActions maxTimebaseAscAc | tions | |
| Pass/Fail Crit | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | USER-ACTION 'Determine the number of action events required as specified in FR ID 3.5.2.1.10.3 of the PRL.' | | | |
| 1.1 | RECORD this information as UserMinTimebaseAscActions. | | | |

| 2 | GET the following objects: maxTimebaseAscActions. | | | | |
|-------------------------|---|--|-----------|--|--|
| 3 | VERIFY maxTimebaseAscActions IS_NOT_LESS_THAN UserMinTimebaseAscActions. | | | | |
| Test Procedure Results | | | | | |
| Tested By: Date Tested: | | | Pass/Fail | | |
| Test Procedure | lotes: | | | | |

C.3.4.30 Determine Action In Effect

| Test Procedure: | | Determine Action In Effect | | |
|------------------------|----------------|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the current time base. | | |
| Requirement(s): | | • 3.5.2.1.10.1.4 Deterr | nine Action In Effect | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: timebaseAscActior | nStatus. | Pass/Fail |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.4.31 Activate Action Plan Remotely

| Test Procedure: | Activate Action Plan Remote | Activate Action Plan Remotely | | |
|---------------------|--|--|--|--|
| Description: | This test case verifies that the station to activate a configur | This test case verifies that the ASC allows a management station to activate a configured action plan. | | |
| Requirement(s): | • 3.5.2.1.10.1.4 Determ | • 3.5.2.1.10.1.4 Determine Action In Effect | | |
| Variable(s): | OriginalPlanControl actionPlanControl TestPlanControl actionPlanControl | | | |
| Pass/Fail Criteria: | The device under test shall p test case to pass the test case | The device under test shall pass every verification step in this test case to pass the test case. | | |

| Test Step Number | Test Procedure | Results |
|---------------------|---|-----------|
| 1 | GET the following objects: actionPlanControl. | Pass/Fail |
| 1.1 | RECORD this information as OriginalPlanControl. | |
| 2 | ASSIGN TestOriginalControl EQUALS RANDOM (0 TO 255). | |
| 2.1 | IF TestPlanControl IS_EQUAL_TO OriginalPlanControl | |
| 2.1.1 | GOTO step 2 | |
| 3 | ASSIGN actionPlanControl EQUALS TestPlanControl. | |
| 4 | SET the following objects: actionPlanControl. | Pass/Fail |
| 5 | GET the following objects: actionPlanControl. | |
| 6 | VERIFY actionPlanControl IS_EQUAL_TO TestPlanControl. | |
| 7 | ASSIGN actionPlanControl EQUALS OriginalPlanControl. | |
| 8 | SET the following objects: actionPlanControl. | Pass/Fail |
| 9 | GET the following objects: actionPlanControl. | Pass/Fail |
| 10 | VERIFY actionPlanControl IS_EQUAL_TO OriginalPlanControl. | Pass/Fail |
| | Test Procedure Results | |
| Tested By: | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | |

C.3.5 Sequences

C.3.5.1 Configure Sequence Data

| Test Procedure: | | Configure Sequence Data | | |
|---------------------|--|---|---|-----------|
| Description: | | This test case verifies that the ASC allows a management station configure the order of phases within a ring. | | |
| Requirement(| s): | • 3.5.2.1.6.1 Configure Sec | juence Data | |
| Variable(s): | | MaxSequences Selected_Sequence OriginalData TestData MaxRings Selected_Ring MaxPhases | maxSequences Int sequenceData sequenceData maxRings Int maxPhases | |
| Pass/Fail Criteria: | | The device under test shall pass test case to pass the test case. | every verification ste | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | PRE-CONDITION 'A valid timing plan is loaded'. | | | |
| 2 | PRE-CONDITION 'The user shall know the maximum number of phases supported by the ASC'. | | | |
| 2.1 | RECO | RD this information as MaxPhases. | | |
| 3 | GET the follow | ing objects: maxSequences. | | Pass/Fail |

| 3.1 | RECORD this information as MaxSequences. | | |
|------------------|--|---------------------------|-----------|
| 4 | GET the following objects: maxRings. | | Pass/Fail |
| 4.1 | RECORD this information as MaxRing | gs. | |
| 5 | ASSIGN Selected_Sequence EQUALS RANE MaxSequences). | DOM (1 TO | |
| 6 | ASSIGN Selected_Ring EQUALS RANDOM (1 TO MaxRings). | | |
| 7 | GET the following objects: sequenceData.Selected_SequenceData | Ring. | Pass/Fail |
| 7.1 | RECORD this information as Original | Data. | |
| 8 | USER-ACTION 'Select a valid value for sequenceData.Selected_Sequence.Selected_ be greater than MaxPhases.' | Ring. Each octet must not | |
| 9 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 10 | ASSIGN sequenceData.Selected_Sequence.SetData. | Selected_Ring EQUALS | |
| 11 | SET the following objects: sequenceData.Selected_Sequence.Selected_ | _Ring. | Pass/Fail |
| 12 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | Database Status and | |
| 13 | GET the following objects: sequence.Selected_Ring. | | Pass/Fail |
| 14 | VERIFY sequenceData.Selected_Sequence.Selected_Ring IS_EQUAL_TO TestData. | | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 16 | ASSIGN sequenceData.Selected_Sequence.Selected_Ring EQUALS OriginalData. | | |
| 17 | SET the following objects: sequence.Selected Ring. | | Pass/Fail |
| 18 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 19 | GET the following objects: sequence.Selected Ring. | | Pass/Fail |
| 20 | VERIFY sequenceData.Selected_Sequence.Selected_Ring IS_EQUAL_TO OriginalData. | | Pass/Fail |
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.5.2 Determine Maximum Number of Rings

| Test Procedure: | Determine Maximum Number of Rings |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to determine the maximum number of rings supported. |
| Requirement(s): | • 3.5.2.1.6.2 Determine Maximum Number of Rings |

| Variable(s): | | UserMinRings | maxRings | |
|------------------------|--|---|-----------------|-----------|
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | USER-ACTION 'Determine the number of rings required as specified in FR ID 3.5.2.1.6.2 of the PRL.' | | | |
| 1.1 | RECORD this information as UserMinRings. | | | |
| 2 | GET the following objects: maxRings. | | Pass/Fail | |
| 3 | VERIFY maxRings IS_NOT_LESS_THAN UserMinRings. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |

C.3.5.3 Determine Maximum Number of Sequences

| Test Procedure: | | Determine Maximum Number of Sequences | | |
|---------------------|--|--|-------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of sequence plans supported by ASC and verifies that the ASC supports the minimum number of sequence plans required by the user. | | |
| Requirement(| s): | • 3.5.2.1.6.3 Determin | e Maximum Number of Sec | quences |
| Variable(s): | | UserMinSequences | UserMinSequences maxSequences | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | | Results |
| 1 | USER-ACTION specified in FR | I 'Determine the number of sec ID 3.5.2.1.6.2.3 of the PRL.' | uence plans required as | |
| 1.1 | RECO | RD this information as UserMir | Sequences. | |
| 2 | GET the follow | ing objects: maxSequences. | | Pass/Fail |
| 3 | 3 VERIFY maxSequences IS_NOT_LESS_THAN_UserMinSequences. | | N UserMinSequences. | Pass/Fail |
| Test Procedur | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.6 Channels

C.3.6.1 Configure Channel Control Source

| Test Procedure: | | Configure Channel Control Source | |
|---------------------|---|--|-------------|
| Description: | | This test case verifies that the ASC allows a manager station to configure which phase or overlap controls channel. | nent the |
| Requirement(| s): | • 3.5.2.1.7.1.1 Configure Channel Control Sourc | e |
| Variable(s): | | MaxChannelRowsmaxChannelsTable_RowIntOriginalControlSourcechannelControlSourceTestControlSourcechannelControlSourceMaxControlSourceInt | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification sto test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | e | Results |
| 1 | GET the follow | ing objects: maxChannels. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxChannelRows. | |
| 2 | GET the follow | ing objects: channelControlType. | Pass/Fail |
| 3 | IF channelControlType IS_EQUAL_TO 4 OR channelControlType IS_EQUAL_TO 5. | | |
| 3.1 | GET the following objects: maxOverlaps. | | Pass/Fail |
| 3.1.1 | RECORD this information as MaxControlSource. | | |
| 3.2 | GOTO step 5 | | |
| 4 | GET the following objects: maxPhases. | | Pass/Fail |
| 4.1 | RECORD this information as MaxControlSource. | | |
| 5 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxChannelRows). | | |
| 6 | GET the follow | ing objects: channelControlSource.Table_Row. | Pass/Fail |
| 6.1 | RECO | RD this information as OriginalControlSource. | |
| 7 | ASSIGN TestC MaxControlSou | ontrolSource EQUALS RANDOM (0 TO urce). | |
| 7.1 | IF Test | ControlSource IS_EQUAL_TO OriginalControlSource. | |
| 7.1.1 | | GOTO step 7. | |
| 8 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 9 | ASSIGN channelControlSource.Table_Row EQUALS TestControlSource. | | |
| 10 | SET the follow | ng objects: channelControlSource.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: channelControlSource.Table_Row. | Pass/Fail |
| 12 | PERFORM the Verify Databas | Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 13 | VERIFY chann TestControlSo | elControlSource.Table_Row IS_EQUAL_TO urce. | Pass/Fail |
| 14 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | Ī |

| Test Procedure Notes: | | | | |
|-------------------------|--|-----------------|-----------|--|
| Tested By: Date Tested: | | | Pass/Fail | |
| Test Procedure Results | | | | |
| 19 | VERIFY channelControlSource.Table_Row IS_EQUAL_TO OriginalControlSource. | | | |
| 18 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 17 | GET the following objects: channelControlSou | rce.Table_Row. | Pass/Fail | |
| 16 | SET the following objects: channelControlSou | irce.Table_Row. | Pass/Fail | |
| 15 | ASSIGN channelControlSource.Table_Row EQUALS OriginalControlSource. | | | |

C.3.6.2 Configure Channel Control Type

| Test Procedure: | | Configure Channel Control Type - Vehicle Phase | | |
|---------------------|--|--|----------------|--|
| Description: | | This test case verifies that the ASC allows a man station to set channel control type. | agement | |
| Requirement(| s): | • 3.5.2.1.7.1.2 Configure Channel Control T | уре | |
| Variable(s): | | MaxRowsmaxChannelsTable_RowIntOriginalControlTypechannelControlTestControlTypechannelControl | Туре Туре | |
| Pass/Fail Criteria: | | The device under test shall pass every verification test case to pass the test case. | n step in this | |
| Test Step Number | Test Procedu | e | Results | |
| 1 | PRE-CONDITION 'A valid timing plan is loaded with phase 2 configured with for vehicles and pedestrians'. | | red | |
| 2 | GET the following objects: maxChannels. | | Pass/Fail | |
| 2.1 | RECO | RD this information as MaxRows. | | |
| 3 | ASSIGN Table | Row EQUALS 2. | | |
| 4 | GET the follow | ng objects: channelControlType.Table_Row. | Pass/Fail | |
| 4.1 | RECO | RD this information as OriginalControlType. | | |
| 5 | ASSIGN TestC | ontrolType EQUALS 3. | | |
| 6 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transactio | n'. | |
| 6.1 | NOTE | 3 = phasePedestrian. | | |
| 7 | ASSIGN channelControlType.Table_Row EQUALS TestControlType. | | | |
| 8 | SET the following objects: channelControlType.Table_Row. | | Pass/Fail | |
| 9 | GET the follow | ng objects: channelControl I ype. I able_Row. | Pass/Fail | |
| 10 | Verify Databas | e Error'. | | |
| 11 | VERIFY chann TestControlTyp | elControlType.Table_Row IS_EQUAL_TO e. | Pass/Fail | |

| 13 ASSIGN channelControlType.Table_Row EQUALS OriginalControlType. 14 SET the following objects: channelControlType.Table_Row. Pass/ 15 GET the following objects: channelControlType.Table_Row. Pass/ 16 PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. PERIFY channelControlType.Table_Row IS_EQUAL_TO OriginalControlType. Pass/ 17 VERIFY channelControlType.Table_Row IS_EQUAL_TO OriginalControlType. Pass/ Date Test Procedure Results Tested By: | Test Procedure N | ocedure Notes: | | | | |
|--|-------------------------|---|--|-----------|--|--|
| 13 ASSIGN channelControlType.Table_Row EQUALS OriginalControlType. 14 SET the following objects: channelControlType.Table_Row. Pass/ 15 GET the following objects: channelControlType.Table_Row. Pass/ 16 PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. VERIFY channelControlType.Table_Row IS_EQUAL_TO OriginalControlType. Pass/ 17 VERIFY channelControlType. Pass/ Test Procedure Results | Tested By: | By: | Date Tested: | Pass/Fail | | |
| 13 ASSIGN channelControlType.Table_Row EQUALS OriginalControlType. 14 SET the following objects: channelControlType.Table_Row. Pass/ 15 GET the following objects: channelControlType.Table_Row. Pass/ 16 PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. VERIFY channelControlType.Table_Row IS_EQUAL_TO OriginalControlType. Pass/ | Test Procedur | Procedure Results | | | | |
| 13 ASSIGN channelControlType.Table_Row EQUALS OriginalControlType. 14 SET the following objects: channelControlType.Table_Row. Pass/ 15 GET the following objects: channelControlType.Table_Row. Pass/ 16 PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. Pass/ | 17 | 17 VERIFY channelControlType.Table_Row IS_ OriginalControlType. | VERIFY channelControlType.Table_Row IS_EQUAL_TO OriginalControlType. | | | |
| 13 ASSIGN channelControlType.Table_Row EQUALS OriginalControlType. 14 SET the following objects: channelControlType.Table_Row. Pass/ 15 GET the following objects: channelControlType.Table_Row. Pass/ | 16 | 16 PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 13 ASSIGN channelControlType.Table_Row EQUALS OriginalControlType. 14 SET the following objects: channelControlType.Table_Row. Pass/ | 15 | 15 GET the following objects: channelControlTyp | pe.Table_Row. | Pass/Fail | | |
| 13 ASSIGN channelControlType.Table_Row EQUALS OriginalControlType. | 14 | 14 SET the following objects: channelControlTyp | e.Table_Row. | Pass/Fail | | |
| | 13 | 13 ASSIGN channelControlType.Table_Row EQ | ASSIGN channelControlType.Table_Row EQUALS OriginalControlType. | | | |
| 12 PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | 12 | 12 PERFORM the Test Procedure 'C.3.1.1 Creat | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |

C.3.6.3 Configure Channel Flash Yellow

| Test Procedure: | | Configure Channel Flash Yellow | |
|---------------------|---|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to set the channel state for Automatic Flash to Flash Yellow. | |
| Requirement(s): | | • 3.5.2.1.7.1.3.1 Configure Channel Flash Yellov | v |
| Variable(s): | | MaxRowsmaxChannelsOriginalChannelFlashchannelFlashTestChannelFlashchannelFlashTable_RowInt | |
| Pass/Fail Criteria: | | The device under test shall pass every verification sto test case to pass the test case. | ep in this |
| Test Step Number | Test Procedure | | Results |
| 1 | GET the following objects: maxChannels. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: channelFlash.Table_Row. | | Pass/Fail |
| 3.1 | RECORD this information as OriginalChannelFlash. | | |
| 4 | ASSIGN TestChannelFlash EQUALS OriginalChannelFlash AND 251. | | |
| 4.1 | NOTE 'Toggle Bit 2 LOW because Bits 1 & 2 of channelFlash cannot both be TRUE'. | | |
| 5 | ASSIGN TestChannelFlash EQUALS TestChannelFlash XOR 2. | | |
| 6 | ASSIGN channelFlash.Table_Row EQUALS TestChannelFlash. | | |
| 7 | SET the following objects: channelFlash.Table_Row. Pass/Fa | | Pass/Fail |
| 8 | GET the follow | ing objects: channelFlash.Table_Row. | Pass/Fail |
| 9 | VERIFY channelFlash.Table_Row IS_EQUAL_TO TestChannelFlash. Pass/F | | Pass/Fail |
| 10 | ASSIGN chann | nelFlash.Table_Row EQUALS OriginalChannelFlash. | |
| 11 | SET the follow | ing objects: channelFlash.Table_Row. | Pass/Fail |
| 12 | GET the follow | GET the following objects: channelFlash.Table Row. Pass/Fail | |

| 13 | VERIFY channelFlash.Table_Row IS_EQUAL_TO OriginalChannelFlash. | | | |
|----------------------------|--|-----------------|-----------|--|
| Test Procedure Results | | | | |
| Tested By: Date Tested: | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.6.4 Enable/Disable Channel Flash - Red

| Test Procedure: | | Enable/Disable Channel Fla | sh - Red | |
|---------------------|---|---|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to set the channel state for Automatic Flash to Flash Red. | | |
| Requirement(s): | | • 3.5.2.1.7.1.3.2 Confi | gure Channel Flash Red | |
| Variable(s): | | MaxRows Table_Row OriginalChannelFlash TestChannelFlash | maxChannels Int channelFlash channelFlash | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the following objects: maxChannels. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: channelFlash.Table_Row. Pa | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalChannelFlash. | | | |
| 4 | ASSIGN TestChannelFlash EQUALS OriginalChannelFlash XOR 4. | | | |
| 5 | ASSIGN channelFlash.Table_Row EQUALS TestChannelFlash. | | | |
| 6 | SET the follow | ing objects: channelFlash.Table | e_Row. | Pass/Fail |
| 7 | GET the following objects: channelFlash.Table_Row. Pass/Fa | | Pass/Fail | |
| 8 | VERIFY channelFlash.Table_Row IS_EQUAL_TO TestChannelFlash. Pass/Fail | | | Pass/Fail |
| 9 | ASSIGN channelFlash.Table_Row EQUALS OriginalChannelFlash. | | | |
| 10 | SET the following objects: channelFlash.Table_Row. Pass/Fail | | | Pass/Fail |
| 11 | GET the follow | ing objects: channelFlash.Tabl | e_Row. | Pass/Fail |
| 12 | VERIFY channelFlash.Table_Row IS_EQUAL_TO OriginalChannelFlash. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | Tested By: Date Pass/Fail Tested: | | | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.6.5 Configure Channel Flash Alternate Hertz

Configure Channel Flash Alternate Hald Hertz

| Test Procedure: | | | | |
|---------------------|---|--|--|-----------|
| Description: | | This test case verifies that the ASC allows a management station to set the channel Alternate Half Hertz during Automatic Flash. | | |
| Requirement(s): | | • 3.5.2.1.7.1.3.3 Config Hertz | gure Channel Flash Altern | ate Half |
| Variable(s): | | MaxRows Table_Row OriginalChannelFlash TestChannelFlash | maxChannels Int channelFlash channelFlash | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the following objects: maxChannels. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: channelFlash.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalChannelFlash. | | | |
| 4 | ASSIGN TestChannelFlash EQUALS OriginalChannelFlash XOR 8. | | | |
| 5 | ASSIGN channelFlash.Table_Row EQUALS TestChannelFlash. | | | |
| 6 | SET the following objects: channelFlash.Table_Row. Pass/Fa | | Pass/Fail | |
| 7 | GET the following objects: channelFlash.Table_Row. Pas | | Pass/Fail | |
| 8 | VERIFY channelFlash.Table_Row IS_EQUAL_TO TestChannelFlash. Pass/Fail | | | Pass/Fail |
| 9 | ASSIGN channelFlash.Table_Row EQUALS OriginalChannelFlash. | | | |
| 10 | SET the following objects: channelFlash.Table_Row. Pass/Fa | | Pass/Fail | |
| 11 | GET the following objects: channelFlash.Table_Row. Pass/Fail | | Pass/Fail | |
| 12 | VERIFY channelFlash.Table_Row IS_EQUAL_TO OriginalChannelFlash. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.6.6 Configure Channel Flash Alternate First or Second

| Test Procedure: | Configure Channel Flash Alter | nate Hald Hertz |
|--------------------|---|--------------------|
| Description: | This test case verifies that the ASC allows a management station to set the channel to flash first or second in an alternating manner during Automatic Flash. | |
| Requirement(s): | 3.5.2.1.7.1.3.4 Configure Channel Flash Alternate Half Hertz | |
| Variable(s): | MaxRows Table_Row | maxChannels Int |

| | | OriginalChannelFlash TestChannelFlash | channelFlash channelFlash | |
|---------------------|---|--|-------------------------------------|------------|
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure Res | | Results | |
| 1 | GET the follow | ing objects: maxChannels. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: channelFlash.Tabl | e_Row. | Pass/Fail |
| 3.1 | RECO | RECORD this information as OriginalChannelFlash. | | |
| 4 | ASSIGN TestChannelFlash EQUALS OriginalChannelFlash XOR 16. | | | |
| 5 | ASSIGN chann | ASSIGN channelFlash.Table_Row EQUALS TestChannelFlash. | | |
| 6 | SET the follow | ing objects: channelFlash.Table | e_Row. | Pass/Fail |
| 7 | GET the follow | GET the following objects: channelFlash.Table_Row. Pass/Fail | | |
| 8 | VERIFY channelFlash.Table_Row IS_EQUAL_TO TestChannelFlash. Pass/Fail | | | |
| 9 | ASSIGN channelFlash.Table_Row EQUALS OriginalChannelFlash. | | | |
| 10 | SET the following objects: channelFlash.Table_Row. Pass/Fail | | | Pass/Fail |
| 11 | GET the following objects: channelFlash.Table_Row. Pass/Fail | | | |
| 12 | VERIFY channelFlash.Table_Row IS_EQUAL_TO OriginalChannelFlash. | | | |
| Test Procedu | re Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.6.7 Determine Maximum Number of Channels

| Test Procedure: | | Determine Maximum Number of Channels Supported | | |
|---------------------------------------|---|--|------------------------------------|------------------------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of channels supported in the channel table. | | |
| Requirement(s): | | • 3.5.2.1.7.2.1 Determine Maximum Number of Channels | | |
| Variable(s): | | UserMinChannels | maxChannels | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verfication ste ase. | p in the |
| Test Step Number | Test Procedu | re | | Results |
| 1 | USER-ACTION 'Determine the number of channels required as specified in FR ID 3.5.2.1.7.2.1 of the PRL.' | | | |
| | specified in FR | ID 3.5.2.1.7.2.1 of the PRL.' | | |
| 1.1 | specified in FR RECO | ID 3.5.2.1.7.2.1 of the PRL.' RD this information as UserMir | nChannels. | |
| 1.1 2 | specified in FR RECO GET the follow | ID 3.5.2.1.7.2.1 of the PRL.' RD this information as UserMir ing objects: maxChannels. | nChannels. | Pass/Fail |
| 1.1 2 3 | specified in FR RECO GET the follow VERIFY maxC | ID 3.5.2.1.7.2.1 of the PRL.' RD this information as UserMir ing objects: maxChannels. hannels IS_NOT_LESS_THAN | nChannels. I UserMinChannels. | Pass/Fail Pass/Fail |
| 1.1 2 3 Test Procedur | specified in FR RECO GET the follow VERIFY maxC e Results | ID 3.5.2.1.7.2.1 of the PRL.' RD this information as UserMir ing objects: maxChannels. hannels IS_NOT_LESS_THAN | nChannels. | Pass/Fail Pass/Fail |

Test Procedure Notes:

C.3.7 Overlaps

C.3.7.1 Configure Overlap Type – Vehicle Normal

| Test Procedure: | | Configure Overlap Type - Normal | | |
|---------------------|--|---|--|------------|
| Description: | | This test case verifies that t station to set the overlap ty | the ASC allows a managen pe to normal. | nent |
| Requirement(s): | | • 3.5.2.1.8.1.1.1 Config | gure Overlap Type - Vehic | e Normal |
| Variable(s): | | MaxRows Table_Row OriginalType TestType | maxOverlaps Int overlapType overlapType | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: overlapType.Table_Row. Pass | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalType. | | | |
| 4 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 5 | ASSIGN TestType EQUALS 2. | | | |
| 5.1 | NOTE 2 = normal. | | | |
| 6 | ASSIGN overlapType.Table_Row EQUALS TestType. | | | |
| 7 | SET the following objects: overlapType.Table_Row. Pass/Fail | | | |
| 8 | GET the following objects: overlapType.Table_Row. Pass/Fail | | | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 10 | VERIFY overla | pType.Table_Row IS_EQUAL_ | _TO TestType. | Pass/Fail |
| 11 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 12 | ASSIGN overlapType.Table Row EQUALS OriginalType. | | | |
| 13 | SET the following objects: overlapType.Table Row. Pass/Fail | | Pass/Fail | |
| 14 | GET the follow | ing objects: overlapType.Table | _Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 16 | VERIFY overla | pType.Table_Row IS_EQUAL_ | TO OriginalType. | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.7.2 Configure Overlap Type – Vehicle Minus Green and Yellow

Configure Overlap Type – Vehicle Minus Green Yellow

| Test Procedure: | | | | |
|---------------------|--|--|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to set the overlap type to Minus Green and Yellow | | |
| Requirement(s): | | • 3.5.2.1.8.1.1.2 Config Green and Yellow | gure Overlap Type - Vehic | e Minus |
| Variable(s): | | MaxRowsmaxOverlapsTable_RowIntOriginalTypeoverlapTypeTestTypeoverlapType | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste use. | ep in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: overlapType.Table_Row. Pass/Fail | | | |
| 3.1 | RECORD this information as OriginalType. | | | |
| 4 | PERFORM the | Test Procedure 'C.3.1.1 Create | e Database Transaction'. | |
| 5 | ASSIGN TestType EQUALS 3. | | | |
| 5.1 | NOTE 3 = minusGreenYellow. | | | |
| 6 | ASSIGN overlapType.Table_Row EQUALS TestType. | | | |
| 7 | SET the following objects: overlapType.Table_Row. Pass/Fail | | | |
| 8 | GET the following objects: overlapType.Table_Row. Pass/Fail | | | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 10 | VERIFY overlapType.Table_Row IS_EQUAL_TO TestType. Pass/Fail | | | |
| 11 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 12 | ASSIGN overla | pType.Table_Row EQUALS O | riginalType. | |
| 13 | SET the following objects: overlapType.Table_Row. Pass/Fail | | | |
| 14 | GET the follow | ing objects: overlapType.Table | _Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 16 | VERIFY overla | pType.Table_Row IS_EQUAL_ | TO OriginalType. | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.7.3 Configure Overlap Type – Pedestrian Normal

| Test Procedure: Configure Overlap Type – Pedestrian Normal |
|---|
|---|

| Description: | | This test case verifies that the ASC allows a management station to set the overlap type to pedestrian normal. | | |
|---------------------|--|--|--|------------|
| Requirement(s): | | 3.5.2.1.8.1.1.3 Configure Overlap Type – Pedestrian Normal | | |
| Variable(s): | | MaxRows Table_Row OriginalType TestType | maxOverlaps Int overlapType overlapType | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | re | | Results |
| 1 | GET the follow | ing objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRow | VS. | |
| 2 | ASSIGN Table | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: overlapType.Table_Row. Pass/Fail | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalType. | | | |
| 4 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 5 | ASSIGN TestT | ASSIGN TestType EQUALS 4. | | |
| 5.1 | NOTE | 4 = pedestrianNormal. | | |
| 6 | ASSIGN overlapType.Table_Row EQUALS TestType. | | | |
| 7 | SET the following objects: overlapType.Table_Row. Pass/Fail | | | |
| 8 | GET the following objects: overlapType.Table_Row. Pass/Fail | | | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 10 | VERIFY overlapType.Table_Row IS_EQUAL_TO TestType. Pass/Fail | | | |
| 11 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 12 | ASSIGN overla | apType.Table_Row EQUALS O | riginalType. | |
| 13 | SET the following objects: overlapType.Table_Row. Pass/Fail | | Pass/Fail | |
| 14 | GET the follow | ing objects: overlapType.Table | Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 16 | VERIFY overla | pType.Table_Row IS_EQUAL_ | _TO OriginalType. | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.7.4 Configure Overlap Type – Flashing Yellow Arrow – 3 Section Head

| Test | Configure Overlap Type – Flashing Yellow Arrow – 3 Section |
|--------------|--|
| Procedure: | Head |
| Description: | This test case verifies that the ASC allows a management station to set the overlap type to a 3 Section Flashing Yellow Arrow. |

| Requirement(s): | | • 3.5.2.1.8.1.1.4 Confi Arrow - 3 Section H | gure Overlap Type - Flash ead | ing Yellow |
|-------------------------|--|---|--|------------|
| Variable(s): | | MaxRows Table_Row OriginalType TestType | maxOverlaps Int overlapType overlapType | |
| Pass/Fail Crit | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedur | - .e | | Results |
| 1 | GET the follow | ing objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 1 | O MaxRows). | |
| 3 | GET the follow | ing objects: overlapType.Table | e_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Туре. | |
| 4 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | te Database Transaction'. | |
| 5 | ASSIGN TestType EQUALS 5. | | | |
| 5.1 | NOTE 5 = fyaThreeSection. | | | |
| 6 | ASSIGN overlapType.Table_Row EQUALS TestType. | | | |
| 7 | SET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 8 | GET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 10 | VERIFY overlapType.Table Row IS EQUAL TO TestType. | | Pass/Fail | |
| 11 | PERFORM the | Test Procedure 'C.3.1.1 Crea | te Database Transaction'. | |
| 12 | ASSIGN overlapType.Table_Row EQUALS OriginalType. | | | |
| 13 | SET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 14 | GET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 16 | VERIFY overla | pType.Table_Row IS_EQUAL | _TO OriginalType. | Pass/Fail |
| Test Procedure Results | | | | |
| Tested By: Date Tested: | | | Pass/Fail | |
| Test Procedure | Notes: | | | |

C.3.7.5 Configure Overlap Type - Flashing Yellow Arrow - 4 Section Head

| Test Procedure: | Configure Overlap Type - Flashing Yellow Arrow - 4 Sec Head | |
|--------------------|--|--|
| Description: | This test case verifies station to set the ove Arrow. | s that the ASC allows a management rlap type to a 4 Section Flashing Yellow |
| Requirement(s): | • 3.5.2.1.8.1.1.5 Arrow - 4 Sec | Configure Overlap Type - Flashing Yellow tion Head |
| | MaxRows | maxOverlaps |

| Variable(s): | | Table_Row OriginalType TestType | Int overlapType overlapType | |
|---------------------|--|---|-----------------------------------|-----------|
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedur | re | | Results |
| 1 | GET the follow | ing objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: overlapType.Table | e_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Туре. | |
| 4 | PERFORM the | Test Procedure 'C.3.1.1 Crea | te Database Transaction'. | |
| 5 | ASSIGN TestT | ype EQUALS 6. | | |
| 5.1 | NOTE | 6 = fyaFourSection. | | |
| 6 | ASSIGN overlapType.Table_Row EQUALS TestType. | | | |
| 7 | SET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 8 | GET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 10 | VERIFY overlapType.Table_Row IS_EQUAL_TO TestType. | | Pass/Fail | |
| 11 | PERFORM the | Test Procedure 'C.3.1.1 Crea | te Database Transaction'. | |
| 12 | ASSIGN overlapType.Table_Row EQUALS OriginalType. | | | |
| 13 | SET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 14 | GET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 16 | VERIFY overlapType.Table_Row IS_EQUAL_TO OriginalType. | | _TO OriginalType. | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | Date Tested: | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.7.6 Configure Overlap Type - Flashing Red Arrow - 3 Section Head

| Test Procedure: | Configure Overlap Type | Configure Overlap Type - Flashing Red Arrow - 3 Section Head | |
|--------------------|---|---|--|
| Description: | This test case verifies the station to set the overla Arrow. | This test case verifies that the ASC allows a management station to set the overlap type to a 3 Section Flashing Red Arrow. | |
| Requirement(s): | • 3.5.2.1.8.1.1.6 C Arrow - 3 Sectio | 3.5.2.1.8.1.1.6 Configure Overlap Type - Flashing Red Arrow - 3 Section Head | |
| Variable(s): | MaxRows Table_Row OriginalType TestType | maxOverlaps Int overlapType overlapType | |

| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
|------------------------|--|---|--------------------------|-----------|
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: overlapType.Table | _Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Туре. | |
| 4 | PERFORM the | Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 5 | ASSIGN TestT | ype EQUALS 7. | | |
| 5.1 | NOTE | 7 = fyaThreeSection. | | |
| 6 | ASSIGN overla | pType.Table_Row EQUALS T | estType. | |
| 7 | SET the followi | ng objects: overlapType.Table | _Row. | Pass/Fail |
| 8 | GET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 10 | VERIFY overla | pType.Table_Row IS_EQUAL_ | _TO TestType. | Pass/Fail |
| 11 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 12 | ASSIGN overla | pType.Table_Row EQUALS C | PriginalType. | |
| 13 | SET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 14 | GET the following objects: overlapType.Table_Row. | | Pass/Fail | |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 16 | VERIFY overlapType.Table_Row IS_EQUAL_TO OriginalType. | | _TO OriginalType. | Pass/Fail |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.7.7 Configure Overlap Type - Flashing Red Arrow - 4 Section Head

| Test Procedure: | Configure Overlap Type | Configure Overlap Type - Flashing Red Arrow - 4 Section Head | | |
|---------------------|---|---|--|--|
| Description: | This test case verifies the station to set the overla Arrow. | This test case verifies that the ASC allows a management station to set the overlap type to a 4 Section Flashing Red Arrow. | | |
| Requirement(s): | • 3.5.2.1.8.1.1.7 Co Arrow - 4 Sectio | 3.5.2.1.8.1.1.7 Configure Overlap Type - Flashing Red Arrow - 4 Section Head | | |
| Variable(s): | MaxRowsmaxOverlapsTable_RowIntOriginalTypeoverlapTypeTestTypeoverlapType | | | |
| Pass/Fail Criteria: | The device under test shall pass every verification step in this test case to pass the test case. | | | |

٦

| Test Step Number | Test Procedure | | |
|---------------------|--|--------------------------|-----------|
| 1 | GET the following objects: maxOverlaps. | | |
| 1.1 | RECORD this information as MaxRov | vs. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | Туре. | |
| 4 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 5 | ASSIGN TestType EQUALS 8. | | |
| 5.1 | NOTE 8 = fraFourSection. | | |
| 6 | ASSIGN overlapType.Table_Row EQUALS T | estType. | |
| 7 | SET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 8 | GET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | Database Status and | |
| 10 | VERIFY overlapType.Table_Row IS_EQUAL_ | _TO TestType. | Pass/Fail |
| 11 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 12 | ASSIGN overlapType.Table_Row EQUALS O | riginalType. | |
| 13 | SET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 14 | GET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | VERIFY overlapType.Table_Row IS_EQUAL_TO OriginalType. | | |
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.7.8 Configure Overlap Type – 2 Section Transit Specific Signal Head

| Test Procedure: | | Configure Overlap Type – 2 Section Transit Specific Signal Head | | |
|---------------------|---|---|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to set the overlap type to support a 2-section head. | | |
| Requirement(| s): | 3.5.2.1.8.1.1.8 Configure Overlap Type – 2 Section Transit Specific Signal Head | | tion |
| Variable(s): | | MaxRows Table_Row OriginalType TestType | maxOverlaps Int overlapType overlapType | |
| Pass/Fail Criteria: | | The device under test test case to pass the t | shall pass every verification ste est case. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxOverlaps. | | Pass/Fail | |

| 1.1 | RECORD this information as MaxRows. | | |
|------------------------|--|--------------------------|-----------|
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | Туре. | |
| 4 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 5 | ASSIGN TestType EQUALS 9. | | |
| 5.1 | NOTE 9 = transit-2. | | |
| 6 | ASSIGN overlapType.Table_Row EQUALS T | estType. | |
| 7 | SET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 8 | GET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | Database Status and | |
| 10 | VERIFY overlapType.Table_Row IS_EQUAL_TO TestType. | | Pass/Fail |
| 11 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 12 | ASSIGN overlapType.Table_Row EQUALS O | priginalType. | |
| 13 | SET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 14 | GET the following objects: overlapType.Table | _Row. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | VERIFY overlapType.Table_Row IS_EQUAL_TO OriginalType. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.7.9 Configure Overlap Type – Minus Green Yellow Alternate

| Test Procedure: | | Configure Overlap Type – Minus Green Yellow Alternate | |
|---------------------|--|---|--------------------------|
| Description: | | This test case verifies that the ASC allows a management station to set the overlap type to Vehicle Minus Green and Yellow Alternate. | |
| Requirement(s): | | 3.5.2.1.8.1.1.9 Configure Overlap Type – Minus Green Yellow Alternate | |
| Variable(s): | | MaxRowsmaxOvTable_RowIntOriginalTypeoverlagTestTypeoverlag | erlaps oType oType |
| Pass/Fail Criteria: | | The device under test shall pass every v test case to pass the test case. | erification step in this |
| Test Step Number | Test Procedu | Test Procedure | |
| 1 | GET the follow | GET the following objects: maxOverlaps. | |
| 1.1 | RECO | RECORD this information as MaxRows. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the follow | GET the following objects: overlapType.Table_Row. | |
| Test Procedure | Notes: | | | |
|----------------|--|---|-----------|--|
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedu | re Results | | | |
| 16 | VERIFY overlapType.Table_Row IS_EQUAL | TO OriginalType. | Pass/Fail | |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 14 | GET the following objects: overlapType.Table_Row. Pas | | | |
| 13 | SET the following objects: overlapType.Table | SET the following objects: overlapType.Table_Row. Pas | | |
| 12 | ASSIGN overlapType.Table_Row EQUALS OriginalType. | | | |
| 11 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 10 | VERIFY overlapType.Table_Row IS_EQUAL_TO TestType. | | | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 8 | GET the following objects: overlapType.Table | _Row. | Pass/Fail | |
| 7 | SET the following objects: overlapType.Table | _Row. | Pass/Fail | |
| 6 | ASSIGN overlapType.Table_Row EQUALS T | estType. | | |
| 5.1 | NOTE 10 = minusGreenYellow. | | | |
| 5 | ASSIGN TestType EQUALS 10. | | | |
| 4 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 3.1 | RECORD this information as Original | Туре. | | |

C.3.7.10 Configure Overlap Included Phases

| Test Procedure: Configure Overlap Included Phases | | | | |
|--|--|--|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure which phases are included in overlaps. | | |
| Requirement(| (s): | • 3.5.2.1.8.1.2 Configure | Overlap Included Phase | es |
| Variable(s): | | MaxRows Table_Row OriginalPhases TestPhases | maxOverlaps Int overlapincludedPhas overlapincludedPhas | ses Ses |
| Pass/Fail Criteria: | | The device under test shall p test case to pass the test cas | ass every verification ste e. | ep in this |
| Test Step Number | Test Procedu | e | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded' | | |
| 2 | GET the follow | ving objects: maxOverlaps. | | Pass/Fail |
| 2.1 | RECO | RECORD this information as MaxRows. | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 4 | GET the follow | ng objects: overlapIncludedPhas | es.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalPh | ases. | |

| 5 | USER-ACTION 'Select a valid value for overlapIncludedPhases.Table_Row.' | | |
|----------------|--|--------------------------|-----------|
| 5.1 | RECORD this information as TestPha | ISES. | |
| 6 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 7 | ASSIGN overlapIncludedPhases.Table_Row | EQUALS TestPhases. | |
| 8 | SET the following objects: overlapIncludedPh | ases.Table_Row. | Pass/Fail |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | v Database Status and | |
| 10 | GET the following objects: overlapIncludedPh | ases.Table_Row. | Pass/Fail |
| 11 | VERIFY overlapIncludedPhases.Table_Row IS_EQUAL_TO TestPhases. | | |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 13 | ASSIGN overlapIncludedPhases.Table_Row EQUALS OriginalPhases. | | |
| 14 | SET the following objects: overlapIncludedPhases.Table_Row. | | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: overlapIncludedPh | ases.Table_Row. | Pass/Fail |
| 17 | VERIFY overlapIncludedPhases.Table_Row IS_EQUAL_TO OriginalPhases. | | |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.7.11 Configure Overlap Modifier Phases

| Test Procedure: | | Configure Overlap Modifier Phases | | |
|---------------------|---|--|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure a modifier phase for an overlap. | | |
| Requirement | Requirement(s): • 3.5.2.1.8.1.3 Configure Overlap Modifier Phases | | S | |
| Variable(s): | | MaxRows Table_Row OriginalPhases TestPhases | maxOverlaps Int overlapModifierPhas overlapModifierPhas | ies ies |
| Pass/Fail Criteria: | | The device under test s test case to pass the te | hall pass every verification ste st case. | ep in this |
| Test Step Number | Test Procedu | e. | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is le | oaded'. | |
| 2 | GET the follow | ving objects: maxOverlaps. | | Pass/Fail |
| 2.1 | RECO | RECORD this information as MaxRows. | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 4 | GET the follow | ing objects: overlapModifie | erPhases.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as Orig | ginalPhases. | |

| 5 | USER-ACTION 'Select a valid value for overlapModifierPhases.Table_Row.' | | |
|------------------|--|--------------------------|-----------|
| 5.1 | RECORD this information as TestPhases. | | |
| 6 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 7 | ASSIGN overlapModifierPhases.Table_Row E | EQUALS TestPhases. | |
| 8 | SET the following objects: overlapModifierPha | ases.Table_Row. | Pass/Fail |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | Database Status and | |
| 10 | GET the following objects: overlapModifierPha | ases.Table_Row. | Pass/Fail |
| 11 | VERIFY overlapModifierPhases.Table_Row IS_EQUAL_TO TestPhases. | | |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 13 | ASSIGN overlapModifierPhases.Table_Row EQUALS OriginalPhases. | | |
| 14 | SET the following objects: overlapModifierPhases.Table_Row. | | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 16 | GET the following objects: overlapModifierPhases.Table_Row. | | Pass/Fail |
| 17 | VERIFY overlapModifierPhases.Table_Row IS_EQUAL_TO OriginalPhases. | | Pass/Fail |
| Test Procedur | e Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.7.12 Configure Pedestrian Modifier Phases

| Test Procedure: | st ocedure: Configure Pedestrian Modifier Phases | | | |
|---|---|--|----------------|-----------|
| Description: | | This test case verifies that the ASC shall allow a management station to configure pedestrian modifier phases for a vehicle overlap. | | |
| Requirement | ment(s): • 3.5.2.1.8.1.4 Configure Pedestrian Modifier Phases | | ases | |
| Variable(s): | | Table_RowIntOriginalConflictingPedPhasesoverlapConflictingPedPhasesTestConflictingPedPhasesoverlapConflictingPedPhasesMaxRowsmaxOverlaps | | |
| Pass/Fail Criteria: The device under test shall pass every verification stores to pass the test case. | | ep in this | | |
| Test Step Number | Test Procedur | e | | Results |
| 1 | PRE-CONDITION | ON The ASC supports overlap modifie | rs. | |
| 2 | GET the followi | ing objects: maxOverlaps. | | Pass/Fail |
| 2.1 | RECO | RECORD this information as MaxRows. | | |
| 3 | ASSIGN Table | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 4 | GET the followi | ng objects: overlapConflictingPedPha | ses.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalConflict | ngPedPhases. | |

| 5 | USER-ACTION Select a valid value for TestC | onflictingPedPhases. | | |
|------------------|---|-----------------------|-----------|--|
| 6 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 7 | ASSIGN overlapConflictingPedPhases.Table_Row EQUALS TestConflictingPedPhases. | | | |
| 8 | SET the following objects: overlapConflictingF | PedPhases.Table_Row. | Pass/Fail | |
| 9 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | | |
| 10 | GET the following objects: overlapConflictingF | PedPhases.Table_Row. | Pass/Fail | |
| 11 | VERIFY overlapConflictingPedPhases.Table_Row IS_EQUAL_TO TestConflictingPedPhases. | | | |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | | |
| 13 | ASSIGN overlapConflictingPedPhases.Table_Row EQUALS OriginalConflictingPedPhases. | | | |
| 14 | SET the following objects: overlapConflictingF | PedPhases.Table_Row. | Pass/Fail | |
| 15 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | | |
| 16 | GET the following objects: overlapConflictingF | PedPhases.Table_Row. | Pass/Fail | |
| 17 | VERIFY overlapConflictingPedPhases.Table_Row IS_EQUAL_TO OriginalConflictingPedPhases. | | | |
| Test Procedur | e Results | | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.7.13 Configure Overlap Trailing Green

| Test Procedure: | Configure Overlap Trailing Green | | | |
|---------------------|--|---|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure the trailing green time for an overlap GREEN indication to be extended. | | |
| Requirement(| Requirement(s): • 3.5.2.1.8.1.5 Configure Overlap Trailing Green | | | |
| Variable(s): | | MaxRows Table_Row OriginalTrailGreen TestTrailGreen | maxOverlaps Int overlapTrailGreen overlapTrailGreen | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedu | e | | Results |
| 1 | GET the follow | ing objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: overlapTrailGreen.Table_Row. Pass/I | | Pass/Fail | |
| 3.1 | RECO | RD this information as Original | TrailGreen. | |
| 4 | ASSIGN TestT | railGreen EQUALS RANDOM | (0 TO 255). | |

| 4.1 | IF TestTrailGreen IS_EQUAL_TO OriginalTrailGreen. | |
|----------------|---|-----------|
| 4.1.1 | GOTO step 4. | |
| 5 | ASSIGN overlapTrailGreen.Table_Row EQUALS TestTrailGreen. | |
| 6 | SET the following objects: overlapTrailGreen.Table_Row. | Pass/Fail |
| 7 | GET the following objects: overlapTrailGreen.Table_Row. | Pass/Fail |
| 8 | VERIFY overlapTrailGreen.Table_Row IS_EQUAL_TO TestTrailGreen. | Pass/Fail |
| 9 | ASSIGN overlapTrailGreen.Table_Row EQUALS OriginalTrailGreen. | |
| 10 | SET the following objects: overlapTrailGreen.Table_Row. | |
| 11 | GET the following objects: overlapTrailGreen.Table_Row. | |
| 12 | VERIFY overlapTrailGreen.Table_Row IS_EQUAL_TO OriginalTrailGreen. | |
| Test Procedur | e Results | |
| Tested By: | Tested By: Date Tested: | |
| Test Procedure | Notes: | - |

C.3.7.14 Configure Overlap Trailing Yellow

| Test Procedure: | | Configure Overlap Trailing Yellow | |
|--|--|--|---|
| Description: | | This test case verifies that the ASC allows a managen station to configure the trailing yellow time for the over the event that the overlap GREEN indication is extend | nent erlap in led. |
| Requirement(| s): | • 3.5.2.1.8.1.6 Configure Overlap Trailing Yellow | , |
| Variable(s): | | MaxRowsmaxOverlapsTable_RowIntOriginalTrailYellowoverlapTrailYellowTestTrailYellowoverlapTrailYellow | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step | | | |
| Number | Test Procedur | e | Results |
| Number 1 | Test Procedur GET the follow | re ing objects: maxOverlaps. | Results Pass/Fail |
| Number 1 1.1 | Test Procedur GET the follow RECO | re ing objects: maxOverlaps. RD this information as MaxRows. | Results Pass/Fail |
| Number 1 1.1 2 | GET the follow RECO ASSIGN Table | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). | Results Pass/Fail |
| Number 1 1.1 2 3 | GET the follow RECO ASSIGN Table GET the follow | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. RD this information as OriginalTrailYellow. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 | Test ProcedurGET the followRECOASSIGN TableGET the followRECOASSIGN TestT | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. RD this information as OriginalTrailYellow. frailYellow EQUALS RANDOM (0 TO 255). | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestT IF Test | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. RD this information as OriginalTrailYellow. frailYellow EQUALS RANDOM (0 TO 255). tTrailYellow IS_EQUAL_TO OriginalTrailYellow. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 4.1.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestT IF Test | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. RD this information as OriginalTrailYellow. railYellow EQUALS RANDOM (0 TO 255). tTrailYellow IS_EQUAL_TO OriginalTrailYellow. GOTO step 4. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 5 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestT IF Test ASSIGN overla | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. RD this information as OriginalTrailYellow. railYellow EQUALS RANDOM (0 TO 255). tTrailYellow IS_EQUAL_TO OriginalTrailYellow. GOTO step 4. apTrailYellow.Table_Row EQUALS TestTrailYellow. | Results Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 5 6 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestT IF Test ASSIGN overla SET the follow | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. RD this information as OriginalTrailYellow. railYellow EQUALS RANDOM (0 TO 255). tTrailYellow IS_EQUAL_TO OriginalTrailYellow. GOTO step 4. apTrailYellow.Table_Row EQUALS TestTrailYellow. ing objects: overlapTrailYellow.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 7 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestT IF Test ASSIGN overla SET the follow GET the follow | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. RD this information as OriginalTrailYellow. railYellow EQUALS RANDOM (0 TO 255). tTrailYellow IS_EQUAL_TO OriginalTrailYellow. GOTO step 4. apTrailYellow.Table_Row EQUALS TestTrailYellow. ing objects: overlapTrailYellow.Table_Row. ing objects: overlapTrailYellow.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 4.1 5 6 7 8 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestT IF Test ASSIGN overla SET the follow GET the follow VERIFY overla | re ing objects: maxOverlaps. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: overlapTrailYellow.Table_Row. RD this information as OriginalTrailYellow. railYellow EQUALS RANDOM (0 TO 255). tTrailYellow IS_EQUAL_TO OriginalTrailYellow. GOTO step 4. apTrailYellow.Table_Row EQUALS TestTrailYellow. ing objects: overlapTrailYellow.Table_Row. ing objects: overlapTrailYellow.Table_Row. pTrailYellow.Table_Row IS_EQUAL_TO TestTrailYellow. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 10 | SET the following objects: overlapTrailYellow.Table_Row. Pass/Fa | | | | |
|--------------|---|--|-----------|--|--|
| 11 | GET the following objects: overlapTrailYellow | GET the following objects: overlapTrailYellow.Table_Row. | | | |
| 12 | VERIFY overlapTrailYellow.Table_Row IS_EQUAL_TO OriginalTrailYellow. | | | | |
| Test Procedu | Test Procedure Results | | | | |
| | | Data | | | |
| Tested By: | | Tested: | Pass/Fail | | |

C.3.7.15 Configure Overlap Trailing Red Clearance

| Test Procedure: | Configure Overlap Trailing Red Clearance | | |
|---------------------|---|--|-----------|
| Description: | | This test case verifies that the ASC allows a management station to configure the trailing red clearance time for the overlap in the event that the overlap GREEN indication is extended. | |
| Requirement(| (s): | • 3.5.2.1.8.1.7 Configure Overlap Trailing Red C | learance |
| Variable(s): | | MaxRowsmaxOverlapsTable_RowIntOriginalTrailRedoverlapTrailRedTestTrailRedoverlapTrailRed | |
| Pass/Fail Crit | eria: | The device under test shall pass every verification step in the test case to pass the test case. | |
| Test Step Number | Test Procedu | re | Results |
| 1 | GET the following objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: overlapTrailRed.Table_Row. | | Pass/Fail |
| 3.1 | RECO | RECORD this information as OriginalTrailRed. | |
| 4 | ASSIGN TestT | railRed EQUALS RANDOM (0 TO 255). | |
| 4.1 | IF Tes | tTrailRed IS_EQUAL_TO OriginalTrailRed. | |
| 4.1.1 | | GOTO step 4. | |
| 5 | ASSIGN overla | apTrailRed.Table_Row EQUALS TestTrailRed. | |
| 6 | SET the follow | ing objects: overlapTrailRed.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: overlapTrailRed.Table_Row. | Pass/Fail |
| 8 | VERIFY overla | pTrailRed.Table_Row IS_EQUAL_TO TestTrailRed. | Pass/Fail |
| 9 | ASSIGN overla | apTrailRed.Table_Row EQUALS OriginalTrailRed. | |
| 10 | SET the follow | ing objects: overlapTrailRed.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: overlapTrailRed.Table_Row. | Pass/Fail |
| 12 | VERIFY overlapTrailRed.Table_Row IS_EQUAL_TO OriginalTrailRed. Pass/Fai | | Pass/Fail |
| Test Procedur | re Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |

Test Procedure Notes:

C.3.7.16 Configure Overlap Walk

| Test Procedure: | | Configure Overlap Walk | | | |
|---------------------|---|--|---|------------|--|
| Description: | | This test case verifies that t station to configure the wal | This test case verifies that the ASC allows a management station to configure the walk time for a pedestrian overlap. | | |
| Requirement(s): | | • 3.5.2.1.8.1.8 Configu | • 3.5.2.1.8.1.8 Configure Overlap Walk | | |
| Variable(s): | | MaxRows Table_Row OriginalOverlapWalk TestOverlapWalk | maxOverlaps Int overlapWalk overlapWalk | | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this | |
| Test Step Number | Test Procedur | e | | Results | |
| 1 | GET the following objects: maxOverlaps. | | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | | | |
| 3 | GET the following objects: overlapWalk.Table_Row. | | Pass/Fail | | |
| 3.1 | RECORD this information as OriginalOverlapWalk. | | | | |
| 4 | ASSIGN TestOverlapWalk EQUALS RANDOM (0 TO 255). | | | | |
| 4.1 | IF TestOverlapWalk IS_EQUAL_TO OriginalOverlapWalk. | | | | |
| 4.1.1 | | GOTO step 4. | - | | |
| 5 | ASSIGN overla | apWalk.Table_Row EQUALS T | estOverlapWalk. | | |
| 6 | SET the follow | ing objects: overlapWalk. I able | _Row. | Pass/Fail | |
| / | GET the following objects: overlapWalk.Table_Row. Pass/F | | Pass/Fail | | |
| 8 | VERIFY overlapWalk.Table_Row IS_EQUAL_TO TestOverlapWalk. Pass/Fail | | | Pass/Fail | |
| 9 | ASSIGN overlapWalk.Table_Row EQUALS OriginalOverlapWalk. | | | Dece/Feil | |
| 10 | SET the following objects: overlapWalk.Table_Row. Pass/Fa | | Pass/Fall | | |
| 10 | GET the following objects: overlap/Walk.Table_Row. | | Pass/Fail | | |
| | | | | rass/rail | |
| rest Flocedur | e Results | | Dato | | |
| Tested By: | | | Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | | |

C.3.7.17 Configure Overlap Pedestrian Clearance

| Test Procedure: | Configure Overlap Pedestrian Clearance |
|--------------------|--|
| | |

| Description: | | This test case verifies that the ASC allows a management station to configure the pedestrian clearance time for an overlap. | | |
|---------------------|--|---|--|------------|
| Requirement(| (s): | • 3.5.2.1.8.1.9 Configure Overlap Pedestrian Clearance | | |
| Variable(s): | | MaxRows Table_Row OriginalOverlapPedCleara TestOverlapPedClearance | maxOverlaps Int nce overlapPedClearanc overlapPedClearanc | ;e ;e |
| Pass/Fail Crit | teria: | The device under test shall test case to pass the test ca | pass every verification sto ase. | ∍p in this |
| Test Step Number | Test Procedur | re | | Results |
| 1 | GET the follow | ing objects: maxOverlaps. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRow | /S. | |
| 2 | ASSIGN Table | ASSIGN Table_Row EQUALS RANDOM (1 to MaxRows). | | |
| 3 | GET the follow | GET the following objects: overlapPedClearance.Table_Row. | | Pass/Fail |
| 3.1 | RECORD this information as OriginalOverlapPedClearance. | | | |
| 4 | ASSIGN TestPedAlternateWalk EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestOverlapPedClearance IS_EQUAL_TO OriginalOverlapPedClearance. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN overla TestOverlapPe | ASSIGN overlapPedClearance.Table_Row EQUALS TestOverlapPedClearance. | | |
| 6 | SET the follow | ing objects: overlapPedClearan | ce.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: overlapPedClearar | nce.Table_Row. | Pass/Fail |
| 8 | VERIFY overla TestOverlapPe | VERIFY overlapPedClearance.Table_Row IS_EQUAL_TO TestOverlapPedClearance. | | Pass/Fail |
| 9 | ASSIGN overla OriginalOverla | ASSIGN overlapPedClearance.Table_Row EQUALS OriginalOverlapPedClearance. | | |
| 10 | SET the follow | SET the following objects: overlapPedClearance.Table Row. | | Pass/Fail |
| 11 | GET the follow | ing objects: overlapPedClearar | nce.Table_Row. | Pass/Fail |
| 12 | VERIFY overlapPedClearance.Table_Row IS_EQUAL_TO OriginalOverlapPedClearance. | | Pass/Fail | |
| Test Procedur | re Results | | | - |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.7.18 Determine Maximum Number of Overlaps

| Test Procedure: | Determine Maximum Number of Overlaps |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to determine the maximum number of overlaps supported by ASC and verifies that the ASC supports the minimum number of overlaps required by the user. |

| Requirement(s): | | • 3.5.2.1.8.3.1 Determine Maximum Number of Overlaps | | |
|---------------------|---|---|-----------------|-----------|
| Variable(s): | | UserMinOverlaps maxOverlaps | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | USER-ACTION 'Determine the number of overlaps required as specified in FR ID 3.5.2.1.8.3.1 of the PRL.' | | | |
| 1.1 | RECORD this information as UserMinOverlaps. | | | |
| 2 | GET the following objects: maxOverlaps. | | Pass/Fail | |
| 3 | VERIFY maxOverlaps IS_NOT_LESS_THAN UserMinOverlaps. | | Pass/Fail | |
| Test Procedur | e Results | | | - |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.7.19 Determine Maximum Number of Overlaps Sets

| Test Procedure: | | Determine Maximum Number of Overlaps | | |
|---------------------|---|--|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of overlap sets supported by ASC and verifies that the ASC supports the minimum number of overlap sets required by the user. | | |
| Requirement(s): | | 3.5.2.1.8.3.2 Determine Maximum Number of Overlap Sets | | |
| Variable(s): | | UserMinOverlapSets maxOverlapSets | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedu | Procedure | | Results |
| 1 | USER-ACTION 'Determine the number of overlaps required as specified in FR ID 3.5.2.1.8.3.2 of the PRL.' | | | |
| 1.1 | RECORD this information as UserMinOverlapSets. | | | |
| 2 | GET the following objects: maxOverlapSets. | | Pass/Fail | |
| 3 | VERIFY maxOverlapSets IS_NOT_LESS_THAN UserMinOverlapSets. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.8 Preemption

C.3.8.1 Enable/Disable Preempt Inputs

| Test Procedure: | | Enable/Disable Preempt Inp | outs | |
|---------------------|--|--|--|-----------|
| Description: | | This test case verifies that t station to enable/disable pr | the ASC allows a managen eempt inputs. | nent |
| Requirement(| s): | • 3.5.2.1.9.1.1 Enable/ | Disable Preempt Inputs | |
| Variable(s): | | MaxRows Table_Row OriginalPreemptControl TestPreemptControl | MaxRowsmaxPreemptsTable_RowIntOriginalPreemptControlpreemptControlTestPreemptControlpreemptControl | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxPreempts. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: preemptControl.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPreemptControl. | | | |
| 4 | ASSIGN TestPreemptControl EQUALS OriginalPreemptControl XOR 16. | | | |
| 5 | ASSIGN preen | nptControl.Table_Row EQUAL | S TestPreemptControl. | |
| 6 | SET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 8 | VERIFY preemptControl.Table_Row IS_EQUAL_TO TestPreemptControl. | | Pass/Fail | |
| 9 | ASSIGN preen | nptControl.Table_Row EQUALS | S OriginalPreemptControl. | |
| 10 | SET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 12 | VERIFY preemptControl.Table_Row IS_EQUAL_TO OriginalPreemptControl. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.8.2 Configure Preempt Control - Non-Locking Memory

| Test Procedure: | Configure Preempt Control - Non-Locking Memory |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to configure if a preempt is to not occur if the preempt |

| | | request terminates prior to time. | the expiration of the preen | npt delay |
|-----------------------|--|--|--|------------|
| Requirement(s): | | 3.5.2.1.9.1.2 Configure Preempt Control - Non-Locking Memory | | |
| Variable(s): | | MaxRows Table_Row OriginalPreemptControl TestPreemptControl | maxPreempts Int preemptControl preemptControl | |
| Pass/Fail Crit | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | ·e | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | O MaxRows). | |
| 3 | GET the following objects: preemptControl.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPreemptControl. | | | |
| 4 | ASSIGN TestPreemptControl EQUALS OriginalPreemptControl XOR 1. | | | |
| 5 | ASSIGN preemptControl.Table_Row EQUALS TestPreemptControl. | | | |
| 6 | SET the following objects: preemptControl. Table_Row. | | Pass/Fail | |
| 7 | GET the following objects: preemptControl.Table_Row. | | | Pass/Fail |
| 8 | VERIFY preemptControl.Table_Row IS_EQUAL_TO TestPreemptControl. | | | Pass/Fail |
| 9 | ASSIGN preen | nptControl.Table_Row EQUALS | S OriginalPreemptControl. | |
| 10 | SET the following objects: preemptControl.Table_Row. Pass/F | | | Pass/Fail |
| 11 | GET the following objects: preemptControl.Table_Row. Pass/Fa | | | Pass/Fail |
| 12 | VERIFY preemptControl.Table_Row IS_EQUAL_TO OriginalPreemptControl. | | Pass/Fail | |
| Test Procedur | e Results | | | - |
| Tested By: | Date Pass/Fail Tested: | | | |
| Test Procedure | Notes: | | | |

C.3.8.3 Configure Preempt Control – Override Automatic Flash

| Test Procedure: | Configure Preempt Control - | Configure Preempt Control - Preempt Override Flash | | |
|--------------------|--|--|--|--|
| Description: | This test case verifies that th station to configure if a preer automatic flash. | This test case verifies that the ASC allows a management station to configure if a preempt is not allowed to override automatic flash. | | |
| Requirement(s): | • 3.5.2.1.9.1.3 Configure Override Flash | 3.5.2.1.9.1.3 Configure Preempt Control - Preempt Override Flash | | |
| Variable(s): | MaxRows Table_Row OriginalPreemptControl TestPreemptControl | maxPreempts Int preemptControl preemptControl | | |

| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
|---------------------|--|---|-------------------------------------|-----------|
| Test Step Number | Test Procedur | - - - | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | PreemptControl. | |
| 4 | ASSIGN TestP | reemptControl EQUALS Origin | alPreemptControl XOR 2. | |
| 5 | ASSIGN preemptControl.Table_Row EQUALS TestPreemptControl. | | | |
| 6 | SET the following objects: preemptControl.Table_Row. | | | Pass/Fail |
| 7 | GET the following objects: preemptControl.Table_Row. | | | Pass/Fail |
| 8 | VERIFY preemptControl.Table_Row IS_EQUAL_TO TestPreemptControl. | | | Pass/Fail |
| 9 | ASSIGN preemptControl.Table_Row EQUALS OriginalPreemptControl. | | | |
| 10 | SET the following objects: preemptControl.Table_Row. | | Pass/Fail | |
| 11 | GET the following objects: preemptControl.Table_Row. | | | Pass/Fail |
| 12 | VERIFY preemptControl.Table_Row IS_EQUAL_TO OriginalPreemptControl. | | | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.8.4 Configure Preempt Control – Override Preempt

| Test Procedure: | | Configure Preempt Control – Override Preempt | |
|---------------------|--|---|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure if is not allowed to override the preempt next in the preemptTable. | |
| Requirement(s): | | 3.5.2.1.9.1.4 Configure Preempt Control – Override Preempt | |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalPreemptControlpreemptControlTestPreemptControlpreemptControl | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | Test Procedure | | Results |
| 1 | GET the following objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the follow | GET the following objects: preemptControl.Table_Row. Pass/Fail | |

| 3.1 | RECORD this information as Original | PreemptControl. | |
|----------------|--|-------------------------|-----------|
| 4 | ASSIGN TestPreemptControl EQUALS Origin | alPreemptControl XOR 4. | |
| 5 | ASSIGN preemptControl.Table_Row EQUALS | S TestPreemptControl. | |
| 6 | SET the following objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 7 | GET the following objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 8 | VERIFY preemptControl.Table_Row IS_EQUAL_TO TestPreemptControl. | | Pass/Fail |
| 9 | ASSIGN preemptControl.Table_Row EQUALS OriginalPreemptControl. | | |
| 10 | SET the following objects: preemptControl.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: preemptControl.Table_Row. | | Pass/Fail |
| 12 | VERIFY preemptControl.Table_Row IS_EQUAL_TO OriginalPreemptControl. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.8.5 Configure Preempt Control – Flash Dwell

| Test Procedure: | | Configure Preempt Control - Flash Dwell | | |
|------------------------------|---|--|--|--|
| Description: | | This test case verifies that the ASC allows a management station for enable or disable the Flash Dwell parameter for a preempt. | | ent er for a |
| Requirement(| s): | 3.5.2.1.9.1.5 Configure Preempt Control - Flash Dwell | | n Dwell |
| Variable(s): | | MaxRows Table_Row OriginalPreemptControl TestPreemptControl | maxPreempts Int preemptControl preemptControl | |
| Pass/Fail Crit | eria: | The device under test shall pass test case to pass the test case. | every verification ste | p in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO Ma | axRows). | |
| 3 | | | | |
| 2.1 | GET the follow | ing objects: preemptControl.Table_F | Row. | Pass/Fail |
| 3.1 | GET the follow RECO | ing objects: preemptControl.Table_F RD this information as OriginalPreer | Row. nptControl. | Pass/Fail |
| 4 | GET the follow RECO ASSIGN TestP | ing objects: preemptControl.Table_F RD this information as OriginalPreer reemptControl EQUALS OriginalPre | Row. nptControl. emptControl XOR 8. | Pass/Fail |
| 4 5 | GET the follow RECO ASSIGN TestP ASSIGN preem | ing objects: preemptControl.Table_F RD this information as OriginalPreer reemptControl EQUALS OriginalPre nptControl.Table_Row EQUALS Tes | Row. nptControl. emptControl XOR 8. tPreemptControl. | Pass/Fail |
| <u> </u> | GET the follow RECO ASSIGN TestP ASSIGN preen SET the follow | ing objects: preemptControl.Table_F RD this information as OriginalPreer reemptControl EQUALS OriginalPre nptControl.Table_Row EQUALS Tes ng objects: preemptControl.Table_F | Row. nptControl. emptControl XOR 8. tPreemptControl. Row. | Pass/Fail Pass/Fail |
| 4 5 6 7 | GET the follow RECO ASSIGN TestP ASSIGN preen SET the follow GET the follow | ing objects: preemptControl.Table_F RD this information as OriginalPreer reemptControl EQUALS OriginalPre nptControl.Table_Row EQUALS Tes ng objects: preemptControl.Table_F ing objects: preemptControl.Table_F | Row. nptControl. emptControl XOR 8. tPreemptControl. Row. | Pass/Fail Pass/Fail Pass/Fail |
| 3.1 4 5 6 7 8 | GET the follow RECO ASSIGN TestP ASSIGN preem SET the follow GET the follow VERIFY preem TestPreemptCo | ing objects: preemptControl.Table_F RD this information as OriginalPreer reemptControl EQUALS OriginalPre nptControl.Table_Row EQUALS Tes ng objects: preemptControl.Table_F ing objects: preemptControl.Table_F ptControl.Table_Row IS_EQUAL_T ontrol. | Row. nptControl. eemptControl XOR 8. tPreemptControl. Row. Row. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 10 | SET the following objects: preemptControl.Table_Row. Pas | | | |
|----------------------------|--|--|-----------|--|
| 11 | GET the following objects: preemptControl.Ta | GET the following objects: preemptControl.Table_Row. | | |
| 12 | VERIFY preemptControl.Table_Row IS_EQUAL_TO OriginalPreemptControl. | | | |
| | | | | |
| Test Procedu | re Results | | | |
| Test Procedu Tested By: | re Results | Date Tested: | Pass/Fail | |

C.3.8.6 Configure Preempt Control – All Red Entry

| Test Procedure: | Test Procedure: Configure Preempt Control – All Red Entry | | | |
|---------------------|--|---|--|-----------------|
| Description: | | This test case verifies that the station for configure the AS start of a preempt. | the ASC allows a managen SC to go to an all-red state | nent and the |
| Requirement | (s): | • 3.5.2.1.9.1.6 Configu | ıre Preempt Control – All F | ed Entry |
| Variable(s): | | MaxRows Table_Row OriginalPreemptControl TestPreemptControl | maxPreempts Int preemptControl preemptControl | |
| Pass/Fail Crit | Pass/Fail Criteria: The device under test shall pass every verification ste test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedu | re | | Results |
| 1 | GET the following objects: maxPreempts. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: preemptControl.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPreemptControl. | | | |
| 4 | ASSIGN TestF | ASSIGN TestPreemptControl EQUALS OriginalPreemptControl XOR 32. | | |
| 5 | ASSIGN preen | nptControl.Table_Row EQUAL | S TestPreemptControl. | |
| 6 | SET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 8 | VERIFY preem TestPreemptC | ptControl.Table_Row IS_EQU | AL_TO | Pass/Fail |
| 9 | ASSIGN preen | nptControl.Table_Row EQUAL | S OriginalPreemptControl. | |
| 10 | SET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: preemptControl.Ta | ble_Row. | Pass/Fail |
| 12 | VERIFY preem OriginalPreem | ptControl.Table_Row IS_EQU | AL_TO | Pass/Fail |
| Test Procedu | re Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.8.7 Configure Preempt Link

| Test Procedure: | | Configure Preempt Link | | |
|---------------------|--|--|--|------------------|
| Description: | | This test case verifies that t station to associate preemp table. | the ASC allows a managen ots with previous preempts | nent s in the |
| Requirement(| s): | • 3.5.2.1.9.1.7 Configu | ire Preempt Link | |
| Variable(s): | | MaxRows Table_Row OriginalPreemptLink TestPreemptLink | maxPreempts Int preemptLink preemptLink | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in thi test case to pass the test case. | | ep in this |
| Test Step Number | Test Procedu | .е | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (2 TO MaxRows). | | | |
| 3 | GET the follow | ing objects: preemptLink.Test_ | Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | PreemptLink. | |
| 4 | ASSIGN TestP | reemptLink EQUALS RANDO | /I (1 TO Table_Row). | |
| 5 | ASSIGN preen | nptLink.Test_Row EQUALS Te | stPreemptLink. | |
| 6 | SET the follow | ing objects: preemptLink.Test_ | Row. | Pass/Fail |
| 7 | GET the follow | ing objects: preemptLink.Test_ | Row. | Pass/Fail |
| 8 | VERIFY preem | ptLink.Test_Row IS_EQUAL_1 | FO TestPreemptLink. | Pass/Fail |
| 9 | ASSIGN preen | nptLink.Test_Row EQUALS Or | iginalPreemptLink. | |
| 10 | SET the follow | ing objects: preemptLink.Test_ | Row. | Pass/Fail |
| 11 | GET the follow | ing objects: preemptLink.Test_ | Row. | Pass/Fail |
| 12 | VERIFY preem | ptLink.Test_Row IS_EQUAL_1 | TO OriginalPreemptLink. | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |

C.3.8.8 Configure Preempt Delay

| Test Procedure: | Configure Preempt Delay | |
|--------------------|--|---|
| Description: | This test case verifies that the ASC allows a management station to configure how long a preempt input shall be active prior to any preempt sequence begins. | ; |
| Requirement(s): | • 3.5.2.1.9.1.8 Configure Preempt Delay | |
| | MaxRows maxPreempts | - |

| Variable(s): | | Table_Row OriginalPreemptDelay TestPreemptDelay | Int preemptDelay preemptDelay | |
|---------------------|--|---|-------------------------------------|------------|
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: preemptDelay.Tab | le_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | PreemptDelay. | |
| 4 | ASSIGN TestP | reemptDelay EQUALS RANDO | DM (0 TO 600). | |
| 4.1 | IF TestPreemptDelay IS_EQUAL_TO OriginalPreemptDelay. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN preem | nptDelay.Table_Row EQUALS | TestPreemptDelay. | |
| 6 | SET the followi | ng objects: preemptDelay.Tabl | e_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: preemptDelay.Tab | le_Row. | Pass/Fail |
| 8 | VERIFY preem | ptDelay.Table_Row IS_EQUA | L_TO TestPreemptDelay. | Pass/Fail |
| 9 | ASSIGN preem | nptDelay.Table_Row EQUALS | OriginalPreemptDelay. | |
| 10 | SET the followi | ng objects: preemptDelay.Tabl | e_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: preemptDelay.Tab | le_Row. | Pass/Fail |
| 12 | VERIFY preemptDelay.Table_Row IS_EQUAL_TO OriginalPreemptDelay. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.8.9 Configure Preempt Minimum Duration

| Test Procedure: | | Configure Preempt Minimum Duration | | |
|---------------------|--------------|---|------------------|--|
| Description: | | This test case verifies that the ASC allows a management station to configure the minimum time a preempt shall be active. | | |
| Requirement(| s): | • 3.5.2.1.9.1.9 Configure Preempt Minimum Duration | | |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalMinimumDurationpreemptMinimumDuTestMinimumDurationpreemptMinimumDu | ration ration | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | p in this | |
| Test Step Number | Test Procedu | re | Results | |

| 1 | GET the following objects: maxPreempts. | | Pass/Fail |
|----------------|---|---------------------|-----------|
| 1.1 | RECORD this information as MaxRov | VS. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: preemptMinimumD | ouration.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | MinimumDuration. | |
| 4 | ASSIGN TestMinimumDuration EQUALS RAM | NDOM (0 TO 65535). | |
| 4.1 | IF TestMinimumDuration IS_EQUAL_ OriginalMinimumDuration. | ТО | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN preemptMinimumDuration.Table_Ro TestMinimumDuration. | w EQUALS | |
| 6 | SET the following objects: preemptMinimumD | uration.Table_Row. | Pass/Fail |
| 7 | GET the following objects: preemptMinimumDuration.Table_Row. | | Pass/Fail |
| 8 | VERIFY preemptMinimumDuration.Table_Row IS_EQUAL_TO TestMinimumDuration. | | Pass/Fail |
| 9 | ASSIGN preemptMinimumDuration.Table_Ro OriginalMinimumDuration. | w EQUALS | |
| 10 | SET the following objects: preemptMinimumD | uration.Table_Row. | Pass/Fail |
| 11 | GET the following objects: preemptMinimumD | ouration.Table_Row. | Pass/Fail |
| 12 | VERIFY preemptMinimumDuration.Table_Row IS_EQUAL_TO OriginalMinimumDuration. | | |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.8.10 Configure Preempt Enter Minimum Green Time

| Test Procedure: | | Configure Preempt Enter Minimum Green Time | |
|---------------------|----------------|--|------------------------------|
| Description: | | This test case verifies that the ASC allows manageme to configure the minimum green time for phases at th preempt. | ent station le start of a |
| Requirement(| s): | 3.5.2.1.9.1.10.1 Configure Preempt Enter Minimum Green Time | |
| Variable(s): | | MaxRows maxPreempts Table_Row Int OriginalMinimumGreen preemptMinimumGreen TestGreenTime preemptMinimumGreen | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification statest case to pass the test case. | ep in this |
| Test Step Number | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | |
| 2 | ASSIGN Table | Row EQUALS RANDOM (1 TO MaxRows). | |
| 3 | GET the follow | ing objects: preemptMinimumGreen.Table_Row. | Pass/Fail |

| 3.1 | RECORD this information as Original | MinimumGreen. | |
|----------------|---|-----------------------|-----------|
| 4 | ASSIGN TestGreenTime EQUALS RANDOM | (0 TO 255). | |
| 4.1 | IF TestGreenTime IS_NOT_EQUAL_ OriginalMinimumGreen. | ТО | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN preemptMinimumGreen.Table_Row | EQUALS TestGreenTime. | |
| 6 | SET the following objects: preemptMinimumG | Freen.Table_Row. | Pass/Fail |
| 7 | GET the following objects: preemptMinimumG | Green.Table_Row. | Pass/Fail |
| 8 | VERIFY preemptMinimumGreen.Table_Row IS_EQUAL_TO TestGreenTime. | | Pass/Fail |
| 9 | ASSIGN preemptMinimumGreen.Table_Row EQUALS OriginalMinimumGreen. | | |
| 10 | SET the following objects: preemptMinimumGreen.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: preemptMinimumG | Green.Table_Row. | Pass/Fail |
| 12 | VERIFY preemptMinimumGreen.Table_Row IS_EQUAL_TO OriginalMinimumGreen. | | Pass/Fail |
| Test Procedur | e Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.8.11 Configure Preempt Enter Minimum Walk Time

| Test Procedure: | | Configure Preempt Enter Minimum Walk Time | |
|---------------------|--|---|--------------|
| Description: | | This test case verifies that the ASC allows a manager station the minimum walk time for phases at the start preempt. | nent of a |
| Requirement(| s): | 3.5.2.1.9.1.10.2 Configure Preempt Enter Minimum Walk Time | |
| Variable(s): | | MaxRows maxPreempts Table_Row Int OriginalMinimumWalk preemptMinimumWalk TestMinimumWalk preemptMinimumWalk | |
| Pass/Fail Crite | eria: The device under test shall pass every verification step in t test case to pass the test case. | | ep in this |
| Test Step Number | Test Procedu | est Procedure | |
| 1 | GET the follow | ing objects: maxPreempts. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | |
| 2 | ASSIGN Table | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | |
| 3 | GET the following objects: preemptMinimumWalk. | | Pass/Fail |
| 3.1 | RECO | RECORD this information as OriginalMinimumWalk. | |
| 4 | ASSIGN TestM | linimumWalk EQUALS RANDOM (0 TO 255). | |
| 4.1 | IF Test | MinimumWalk IS_EQUAL_TO OriginalMinimumWalk. | |
| 4.1.1 | | GOTO step 4. | |

| 5 | ASSIGN preemptMinimumWalk.Table_Row EQUALS TestMinimumWalk. | | | |
|----------------|---|-----------------|-----------|--|
| 6 | SET the following objects: preemptMinimumV | Valk.Table_Row. | Pass/Fail | |
| 7 | GET the following objects: preemptMinimumV | Valk.Table_Row. | Pass/Fail | |
| 8 | VERIFY preemptMinimumWalk.Table_Row IS_EQUAL_TO TestMinimumWalk. | | | |
| 9 | ASSIGN preemptMinimumWalk.Table_Row EQUALS OriginalMinimumWalk. | | | |
| 10 | SET the following objects: preemptMinimumWalk.Table_Row. Pass/F | | | |
| 11 | GET the following objects: preemptMinimumWalk.Table_Row. Pa | | | |
| 12 | VERIFY preemptMinimumWalk.Table_Row IS_EQUAL_TO OriginalMinimumWalk. | | | |
| Test Procedu | re Results | | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure | Notes: | | | |

C.3.8.12 Configure Preempt Enter Pedestrian Clearance Time

| Test Procedure: | | Configure Preempt Enter Pedestrian Clearance Time | | |
|--|--|--|---|---|
| Description: | This test case verifies that the ASC allows a management Description: station to configure the clearance time for a WALK display is terminated by a preempt. | | nent splay that | |
| Requirement(| s): | 3.5.2.1.9.1.10.3 Configu Clearance Time | re Preempt Enter Pede | strian |
| Variable(s): | | MaxRows Table_Row OriginalPedClear TestPedClear | maxPreempts Int preemptEnterPedCle preemptEnterPedCle | ear ear |
| Pass/Fail Criteria: | | The device under test shall parties test case to pass the test case | ss every verification ste | ep in this |
| | | | | |
| Test Step Number | Test Procedur | re | | Results |
| Test Step Number 1 | Test Procedur | re ing objects: maxPreempts. | | Results Pass/Fail |
| Test Step Number 1 1.1 | Test Procedur GET the follow RECO | re ing objects: maxPreempts. RD this information as MaxRows. | | Results Pass/Fail |
| Test StepNumber11.12 | Test Procedur GET the follow RECO ASSIGN Table | re ing objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO N | /laxRows). | Results Pass/Fail |
| Test Step Number 1 1.1 2 3 | Test Procedur GET the follow RECO ASSIGN Table GET the follow | re ing objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO N ing objects: preemptEnterPedClea | /laxRows). r.Table_Row. | Results Pass/Fail Pass/Fail |
| 1 1.1 2 3 3.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO | re ing objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO N ing objects: preemptEnterPedClea RD this information as OriginalPec | /laxRows). r.Table_Row. Clear. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP | re ing objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO Noing objects: preemptEnterPedClear RD this information as OriginalPed PedClear EQUALS RANDOM (0 TO | /axRows). r.Table_Row. Clear.) 255). | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test | re ing objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO N ing objects: preemptEnterPedClea RD this information as OriginalPec PedClear EQUALS RANDOM (0 TO tPedClear IS_EQUAL_TO Original | MaxRows). r.Table_Row. Clear. D 255). PedClear. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test | re ring objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO N ring objects: preemptEnterPedClea RD this information as OriginalPec PedClear EQUALS RANDOM (0 TO tPedClear IS_EQUAL_TO Original GOTO step 4. | /axRows). r.Table_Row. Clear. 0 255). PedClear. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test ASSIGN preen | re ing objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO N ing objects: preemptEnterPedClea RD this information as OriginalPec PedClear EQUALS RANDOM (0 TO tPedClear IS_EQUAL_TO Original GOTO step 4. nptEnterPedClear.Table_Row EQU | MaxRows). r.Table_Row. Clear. 0 255). PedClear. JALS TestPedClear. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test ASSIGN preen SET the follow | re ing objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO N ing objects: preemptEnterPedClea RD this information as OriginalPed PedClear EQUALS RANDOM (0 TO tPedClear IS_EQUAL_TO Original GOTO step 4. nptEnterPedClear.Table_Row EQU ing objects: preemptEnterPedClea | MaxRows). r.Table_Row. Clear. D 255). PedClear. JALS TestPedClear. r.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 7 | Test Procedur GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestP IF Test ASSIGN preem SET the follow GET the follow | re ring objects: maxPreempts. RD this information as MaxRows. Row EQUALS RANDOM (1 TO N ring objects: preemptEnterPedClea RD this information as OriginalPed PedClear EQUALS RANDOM (0 TO tPedClear IS_EQUAL_TO Original GOTO step 4. nptEnterPedClear.Table_Row EQU ing objects: preemptEnterPedClea ring objects: preemptEnterPedClea | MaxRows). r.Table_Row. Clear. 0 255). PedClear. JALS TestPedClear. r.Table_Row. r.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| | | Testeu. | | | |
|--------------|--|-----------------|-----------|--|--|
| Tested By: | | Date Tostod: | Pass/Fail | | |
| Test Procedu | re Results | | | | |
| 12 | VERIFY preemptEnterPedClear.Table_Row IS_EQUAL_TO OriginalPedClear. | | | | |
| 11 | GET the following objects: preemptEnterPedClear.Table_Row. | | | | |
| 10 | SET the following objects: preemptEnterPedClear.Table_Row. | | | | |
| 9 | ASSIGN preemptEnterPedClear.Table_Row EQUALS OriginalPedClear. | | | | |

C.3.8.13 Configure Preempt Enter Yellow Change Time

| Test Procedure: | | Configure Preempt Enter Yellow Change Time | | |
|---------------------|--|---|--|------------------|
| Description: | | This test case verifies that t station to configure the min yellow change interval term | he ASC allows a managen imum yellow change time inated by a preempt. | ient for a |
| Requirement(| s): | • 3.5.2.1.9.1.10.4 Conf Time | igure Preempt Enter Yellov | w Change |
| Variable(s): | | MaxRows Table_Row OriginalYellowChange TestYellowChange | maxPreempts Int preemptEnterYellow preemptEnterYellow | Change Change |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste se. | p in this |
| Test Step Number | Test Procedur | 'e | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: preemptEnterYellowChange.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalYellowChange. | | | |
| 4 | ASSIGN TestYellowChange EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF Test | tYellowChange IS_EQUAL_TO | OriginalYellowChange. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN preem TestYellowCha | nptEnterYellowChange.Table_F inge. | Row EQUALS | |
| 6 | SET the followi | ing objects: preemptEnterYellov | vChange.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: preemptEnterYellov | wChange.Table_Row. | Pass/Fail |
| 8 | VERIFY preem TestYellowCha | nptEnterYellowChange.Table_R nge. | ow IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN preem OriginalYellow | nptEnterYellowChange.Table_F Change. | Row EQUALS | |
| 10 | SET the following objects: preemptEnterYellowChange.Table_Row. | | | Pass/Fail |
| 11 | GET the follow | ing objects: preemptEnterYellov | wChange.Table_Row. | Pass/Fail |
| 12 | VERIFY preem OriginalYellow | emptEnterYellowChange.Table_Row IS_EQUAL_TO | | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

| C.3.8.14 | Configure | Preempt | Enter F | Red | Clearance | Time |
|----------|-----------|---------|---------|-----|-----------|------|
|----------|-----------|---------|---------|-----|-----------|------|

| Test Procedure: | | Configure Preempt Enter Red Clearance Time | | |
|---------------------|--|--|---|------------|
| Description: | | This test case verifies that t station to configure the min clearance interval terminate | he ASC allows a managen himum red clearance time f ed by a preempt. | or a red |
| Requirement(| s): | • 3.5.2.1.9.1.10.5 Conf Time | igure Preempt Enter Red 0 | Clearance |
| Variable(s): | | MaxRows Table_Row OriginalRedClear TestRedClear | maxPreempts Int preemptEnterRedCle preemptEnterRedCle | ear ear |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | /S. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: preemptEnterRedClear.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalRedClear. | | | |
| 4 | ASSIGN TestRedClear EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestRedClear IS_EQUAL_TO OriginalRedClear. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN preem | nptEnterRedClear.Table_Row I | EQUALS TestRedClear. | |
| 6 | SET the followi | ng objects: preemptEnterRedC | lear.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: preemptEnterRed0 | Clear.Table_Row. | Pass/Fail |
| 8 | VERIFY preem TestRedClear. | ptEnterRedClear.Table_Row I | S_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN preem OriginalRedCle | nptEnterRedClear.Table_Row I ear. | EQUALS | |
| 10 | SET the followi | ng objects: preemptEnterRedC | lear.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: preemptEnterRed0 | Clear.Table_Row. | Pass/Fail |
| 12 | VERIFY preem OriginalRedCle | emptEnterRedClear.Table_Row IS_EQUAL_TO Pas | | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.8.15 Configure Preempt Track Clearance Green Time

| Test Procedure: Configure Preempt Track Clearance Green Time |
|--|
|--|

| Description: | | This test case verifies that the ASC allows a management station to configure the minimum time that preempt track clearance phases display a GREEN indication during the track clearance interval. | | |
|---------------------|---|---|-------------------------------------|-----------|
| Requirement(| s): | • 3.5.2.1.9.1.11.1 Cont Green Time | igure Preempt Track Clear | ance |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalTrackGreenpreemptTrackGreenTestTrackTimepreemptTrackGreen | | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedur | - 'e | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | /S. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: preemptTrackGreen.Table_Row. P | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalTrackGreen. | | | |
| 4 | ASSIGN Table_Row EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestTrackTime IS_EQUAL_TO OriginalTrackGreen. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN preemptTrackGreen.Table_Row EQUALS TestTrackTime. | | | |
| 6 | SET the following objects: preemptTrackGreen.Table_Row. | | Pass/Fail | |
| 7 | GET the follow | ing objects: preemptTrackGree | n.Table_Row. | Pass/Fail |
| 8 | VERIFY preemptTrackGreen.Table_Row IS_EQUAL_TO TestTrackTime. | | Pass/Fail | |
| 9 | ASSIGN preem | nptTrackGreen.Table_Row EQ | UALS OriginalTrackGreen. | |
| 10 | SET the followi | ng objects: preemptTrackGree | n.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: preemptTrackGree | n.Table_Row. | Pass/Fail |
| 12 | VERIFY preemptTrackGreen.Table_Row IS_EQUAL_TO OriginalTrackGreen. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.8.16 Configure Preempt Track Clearance Yellow Change Time

| Test Procedure: | Configure Preempt Track Clearance Yellow Change Time |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to configure the minimum time that preempt track clearance phases display track a YELLOW indication during the track clearance interval. |
| Requirement(s): | 3.5.2.1.9.1.11.2 Configure Preempt Track Clearance Yellow Change Time |

| Variable(s): | | MaxRows OriginalYellowChange TestYellowChange Table_Row | maxPreempts preemptTrackYellov preemptTrackYellov Int | wChange wChange |
|---------------------|--|--|--|--------------------|
| Pass/Fail Crit | eria: | The device under test shall test case to pass the test ca | pass every verification st ase. | ep in this |
| Test Step Number | Test Procedu | re | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: preemptTrackYellc | wChange.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | YellowChange. | |
| 4 | ASSIGN TestY | ASSIGN TestYellowChange EQUALS RANDOM (0 TO 255). | | |
| 4.1 | IF TestYellowChange IS_EQUAL_TO OriginalYellowChange. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN preen TestYellowCha | nptTrackYellowChange.Table_l ange. | Row EQUALS | |
| 6 | SET the follow | SET the following objects: preemptTrackYellowChange.Table_Row. | | Pass/Fail |
| 7 | GET the follow | ing objects: preemptTrackYellc | wChange.Table_Row. | Pass/Fail |
| 8 | VERIFY preem TestYellowCha | nptTrackYellowChange.Table_F ange. | Row IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN preen OriginalYellow | nptTrackYellowChange.Table_l Change. | Row EQUALS | |
| 10 | SET the following objects: preemptTrackYellowChange.Table Row. | | Pass/Fail | |
| 11 | GET the follow | ing objects: preemptTrackYellc | wChange.Table_Row. | Pass/Fail |
| 12 | VERIFY preemptTrackYellowChange.Table_Row IS_EQUAL_TO OriginalYellowChange. | | | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.8.17 Configure Preempt Track Red Clearance Time

| Test Procedure: | Configure Preempt Track | Configure Preempt Track Red Clearance Time | | |
|--------------------|--|---|--|--|
| Description: | This test case verifies tha station to configure the m clearance phases display track clearance interval. | This test case verifies that the ASC allows a management station to configure the minimum time that preempt track clearance phases display track a RED indication during the track clearance interval. | | |
| Requirement(s): | • 3.5.2.1.9.1.11.3 Co Time | onfigure Preempt Track Red Clearance | | |
| Variable(s): | MaxRows Table_Row OriginalRedClear | maxPreempts Int preemptTrackRedClear | | |

| | | TestRedClear | preemptTrackRedCl | ear |
|---------------------|---|--|-------------------|-----------|
| Pass/Fail Crit | Pass/Fail Criteria:The device under test shall pass every verification step test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedu | re | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 1 | ΓΟ MaxRows). | |
| 3 | GET the follow | ing objects: preemptTrackRed | Clear.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | RedClear. | |
| 4 | ASSIGN TestF | RedClear EQUALS RANDOM (| 0 TO 255). | |
| 4.1 | IF Tes | tRedClear IS_EQUAL_TO Orig | ginalRedClear. | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN preemptTrackRedClear.Table_Row EQUALS TestRedClear. | | | |
| 6 | SET the follow | ing objects: preemptTrackRed0 | Clear.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: preemptTrackRed | Clear.Table_Row. | Pass/Fail |
| 8 | VERIFY preem TestRedClear. | VERIFY preemptTrackRedClear.Table_Row IS_EQUAL_TO TestRedClear. | | |
| 9 | ASSIGN preemptTrackRedClear.Table_Row EQUALS OriginalRedClear. | | | |
| 10 | SET the follow | SET the following objects: preemptTrackRedClear.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: preemptTrackRedClear.Table_Row. | | Pass/Fail | |
| 12 | VERIFY preemptTrackRedClear.Table_Row IS_EQUAL_TO OriginalRedClear. | | Pass/Fail | |
| Test Procedu | re Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.8.18 Configure Preempt Track Clearance Phases

| Test Procedure: | Configure Preempt Track Clearance Phases | |
|---------------------|---|---|
| Description: | This test case verifies that the ASC allows a management station to configure which phases are be active for preempts during the preempt track clearance intervals. | |
| Requirement(s): | 3.5.2.1.9.1.11.4 Configure Preempt Track Clearance Phases | |
| Variable(s): | MaxRows Table_Row OriginalTrackPhase TestTrackPhase MaxPhases | maxPreempts Int preemptTrackPhase preemptTrackPhase maxPhases |
| Pass/Fail Criteria: | The device under test shall p test case to pass the test case | bass every verification step in this se. |

| Test Step Number | Test Procedure | | Results |
|---------------------|---|--------------------------|-----------|
| 1 | PRE-CONDITION 'A valid timing plan is loaded'. | | |
| 2 | PRE-CONDITION 'The user shall know the m supported by the ASC'. | aximum of phases | |
| 2.1 | RECORD this information as MaxPha | ses. | |
| 3 | GET the following objects: maxPreempts. | | Pass/Fail |
| 3.1 | RECORD this information as MaxRov | vs. | |
| 4 | ASSIGN Table_Row EQUALS RANDOM (1 T | O MaxRows). | |
| 5 | GET the following objects: preemptTrackPhas | e.Table_Row. | Pass/Fail |
| 6 | USER-ACTION 'Select a valid value for preemptTrackPhase.Table_Row, such as '02 exceed MaxPhases.' | 05'. Each phase cannot | |
| 6.1 | RECORD this information as TestTrac | ckPhase. | |
| 7 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 8 | ASSIGN preemptTrackPhase.Table_Row EQ | UALS TestTrackPhase. | |
| 9 | SET the following objects: preemptTrackPhas | e.Table_Row. | Pass/Fail |
| 10 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 11 | GET the following objects: preemptTrackPhase.Table_Row. | | Pass/Fail |
| 12 | VERIFY preemptTrackPhase.Table_Row IS_EQUAL_TO TestTrackPhase. | | Pass/Fail |
| 13 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 14 | ASSIGN preemptTrackPhase.Table_Row EQ | UALS OriginalTrackPhase. | |
| 15 | SET the following objects: preemptTrackPhase.Table_Row. | | Pass/Fail |
| 16 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 17 | GET the following objects: preemptTrackPhas | e.Table_Row. | Pass/Fail |
| 18 | VERIFY preemptTrackPhase.Table_Row IS_EQUAL_TO OriginalTrackPhase. | | Pass/Fail |
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.8.19 Configure Preempt Track Clearance Overlaps

| Test Procedure: | Configure Preempt Track C | Configure Preempt Track Clearance Overlaps | |
|--------------------|--|--|--|
| Description: | This test case verifies that t station configure which ove preempt track clearance inter | This test case verifies that the ASC allows a management station configure which overlaps shall be active during a preempt track clearance interval. | |
| Requirement(s): | • 3.5.2.1.9.1.11.5 Conf Overlaps | igure Preempt Track Clearance | |
| Variable(s): | MaxRows Table_Row OriginalTrackOverlap TestTrackOverlap | maxPreempts Int preemptTrackOverlap preemptTrackOverlap | |

| | | MaxOverlaps | maxOverlaps | |
|------------------------|--|---|-------------------------------------|------------|
| Pass/Fail Crit | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedu | - .e | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loade | ed'. | |
| 2 | PRE-CONDITI supported by the | ON 'The user shall know the m าe ASC'. | aximum of overlaps | |
| 2.1 | RECO | RD this information as MaxOve | erlaps. | |
| 3 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 3.1 | RECO | RD this information as MaxRov | VS. | |
| 4 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 5 | GET the follow | ing objects: preemptTrackOver | lap.Table_Row. | Pass/Fail |
| 5.1 | RECO | RD this information as Original | TrackOverlap. | |
| 6 | USER-ACTION preemptTrackO exceed MaxOv | USER-ACTION 'Select a valid value for preemptTrackOverlap.Table_Row, such as '01'. Each phase cannot exceed MaxOverlaps.' | | |
| 6.1 | RECO | RECORD this information as TestTrackOverlap. | | |
| 7 | PERFORM the | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 8 | ASSIGN preen | ASSIGN preemptTrackOverlap.Table_Row EQUALS TestTrackOverlap. | | |
| 9 | SET the following objects: preemptTrackOverlap.Table_Row. Pass/F | | | Pass/Fail |
| 10 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify e Error'. | Database Status and | |
| 11 | GET the follow | ing objects: preemptTrackOver | lap.Table_Row. | Pass/Fail |
| 12 | VERIFY preem TestTrackOver | ptTrackOverlap.Table_Row IS lap. | _EQUAL_TO | Pass/Fail |
| 13 | PERFORM the | Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 14 | ASSIGN preen OriginalTrackC | nptTrackOverlap.Table_Row E overlap. | QUALS | |
| 15 | SET the follow | ing objects: preemptTrackOver | lap.Table_Row. | Pass/Fail |
| 16 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 17 | GET the following objects: preemptTrackOverlap.Table_Row. | | Pass/Fail | |
| 18 | VERIFY preemptTrackOverlap.Table_Row IS_EQUAL_TO OriginalTrackOverlap. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.8.20 Configure Preempt Minimum Green Dwell Time

| Test Procedure: | Configure Preempt Minimum Green Dwell Time |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to configure the minimum green time for the preempt dwell interval before phases are allowed to cycle. |

| Requirement(| s): | 3.5.2.1.9.1.12.1 Configure Preempt Minimum Green Dwell Time | | |
|---------------------|---|--|--|------------|
| Variable(s): | | MaxRows Table_Row OriginalDwellGreen TestDwellGreen | maxPreempts Int preemptDwellGreen preemptDwellGreen | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | e | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: preemptDwellGree | n.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as OriginalDwellGreen. | | | |
| 4 | ASSIGN TestDwellGreen EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestDwellGreen IS_EQUAL_TO OriginalDwellGreen. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN preemptDwellGreen.Table_Row EQUALS TestDwellGreen. | | | |
| 6 | SET the following objects: preemptDwellGreen.Table_Row. | | n.Table_Row. | Pass/Fail |
| 7 | GET the following objects: preemptDwellGreen.Table_Row. | | Pass/Fail | |
| 8 | VERIFY preemptDwellGreen.Table_Row IS_EQUAL_TO TestDwellGreen. | | Pass/Fail | |
| 9 | ASSIGN preemptDwellGreen.Table_Row EQUALS OriginalDwellGreen. | | | |
| 10 | SET the following objects: preemptDwellGreen.Table_Row. Pas | | Pass/Fail | |
| 11 | GET the following objects: preemptDwellGreen.Table_Row. Pass/Fa | | Pass/Fail | |
| 12 | VERIFY preem OriginalDwellG | / preemptDwellGreen.Table_Row IS_EQUAL_TO IDwellGreen. | | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | Date Tested: | | |
| Test Procedure N | Notes: | | | |

C.3.8.21 Configure Preempt Dwell Phases

| Test Procedure: | Configure Preempt Dwell P | Configure Preempt Dwell Phases | |
|--------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to configure the phases are to be serviced by a preempt during the preempt dwell interval. | | |
| Requirement(s): | • 3.5.2.1.9.1.12.2 Cont | 3.5.2.1.9.1.12.2 Configure Preempt Dwell Phases | |
| Variable(s): | MaxRows Table_Row OriginalDwellPhase TestDwellPhase | maxPreempts Int preemptDwellPhase preemptDwellPhase | |

| | | MaxPhases | maxPhases | |
|---------------------|---|--|--|-----------|
| Pass/Fail Cri | teria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedu | re | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loade | ed'. | |
| 2 | PRE-CONDITI supported by t | ON 'The user shall know the m he ASC'. | aximum of phases | |
| 2.1 | RECO | RD this information as MaxPha | ISES. | |
| 3 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 3.1 | RECO | RD this information as MaxRov | VS. | |
| 4 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 5 | GET the follow | ring objects: preemptDwellPhas | e.Table_Row. | Pass/Fail |
| 5.1 | RECO | RD this information as Original | DwellPhase. | |
| 6 | USER-ACTION such as '02 05 | N 'Select a valid value for preen '. Each phase cannot exceed N | nptDwellPhase.Table_Row laxPhases.' | |
| 6.1 | RECO | RECORD this information as TestDwellPhase. | | |
| 7 | PERFORM the | e Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 8 | ASSIGN preen | ASSIGN preemptDwellPhase.Table_Row EQUALS TestDwellPhase. | | |
| 9 | SET the follow | SET the following objects: preemptDwellPhase.Table_Row. | | Pass/Fail |
| 10 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify se Error'. | Database Status and | |
| 11 | GET the follow | ing objects: preemptDwellPhas | e.Table_Row. | Pass/Fail |
| 12 | VERIFY preem TestDwellPhas | VERIFY preemptDwellPhase.Table_Row IS_EQUAL_TO TestDwellPhase. | | Pass/Fail |
| 13 | PERFORM the | e Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 14 | ASSIGN preen | nptDwellPhase.Table_Row EQ | UALS OriginalDwellPhase. | |
| 15 | SET the follow | ing objects: preemptDwellPhas | e.Table_Row. | Pass/Fail |
| 16 | PERFORM the Verify Databas | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 17 | GET the follow | GET the following objects: preemptDwellPhase.Table_Row. | | Pass/Fail |
| 18 | VERIFY preemptDwellPhase.Table_Row IS_EQUAL_TO OriginalDwellPhase. | | Pass/Fail | |
| Test Procedu | re Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.8.22 Configure Preempt Pedestrian Movements

| Test Procedure: | Configure Preempt Dwell Pedestrian Movements |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to configure pedestrian movements to dwell in WALK during the preempt dwell interval. |
| Requirement(s): | 3.5.2.1.9.1.12.3 Configure Preempt Pedestrian Movements |

| Variable(s): | | MaxRows Table_Row OriginalDwellPed TestDwellPed MaxPhases | maxPreempts Int preemptDwellPed preemptDwellPed maxPhases | |
|---------------------|--|---|---|------------|
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification st ase. | ep in this |
| Test Step Number | Test Procedu | e | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loade | ed'. | |
| 2 | PRE-CONDITI supported by the | ON 'The user shall know the m ne ASC'. | aximum of phases | |
| 2.1 | RECO | RD this information as MaxPha | ISES. | |
| 3 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 3.1 | RECO | RD this information as MaxRov | VS. | |
| 4 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 5 | GET the follow | ing objects: preemptDwellPed. | Table_Row. | Pass/Fail |
| 5.1 | RECORD this information as OriginalDwellPed. | | | |
| 6 | USER-ACTION 'Select a valid value for preemptDwellPed.Table_Row, such as '02'. Each phase must not be greater than MaxPhases.' | | | |
| 7 | PERFORM the | Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 8 | ASSIGN preemptDwellPed.Table_Row EQUALS TestDwellPed. | | | |
| 9 | SET the following objects: preemptDwellPed.Table_Row. | | Pass/Fail | |
| 10 | PERFORM the Verify Databas | Test Procedure 'C.3.1.2 Verify e Error'. | Database Status and | |
| 11 | GET the follow | ing objects: preemptDwellPed. | Table_Row. | Pass/Fail |
| 12 | VERIFY preem | ptDwellPed.Table_Row IS_EC | UAL_TO TestDwellPed. | Pass/Fail |
| 13 | PERFORM the | Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 14 | ASSIGN preen | nptDwellPed.Table_Row EQUA | LS OriginalDwellPed. | |
| 15 | SET the follow | ng objects: preemptDwellPed. | Table_Row. | Pass/Fail |
| 16 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | | |
| 17 | GET the follow | GET the following objects: preemptDwellPed.Table_Row. | | Pass/Fail |
| 18 | VERIFY preemptDwellPed.Table_Row IS_EQUAL_TO OriginalDwellPed. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.8.23 Configure Preempt Dwell Overlaps

| Test Procedure: | Configure Preempt Dwell Overlaps |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to configure which overlaps shall be active during the preempt dwell interval. |

.

| Requirement(| s): | • 3.5.2.1.9.1.12.4 Configure Preempt Dwell Overlaps | | laps |
|---------------------|--|--|---|------------|
| Variable(s): | | MaxRows Table_Row OriginalDwellOverlap TestDwellOverlap MaxOverlaps | maxPreempts Int preemptDwellOverla preemptDwellOverla maxOverlaps | ip |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedur | re | | Results |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loade | ed'. | |
| 2 | PRE-CONDITI overlaps suppo | ON 'The user shall know the m orted by the ASC'. | aximum number of | |
| 2.1 | RECO | RD this information as MaxOve | erlaps. | |
| 3 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 3.1 | RECO | RD this information as MaxRov | VS. | |
| 4 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 5 | GET the follow | ing objects: preemptDwellOver | lap.Table_Row. | Pass/Fail |
| 5.1 | RECORD this information as OriginalDwellOverlap. | | | |
| 6 | USER-ACTION 'Select a valid value for preemptDwellOverlap.Table_Row, such as '01'. Each overlap must not be greater than MaxOverlaps.' | | | |
| 6.1 | RECO | ORD this information as TestDwellOverlap. | | |
| 7 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 8 | ASSIGN preem | preemptDwellOverlap.Table_Row EQUALS TestDwellOverlap. | | |
| 9 | SET the followi | ing objects: preemptDwellOver | lap.Table_Row. | Pass/Fail |
| 10 | PERFORM the Verify Database | e Test Procedure 'C.3.1.2 Verify e Error'. | / Database Status and | |
| 11 | GET the follow | ing objects: preemptDwellOver | lap.Table_Row. | Pass/Fail |
| 12 | VERIFY preem TestDwellOver | nptDwellOverlap.Table_Row IS lap. | _EQUAL_TO | Pass/Fail |
| 13 | PERFORM the | Test Procedure 'C.3.1.1 Creat | te Database Transaction'. | |
| 14 | ASSIGN preemptDwellOverlap.Table_Row EQUALS OriginalDwellOverlap. | | | |
| 15 | SET the followi | ing objects: preemptDwellOver | lap.Table_Row. | Pass/Fail |
| 16 | PERFORM the Verify Databas | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 17 | GET the follow | ing objects: preemptDwellOver | lap.Table_Row. | Pass/Fail |
| 18 | VERIFY preem OriginalDwellO | nptDwellOverlap.Table_Row IS vverlap. | _EQUAL_TO | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.8.24 Configure Preempt Cycling Phases

| Test Procedure: | | Configure Preempt Cycling Phases | |
|---------------------|---|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure which phases are allowed to cycle during the preempt dwell interval. | |
| Requirement(s): | | • 3.5.2.1.9.1.12.5 Configure Preempt Cycling Ph | ases |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalCyclingPhasepreemptCyclingPhaTestCyclingPhasepreemptCyclingPhaMaxPhasesmaxPhases | se se |
| Pass/Fail Cr | iteria: | The device under test shall pass every verification sto test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | Test Procedure | |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded'. | |
| 2 | PRE-CONDITI supported by t | ON 'The user shall know maximum number of phases he ASC'. | |
| 2.1 | RECO | RD this information as MaxPhases. | |
| 3 | GET the follow | ing objects: maxPreempts. | Pass/Fail |
| 3.1 | RECO | RD this information as MaxRows. | |
| 4 | ASSIGN Table | Row EQUALS RANDOM (1 TO MaxRows). | |
| 5 | GET the follow | ing objects: preemptCyclingPhase.Table_Row. | Pass/Fail |
| 5.1 | RECO | RD this information as OriginalCyclingPhase. | |
| 6 | USER-ACTION preemptCyclin be greater than | USER-ACTION 'Select a valid value for preemptCyclingPhase.Table_Row, such as '02'. Each phase must not be greater than MaxPhases.' | |
| 6.1 | RECO | RD this information as TestCyclingPhase. | |
| 7 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 8 | ASSIGN preen | nptCyclingPhase.Table_Row EQUALS TestCyclingPhase. | |
| 9 | SET the follow | ing objects: preemptCyclingPhase.Table_Row. | Pass/Fail |
| 10 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 11 | GET the follow | ing objects: preemptCyclingPhase.Table_Row. | Pass/Fail |
| 12 | VERIFY preem TestCyclingPh | nptCyclingPhase.Table_Row IS_EQUAL_TO ase. | Pass/Fail |
| 13 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 14 | ASSIGN preen OriginalCyclind | nptCyclingPhase.Table_Row EQUALS pPhase. | |
| 15 | SET the follow | ing objects: preemptCyclingPhase.Table Row. | Pass/Fail |
| 16 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 17 | GET the follow | ing objects: preemptCvclingPhase.Table Row. | Pass/Fail |

| 18 | VERIFY preemptCyclingPhase.Table_Row IS OriginalCyclingPhase. | EQUAL_TO | Pass/Fail |
|------------------|--|-----------------|-----------|
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.8.25 Configure Preempt Cycling Pedestrian Movements

| Test Procedure: | | Configure Preempt Cycling Pedestrian Movements | |
|---------------------|---|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure which phases with pedestrian movements are allowed to cycle during the preempt dwell interval. | |
| Requirement(s): | | • 3.5.2.1.9.1.12.6 Configure Preempt Dwell Overlaps | |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalCyclingPedpreemptCyclingPedTestCyclingPedpreemptCyclingPedMaxPhasesmaxPhases | |
| Pass/Fail Cr | iteria: | The device under test shall pass every verification sto test case to pass the test case. | ep in this |
| Test Step Number | Test Procedu | Test Procedure | |
| 1 | PRE-CONDITI | PRE-CONDITION 'A valid timing plan is loaded'. | |
| 2 | PRE-CONDITI supported by t | PRE-CONDITION 'The user shall know the maximum number of phases supported by the ASC'. | |
| 2.1 | RECO | RD this information as MaxPhases. | |
| 3 | GET the follow | GET the following objects: maxPreempts. | |
| 3.1 | RECO | RD this information as MaxRows. | |
| 4 | ASSIGN Table | Row EQUALS RANDOM (1 TO MaxRows). | |
| 5 | GET the follow | ing objects: preemptCyclingPed.Table_Row. | Pass/Fail |
| 5.1 | RECO | RD this information as OriginalCyclingPed. | |
| 6 | USER-ACTION such as '02'. E | N 'Select a valid value for preemptCyclingPed.Table_Row, ach phase must not be greater than MaxPhases.' | |
| 6.1 | RECO | RD this information as TestCyclingPed. | |
| 7 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 8 | ASSIGN preen | nptCyclingPed.Table_Row EQUALS TestCyclingPed. | |
| 9 | SET the follow | ing objects: preemptCyclingPed.Table_Row. | Pass/Fail |
| 10 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 11 | GET the follow | ing objects: preemptCyclingPed.Table_Row. | Pass/Fail |
| 12 | VERIFY preemptCyclingPed.Table_Row IS_EQUAL_TO TestCyclingPed. | | Pass/Fail |
| 13 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 14 | ASSIGN preen | nptCyclingPed.Table_Row EQUALS OriginalCyclingPed. | |

| 15 | SET the following objects: preemptCyclingPec | d.Table_Row. | Pass/Fail |
|----------------|---|-----------------------|-----------|
| 16 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error". | / Database Status and | |
| 17 | GET the following objects: preemptCyclingPe | d.Table_Row. | Pass/Fail |
| 18 | VERIFY preemptCyclingPed.Table_Row IS_EQUAL_TO OriginalCyclingPed. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.8.26 Configure Preempt Cycling Phases Sequence

| Test Procedure: | | Configure Preempt Cycling Phases Sequence | | |
|---------------------|--|--|----------------------------------|----------------|
| Description: | | This test case verifies that the ASC allows a management station to configure the sequence for phases to cycle during a preempt dwell interval after the minimum green dwell time. | | |
| Requirement(s): | | 3.5.2.1.9.1.12.7 Configure Preempt Cycling Phases Sequence | | |
| Variable(s): | | MaxPreemptsmaxPreemptsMaxSequencesmaxSequencesTable_RowIntOriginalSequencepreemptSequenceNumberTestSequencepreemptSequenceNumber | | umber umber |
| Pass/Fail Criteria: | | The device under test shall pattern test case to pass the test case | ass every verification ste e. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxPreempts. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxPreempts. | | | |
| 2 | GET the following objects: maxSequences. | | Pass/Fail | |
| 2.1 | RECORD this information as MaxSequences. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxPreempts). | | | |
| 3 | GET the follow | ing objects: preemptSequenceNumber.Table_Row. | | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalSe | quence. | |
| 4 | ASSIGN TestS | equence EQUALS RANDOM (0 | TO MaxSequences). | |
| 4.1 | IF Te | stSequence IS_EQUAL_TO Orig | nalSequence. | |
| 4.1.1 | | GOTO step 4. | | ļ |
| 5 | ASSIGN preem TestSequence | nptSequenceNumber.Table_Row | EQUALS | |
| 6 | SET the followi | ng objects: preemptSequenceNu | mber.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: preemptSequenceNu | mber.Table_Row. | Pass/Fail |
| 8 | VERIFY preem TestSequence | ptSequenceNumber.Table_Row | IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN preem OriginalSequer | nptSequenceNumber.Table_Row nce. | EQUALS | |

| 10 | SET the following objects: preemptSequence | Number.Table_Row. | Pass/Fail |
|------------------|---|------------------------|-----------|
| 11 | GET the following objects: preemptSequence | Number.Table_Row. | Pass/Fail |
| 12 | VERIFY preemptSequenceNumber.Table_Ro Sequence. | w IS_EQUAL_TO Original | Pass/Fail |
| Test Procedur | e Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.8.27 Configure Preempt Cycling Overlaps

| Test Procedure: | | Configure Preempt Cycling Overlaps | |
|---------------------|---|--|------------|
| Description: | | This test case verifies that the ASC allows a management station to configure which overlaps are allowed to cycle during the preempt dwell interval. | |
| Requirement(s): | | 3.5.2.1.9.1.12.8 Configure Preempt Cycling Overlaps | |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalCyclingOverlappreemptCyclingOverlapTestCyclingOverlappreemptCyclingOverlapMaxOverlapsmaxOverlaps | |
| Pass/Fail Criteria: | | The device under test shall pass every verification st test case to pass the test case. | ep in this |
| Test Step Number | Test Procedur | Test Procedure | |
| 1 | PRE-CONDITI | ON 'A valid timing plan is loaded'. | |
| 2 | PRE-CONDITI overlaps suppo | PRE-CONDITION 'The user shall know the maximum number of overlaps supported by the ASC'. | |
| 2.1 | RECO | RD this information as MaxOverlaps. | |
| 3 | GET the follow | ing objects: maxPreempts. | Pass/Fail |
| 3.1 | RECO | RD this information as MaxRows. | |
| 4 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 5 | GET the follow | ing objects: preemptCyclingOverlap.Table_Row. | Pass/Fail |
| 5.1 | RECO | RD this information as OriginalCyclingOverlap. | |
| 6 | USER-ACTION preemptDwellC be greater than | USER-ACTION 'Select a valid value for preemptDwellOverlap.Table_Row, such as '01'. Each overlap must not be greater than MaxOverlaps.' | |
| 6.1 | RECO | RD this information as TestCyclingOverlap. | |
| 7 | PERFORM the | Test Procedure "C.3.1.1 Create Database Transaction". | |
| 8 | ASSIGN preen TestCyclingOv | nptCyclingOverlap.Table_Row EQUALS erlap. | |
| 9 | SET the follow | ing objects: preemptCyclingOverlap.Table_Row. | Pass/Fail |
| 10 | PERFORM the Verify Databas | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | |
| 11 | GET the follow | ing objects: preemptCyclingOverlap.Table_Row. | Pass/Fail |

| 12 | VERIFY preemptCyclingOverlap.Table_Row TestCyclingOverlap. | IS_EQUAL_TO | Pass/Fail |
|----------------|--|--------------------------|-----------|
| 13 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 14 | ASSIGN preemptCyclingOverlap.Table_Row OriginalCyclingOverlap. | EQUALS | |
| 15 | SET the following objects: preemptCyclingOv | erlap.Table_Row. | Pass/Fail |
| 16 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 17 | GET the following objects: preemptCyclingOv | erlap.Table_Row. | Pass/Fail |
| 18 | VERIFY preemptCyclingOverlap.Table_Row IS_EQUAL_TO OriginalCyclingOverlap. | | Pass/Fail |
| Test Procedu | re Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.8.28 Configure Preempt Exit Phases

| Test Procedure: | | Configure Preempt Exit Phases | |
|---|--|---|--|
| Description: | | This test cases verifies that the ASC allows a management station to configure the phases to be serviced after a preempt dwell interval. | |
| Requirement(| ment(s): • 3.5.2.1.9.1.13.1 Configure Preempt Exit Phases | | es |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalExitPhasepreemptExitPhaseTestExitPhasepreemptExitPhaseMaxPhasesmaxPhases | |
| Pass/Fail Criteria: The device under test shall pass every verifica test case to pass the test case. | | The device under test shall pass every verification s test case to pass the test case. | tep in this |
| | | | |
| Test Step Number | Test Procedu | re | Results |
| Test Step Number 1 | Test Procedur | r e ON 'A valid timing plan is loaded'. | Results |
| Test StepNumber12 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. | Results |
| Test Step Number 1 2 2.1 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th RECO | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. RD this information as MaxPhases. | Results |
| Test Step Number122.13 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th RECO GET the follow | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. RD this information as MaxPhases. ing objects: maxPreempts. | Results |
| Test Step Number 1 2 2.1 3 3.1 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th RECO GET the follow RECO | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. RD this information as MaxPhases. ing objects: maxPreempts. RD this information as MaxRows. | Results Pass/Fail |
| Test Step Number 1 2 2.1 3 3.1 4 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th RECO GET the follow RECO ASSIGN Table | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. RD this information as MaxPhases. ing objects: maxPreempts. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). | Results Pass/Fail |
| Test Step Number 1 2 2.1 3 3.1 4 5 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th RECO GET the follow RECO ASSIGN Table GET the follow | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. RD this information as MaxPhases. ing objects: maxPreempts. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptExitPhase.Table_Row. | Results Results Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 3.1 4 5 5.1 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th RECO GET the follow RECO ASSIGN Table GET the follow RECO | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. RD this information as MaxPhases. ing objects: maxPreempts. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptExitPhase.Table_Row. RD this information as OriginalExitPhase. | Results Results Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 3.1 4 5 5.1 6 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th RECO GET the follow RECO ASSIGN Table GET the follow RECO USER-ACTION such as '02 06' | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. RD this information as MaxPhases. ing objects: maxPreempts. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptExitPhase.Table_Row. RD this information as OriginalExitPhase. N 'Select a valid value for preemptExitPhase.Table_Row, '. Each phase cannot exceed MaxPhases.' | Results Results Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 3 3.1 4 5 5.1 6 6.1 | Test Procedur PRE-CONDITI PRE-CONDITI supported by th RECO GET the follow RECO ASSIGN Table GET the follow RECO USER-ACTION such as '02 06' RECO | re ON 'A valid timing plan is loaded'. ON 'The user shall know the maximum number of phases ne ASC'. RD this information as MaxPhases. ing objects: maxPreempts. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptExitPhase.Table_Row. RD this information as OriginalExitPhase. N 'Select a valid value for preemptExitPhase.Table_Row, . Each phase cannot exceed MaxPhases.' RD this information as TestExitPhase. | Results Result |
| 8 | ASSIGN preemptExitPhase.Table_Row EQU | ALS TestExitPhase. | |
|------------------|--|--------------------------|-----------|
| 9 | SET the following objects: preemptExitPhase. | .Table_Row. | Pass/Fail |
| 10 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
| 11 | GET the following objects: preemptExitPhase | .Table_Row. | Pass/Fail |
| 12 | VERIFY preemptExitPhase.Table_Row IS_E0 | QUAL_TO TestExitPhase. | Pass/Fail |
| 13 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 14 | ASSIGN preemptExitPhase.Table_Row EQU | ALS OriginalExitPhase. | |
| 15 | SET the following objects: preemptExitPhase. | Table_Row. | Pass/Fail |
| 16 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 17 | GET the following objects: preemptExitPhase.Table_Row. Pa | | Pass/Fail |
| 18 | VERIFY preemptExitPhase.Table_Row IS_EQUAL_TO OriginalExitPhase. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.8.29 Configure Preempt Exit Phase Strategy

| Test Procedure: Configure Preempt Exit Phase Strategy | | | |
|--|----------------|--|-----------------|
| T Description: s u | | This test case verifies that the ASC allows a manager station to configure the ASC to resume a coordinatio upon exiting a preempt. | nent n cycle |
| Requirement(| s): | 3.5.2.1.9.1.13.2 Configure Preempt Exit Phase Strategy | |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalExitTypepreemptExitTypeTestExitTypepreemptExitType | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | |
| Test Step Number | Test Procedur | e | Results |
| 1 | GET the follow | ing objects: maxPreempts. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 3 | GET the follow | ing objects: preemptExitType.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalExitType. | |
| 4 | ASSIGN TestE | xitType EQUALS RANDOM (1 TO 4). | |
| 4.1 | IF Te | stExitType IS_EQUAL_TO OriginalExitType. | |
| 4.1.1 | | GOTO step 4. | |
| 5 | ASSIGN preen | nptExitType.Table_Row _Row EQUALS TestExitType. | |
| 6 | SET the follow | ng objects: preemptExitType.Table_Row. | Pass/Fail |

| 7 | GET the following objects: preemptExitType.1 | Table_Row. | Pass/Fail |
|----------------|--|--------------------------|-----------|
| 8 | VERIFY preemptExitType.Table_Row IS_EQ | UAL_TO TestExitType. | Pass/Fail |
| 9 | ASSIGN preemptExitType.Table_Row _Row | EQUALS OriginalExitType. | |
| 10 | SET the following objects: preemptExitType.T | able_Row. | Pass/Fail |
| 11 | GET the following objects: preemptExitType.Table_Row. | | Pass/Fail |
| 12 | VERIFY preemptExitType.Table_Row IS_EQUAL_TO OriginalExitType. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.8.30 Configure Preempt Exit Priority Levels

| Test Procedure: | | Configure Preempt Exit Priority | Levels | |
|---------------------|--|---|--|-----------------|
| Description: | | This test case verifies that the A station to associate a priority lev Delay Recovery exit strategy du | SC allows a managen vel with a detector for ring a preempt. | ient a Queue |
| Requirement(| Requirement(s): • 3.5.2.1.9.1.13.3 Configure Preempt Exit Priority I | | v Levels | |
| Variable(s): | | MaxPreempts OriginalDetectorWeight TestDetectorWeight MaxVehicleDetectors Selected_Preempt Selected_Detector | maxPreempts preemptDetectorWei preemptDetectorWei maxVehicleDetectors Int Int | ght ght s |
| Pass/Fail Crite | eria: | The device under test shall pass test case to pass the test case. | every verification ste | p in this |
| Test Step Number | Test Procedu | re | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxPreempts | 3. | |
| 1 | GET the follow | ing objects: maxVehicleDetectors. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxVehicleD | etectors. | |
| 2 | ASSIGN Selec | ted_Preempt EQUALS RANDOM (1 | TO MaxPreempts). | |
| 2 | ASSIGN Select MaxVehicleDet | ted_Detector EQUALS RANDOM (1 tectors). | ТО | |
| 3 | GET the follow preemptDetect | ing objects: orWeight.SelectedPreempt.Selected | l.Detector. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalDeted | ctorWeight. | |
| 4 | ASSIGN TestD | etectorWeight EQUALS RANDOM (| (0 TO 1000). | |
| 4.1 | IF Test | tDetectorWeight IS_EQUAL_TO Ori | ginalDetectorWeight. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN preen | nptDetectorWeight.SelectedPreempt | Selected.Detector | |

| 6 | SET the following objects: preemptDetectorWeight.SelectedPreempt.Se | lected.Detector. | Pass/Fail |
|----------------|--|-------------------------|-----------|
| 7 | GET the following objects preemptDetectorWeight.SelectedPreempt.Se | lected.Detector. | Pass/Fail |
| 8 | VERIFY preemptDetectorWeight.SelectedPre IS_EQUAL_TO TestDetectorWeight. | empt.Selected.Detector | Pass/Fail |
| 9 | ASSIGN preemptDetectorWeight.SelectedPre EQUALS OriginalDetectorWeight. | eempt.Selected.Detector | |
| 10 | SET the following objects: preemptDetectorWeight.SelectedPreempt.Se | lected.Detector. | Pass/Fail |
| 11 | GET the following objects: preemptDetectorWeight.SelectedPreempt.Selected.Detector. | | Pass/Fail |
| 12 | VERIFY preemptDetectorWeight.SelectedPre IS_EQUAL_TO OriginalDetectorWeight. | empt.Selected.Detector | Pass/Fail |
| Test Procedui | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.8.31 Configure Preempt Maximum Presence Time

| Test Procedure: | Configure Preempt Maximum Presence Time | | | |
|---|---|---|--|------------------|
| This test case verifies that the ASC allows a managementDescription:station to configure the maximum time a preempt call ma remain active and be considered valid. | | nent may | | |
| Requirement(| s): | • 3.5.2.1.9.1.14.1 Configu Time | re Preempt Maximum F | Presence |
| Variable(s): | | MaxRows Table_Row OriginalMaxPresence TestMaxPresence | maxPreempts Int preemptMaximumPr preemptMaximumPr | esence esence |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | ep in this |
| Test Step Number | Test Procedur | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO M | axRows). | |
| 3 | GET the follow | ing objects: preemptMaximumPres | ence. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalMax | Presence. | |
| 4 | ASSIGN TestM | laxPresence EQUALS RANDOM (0 |) TO 65535). | |
| 4.1 | IF Test | tMaxPresence IS_EQUAL_TO Orig | inalMaxPresence. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN preem | nptMaximumPresence.Table_Row nce | EQUALS | |
| | I ESUMAAT I ESE | | | |

| 7 | GET the following objects: preemptMaximum | Presence.Table_Row. | Pass/Fail |
|----------------|---|---------------------|-----------|
| 8 | VERIFY preemptMaximumPresence.Table_R TestMaxPresence. | low IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN preemptMaximumPresence.Table_F OriginalMaxPresence. | Row EQUALS | |
| 10 | SET the following objects: preemptMaximumF | Presence.Table_Row. | Pass/Fail |
| 11 | GET the following objects: preemptMaximum | Presence.Table_Row. | Pass/Fail |
| 12 | VERIFY preemptMaximumPresence.Table_R OriginalMaxPresence. | low IS_EQUAL_TO | Pass/Fail |
| Test Procedu | re Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.8.32 Configure Preempt Maximum Presence Action

| Test Procedure: | | Configure Preempt Maximum Prescence Action | |
|--|---|--|--|
| Description: | | This test case verifies that the ASC allows a management station to configure if an all-red flash action is executed after the preempt's maximum presence time or if it services exit phases. | |
| Requirement(| s): | • 3.5.2.1.9.1.14.2 Configure Preempt Max Presence Action | |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalPreemptControlpreemptControlTestPreemptControlpreemptControl | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in thi test case to pass the test case. | |
| Test Step Number | Test Procedur | e | Results |
| 1 | GET the follow | ing objects: maxPreempts. | Docc/Eail |
| | - | | Fass/Fall |
| 1.1 | RECO | RD this information as MaxRows. | Fass/Fail |
| 1.1 2 | RECO ASSIGN Table | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). | rass/raii |
| 1.1 2 3 | RECO ASSIGN Table GET the follow | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. | Pass/Fail |
| 1.1 2 3 3.1 | RECO ASSIGN Table GET the follow RECO | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. RD this information as OriginalPreemptControl. | Pass/Fail |
| 1.1 2 3 3.1 4 | RECO ASSIGN Table GET the follow RECO ASSIGN TestP | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. RD this information as OriginalPreemptControl. reemptControl EQUALS OriginalPreemptControl XOR 32. | Pass/Fail |
| 1.1 2 3 3.1 4 5 | RECO ASSIGN Table GET the follow RECO ASSIGN TestP ASSIGN preen | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. RD this information as OriginalPreemptControl. PreemptControl EQUALS OriginalPreemptControl XOR 32. hptControl.Table_Row EQUALS TestPreemptControl. | Pass/Fail |
| 1.1 2 3 3.1 4 5 6 | RECO ASSIGN Table GET the follow RECO ASSIGN TestP ASSIGN preen SET the follow | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. RD this information as OriginalPreemptControl. reemptControl EQUALS OriginalPreemptControl XOR 32. hptControl.Table_Row EQUALS TestPreemptControl. ing objects: preemptControl.Table_Row. | Pass/Fail Pass/Fail Pass/Fail |
| 1.1 2 3 3.1 4 5 6 7 | RECO ASSIGN Table GET the follow RECO ASSIGN TestP ASSIGN preen SET the follow GET the follow | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. RD this information as OriginalPreemptControl. PreemptControl EQUALS OriginalPreemptControl XOR 32. hptControl.Table_Row EQUALS TestPreemptControl. ing objects: preemptControl.Table_Row. ing objects: preemptControl.Table_Row. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 1.1 2 3 3.1 4 5 6 7 8 | RECO ASSIGN Table GET the follow RECO ASSIGN TestP ASSIGN preem SET the follow VERIFY preem TestPreemptCo | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. RD this information as OriginalPreemptControl. PreemptControl EQUALS OriginalPreemptControl XOR 32. hptControl.Table_Row EQUALS TestPreemptControl. ing objects: preemptControl.Table_Row. ing objects: preemptControl.Table_Row. ptControl.Table_Row IS_EQUAL_TO ontrol. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 1.1 2 3 3.1 4 5 6 7 8 9 | RECO ASSIGN Table GET the follow RECO ASSIGN TestP ASSIGN preen SET the follow GET the follow VERIFY preem TestPreemptCo ASSIGN preen | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. RD this information as OriginalPreemptControl. PreemptControl EQUALS OriginalPreemptControl XOR 32. hptControl.Table_Row EQUALS TestPreemptControl. ing objects: preemptControl.Table_Row. ing objects: preemptControl.Table_Row. hptControl.Table_Row IS_EQUAL_TO ontrol. hptControl.Table_Row EQUALS OriginalPreemptControl. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 1.1 2 3 3.1 4 5 6 7 8 9 10 | RECO ASSIGN Table GET the follow RECO ASSIGN TestP ASSIGN preem SET the follow VERIFY preem TestPreemptCo ASSIGN preem SET the follow | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: preemptControl.Table_Row. RD this information as OriginalPreemptControl. PreemptControl EQUALS OriginalPreemptControl XOR 32. hptControl.Table_Row EQUALS TestPreemptControl. ing objects: preemptControl.Table_Row. ing objects: preemptControl.Table_Row. hptControl.Table_Row IS_EQUAL_TO ontrol. hptControl.Table_Row EQUALS OriginalPreemptControl. ing objects: preemptControl.Table_Row. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 12 | VERIFY preemptControl.Table_Row IS_EQUAL_TO OriginalPreemptControl. | | |
|------------------|--|-----------------|-----------|
| Test Procedure | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.8.33 Configure Preempt Gate Description

| Test Procedure: | Configure Preempt Gate Description | | | |
|---------------------|---|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to configure descriptions for preempt gates. | | |
| Requirement(s): | | • 3.5.2.1.9.1.15 Config | ure Preempt Gate Descrip | tion |
| Variable(s): | | MaxRowsmaxPreemptsTable_RowIntOriginalDescriptionpreemptGateDescription | | tion |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRow | /S. | |
| 2 | ASSIGN Table | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: preemptGateDescription.Table_Row. Pass/F | | Pass/Fail | |
| 3.1 | RECO | RECORD this information as OriginalDescription | | |
| 4 | ASSIGN preem | nptGateDescription.Table_Row | EQUALS 'Test'. | |
| 5 | SET the followi | ing objects: preemptGateDescr | iption.Table_Row. | Pass/Fail |
| 6 | GET the follow | ing objects: preemptGateDescr | iption.Table_Row. | Pass/Fail |
| 7 | VERIFY preem | ptGateDescription.Table_Row | IS_EQUAL_TO 'Test'. | Pass/Fail |
| 8 | ASSIGN preem OrignalDescrip | ASSIGN preemptGateDescription.Table_Row EQUALS OrignalDescription. | | |
| 9 | SET the followi | ing objects: preemptGateDescr | iption.Table_Row. | Pass/Fail |
| 10 | GET the follow | ing objects: preemptGateDescr | iption.Table_Row. | Pass/Fail |
| 11 | VERIFY preem OriginalDescrip | ptGateDescription.Table_Row otion. | IS_EQUAL_TO | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.8.34 Determine Maximum Number of Preempts

| Test Procedure: | Determine Maximum Number of Preempts |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to determine the maximum number of preempts |

| | supported by ASC and verifies that the ASC support minimum number of preempts required by the user. | | the | |
|--|--|--|-------------------------------------|--|
| Requirement(s): | | • 3.5.2.1.9.2 Determin | e Maximum Number of Pre | empts |
| Variable(s): | | UserMinPreempts | maxPreempts | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedu | st Procedure Resul | | Results |
| | USER-ACTION 'Determine the number of preempts required as specified in FR ID 3.5.2.1.9.2 of the PRL.' | | | |
| 1 | specified in FR | ID 3.5.2.1.9.2 of the PRL.' | | |
| 1 | specified in FR RECO | Dotation and the first of pro- ID 3.5.2.1.9.2 of the PRL.' RD this information as UserMir | Preempts. | |
| 1 1.1 2 | specified in FR RECO GET the follow | RD this information as UserMir ing objects: maxPreempts. | nPreempts. | Pass/Fail |
| 1 1.1 2 3 | specified in FR RECO GET the follow VERIFY maxP | RD this information as UserMir ing objects: maxPreempts. reempts IS_NOT_LESS_THAN | Preempts. | Pass/Fail Pass/Fail |
| 1 1.1 2 3 Test Procedure | specified in FR RECO GET the follow VERIFY maxPi e Results | TD 3.5.2.1.9.2 of the PRL.' RD this information as UserMir ing objects: maxPreempts. reempts IS_NOT_LESS_THAN | NUserMinPreempts. | Pass/Fail Pass/Fail |
| 1 1.1 2 3 Test Procedure Tested By: | specified in FR RECO GET the follow VERIFY maxPi e Results | TD 3.5.2.1.9.2 of the PRL.' RD this information as UserMir ing objects: maxPreempts. reempts IS_NOT_LESS_THAN | N UserMinPreempts. | Pass/Fail Pass/Fail Pass/Fail |

C.3.9 Communication and IO

C.3.9.1 Determine Number of ASC Communications Ports

| Test Procedure: | | Determine Number of ASC | Communications Ports | |
|---------------------|---|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the number of physical communications ports on the device. | | |
| Requirement(s): | | 3.5.1.2.2.1 Determine Number of ASC Communications Ports | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure Resu | | Results | |
| 1 | GET the following objects: maxCommPorts | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

| Test Procedure: | | Retrieve Maximum Number | of I/O Maps | |
|---------------------|-------------------|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of I/O maps supported. | | |
| Requirement(s): | | • 3.5.2.1.11.2.1 Retrie | ve Maximum Number I/O M | aps |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure Re | | Results | |
| 1 G | GET the followi | ng objects: asclOmaxMaps. | | Pass/Fail |
| Test Procedure R | Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure Not | tes: | | | |

C.3.9.2 Retrieve Maximum Number of I/O Maps

C.3.9.3 Retrieve Maximum Number of I/O Map Inputs

| Test Procedure: | Retrieve Maximum Number of I/O Map Inputs | | of I/O Map Inputs | |
|---------------------|---|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of I/O map inputs supported. | | |
| Requirement(s): | | • 3.5.2.1.11.2.2 Retrie Inputs | ve Maximum Number I/O M | aps |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure Re | | Results | |
| 1 | GET the follow | ing objects: asclOmapMaxInpu | its. | Pass/Fail |
| Test Procedure | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.9.4 Retrieve Maximum Number of I/O Map Outputs

| Test Procedure: | | Retrieve Maximum Number | of I/O Map Outputs | |
|---------------------|--|---|--|-----------------|
| Description: | | This test case verifies that station to determine the ma supported. | the ASC allows a managen iximum number of I/O map | nent outputs |
| Requirement(s): | | • 3.5.2.1.11.2.3 Retrie Outputs | ve Maximum Number I/O M | aps |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure F | | Results | |
| 1 | GET the following objects: ascIOmapMaxOutputs. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | | |

C.3.9.5 Retrieve I/O Mapping Activate Conditions

| Test Procedure: | Retrieve I/O Mapping Activate Conditions | | ate Conditions | |
|---|--|--|---|---------------|
| Description: | | This test case verifies that station to determine the co takes effect. | the ASC allows a managen nditions for when a new I/C | nent D map |
| Requirement(| s): | • 3.5.2.1.11.2.4 Retrieve I/O Mapping Activate Conditions | | onditions |
| Variable(s): | | | | |
| Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this | | |
| Test Step Number | Test Procedu | re | | Results |
| 1 | GET the follow | ing objects: asclOactivateRequ | uirement | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.9.6 Retrieve I/O Mapping Input Functions

| Test Procedure: | | Retrieve I/O Mapping Input | Functions | |
|---------------------|--|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the I/O Mapping input Functions. | | |
| Requirement(s): | | • 3.5.2.1.11.2.5 Retrieve I/O Mapping Input Functions | | |
| Variable(s): | | MaxRows Table_Row | asclOmapMaxInputFunctions Int | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedu | e | | Results |
| 1 | GET the follow | ing objects: asclOmapMaxInpu | utFunctions. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 2.1 | GET the following objects: asclOinputMaxFuncIndex.Table_Row, asclOinputFunctionName.Table_Row. | | | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.9.7 Retrieve I/O Mapping Output Functions

| Test Procedure: | | Retrieve I/O Mapping | Output Functions | |
|---------------------|---|--|---|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the I/O Mapping output Functions. | | |
| Requirement(s | • 3.5.2.1.11.2.6 Retrieve I/O Mapping Output Function | | ctions | |
| | | MaxRows ascIOmapMaxOutputFunctio | | |
| Variable(s): | | Table Dave | ns | |
| | | I able_Row | Int | |
| Pass/Fail Criteria: | | The device under test test case to pass the | t shall pass every verification ste test case. | p in this |
| Test Step Number | Test Procedur | re | | Results |
| 1 | GET the follow | ing objects: asclOmapM | axOutputFunctions. | Pass/Fail |
| 1.1 | RECO | RD this information as N | laxRows. | |

| 2 | FOR Table_Row from 1 to MaxRows. | | |
|----------------|--|-----------------|-----------|
| 2.1 | GET the following objects: asclOoutputMaxFuncIndex.Table_Row, asclOoutputFunctionName.Table_Row. | | |
| Test Procedur | e Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.9.8 Retrieve I/O Map Input Device Pin Status

| Test Procedure: | | Retrieve I/O Map Input Devi | ce Pin Status | |
|--|--|---|---|-------------------------------|
| Description: | | This test case verifies that station to view the status o | the ASC allows a managen f I/O map input device pin. | nent |
| Requirement(| s): | • 3.5.2.1.11.2.7 Retrieve I/O Map Input Device Pin Status | | |
| Variable(s): | | MaxMapsasclOmaxMapsMaxInputsasclOmapMaxInputsMapIntInputInt | | i |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: asclOmaxMaps. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxMar | 20 | |
| 0 | | | JS. | |
| Ζ | GET the follow | ing objects: asclOmapMaxInpu | its. | Pass/Fail |
| 2.1 | GET the follow RECO | ing objects: asclOmapMaxInpu RD this information as MaxInpu | its. | Pass/Fail |
| 2 2.1 2 | GET the follow RECO FOR Map from | ing objects: asclOmapMaxInpu RD this information as MaxInpu 1 to MaxMaps. | its. uts. | Pass/Fail |
| 2 2.1 2 2.1 | GET the follow RECO FOR Map from FOR | ing objects: asclOmapMaxInpu RD this information as MaxInpu 1 to MaxMaps. Input from 1 to MaxInputs. | its. its. | Pass/Fail |
| 2 2.1 2 2.1 2.1.1 | GET the follow RECO FOR Map from FOR asclOinputMap asclOinputMap | ing objects: ascIOmapMaxInpu RD this information as MaxInpu 1 to MaxMaps. Input from 1 to MaxInputs. GET the following objects: DevPinDesc.Map.Input, DevPinStatus.Map.Input. | its. | Pass/Fail Pass/Fail |
| 2 2.1 2.1 2.1 2.1.1 Test Procedur | GET the follow RECO FOR Map from FOR asclOinputMap asclOinputMap e Results | ing objects: asclOmapMaxInpu RD this information as MaxInpu 1 to MaxMaps. Input from 1 to MaxInputs. GET the following objects: DevPinDesc.Map.Input, DevPinStatus.Map.Input. | its. | Pass/Fail Pass/Fail |
| 2 2.1 2.1 2.1 2.1.1 Test Procedur Tested By: | GET the follow RECO FOR Map from FOR asclOinputMap asclOinputMap e Results | ing objects: ascIOmapMaxInpu RD this information as MaxInpu 1 to MaxMaps. Input from 1 to MaxInputs. GET the following objects: DevPinDesc.Map.Input, DevPinStatus.Map.Input. | Date Tested: | Pass/Fail Pass/Fail Pass/Fail |

C.3.9.9 Retrieve I/O Map Output Device Pin Status

| Test Procedure: | Retrieve I/O Map Output Device Pin Status |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to view the status of I/O map output device pin. |

| Requirement(s): | | • 3.5.2.1.11.2.8 Retriev | ve I/O Map Output Device I | Pin Status |
|---------------------|--|---|--|------------|
| Variable(s): | | MaxMaps MaxOutputs Map Output | asclOmaxMaps asclOmapMaxOutpu Int Int | ts |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedu | e | | Results |
| 1 | GET the follow | ing objects: asclOmaxMaps. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxMap |)S. | |
| 2 | GET the follow | ing objects: asclOmapMaxOut | outs. | Pass/Fail |
| 2.1 | RECO | RD this information as MaxOut | puts. | |
| 2 | FOR Map from | 1 to MaxMaps. | | |
| 2.1 | FOR | Output from 1 to MaxOutputs. | | |
| 2.1.1 | GET the following objects: asclOoutputMapDevPinDesc.Map.Output, asclOoutputMapDevPinStatus.Map.Output. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | | |

C.3.9.10 Determine Serial Bus 1 Device Present

| Test Procedure: | | Determine Serial Bus 1 Device Present | |
|---|--|---|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify if a device is present for a Serial Bus 1 address in an ATC cabinet. | |
| Requirement | (s): | • 3.5.2.1.12.1.1 Determine Serial Bus 1 Device Present | |
| Variable(s): | | MaxRowsmaxSIUPort1AddressesTable_RowIntUserMinSerialBus1AddressemaxSIUPort1Addressesss | |
| Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedu | re | Results |
| 1 | USER-ACTION Determine the number of channels required as specified in FR ID 3.5.2.1.12.1.1 of the PRL. | | |
| 1.1 | RECO | ORD this information as UserMinSerialBus1Addresses | |
| 2 | GET the follow | ing objects: maxSIUPort1Addresses. | Pass/Fail |

| 2.1 | RECORD this information as MaxRows. | | |
|------------------------|---|-----------------|-----------|
| 3 | VERIFY MaxRows IS_NOT_LESS_THAN UserMinSerialBus1Addresses | | |
| 4 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 5 | GET the following objects: siuport1Number.Table_Row, siuPort1DevicePresent.Table_Row. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.9.11 Determine TS2 Port 1 Device Present

| Test Procedure: | | Determine TS2 Port 1 Devic | e Present | |
|------------------------------------|--|---|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify if a device is present for a TS2 Port 1. | | |
| Requirement(s): | | • 3.5.2.1.12.2.1 Determine TS2 Port 1 Device Present | | |
| Variable(s): | | MaxRows maxPort1Addresses Table_Row Int | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedu | - .e | | Results |
| 1 | GET the follow | ing objects: maxPort1Addresse | es. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | vs. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 GET the following port1DevicePre | | ing objects: port1Number.Table esent.Table_Row. | e_Row, | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.9.12 Enable/Disable TS2 Port 1 Frame 40 Messages

| Test Procedure: | Enable/Disable TS2 Port 1 Frame 40 Messages |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to enable or disable TS2 Port 1 Frame 40 messages. |
| Requirement(s): | 3.5.2.1.12.2.2 Enable/Disable TS2 Port 1 Frame 40 Messages |

| Variable(s): | | MaxRows Table_Row OriginalFrame40Enable TestFrame40Enable | maxPort1Addresses Int port1Frame40Enable port1Frame40Enable | 6 9 |
|---------------------|---------------------------------|--|--|------------|
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | bass every verification sto se. | ep in this |
| Test Step Number | Test Procedur | re | | Results |
| 1 | GET the follow | ing objects: maxPort1Addresse | S. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRow | S. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO | D MaxRows). | |
| 3 | GET the follow | ing objects: port1Frame40Enab | le.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalF | rame40Enable. | |
| 4 | IF OriginalFran | ne40Enable IS_EQUAL_TO 0. | | |
| 4.1 | ASSIG | N TestFrame40Enable EQUAL | S 1. | |
| 4.2 | GOTO | step 6. | | |
| 5 | ASSIGN TestF | rame40Enable EQUALS 0. | | |
| 6 | ASSIGN port1 TestFrame40E | Frame40Enable.Table_Row EQ nable. | UALS | |
| 7 | SET the follow | ing objects: port1Frame40Enab | le.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: port1Frame40Enab | le.Table_Row. | Pass/Fail |
| 9 | VERIFY port1F TestFrame40E | Frame40Enable.Table_Row IS_ nable. | EQUAL_TO | Pass/Fail |
| 10 | ASSIGN port18 OriginalFrame | Frame40Enable.Table_Row EQ 40Enable. | UALS | |
| 11 | SET the follow | ing objects: port1Frame40Enab | e.Table_Row. | Pass/Fail |
| 12 | GET the follow | ing objects: port1Frame40Enab | le.Table_Row. | Pass/Fail |
| 13 | VERIFY port1F OriginalFrame4 | rame40Enable.Table_Row IS_ 40Enable. | EQUAL_TO | Pass/Fail |
| Test Procedu | re Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.9.13 Monitor TS2 Port 1 Status

| Test Procedure: | Monitor TS2 Port 1 Status | |
|--------------------|---|--------------------------|
| Description: | This test case verifies that the ASC allows a management station to identify if a device is present for a TS2 Port 1 address. | |
| Requirement(s): | • 3.5.2.2.9.1 Monitor TS2 Port 1 Status | |
| Variable(s): | MaxRows Table_Row | maxPort1Addresses Int |

| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | p in this | |
|-------------------------|--|---|-----------|-----------|--|
| Test Step Number | Test Procedure | | Results | | |
| 1 | GET the following objects: maxPort1Addresses. | | Pass/Fail | | |
| 1.1 | RECO | RECORD this information as MaxRows. | | | |
| 2 | FOR Table_Ro | FOR Table_Row from 1 to MaxRows. | | | |
| 3 | GET the following objects: port1Frame40Enable.Table_Row. | | | Pass/Fail | |
| | | Test Procedure Results | | | |
| Tested By: Date Tested: | | | Pass/Fail | | |
| Test Procedure N | Test Procedure Notes: | | | | |

C.3.9.14 Monitor TS2 Port 1 Fault Frame

| Test Procedure: | | Monitor TS2 Port 1 Fault Fr | ame | |
|---------------------|------------------------------|---|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to identify if Frame 40 message to the device is enabled for a TS2 Port 1 address. | | |
| Requirement(s): | | • 3.5.2.2.9.2 Monitor TS2 Port 1 Fault Frame | | |
| Variable(s): | | MaxRows Table_Row | maxTimebaseAscActions Int | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPort1Address | es. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRo | WS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 3 | GET the follow port1Frame40E | ing objects: port1Number.Tabl Enable.Table_Row. | e_Row, | Pass/Fail |
| | - | Test Procedure Results | ; ; | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.9.15 Monitor ATC Serial Bus 1 Status

| Test Procedure: | Monitor ATC Serial Bus 1 Status |
|--------------------|---------------------------------|
|--------------------|---------------------------------|

| Description: | | This test case verifies that the ASC allows a management station to identify the communications status of devices with a Serial Bus 1 address. | | |
|---|--|--|--|-------------------------------|
| Requirement(s): | | • 3.5.2.2.9.3 Monitor ATC Serial Bus 1 Status | | |
| Variable(s): | | MaxRows maxSIUPort1Addresses Table_Row Int | | ses |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | p in this |
| Test Step | Test Procedure | | Results | |
| number | | | | |
| 1 | GET the follow | ing objects: maxSIUPort1Addr | esses. | Pass/Fail |
| 1 1.1 | GET the follow RECO | ing objects: maxSIUPort1Addr RD this information as MaxRov | esses. NS. | Pass/Fail |
| 1 1.1 2 | GET the follow RECO FOR Table_Ro | ing objects: maxSIUPort1Addr RD this information as MaxRov ow from 1 to MaxRows. | esses. NS. | Pass/Fail |
| 1 1.1 2 3 | GET the follow RECO FOR Table_RC GET the follow siuPort1Status | ing objects: maxSIUPort1Addr RD this information as MaxRov ow from 1 to MaxRows. ing objects: siuPort1DevicePre .Table_Row. | esses. ws. esent.Table_Row, | Pass/Fail Pass/Fail |
| 1 1.1 2 3 | GET the follow RECO FOR Table_Ro GET the follow siuPort1Status | ing objects: maxSIUPort1Addr RD this information as MaxRov ow from 1 to MaxRows. ing objects: siuPort1DevicePre .Table_Row. Test Procedure Results | esses. ws. esent.Table_Row, | Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 Tested By: | GET the follow RECO FOR Table_Ro GET the follow siuPort1Status | ing objects: maxSIUPort1Addr RD this information as MaxRov ow from 1 to MaxRows. ing objects: siuPort1DevicePre .Table_Row. Test Procedure Results | esses. ws. esent.Table_Row, Date Tested: | Pass/Fail Pass/Fail Pass/Fail |

C.3.9.16 Monitor Signal Monitoring Unit Channel Voltage

| Test Procedure: | | Monitor Signal Monitoring Unit Channel Voltage | |
|---------------------|---|---|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view voltage levels in the SMU. | |
| Requirement(| ement(s): • 3.5.2.2.10.1 Monitor Signal Monitoring Unit Channel Voltage | | annel |
| Variable(s): | | MaxChannelsmaxChannelsTable_Index1IntTable_Index2Int | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | p in this |
| Test Step Number | | Test Procedure | Results |
| 1 | GET the follow | ing objects: maxChannels. | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxChannels. | |
| 2 | FOR Table_Index1 from 1 to MaxRows. | | |
| 2.1 | FOR Table | FOR Table_Index1 from 1 to 3. | |
| 2.1.1 | GET the following objects: ascSmuVoltage.Table_Index1.Table_Index2 | | Pass/Fail |
| | | Test Procedure Results | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.9.17 Monitor Signal Monitoring Unit Channel Current

| Test Procedure: | | Monitor Signal Monitoring | Unit Channel Current | |
|---------------------|-------------------------------|---|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to view current levels in the SMU. | | |
| Requirement(s): | | 3.5.2.2.10.2 Monitor Signal Monitoring Unit Channel Current | | annel |
| Variable(s): | | MaxChannels Table_Index1 Table_Index2 | maxChannels Int Int | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test case | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxChannels. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxCha | annels. | |
| 2 | FOR Table_Inc | lex1 from 1 to MaxRows. | | |
| 2.1 | FOR Table_Index1 from 1 to 3. | | | |
| 211 | GET the following objects: | | | Pass/Fail |
| 2.1.1 | ascSmuCurren | t.Table_Index1.Table_Index2 | | Fass/Faii |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.9.18 Enable/Disable ECLA Communications

| Test Procedure: | Enable/Disable ECLA Communic | cations |
|--------------------|---|--|
| Description: | This test case verifies that the A station to enable or disable com | SC allows a management munication with an ECLA |
| Requirement(s): | 3.5.1.10.1.1 Enable ECLA 3.5.1.10.1.2 Disable ECLA | A Communications A Communications |
| Variable(s): | OriginalSetting TestSetting | eclaCommEnable eclaCommEnable |

| Pass/Fail Crite | Criteria: The device under test shall pass every verification sterest case to pass the test case. | | ep in this | |
|---------------------|---|--|-------------------------|-----------|
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the followi | ing objects: eclaCommEnable | | Pass/Fail |
| 1.1 | RECO | RD this information as Original | Setting. | |
| 2 | IF OriginalSetti | ng IS_EQUAL_TO 0. | | |
| 2.1 | ASSIG | N TestSetting EQUALS 1. | | |
| 2.2 | GO ste | p 4. | | |
| 3 | ASSIGN TestO | utputControl EQUALS 0. | | |
| 4 | ASSIGN specialFunctionOutputControl.Table_Row EQUALS TestOutputControl. | | | |
| 5 | SET the followi | ng objects: specialFunctionOu | tputControl.Table_Row. | Pass/Fail |
| 6 | GET the follow | ing objects: specialFunctionOu | itputControl.Table_Row. | Pass/Fail |
| 7 | VERIFY specia TestOutputCon | IFunctionOutputControl.Table_ trol. | _Row IS_EQUAL_TO | Pass/Fail |
| 8 | ASSIGN specia OriginalOutput | alFunctionOutputControl.Table | _Row EQUALS | |
| 9 | SET the followi | ng objects: specialFunctionOu | tputControl.Table_Row. | Pass/Fail |
| 10 | GET the followi | ing objects: specialFunctionOu | itputControl.Table_Row. | Pass/Fail |
| 11 | VERIFY specialFunctionOutputControl.Table_Row IS_EQUAL_TO OriginalOutputControl. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.10 Compressed Block Objects

C.3.10.1 Configure Phase Data Block

| Test Procedure: | | Configure Phase Data Block | |
|---|---|--|------------------------------|
| Description: | This test case verifies that the ASC allows a management station to use the block object to configure phase sets. If the ASC does not support multiple phase sets, it configures the phase table instead of the phase set table. | | าent s. If the ′es the |
| 3.5.2.1.14.1.1 Configure Block Object Get Contro Phase Data 3.5.2.1.14.2 Monitor Block Error – Error-causing Element | | trol - ıg Data | |
| Variable(s): | | OriginalBlockascBlockDataTestBlockascBlockDataMaxPhasesmaxPhasesPhaseOctet String | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification ste test case to pass the test case. This case fails if the te procedure step 11.1.1 is executed because the SET fu Step 11 resulted in an error. | p in this st nction in |
| Test Step Number | Test Procedur | e | Results |
| 1 | PRE-CONDITION 'The user shall know the maximum number of phases supported by the ASC' | | |
| 1.1 | RECORD this information as MaxPhases. | | |
| 2 | ASSIGN Phase EQUALS RANDOM (1 TO MaxPhases). | | |
| 3 | ASSIGN ascBlockGetControl EQUALS '00 00 Phase 01 01 01'. | | |
| 3.1 | NOTE 'ascBlockDataType = 0 (standard block)'. | | |
| 3.2 | NOTE 'ascBlockDataID = 0 (phase data)'. | | |
| 3.3 | NOTE 'ascBlockIndex1 = Phase in hexadecimal format'. | | |
| 3.4 | NOTE 'ascBlockQuantity1 = 1 (number of phases = 1)'. | | |
| 3.5 | NOTE | 'ascBlockIndex2 = 1 (phase set = 1)'. | |
| 3.6 | NOTE | 'ascBlockQuantity2 = 1 (number of phase sets = 1)'. | |
| 4 | SET the followi | ng objects: ascBlockGetControl. | Pass/Fail |
| 5 | GET the follow | ing objects: ascBlockErrorStatus. | Pass/Fail |
| 6 | VERIFY ascBlo | ockErrorStatus IS_EQUAL_TO 0. | Pass/Fail |
| 7 | GET the follow | ing objects: ascBlockData. | Pass/Fail |
| 7.1 | RECO | RD this information as OriginalBlock. | |
| 8 | USER-ACTION 'Generate an octet string with the modified valid values for AscPhaseBlockData. Valid values must pass a consistency check.' | | |
| 8.1 | RECORD this information as TestBlock. | | |
| 9 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 10 | ASSIGN ascBl | ockData EQUALS TestBlock. | |
| 11 | SET the followi | ng objects: ascBlockData. | Pass/Fail |
| 11.1 | IF ERI | ROR INDEX IS_NOT_EQUAL_TO 0. | |
| 11.1.1 | G | ET the following objects: ascBlockErrorStatus. | Pass/Fail |
| 11.1.2 | E | XIT | |

| 12 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
|----------------|--|--------------------------|-----------|
| 13 | GET the following objects: ascBlockData. | | |
| 14 | VERIFY ascBlockData IS_EQUAL_TO TestB | lock. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 16 | ASSIGN ascBlockData EQUALS OriginalBlock. | | |
| 17 | SET the following objects: ascBlockData. | | Pass/Fail |
| 18 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 19 | GET the following objects: ascBlockData. | | Pass/Fail |
| 20 | VERIFY ascBlockData IS_EQUAL_TO OriginalBlock. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.10.2 Configure Vehicle Detector Data Block

| Test Procedure: | re: Configure Vehicle Detector Data Block | | |
|---|--|---|---------|
| This test case verifies that the ASC allows a management station to use the block object to configure vehicle detect sets. If the ASC does not support multiple vehicle detect it configures the vehicle detector table instead of the vehicle | | nent etector tector sets, vehicle | |
| Requirement(| s): | 3.5.2.1.14.1.2 Configure Block Object Get Control – Vehicle Detector Data | |
| Variable(s): | | OriginalBlockascBlockDataTestBlockascBlockDataMaxVehicleDetectorsmaxVehicleDetectorsVehicleDetectorOctet String | |
| Pass/Fail Crit | eria: | ia: The device under test shall pass every verification step in the test case to pass the test case. This case fails if the test procedure step 12.1.1 is executed because the SET function Step 12 resulted in an error. | |
| Test Step Number | Test Procedu | e. | Results |
| 1 | PRE-CONDITI | ON 'The ASC supports vehicle detectors'. | |
| 2 | PRE-CONDITI detectors supp | ON 'The user shall know the maximum number of vehicle orted by the ASC'. | |
| 2.1 | RECO | RD this information as MaxVehicleDetectors. | |
| 3 | ASSIGN VehicleDetector EQUALS RANDOM (1 TO MaxVehicleDetectors). | | |
| 4 | ASSIGN ascBl | ockGetControl EQUALS '00 01 VehicleDetector 01 01 01'. | |
| 4.1 | NOTE | 'ascBlockDataType = 0 (standard block)'. | |
| 4.2 | NOTE | 'ascBlockDataID = 1 (vehicle detector data)'. | |
| 4.3 | NOTE format'. | 'ascBlockIndex1 = VehicleDetector in hexadecimal | |

| 4.4 | NOTE 'ascBlockQuantity1 = 1 (numbe | er of vehicle detectors = 1)'. | |
|------------------|--|---|-------------|
| 4.5 | NOTE 'ascBlockIndex2 = 1 (vehicle de | etector set = 1)'. | |
| 4.6 | NOTE 'ascBlockQuantity2 = 1 (numbe | er of vehicle detector sets = | |
| 1.0 | 1)'. | | |
| 5 | SET the following objects: ascBlockGetContro | ol. | Pass/Fail |
| 6 | GET the following objects: ascBlockErrorStatu | IS. | Pass/Fail |
| 7 | VERIFY ascBlockErrorStatus IS_EQUAL_TO | 0. | Pass/Fail |
| 8 | GET the following objects: ascBlockData. | | Pass/Fail |
| 8.1 | RECORD this information as Original | Block. | |
| 9 | USER-ACTION 'Generate an octet string with for AscVehDetectorBlock. Valid values must p | the modified valid values bass a consistency check.' | |
| 9.1 | RECORD this information as TestBloo | ck. | |
| 10 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 11 | ASSIGN ascBlockData EQUALS TestBlock. | | |
| 12 | SET the following objects: ascBlockData. | | Pass/Fail |
| 12.1 | IF ERROR INDEX IS_NOT_EQUAL_ | TO 0. | |
| 12.1.1 | GET the following objects: ascB | lockErrorStatus. | Pass/Fail |
| 12.1.2 | EXIT | | |
| 13 | PERFORM the Test Procedure 'C.3.1.2 Verify | Database Status and | |
| 14 | GET the following objects: ascBlockData | | Pass/Fail |
| 15 | VEDIEV appPloakData IS EQUAL TO ToatPloak | | Pass/Fail |
| 16 | PERFORM the Test Procedure 'C 3.1.1 Creat | e Database Transaction' | 1 433/1 41 |
| 17 | ASSIGN ascBlockData EOUALS OriginalBloc | | |
| 18 | SET the following objects: ascBlockData | ASSIGN ASCHOURDAID EQUALS UNUITAIDIUCK. | |
| 10 | DEPENDENT the Test Procedure (C 3.1.2 Verify | Database Status and | 1 a33/1 ali |
| 19 | Verify Database Error'. | Database Status and | |
| 20 | GET the following objects: ascBlockData. | | Pass/Fail |
| 21 | VERIFY ascBlockData IS_EQUAL_TO Origina | alBlock. | Pass/Fail |
| Test Procedure | e Results | | |
| Tested By: | | Date | Dass/Fail |
| resteu by. | | Tested: | Fass/Fall |
| Test Procedure N | lotes: | | |

C.3.10.3 Configure Pedestrian Detector Data Block

| Test Procedure: | Configure Pedestrian Detecto | Configure Pedestrian Detector Data Block | | |
|--------------------|---|--|--|--|
| Description: | This test case verifies that the station to use the block object detectors. | e ASC allows a management ct to configure pedestrian | | |
| Requirement(s): | 3.5.2.1.14.1.3 Configu Pedestrian Detector D | 3.5.2.1.14.1.3 Configure Block Object Get Control – Pedestrian Detector Data | | |
| Variable(s): | OriginalBlock TestBlock MaxPedestrianDetectors PedestrianDetector | ascBlockData ascBlockData maxPedestrianDetectors Octet String | | |

| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step procedure step 12.1.1 is executed because the SET fur Step 12 resulted in an error. | | p in this st nction in | |
|---------------------|--|---|--|-----------|
| Test Step Number | Test Procedur | e | | Results |
| 1 | PRE-CONDITION | ON 'The ASC supports pedestr | ian detectors'. | |
| 2 | PRE-CONDITIO | ON 'The user shall know the ma ectors supported by the ASC'. | aximum number of | |
| 2.1 | RECO | RD this information as MaxPed | estrianDetectors. | |
| 3 | ASSIGN Pedes MaxPedestrian | strianDetector EQUALS RAND(Detectors). | OM (1 TO | |
| 4 | ASSIGN ascBl | ockGetControl EQUALS '00 02 | PedestrianDetector 01'. | |
| 4.1 | NOTE | 'ascBlockDataType = 0 (standa | ard block)'. | |
| 4.2 | NOTE | 'ascBlockDataID = 2 (pedestria | n detector data)'. | |
| 4.3 | NOTE format'. | 'ascBlockIndex1 = PedestrianD | etector in hexadecimal | |
| 4.4 | NOTE 1)'. | 'ascBlockQuantity1 = 1 (numbe | er of pedestrian detectors = | |
| 5 | SET the followi | ing objects: ascBlockGetContro | l. | Pass/Fail |
| 6 | GET the follow | ing objects: ascBlockErrorStatu | IS. | Pass/Fail |
| 7 | VERIFY ascBlo | ockErrorStatus IS_EQUAL_TO | 0. | Pass/Fail |
| 8 | GET the follow | ing objects: ascBlockData. | | Pass/Fail |
| 8.1 | RECO | RD this information as Original | Block. | |
| 9 | USER-ACTION for AscPedDete | I 'Generate an octet string with ectorBlock. Valid values must p | the modified valid values ass a consistency check.' | |
| 9.1 | RECO | RD this information as TestBloo | xk. | |
| 10 | PERFORM the | Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 11 | ASSIGN ascBl | ockData EQUALS TestBlock. | | |
| 12 | SET the followi | ing objects: ascBlockData. | | Pass/Fail |
| 12.1 | IF ERI | ROR INDEX IS_NOT_EQUAL_ | TO 0. | |
| 12.1.1 | G | GET the following objects: ascB | lockErrorStatus. | Pass/Fail |
| 12.1.2 | E | TIX | | |
| 13 | PERFORM the Verify Database | e Test Procedure 'C.3.1.2 Verify e Error'. | Database Status and | |
| 14 | GET the follow | ing objects: ascBlockData. | | Pass/Fail |
| 15 | VERIFY ascBlo | ockData IS_EQUAL_TO TestBl | ock. | Pass/Fail |
| 16 | PERFORM the | Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 17 | ASSIGN ascBl | ockData EQUALS OriginalBloc | K. | |
| 18 | SET the followi | ing objects: ascBlockData. | | Pass/Fail |
| 19 | PERFORM the Verify Database | e Test Procedure 'C.3.1.2 Verify e Error'. | Database Status and | |
| 20 | GET the follow | ing objects: ascBlockData. | | Pass/Fail |
| 21 | VERIFY ascBlo | ockData IS_EQUAL_TO Origina | alBlock. | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.10.4 Configure Pattern Data Block

| Test Procedure: | | Configure Pattern Data Block | |
|---------------------|--|---|-----------|
| Description: | | This test case verifies that the ASC allows a management station to use the block object to configure timing patterns. | |
| Requirement(| s): | 3.5.2.1.14.1.4 Configure Block Object Get Control – Pattern Data | |
| Variable(s): | | OriginalBlockascBlockDataTestBlockascBlockDataMaxPatternsmaxPatternsPatternOctet String | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. This case fails if the test procedure step 12.1.1 is executed because the SET function in Step 12 resulted in an error. | |
| Test Step Number | Test Procedur | e | Results |
| 1 | PRE-CONDITI | ON 'The ASC supports timing patterns'. | |
| 2 | PRE-CONDITI patterns suppo | ON 'The user shall know the maximum number of rted by the ASC'. | |
| 2.1 | RECO | RD this information as MaxPatterns. | |
| 3 | ASSIGN Patter | n EQUALS RANDOM (1 TO MaxPatterns). | |
| 4 | ASSIGN ascBlockGetControl EQUALS '00 03 Pattern 01'. | | |
| 4.1 | NOTE 'ascBlockDataType = 0 (standard block)'. | | |
| 4.2 | NOTE 'ascBlockDataID = 3 (pattern data)'. | | |
| 4.3 | NOTE 'ascBlockIndex1 = Pattern in hexadecimal format'. | | |
| 4.4 | NOTE 'ascBlockQuantity1 = 1 (number of patterns = 1)'. | | |
| 5 | SET the followi | ng objects: ascBlockGetControl. | Pass/Fail |
| 6 | GET the follow | ing objects: ascBlockErrorStatus. | Pass/Fail |
| 7 | VERIFY ascBlo | ockErrorStatus IS_EQUAL_TO 0. | Pass/Fail |
| 8 | GET the follow | ing objects: ascBlockData. | Pass/Fail |
| 8.1 | RECO | RD this information as OriginalBlock. | |
| 9 | USER-ACTION for AscPattern | I 'Generate an octet string with the modified valid values Block. Valid values must pass a consistency check.' | |
| 9.1 | RECO | RD this information as TestBlock. | |
| 10 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 11 | ASSIGN ascBlockData EQUALS TestBlock. | | |
| 12 | SET the following objects: ascBlockData. Pass/Fa | | Pass/Fail |
| 12.1 | IF ER | ROR INDEX IS_NOT_EQUAL_TO 0. | |
| 12.1.1 | Ģ | ET the following objects: ascBlockErrorStatus. | Pass/Fail |
| 12.1.2 | E | XIT | |
| 13 | PERFORM the Verify Databas | Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 14 | GET the follow | ing objects: ascBlockData. | Pass/Fail |
| 15 | VERIFY ascBlo | ockData IS_EQUAL_TO TestBlock. | Pass/Fail |
| 16 | PERFORM the | Test Procedure 'C.3.1.1 Create Database Transaction'. | |

| 17 | ASSIGN ascBlockData EQUALS OriginalBlock. | | |
|-------------------------|--|--|-----------|
| 18 | SET the following objects: ascBlockData. | | Pass/Fail |
| 19 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 20 | GET the following objects: ascBlockData. | | Pass/Fail |
| 21 | VERIFY ascBlockData IS_EQUAL_TO OriginalBlock. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: Date Tested: | | | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.10.5 Configure Split Data Block

| Test Procedure: | | Configure Split Data Block | |
|---|--|---|---|
| Description: | | This test case verifies that the ASC allows a management station to use the block object to configure splits. | |
| Requirement(s): | | 3.5.2.1.14.1.5 Configure Block Object Get Control – Split Data | |
| Variable(s): | | OriginalBlockascBlockDataTestBlockascBlockDataMaxPhasesmaxPhasesMaxSplitsmaxSplitsPhaseOctet StringSplitOctet String | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step i test case to pass the test case. This case fails if the test procedure step 14.1.1 is executed because the SET function Step 14 resulted in an error. | | on step in this the test ET function in |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITI | ON 'The ASC supports splits'. | |
| 2 | PRE-CONDITI | ON 'The user shall know the maximum number of | |
| 0.4 | patterns suppo | rted by the ASC'. | |
| 2.1 | patterns suppo RECO | rted by the ASC'. RD this information as MaxPatterns. | |
| 2.1 | patterns suppo RECO PRE-CONDITI supported by t | RD this information as MaxPatterns. ON 'The user shall know the maximum number of spline ne ASC'. | ts |
| 2.1 3 3.1 | patterns suppo RECO PRE-CONDITI supported by th RECO | rted by the ASC'. RD this information as MaxPatterns. ON 'The user shall know the maximum number of spline ASC'. RD this information as MaxSplits. | ts |
| 2.1 3 3.1 4 | patterns suppo RECO PRE-CONDITI supported by th RECO ASSIGN Patter | The user shall know the maximum humber of RD this information as MaxPatterns. ON 'The user shall know the maximum number of splin ne ASC'. RD this information as MaxSplits. rn EQUALS RANDOM (1 TO MaxPatterns). | ts |
| 2.1 3 3.1 4 5 | patterns suppo RECO PRE-CONDITI supported by the RECO ASSIGN Patter ASSIGN Split | rted by the ASC'. RD this information as MaxPatterns. ON 'The user shall know the maximum number of spline ASC'. RD this information as MaxSplits. rn EQUALS RANDOM (1 TO MaxPatterns). EQUALS RANDOM (1 TO MaxSplits). | ts |
| 2.1 3 3.1 4 5 6 | patterns suppo RECO PRE-CONDITI supported by th RECO ASSIGN Patter ASSIGN Split F ASSIGN ascBI | The user shall know the maximum humber of RD this information as MaxPatterns. ON 'The user shall know the maximum number of splin the ASC'. RD this information as MaxSplits. rn EQUALS RANDOM (1 TO MaxPatterns). EQUALS RANDOM (1 TO MaxSplits). ockGetControl EQUALS '00 04 Phase 01 Split 01'. | ts |
| 2.1 3 3.1 4 5 6 6.1 | patterns suppo RECO PRE-CONDITI supported by th RECO ASSIGN Patter ASSIGN Split I ASSIGN ascBI NOTE | RD this information as MaxPatterns. ON 'The user shall know the maximum number of spline ASC'. RD this information as MaxSplits. RD this information as MaxSplits. RD EQUALS RANDOM (1 TO MaxPatterns). EQUALS RANDOM (1 TO MaxSplits). ockGetControl EQUALS '00 04 Phase 01 Split 01'. 'ascBlockDataType = 0 (standard block)'. | ts |
| 2.1 3 3.1 4 5 6 6 6.1 6.2 | patterns suppo RECO PRE-CONDITI supported by the RECO ASSIGN Patter ASSIGN Split for ASSIGN ascBI NOTE NOTE | A The dser shall know the maximum humber of arted by the ASC'. RD this information as MaxPatterns. ON 'The user shall know the maximum number of spline ASC'. RD this information as MaxSplits. rn EQUALS RANDOM (1 TO MaxPatterns). EQUALS RANDOM (1 TO MaxSplits). ockGetControl EQUALS '00 04 Phase 01 Split 01'. 'ascBlockDataType = 0 (standard block)'. 'ascBlockDataID = 4 (split data)'. | ts |
| 2.1 3 3.1 4 5 6 6 6.1 6.2 6.3 | patterns suppo RECO PRE-CONDITI supported by th RECO ASSIGN Patter ASSIGN Split I ASSIGN ascBl NOTE NOTE NOTE | A rife dser shall know the maximum humber of rifed by the ASC'. RD this information as MaxPatterns. ON 'The user shall know the maximum number of split the ASC'. RD this information as MaxSplits. rn EQUALS RANDOM (1 TO MaxPatterns). EQUALS RANDOM (1 TO MaxSplits). ockGetControl EQUALS '00 04 Phase 01 Split 01'. 'ascBlockDataType = 0 (standard block)'. 'ascBlockDataID = 4 (split data)'. 'ascBlockIndex1 = Phase in hexadecimal format'. | ts |
| $ \begin{array}{r} 2.1 \\ 3 \\ 3.1 \\ 4 \\ 5 \\ 6 \\ 6.1 \\ 6.2 \\ 6.3 \\ 6.4 \\ \end{array} $ | patterns suppo RECO PRE-CONDITI supported by the RECO ASSIGN Patter ASSIGN Split I ASSIGN ascBI NOTE NOTE NOTE NOTE | A The dser shall know the maximum humber of splinted by the ASC'. RD this information as MaxPatterns. ON 'The user shall know the maximum number of splinted ASC'. RD this information as MaxSplits. rn EQUALS RANDOM (1 TO MaxPatterns). EQUALS RANDOM (1 TO MaxSplits). ockGetControl EQUALS '00 04 Phase 01 Split 01'. 'ascBlockDataType = 0 (standard block)'. 'ascBlockIndex1 = Phase in hexadecimal format'. 'ascBlockQuantity1 = 1 (number of phases = 1)'. | ts |
| $ \begin{array}{r} 2.1 \\ 3 \\ 3.1 \\ 4 \\ 5 \\ 6 \\ 6.1 \\ 6.2 \\ 6.3 \\ 6.4 \\ 6.5 \\ \end{array} $ | patterns suppo RECO PRE-CONDITI supported by the RECO ASSIGN Patter ASSIGN Split R ASSIGN ascBI NOTE NOTE NOTE NOTE | A The dser shall know the maximum humber of arted by the ASC'. RD this information as MaxPatterns. ON 'The user shall know the maximum number of spline ASC'. RD this information as MaxSplits. The EQUALS RANDOM (1 TO MaxPatterns). EQUALS RANDOM (1 TO MaxSplits). OckGetControl EQUALS '00 04 Phase 01 Split 01'. 'ascBlockDataType = 0 (standard block)'. 'ascBlockDataID = 4 (split data)'. 'ascBlockIndex1 = Phase in hexadecimal format'. 'ascBlockIndex2 = Split in hexadecimal format'. | ts |

| 7 | SET the following objects: ascBlockGetContro | ol. | Pass/Fail |
|------------------|--|--|-----------|
| 8 | GET the following objects: ascBlockErrorStatu | IS. | Pass/Fail |
| 9 | VERIFY ascBlockErrorStatus IS_EQUAL_TO | 0. | Pass/Fail |
| 10 | GET the following objects: ascBlockData. | | Pass/Fail |
| 10.1 | RECORD this information as Original | Block. | |
| 11 | USER-ACTION 'Generate an octet string with for AscSplitBlock. Valid values must pass a co | the modified valid values onsistency check.' | |
| 11.1 | RECORD this information as TestBloo | ck. | |
| 12 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 13 | ASSIGN ascBlockData EQUALS TestBlock. | | |
| 14 | SET the following objects: ascBlockData. | | Pass/Fail |
| 14.1 | IF ERROR INDEX IS_NOT_EQUAL_ | _TO 0. | |
| 14.1.1 | GET the following objects: ascB | lockErrorStatus. | Pass/Fail |
| 14.1.2 | EXIT | | |
| 16 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 17 | GET the following objects: ascBlockData. | | Pass/Fail |
| 18 | VERIFY ascBlockData IS_EQUAL_TO TestBlock. | | Pass/Fail |
| 19 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 20 | ASSIGN ascBlockData EQUALS OriginalBloc | k. | |
| 21 | SET the following objects: ascBlockData. | | Pass/Fail |
| 22 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 23 | GET the following objects: ascBlockData. | | Pass/Fail |
| 24 | VERIFY ascBlockData IS_EQUAL_TO Origina | alBlock. | Pass/Fail |
| Test Procedure | e Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.10.6 Configure Overlap Data Block

| Test Procedure: | Configure Overlap Data Block | |
|---------------------|--|--|
| Description: | This test case verifies that the A station to use the block object t ASC does not support multiple configures the overlap table ins | ASC allows a management o configure overlap sets. If the overlap detector sets, it tead of the overlap set table. |
| Requirement(s): | • 3.5.2.1.14.1.6 Configure Overlap Data | Block Object Get Control – |
| Variable(s): | OriginalBlock TestBlock MaxOverlaps Overlap | ascBlockData ascBlockData maxOverlaps Octet String |
| Pass/Fail Criteria: | The device under test shall pass every verification step in this test case to pass the test case. This case fails if the test | |

| procedure step 12.1.1 is executed because the SET function in Step 12 resulted in an error. | | | |
|---|---|--|-----------|
| Test Step Number | Test Procedure | | Results |
| 1 | PRE-CONDITION 'The ASC supports overlap | s'. | |
| 2 | PRE-CONDITION 'The user shall know the m overlaps supported by the ASC'. | aximum number of | |
| 2.1 | RECORD this information as MaxOve | erlaps. | |
| 3 | ASSIGN Overlap EQUALS RANDOM (1 TO M | /laxOverlaps). | |
| 4 | ASSIGN ascBlockGetControl EQUALS '00 06 | Overlap 01 01 01'. | |
| 4.1 | NOTE 'ascBlockDataType = 0 (standa | ard block)'. | |
| 4.2 | NOTE 'ascBlockDataID = 5 (overlap o | data)'. | |
| 4.3 | NOTE 'ascBlockIndex1 = Overlap in h | nexadecimal format'. | |
| 4.4 | NOTE 'ascBlockQuantity1 = 1 (number | er of overlaps = 1)'. | |
| 4.5 | NOTE 'ascBlockIndex2 = 1 (overlap s | set = 1)'. | |
| 4.6 | NOTE 'ascBlockQuantity2 = 1 (numbe | er of overlap sets = 1)'. | |
| 5 | SET the following objects: ascBlockGetContro | ol. | Pass/Fail |
| 6 | GET the following objects: ascBlockErrorStatu | JS. | Pass/Fail |
| 7 | VERIFY ascBlockErrorStatus IS_EQUAL_TO | 0. | Pass/Fail |
| 8 | GET the following objects: ascBlockData. | | Pass/Fail |
| 8.1 | RECORD this information as Original | Block. | |
| 9 | USER-ACTION 'Generate an octet string with for AscOverlapBlock. Valid values must pass | the modified valid values a consistency check.' | |
| 9.1 | RECORD this information as TestBlock. | | |
| 10 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 11 | ASSIGN ascBlockData EQUALS TestBlock. | | |
| 12 | SET the following objects: ascBlockData. Pass/Fail | | |
| 12.1 | IF ERROR INDEX IS NOT EQUAL TO 0. | | |
| 12.1.1 | GET the following objects: ascB | llockErrorStatus. | Pass/Fail |
| 12.1.2 | EXIT | | |
| 13 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
| 14 | GET the following objects: ascBlockData. | | Pass/Fail |
| 15 | VERIFY ascBlockData IS_EQUAL_TO TestBl | lock. | Pass/Fail |
| 16 | PERFORM the Test Procedure 'C.3.1.1 Creat | te Database Transaction'. | |
| 17 | ASSIGN ascBlockData EQUALS OriginalBloc | k. | |
| 18 | SET the following objects: ascBlockData. | | Pass/Fail |
| 19 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
| 20 | GET the following objects: ascBlockData. Pass/Fail | | Pass/Fail |
| 21 | VERIFY ascBlockData IS_EQUAL_TO Origin | alBlock. | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | |

C.3.10.7 Configure Preempt Data Block

| Test Procedure: | | Configure Preempt Data Block | |
|---------------------|---|---|---------------------------------|
| Description: | | This test case verifies that the ASC allows a manager station to use the block object to configure preempts | nent |
| Requirement | t(s): | 3.5.2.1.14.1.7 Configure Block Object Get Con Preempt Data | trol - |
| Variable(s): | | OriginalBlockascBlockDataTestBlockascBlockDataMaxPreemptsmaxPreemptsPreemptOctet String | |
| Pass/Fail Cri | Ass/Fail Criteria: The device under test shall pass every verification step in t test case to pass the test case. This case fails if the test procedure step 12.1.1 is executed because the SET functio Step 12 resulted in an error. | | ep in this est inction in |
| Test Step Number | Test Procedu | re | Results |
| 1 | PRE-CONDITI | ON 'The ASC supports preempts'. | |
| 2 | PRE-CONDITI preempts supp | ON 'The user shall know the maximum number of orted by the ASC'. | |
| 2.1 | RECO | RECORD this information as MaxPreempts. | |
| 3 | ASSIGN Preer | ASSIGN Preempt EQUALS RANDOM (1 TO MaxPreempts). | |
| 4 | ASSIGN ascBlockGetControl EQUALS '00 06 Preempt 01'. | | |
| 4.1 | NOTE 'ascBlockDataType = 0 (standard block)'. | | |
| 4.2 | NOTE 'ascBlockDataID = 6 (preempt data)'. | | |
| 4.3 | NOTE 'ascBlockIndex1 = Preempt in hexadecimal format'. | | |
| 4.4 | NOTE | NOTE 'ascBlockQuantity1 = 1 (number of preempts = 1)'. | |
| 5 | SET the following objects: ascBlockGetControl. Pass/Fai | | Pass/Fail |
| 6 | GET the follow | ing objects: ascBlockErrorStatus. | Pass/Fail |
| 7 | VERIFY ascBl | ockErrorStatus IS_EQUAL_TO 0. | Pass/Fail |
| 8 | GET the follow | ing objects: ascBlockData. | Pass/Fail |
| 8.1 | RECO | RD this information as OriginalBlock. | |
| 9 | USER-ACTION for AscPreemp | V 'Generate an octet string with the modified valid values tBlock. Valid values must pass a consistency check.' | |
| 9.1 | RECO | RD this information as TestBlock. | |
| 10 | PERFORM the | e Test Procedure 'C.3.1.1 Create Database Transaction'. | |
| 11 | ASSIGN ascBl | ASSIGN ascBlockData EQUALS TestBlock. | |
| 12 | SET the follow | SET the following objects: ascBlockData. Pass/Fail | |
| 12.1 | IF ER | ROR INDEX IS_NOT_EQUAL_TO 0. | |
| 12.1.1 | (| GET the following objects: ascBlockErrorStatus. Pass/Fail | |
| 12.1.2 | E | EXIT | |
| 13 | PERFORM the Verify Databas | e Test Procedure 'C.3.1.2 Verify Database Status and e Error'. | |
| 14 | GET the follow | ing objects: ascBlockData. | Pass/Fail |
| 15 | VERIFY ascBl | ockData IS EQUAL TO TestBlock. | Pass/Fail |

| 16 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
|-------------------------|--|----|-----------|
| 17 | ASSIGN ascBlockData EQUALS OriginalBloc | k. | |
| 18 | SET the following objects: ascBlockData. | | Pass/Fail |
| 19 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 20 | GET the following objects: ascBlockData. Pass/Fa | | Pass/Fail |
| 21 | VERIFY ascBlockData IS_EQUAL_TO OriginalBlock. | | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: Date Pass/Fa | | | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.10.8 Configure Sequence Data Block

| Test Procedure: | | Configure Sequence Data Block | | |
|------------------------------------|--|---|--|------------------------------|
| Description: | | This test case verifies that the ASC station to use the block object to co data. | allows a managem onfigure phase seq | ent uence |
| Requirement(| s): | 3.5.2.1.14.1.8 Configure Bloc Sequence | ck Object Get Cont | rol – |
| Variable(s): | | OriginalBlockasTestBlockasMaxSequencesmaMaxRingsmaRingOcSequenceOc | cBlockData cBlockData axSequences axRings ctet String ctet String | |
| T Pass/Fail Criteria: p s | | The device under test shall pass events test case to pass the test case. This procedure step 13.1.1 is executed b Step 13 resulted in an error. | ery verification ste case fails if the te ecause the SET fu | p in this st nction in |
| Test Step Number | Test Procedu | e | | Results |
| 1 | PRE-CONDITI sequences sup | ON 'The user shall know the maximum ı ported by the ASC'. | number of | |
| 1.1 | RECO | RD this information as MaxSequences. | | |
| 2 | PRE-CONDITI supported by the support of the support | PRE-CONDITION 'The user shall know the maximum number of rings supported by the ASC'. | | |
| 2.1 | RECO | RD this information as MaxRings. | | |
| 3 | ASSIGN Ring | ASSIGN Ring EQUALS RANDOM (1 TO MaxRings). | | |
| 4 | ASSIGN Sequence EQUALS RANDOM (1 TO MaxSequence). | | | |
| 5 | ASSIGN ascBl | ASSIGN ascBlockGetControl EQUALS '00 07 Ring 01 Sequence 01'. | | |
| 5.1 | NOTE | 'ascBlockDataType = 0 (standard block |)'. | |
| 5.2 | NOTE 'ascBlockDataID = 7 (sequence data)'. | | | |
| | NUTE | acceletiere addite | NOTE 'ascBlockIndex1 = Ring in hexadecimal format'. | |
| 5.3 | NOTE | ascBlockIndex1 = Ring in hexadecimal | format'. | |
| 5.3 5.4 | NOTE NOTE NOTE | 'ascBlockIndex1 = Ring in hexadecimal 'ascBlockQuantity1 = 1 (number of rings | format'. s = 1)'. | |

| 5.6 | NOTE 'ascBlockQuantity2 = 1 (numbe | er of sequences = 1)'. | |
|-------------------------|---|---|-----------|
| 6 | SET the following objects: ascBlockGetContro | bl. | Pass/Fail |
| 7 | GET the following objects: ascBlockErrorStatu | JS. | Pass/Fail |
| 8 | VERIFY ascBlockErrorStatus IS_EQUAL_TO | 0. | Pass/Fail |
| 9 | GET the following objects: ascBlockData. | | Pass/Fail |
| 9.1 | RECORD this information as Original | Block. | |
| 10 | USER-ACTION 'Generate an octet string with for AscSequenceBlock. Valid values must pas | the modified valid values as a consistency check.' | |
| 10.1 | RECORD this information as TestBlo | ck. | |
| 11 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 12 | ASSIGN ascBlockData EQUALS TestBlock. | | |
| 13 | SET the following objects: ascBlockData. | | Pass/Fail |
| 13.1 | IF ERROR INDEX IS_NOT_EQUAL | _TO 0. | |
| 13.1.1 | GET the following objects: ascBlockErrorStatus. Pass/Fail | | Pass/Fail |
| 13.1.2 | EXIT | | |
| 14 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | / Database Status and | |
| 15 | GET the following objects: ascBlockData. | | Pass/Fail |
| 16 | VERIFY ascBlockData IS_EQUAL_TO TestB | lock. | Pass/Fail |
| 17 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 18 | ASSIGN ascBlockData EQUALS OriginalBlock. | | |
| 19 | SET the following objects: ascBlockData. Pass/Fa | | Pass/Fail |
| 20 | PERFORM the Test Procedure 'C.3.1.2 Verify Database Status and Verify Database Error'. | | |
| 21 | GET the following objects: ascBlockData. | | Pass/Fail |
| 22 | VERIFY ascBlockData IS_EQUAL_TO Origin | alBlock. | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.10.9 Configure Channel Data Block

| Test Procedure: | Configure Channel Data Block | |
|---------------------|---|---|
| Description: | This test case verifies that the ASC allows a management station to use the block object to configure channels. | |
| Requirement(s): | 3.5.2.1.14.1.9 Configure Block Object Get Control – Channels Data | |
| Variable(s): | OriginalBlock TestBlock MaxChannels Channel | ascBlockData ascBlockData maxChannels Octer String |
| Pass/Fail Criteria: | The device under test shall pa test case to pass the test case | ass every verification step in this e. This case fails if the test |

| procedure step 11.1.1 is executed because the SET function in Step 11 resulted in an error. | | | |
|---|---|---|-----------|
| Test Step Number | Test Procedure | | Results |
| 1 | PRE-CONDITION 'The user shall know the m channels supported by the ASC'. | aximum number of | |
| 1.1 | RECORD this information as MaxCha | nnels. | |
| 2 | ASSIGN Channel EQUALS RANDOM (1 TO I | MaxChannel). | |
| 3 | ASSIGN ascBlockGetControl EQUALS '00 08 | Channel 01'. | |
| 3.1 | NOTE 'ascBlockDataType = 0 (standa | ard block)'. | |
| 3.2 | NOTE 'ascBlockDataID = 8 (channel | data)'. | |
| 3.3 | NOTE 'ascBlockIndex1 = Channel in | hexadecimal format'. | |
| 3.4 | NOTE 'ascBlockQuantity1 = 1 (number | er of channels = 1)'. | |
| 4 | SET the following objects: ascBlockGetContro | ol. | Pass/Fail |
| 5 | GET the following objects: ascBlockErrorState | JS. | Pass/Fail |
| 6 | VERIFY ascBlockErrorStatus IS_EQUAL_TO | 0. | Pass/Fail |
| 7 | GET the following objects: ascBlockData. | | Pass/Fail |
| 7.1 | RECORD this information as Original | Block. | |
| 8 | USER-ACTION 'Generate an octet string with for AscChannelBlock. Valid values must pass | the modified valid values a consistency check.' | |
| 8.1 | RECORD this information as TestBlo | ck. | |
| 9 | PERFORM the Test Procedure 'C.3.1.1 Create Database Transaction'. | | |
| 10 | ASSIGN ascBlockData EQUALS TestBlock. | | |
| 11 | SET the following objects: ascBlockData. Pass/Fail | | Pass/Fail |
| 11.1 | IF ERROR INDEX IS_NOT_EQUAL_TO 0. | | |
| 11.1.1 | GET the following objects: ascB | lockErrorStatus. | Pass/Fail |
| 11.1.2 | EXIT | | |
| 12 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | v Database Status and | |
| 13 | GET the following objects: ascBlockData. | | Pass/Fail |
| 14 | VERIFY ascBlockData IS_EQUAL_TO TestBl | ock. | Pass/Fail |
| 15 | PERFORM the Test Procedure 'C.3.1.1 Creat | e Database Transaction'. | |
| 16 | ASSIGN ascBlockData EQUALS OriginalBloc | k. | |
| 17 | SET the following objects: ascBlockData. Pass/Fail | | Pass/Fail |
| 18 | PERFORM the Test Procedure 'C.3.1.2 Verify Verify Database Error'. | v Database Status and | |
| 19 | GET the following objects: ascBlockData. | | Pass/Fail |
| 20 | VERIFY ascBlockData IS_EQUAL_TO Origin | alBlock. | Pass/Fail |
| Test Procedur | e Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.11 Monitor Signal Operations

C.3.11.1 Monitor External Alarm States

| Test Procedure: | Monitor External Alarm States | | | |
|-------------------------|---|---|--|-------------------|
| Description: | | This test case verifies that station to view active state | the ASC allows a managen of all user configured alarr | nent n inputs. |
| Requirement(| (s): • 3.5.2.2.1.1 Monitor External Alarm States | | | |
| Variable(s): | | MaxRows maxAlarmGroups Table_Row Int | | |
| Pass/Fail Crite | I Criteria: The device under test shall pass every verification step in this test case to pass the test case. | | p in this | |
| Test Step Number | Test Procedure Re: | | Results | |
| 1 | GET the follow | ing objects: maxAlarmGroups. | | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: alarmGroupState.Table_Row. | | Pass/Fail | |
| Test Procedure | e Results | | | |
| Tested By: Date Tested: | | Pass/Fail | | |
| Test Procedure N | lotes: | | | |

C.3.11.2 Monitor Unit Level Alarms - Short

| Test Procedure: | Monitor Unit Level Alarms - Short |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management this test case verifies that the ASC allows to view alarms if any of the following conditions are active: • External Alarm Active • Preempt Active • Flash Active • Offset Transitioning • Local Override • Detector Fault • Stop Time Active |
| Requirement(s): | 3.5.2.2.1.2 Monitor External Alarm Active 3.5.2.2.1.3 Monitor Flash Active 3.5.2.2.1.4 Monitor Local Override 3.5.2.2.1.5 Monitor Coordination Alarm 3.5.2.2.1.6 Monitor Detector Fault |

| Tested By: | Notes: | Date Tested: | Pass/Fail |
|---------------------|---------------------|--|------------|
| Test Proced | ure Results | | |
| 1 | GET the follow | GET the following objects: shortAlarmStatusV4. Pass/F | |
| Test Step Number | Test Procedure Resu | | Results |
| Pass/Fail Cı | riteria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Variable(s): | | | |
| | | 3.5.2.2.2.3 Monitor Offset Transitioning 3.5.2.2.2.4 Monitor Priority Call Active | |
| | | 3.5.2.2.1.7 Monitor Stop Time Active 3.5.2.2.2.2 Monitor Preempt Active | |

C.3.11.3 Monitor Unit Level Alarms

| Test Procedure: | Monitor Unit Level Alarms |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management this test case verifies that the ASC allows to view alarms if any of the following conditions are active: • Cycle Fault • Coordination Fault • Coordination Fail • Cycle Fail • Free Mode • Coordination Active • Cabinet IO Link Error • SMU Communications Error • Preempt Maximum Presence • ECLA Active • RSU Interface Watchdog • CV Certificate Faults |
| Requirement(s): | 3.5.2.2.1.8 Monitor Cycle Fault 3.5.2.2.1.9 Monitor Coordination Fault 3.5.2.2.1.10 Monitor Coordination Fail Alarm 3.5.2.2.1.10 Monitor Cycle Fail Alarm 3.5.2.2.1.11 Monitor Cabinet IO Link Alarm 3.5.2.2.1.12 Monitor Cabinet IO Link Alarm 3.5.2.2.1.13 Monitor SMU Communications Error 3.5.2.2.1.14 Monitor Preempt Maximum Presence Alarm 3.5.2.2.2.5 Monitor Local Free Status 3.5.2.2.2.6 Monitor Coordination Active 3.5.2.2.7 Monitor ECLA Active 3.5.4.1.6 Monitor Interface Watchdog Alarm 3.5.4.3.4 Monitor CV Certificate Fauts |
| Variable(s): | |

| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | | |
|---------------------|---|---|-----------------|-----------|--|
| Test Step Number | Test Procedure | | Results | | |
| 1 | GET the following objects: unitAlarmStatus. | | Pass/Fail | | |
| Test Procedur | Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure | lotes: | | | | |

C.3.11.4 Monitor Flash Status

| Test Procedure: | | Monitor Flash Status | | |
|---------------------|--|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify why the ASC is in Flash. | | |
| Requirement(s): | | • 3.5.2.2.1.3 Monitor Flash Status | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | PERFORM the Test Procedure 'C.3.11.2 Monitor Unit Level Alarms - Short'. | | | |
| 2 GET the followi | | ing objects: unitFlashStatus. | | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.11.5 Monitor Unit Control Status

| Test Procedure: | Monitor Unit Control Status |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to monitor the control mode for pattern, flash, or free. |
| Requirement(s): | • 3.5.2.2.1 Monitor Unit Control Status |

| Variable(s): | | | | |
|-----------------------|---|---|-----------|--|
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: unitControlStatus. | | Pass/Fail | |
| | Test Procedure Results | | | |
| Tested By: Date Pa | | | Pass/Fail | |
| Test Procedure Notes: | | | | |

C.3.11.6 Monitor Local Free Status

| Test Procedure: | | Monitor Local Free Status | | |
|------------------------|--|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify why an ASC is running in Free Mode. | | |
| Requirement(s): | | • 3.5.2.2.5 Monitor Local Free Status | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedur | Э | | Results |
| 1 | PERFORM the Test Procedure 'C.3.11.3 Monitor Unit Level Alarms'. | | | |
| 2 GET the follow | | ing objects: localFreeStatus. | | Pass/Fail |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.11.7 Monitor Current Pattern Status

| Test Procedure: | Monitor Current Pattern Status |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to identify the current coordination pattern/mode |
| Requirement(s): | • 3.5.2.2.8.1 Monitor Current Pattern Status |

| Variable(s): | | | | |
|------------------------|--|---|-----------------|-----------|
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: coordPatternStatus. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | | | |

C.3.11.8 Monitor Current Pattern Command Source

| Test Procedure: | | Monitor Current Pattern Co | mmand Source | |
|-----------------------|--|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify why the currently active pattern is active. | | |
| Requirement(s): | | • 3.5.2.2.8.2 Monitor | r Current Pattern Comman | d Source |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: coordPatternSource. | | e. | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | | | |

C.3.11.9 Monitor Current Pattern Fault Status

| Test Procedure: | Monitor Current Pattern Fault Status |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to identify why the current pattern is invalid if it is invalid. |
| Requirement(s): | • 3.5.2.2.8.3 Monitor Current Pattern Fault Status |
| | |

| Variable(s): | | | | |
|-------------------------|---|---|-----------|---------|
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure R | | | Results |
| 1 | GET the following objects: coordPatternFaultStatus. | | Pass/Fail | |
| Test Procedur | e Results | | | |
| Tested By: Date Tested: | | Pass/Fail | | |
| Test Procedure N | lotes: | | | |

C.3.11.10 Monitor Coordination Cycle Status

| Test Procedure: | | Monitor Coordination Cycle | e Status | |
|------------------------|--|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify how many seconds are remaining in the local cycle. | | |
| Requirement(s): | | • 3.5.2.2.9.1 Monitor | r Coordination Cycle Status | 5 |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure Resu | | Results | |
| 1 | GET the following objects: coordCycleStatus. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure Not | es: | | | |

C.3.11.11 Monitor Coordination Synchronization Status

| Test Procedure: | Monitor Coordination Synchronization Status |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to identify the current position of the master cycle in seconds. |
| Requirement(s): | 3.5.2.2.2.9.2 Monitor Coordination Synchronization Status |

| Variable(s): | | | | | | |
|------------------------|---|---|-----------------|-----------|--|--|
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | | | |
| Test Step Number | Test Procedure | | Results | | | |
| 1 | GET the following objects: coordSyncStatus. | | Pass/Fail | | | |
| Test Procedure Results | | | | | | |
| Tested By: | | | Date Tested: | Pass/Fail | | |
| Test Procedure N | lotes: | | | | | |

C.3.11.12 Monitor Current Offset

| Test Procedure: | | Monitor Coordination Synchronization Status | | | |
|------------------------|--|---|-----------------|-----------|--|
| Description: | | This test case verifies that the ASC allows a management station to identify the current coordination offset in effect. | | | |
| Requirement(s): | | • 3.5.2.2.9.3 Monitor Current Offset | | | |
| Variable(s): | | | | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | | |
| Test Step Number | Test Procedure Resu | | Results | | |
| 1 | GET the following objects: coordCurrentOffset. | | Pass/Fail | | |
| Test Procedure Results | | | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure Notes: | | | | | |

C.3.11.13 Monitor Active Red Phases

| Test Procedure: | Monitor Active Red Phases | |
|--------------------|--|--|
| Description: | This test case verifies that the ASC allows a management station to identify the phases with active RED indications. | |
| Requirement(s): | 3.5.2.2.3.1 Monitor Active Red Phases | |
-

| Variable(s): | | MaxRows Table_Row | maxPhaseGroups Int | |
|--|--|--|-------------------------------------|------------|
| Pass/Fail Criteria: The device test case t | | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxPhaseGroups. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: phaseStatusGroupReds.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | ; ; | - |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

| C.3.11.14 Monitor Active Ye | ellow Phases |
|-----------------------------|--------------|
|-----------------------------|--------------|

| Test Procedure: | est Procedure: Monitor Active Yellow Phases | | | |
|---------------------|--|---|--|--------------------|
| Description: | | This test case verifies tha station to identify the pha | t the ASC allows a managen ses with active YELLOW inc | nent lications. |
| Requirement(| • 3.5.2.2.3.2 Monitor Active Yellow Phases | | Active Yellow Phases | |
| Variable(s): | MaxRows maxPhaseGroups Table_Row Int | | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxPhaseGroup | S. | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | FOR Table_Ro | ow from 1 to MaxRows. | | |
| 2.1 | GET the following objects: phaseStatusGroupYellows.Table_Row. | | Pass/Fail | |
| | | Test Procedure Resul | ts | - |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.11.15 Monitor Active Green Phases

| Test Procedure: | est Procedure: Monitor Active Green Phases | | | |
|---------------------|---|---|--|-----------|
| Description: | | This test case verifies that station to identify the phase | This test case verifies that the ASC allows a management station to identify the phases with active GREEN indications. | |
| Requirement(| (s): • 3.5.2.2.3.3 Monitor A | | Active Green Phases | |
| Variable(s): | MaxRows Table_Row | | maxPhaseGroups Int | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxPhaseGroups. | | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 2.1 | GET the following objects: phaseStatusGroupGreens.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.11.16 Monitor Active Don't Walk Phases

| Test Procedure: | | Monitor Active Don't Walk Phases | | |
|---------------------|--|---|--------------------------------|---------|
| Description: | | This test case verifies that the ASC allows a management station to identify the phases with active DON'T WALK indications. | | |
| Requirement(s): | | • 3.5.2.2.3.4 Mo | nitor Active Don't Walk Phases | |
| Variable(s): | | MaxRows maxPhaseGroups Table_Row Int | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure Results | | Results |

| 1 | GET the following objects: maxPhaseGroups. | Pass/Fail | |
|--|--|-----------------|-----------|
| 1.1 | RECORD this information as MaxRov | WS. | |
| 2 | FOR Table_Row from 1 to MaxRows. | | |
| 2.1 GET the following objects: phaseStatusGroupDontWalks.Table Row. | | | Pass/Fail |
| | | | |
| | Test Procedure Results | i | |
| Tested By: | Test Procedure Results | Date Tested: | Pass/Fail |

C.3.11.17 Monitor Active Pedestrian Clearance Phases

| Test Procedure: | Monitor Active Pedestrian Clearance Phases | | | |
|---------------------|--|--|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify the phases with active flashing DON'T WALK indications. | | |
| Requirement(| Requirement(s): • 3.5.2.2.3.5 Monitor Active Pedestrian Clearance Ph | | e Phases | |
| Variable(s): | : MaxRows maxPhaseGroups Table_Row Int | | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxPhaseGroups | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | NS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 2.1 | GET the following objects: phaseStatusGroupPedClears.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.11.18 Monitor Active Walk Phases

| Test Procedure: Monitor Active Walk Phases | | | S | |
|---|---|---|--------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify the phases with active WALK indications. | | |
| Requirement(| t(s): • 3.5.2.2.3.6 Monitor Active Walk Phases | | Active Walk Phases | |
| Variable(s): | MaxRows maxPhaseGroups Table_Row Int | | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure | | |
| 1 | GET the following objects: maxPhaseGroups. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: phaseStatusGroupWalks.Table_Row. | | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

| | C.3.11.19 | Monitor | Active | On | Phases |
|--|-----------|---------|--------|----|--------|
|--|-----------|---------|--------|----|--------|

| Test Procedure: | | Monitor Active On Phases | |
|---|----------------|---|-----------|
| Description: This test case verifies that the ASC allows a management station to identify the phases that are currently active. | | ent | |
| Requirement(| s): | • 3.5.2.2.3.7 Monitor Active On Phases | |
| Variable(s): | | MaxRows maxPhaseGroups Table_Row Int | |
| Pass/Fail Criteria:The device under test shall pass every verification step test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure | |
| 1 | GET the follow | ing objects: maxPhaseGroups. | Pass/Fail |

| 1.1 | 1.1 RECORD this information as MaxRows. | | | |
|---|---|--|-----------|--|
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 GET the following objects: phaseStatusGroupPhaseOns.Table_Row. | | | Pass/Fail | |
| - | Test Procedure Results | | - | |
| Tested By: Date Tested: | | | | |
| Test Procedure | Notes: | | | |

C.3.11.20 Monitor Next Phases

| Test Procedure: | Fest Monitor Next Phases Procedure: Monitor Next Phases | | | |
|---------------------|---|--|--|---------------------|
| Description: | tion: | | the ASC allows a managen es that are currently comm | ient itted to be |
| Requirement(| (s): • 3.5.2.2.3.8 Monitor N | | • 3.5.2.2.3.8 Monitor Next Phases | |
| Variable(s): | MaxRows maxPhaseGr Table_Row Int | | maxPhaseGroups Int | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxPhaseGroups. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: phaseStatusGroupPhaseOns.Table_Row. | | Pass/Fail | |
| | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.11.21 Monitor Phase Vehicle Calls

| Test Procedure: | Monitor Phase Vehicle Calls |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to identify the phases currently have calls for vehicle, bicycle, or transit service placed on them. |

| Requirement(s): | | • 3.5.2.2.3.9 Monitor F | Phase Vehicle Calls | |
|-------------------------|---|---|---------------------|-----------|
| Variable(s): | | MaxRows maxPhaseGroups Table_Row Int | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | GET the following objects: maxPhaseGroups. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | vs. | |
| 2 | FOR Table_Ro | ow from 1 to MaxRows. | | |
| 2.1 | GET the following objects: phaseStatusGroupVehCalls.Table_Row. | | | Pass/Fail |
| - | - | Test Procedure Results | | - |
| Tested By: Date Tested: | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.11.22 Monitor Phase Pedestrian Calls

| Test Procedure: | | Monitor Phase Vehicle Calls | | | | |
|--|---|---|--------|-----------|--|--|
| Description: | | This test case verifies that the ASC allows a management station to identify the phases currently have calls for pedestrian service placed on them. | | | | |
| Requirement(| (s): | • 3.5.2.2.3.9 Monitor Phase Vehicle Calls | | | | |
| Variable(s): | MaxRows maxPhaseGroups Table_Row Int | | | | | |
| Pass/Fail Criteria: The device under test shall pass every verification ste test case to pass the test case. | | p in this | | | | |
| Test Step Number | Test Procedure Results | | | Results | | |
| 1 | GET the following objects: maxPhaseGroups. Pass/Fai | | | Pass/Fail | | |
| 1.1 | RECORD this information as MaxRows. | | | | | |
| 2 | FOR Table_Ro | ow from 1 to MaxRows. | | | | |
| 2.1 | GET th phaseStatusG | ne following objects: roupPedCalls.Table_Ro | W. | Pass/Fail | | |
| | - | Test Procedure R | esults | | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.11.23 Monitor Ring Status

| Test Procedure: | Monitor Ring Status | | | |
|---------------------|---|---|------------|---------------------|
| Description: | This test case verifies that the ASC allows a management n: station to identify the status of each ring in the currently acti sequence. | | | nent htly active |
| Requirement(| s): | 3.5.2.2.4.1 Monitor Ring Status 3.5.2.2.4.2 Monitor Ring Termination Cause | | |
| Variable(s): | MaxRows maxRings Table_Row Int | | | |
| Pass/Fail Crit | Pass/Fail Criteria:The device under test shall pass every verification step test case to pass the test case. | | ep in this | |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxRings. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 2.1 | GET the following objects: ringStatus.Table_Row. | | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | Tested By: Date Tested: | | | Pass/Fail |
| Test Procedure Not | tes: | | | - |

C.3.11.24 Monitor Current Phase On Time

| Test Procedure: | Monitor Current Phas | Monitor Current Phase On Time | | |
|---------------------|--|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to identify how the time each currently active phase has been on for. | | | |
| Requirement(s): | • 3.5.2.2.4.3 Mo | • 3.5.2.2.4.3 Monitor Current Phase On Time | | |
| Variable(s): | MaxRows Table_Row | maxRings Int | | |
| Pass/Fail Criteria: | The device under tes test case to pass the | The device under test shall pass every verification step in this test case to pass the test case. | | |

| Test Step Number | | Test Procedure | | | | |
|------------------------|--------------------------------------|---|-----------------|-----------|--|--|
| 1 | GET the follow | ving objects: maxRings. | | Pass/Fail | | |
| 1.1 | RECO | RD this information as Max | Rows. | | | |
| 2 | FOR Table_Ro | FOR Table_Row from 1 to MaxRows. | | | | |
| 2.1 | GET th ringCurrentOn ⁻ | GET the following objects: ringCurrentPhase.Table_Row, ringCurrentOnTime.Table_Row. | | | | |
| Test Procedure Results | | | | | | |
| Tested By: | | | Date Tested: | Pass/Fail | | |
| Test Procedure No | Test Procedure Notes: | | | | | |

C.3.11.25 Monitor Active Red Channels

| Test Monitor Active Red Channels | | | | |
|--------------------------------------|--|--|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to identify channels currently outputting a RED indication. | | |
| Requirement(| s): | • 3.5.2.2.5.1 Monitor Active Red Channels | | |
| Variable(s): | MaxRows maxChannelStatusGroups Table_Row Int | | | iroups |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxChannelStatu | sGroups. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 2.1 | GET the following objects: channelStatusGroupReds. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure Not | ·ec• | | | |

C.3.11.26 Monitor Active Yellow Channels

E

| Test Procedure: | nitor Active Yellow Channels |
|--------------------|------------------------------|
|--------------------|------------------------------|

| Description: | | This test case verifies that the ASC allows a management station to identify channels currently outputting a YELLOW indication. | | | |
|--|--|--|---|---------------------------------------|--|
| Requirement(s): | | 3.5.2.2.5.2 Monitor Active Yellow Channels | | | |
| Variable(s): | | MaxRows maxChannelStatusGroups Table_Row Int | | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | | |
| | Test Procedure | | | | |
| Test Step Number | | Test Procedure | | Results | |
| Test Step Number 1 | GET the follow | Test Procedure ring objects: maxChannelStatu | sGroups. | Results Pass/Fail | |
| Test Step Number 1 1.1 | GET the follow RECO | Test Procedure ring objects: maxChannelStatu RD this information as MaxRo | sGroups. ws. | Results Pass/Fail | |
| Test Step Number 1 1.1 2 | GET the follow RECO FOR Table_Ro | Test Procedure ring objects: maxChannelStatu RD this information as MaxRo ow from 1 to MaxRows. | sGroups. ws. | Results Pass/Fail | |
| Test Step Number 1 1.1 2 2.1 | GET the follow RECO FOR Table_Ro GET th | Test Procedure ring objects: maxChannelStatu RD this information as MaxRo ow from 1 to MaxRows. ne following objects: channelSt | sGroups. ws. atusGroupYellows. | Results Pass/Fail Pass/Fail | |
| Test Step Number 1 2 2.1 | GET the follow RECO FOR Table_Ro GET th | Test Procedure ring objects: maxChannelStatu RD this information as MaxRor ow from 1 to MaxRows. ne following objects: channelSt Test Procedure Results | sGroups. ws. atusGroupYellows. | Results Pass/Fail Pass/Fail | |
| Test Step Number 1 1.1 2 2.1 Tested By: | GET the follow RECO FOR Table_Ro GET th | Test Procedure ring objects: maxChannelStatu RD this information as MaxRor ow from 1 to MaxRows. ne following objects: channelSt Test Procedure Results | sGroups. ws. atusGroupYellows. Date Tested: | Results Pass/Fail Pass/Fail Pass/Fail | |

C.3.11.27 Monitor Active Green Channels

| Test Procedure: | | Monitor Active Gree | n Chan | nels | |
|---------------------|--|--|-----------------------|---------------------------------|-------------|
| Description: | | This test case verifies that the ASC allows a management station to identify channels currently outputting a GREEN indication. | | | |
| Requirement(| s): | • 3.5.2.2.5.3 Monitor Active Green Channels | | | |
| Variable(s): | MaxRows maxChannelStatusGro Table_Row Int | | Groups | | |
| Pass/Fail Criteria: | | The device under te test case to pass the | st shall e test ca | pass every verification states. | tep in this |
| Test Step Number | | Test Pro | cedure | | Results |
| 1 | GET the follow | ing objects: maxChann | elStatus | Groups. | Pass/Fail |
| 1.1 | RECO | RD this information as | MaxRov | VS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | | |
| 2.1 | GET th | ne following objects: ch | annelSta | atusGroupGreens. | Pass/Fail |
| | • | Test Procedure | Results | | |
| Tested By: | | | | Date Tested: | Pass/Fail |
| Test Procedure Not | es: | | | | |

C.3.11.28 Monitor Active Red Overlaps

| Test Procedure: | | Monitor Active Red Overlaps | | | |
|---------------------|--|--|---------------|------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to identify the overlaps with active RED indications. | | | |
| Requirement(| (s): • 3.5.2.2.6.1 Monitor Active Red Overlaps | | | | |
| Variable(s): | | MaxRows Table_Row | | maxOverlapStatusGroups Int | |
| Pass/Fail Criteria: | | The device under test sh test case to pass the tes | all p t ca | bass every verification ste se. | ep in this |
| Test Step Number | | Test Procedu | re | | Results |
| 1 | GET the follow | ing objects: maxOverlapSta | tusC | Groups. | Pass/Fail |
| 1.1 | RECO | RD this information as Max | Row | S. | |
| 2 | FOR Table_Ro | ow from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: overlapSt | | Stat | usGroupReds. | Pass/Fail |
| | | Test Procedure Resu | llts | | - |
| Tested By: | | | | Date Tested: | Pass/Fail |
| Test Procedure Not | es: | | | | |

| C.3.11.29 | Monitor | Active | Yellow | Overlaps |
|-----------|---------|--------|--------|---|
| | | | | ••••••••••••••••••••••••••••••••••••••• |

| Test Procedure: | | Monitor Active Yellow Overlaps | | |
|---------------------|----------------|--|-----------|--|
| Description: | | This test case verifies that the ASC allows a management station to identify the overlaps with active YELLOW indications | | |
| Requirement(| s): | • 3.5.2.2.6.2 Monitor Active Red Overlaps | | |
| Variable(s): | | MaxRows maxOverlapStatusGroups Table_Row Int | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure Resul | | |
| 1 | GET the follow | ing objects: maxOverlapStatusGroups. | Pass/Fail | |
| 1.1 | RECO | RD this information as MaxRows. | | |

| 2 | FOR Table_Row from 1 to MaxRows. | | |
|-------------------------|---|--|--|
| 2.1 | GET the following objects: overlapStatusGroupYellows. | | |
| Test Procedure Results | | | |
| Tested By: Date Tested: | | | |
| Test Procedure Not | es: | | |

C.3.11.30 Monitor Active Green Overlaps

| Test Procedure: | | Monitor Active Green Overlaps | | |
|---------------------|--|---|-----------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to identify the overlaps with active GREEN indications | | |
| Requirement(| s): | • 3.5.2.2.6.3 Monitor Active Green Overlaps | | |
| Variable(s): | | MaxRows maxOverlapStatusGroups Table_Row Int | | roups |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in the test case to pass the test case. | | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxOverlapStatus | Groups. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 2.1 | GET the following objects: overlapStatusGroupGreens. | | Pass/Fail | |
| | | Test Procedure Results | 3 | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure Not | es: | | | |

C.3.11.31 Monitor Active Overlap Flashing Yellow Arrows

| Test Procedure: | Monitor Active Overlap Flashing | g Yellow Arrow |
|--------------------|--|-------------------------------|
| Description: | This test case verifies that the ASC allows a management station to identify active Flashing YELLOW Arrow indications. | |
| Requirement(s): | 3.5.2.2.6.4 Monitor Activ Arrows | e Overlap Flashing Yellow |
| Variable(s): | MaxRows Table_Row | maxOverlapStatusGroups Int |

| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verific ase. | ation step in this |
|---------------------|--|--|----------------------------|--------------------|
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxOverlapStatus | Groups. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | IF FYA IS NOT | WIRED | | |
| 2.1 | GOT | GOTO step 4. | | |
| 3 | FOR Table_Row from 1 to MaxRows. | | | |
| 3.1 | GET the following objects: overlapStatusGroupYellows.Table Row. | | Pass/Fail | |
| 3.2 | EXIT | | | |
| 4 | FOR Table_Ro | ow from 1 to MaxRows. | | |
| 4.1 | GET the following objects: overlapStatusGroupGreens.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | ; | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.11.32 Monitor Active Overlap Flashing Red Arrows

| Test Procedure: | | Monitor Active Overlap Flashing Red Arrow | | | |
|---------------------|--|---|--------------|---------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify active Flashing RED Arrow indications. | | | |
| Requirement(| s): | 3.5.2.2.6.5 Monitor Active Overlap Flashing Yellow Arrows | | | llow |
| Variable(s): | | MaxRows maxOverlapStatusGroups Table_Row Int | | roups | |
| Pass/Fail Crite | Pass/Fail Criteria:The device under test shall pass every verification step in test case to pass the test case. | | p in this | | |
| Test Step Number | | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxOverl | apStatusGrou | DS. | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | | |
| 2 | IF FYA IS NOT WIRED | | | | |
| 2.1 | GOTO step 4. | | | | |
| 3 | FOR Table_Row from 1 to MaxRows. | | | | |
| 3.1 | GET the following objects: Pa overlapStatusGroupReds.Table_Row. | | Pass/Fail | | |
| 3.2 | EXIT | | | | |
| 4 | FOR Table_Ro | w from 1 to MaxRows | | | |

| 4.1 GET the following objects: overlapStatusGroupGreens.Table_Row. | | | Pass/Fail |
|---|------------------------|-----------------|-----------|
| | Test Procedure Results | ; | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.11.33 Monitor Currently Active Preempt

| Test Procedure: | | Monitor Currently Active Preempt | | |
|------------------------|---|--|-----------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the currently active preempt. | | ient |
| Requirement(| s): | • 3.5.2.2.7.1 Monitor Currently Active Preempt | | |
| Variable(s): | | | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: preemptStatus. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | Date Tested: | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.11.34 Monitor Currently Preempt Inputs

| Test Procedure: | Monitor Current Preer | npt Inputs | | |
|---------------------|---|--|--|--|
| Description: | This test case verifies station to view the inp | This test case verifies that the ASC allows a management station to view the input state for each preempt. | | |
| Requirement(s): | • 3.5.2.2.7.2 Mor | 3.5.2.2.7.2 Monitor Current Preempt Inputs | | |
| Variable(s): | MaxRows Table_Row | maxPreemptGroups Int | | |
| Pass/Fail Criteria: | The device under test test case to pass the t | shall pass every verification step in this test case. | | |

| Test Step Number | Test Procedure | | Results |
|---------------------|--|-----------------|-----------|
| 1 | GET the following objects: maxPreemptGroup | DS. | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | |
| 2.1 | GET the following objects: preemptStatusGroup.Table_Row. | | Pass/Fail |
| | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.11.35 Monitor Currently Preempt State

| Test Procedure: | | 7.39 Monitor Current Preempt State | | |
|---------------------|---|---|-----------------|-----------|
| Description: | This test case verifies that the ASC allows a manageme station to identify the preempt status of the current act preempt. | | nent ≿tive | |
| Requirement(| s): | • 3.5.2.2.7.3 Monitor Current Preempt State | | |
| Variable(s): | MaxRows maxPreempts Table_Row Int | | | |
| Pass/Fail Crite | ass/Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | FOR Table Row from 1 to MaxRows. | | | |
| 2.1 | GET th | e following objects: preemptSt | ate.Table_Row. | Pass/Fail |
| | - | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.11.36 Monitor Current Gate Status

| Test Procedure: | Monitor Current Gate Status |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to view the status and description of each gate. |
| | 3.5.2.2.7.4 Monitor Current Gate Status |

| Requirement(| s): | | | |
|----------------------|--|----------------------|------------------------|-----------|
| Variable(s): | | MaxRows Table_Row | maxPreemptGates Int | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | | p in this |
| Test Step Number | Test Procedure Results | | | Results |
| 1 | GET the following objects: maxPreemptGates. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1.1 | GET the following objects: preemptGateStatus.Table_Row, preemptGateDescription.Table_Row. | | Pass/Fail | |
| - | Test Procedure Results | | | |
| Tested By: Date Pass | | | Pass/Fail | |
| Test Procedure Not | es: | | | |

C.3.11.37 Determine Maximum Number of Special Functions

| Test Procedure: | | Determine Maximum Number of Special Functions | | |
|---|---|--|-------------------------|-----------|
| Description: This test case verifies that the ASC allows a manager station to determine the maximum number of special supported by ASC and verifies that the ASC supports minimum number of special functions required by the | | ient functions the user. | | |
| Requirement(| s): | 3.5.2.2.8.1 Determine Maximum Number of Special Functions | | cial |
| Variable(s): | UserMinSpecialFunctionOutputs maxSpecialFunctionOutputs | | utputs | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step i test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure F | | |
| 1 | USER-ACTION specified in FR | SER-ACTION 'Determine the number of special functions required as pecified in FR ID 3.5.2.2.8.1 of the PRL.' | | |
| 1.1 | RECO | RD this information as UserMir | SpecialFunctionOutputs. | |
| 2 | GET the following objects: maxSpecialFunctionOutputs. | | Pass/Fail | |
| 3 | VERIFY maxSpecialFunctionOutputs IS_NOT_LESS_THAN UserMinSpecialFunctionOutputs. | | Pass/Fail | |
| | | Test Procedure Results | i | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | | |

C.3.11.38 Monitor Special Function Status

| Test Procedure: | | Monitor Special Function Status | | |
|---------------------|--|---|----------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows the management station to identify whether special functions are on or off. | | |
| Requirement(| s): | • 3.5.2.2.8.2 Monitor Special Function Status | | |
| Variable(s): | | MaxRows Table_Row | maxSpecialFunctionOutputs Int | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure | | |
| 1 | GET the follow | ing objects: maxSpecialFunct | ionOutputs. | Pass/Fail |
| 1.1 | RECO | RD this information as Table_ | Row. | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: specialFunctionOutputStatus.Table_Row. | | Pass/Fail | |
| | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.11.39 Monitor Special Function Control Source

| Test Procedure: | | Monitor Special Function Control Source | | |
|---------------------|-------|---|--|-----------|
| Description: | | This test case verifies that the ASC allows the management station to identify whether special functions are on or off and why any active special functions are on. | | |
| Requirement(s | s): | 3.5.2.2.8.2 Monitor Special Function Status | | |
| Variable(s): | | MaxRows maxSpecialFunctionOutputs Table_Row Int | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure Results | | Results |

| 1 | GET the following objects: maxSpecialFunctionOutputs. | Pass/Fail | | |
|-------------------------|---|-----------|--|--|
| 1.1 | RECORD this information as Table_Row. | | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: specialFunctionOutputControlSource.Table_Row. | Pass/Fail | | |
| Test Procedure Results | | | | |
| Tested By: Date Tested: | | | | |
| Test Procedure | e Notes: | | | |

C.3.12 Control Signal Operations

C.3.12.1 Enable/Disable Manual Backup

| Test Procedure: | | Enable/Disable Manual Backup | | |
|---------------------|--|--|--|-----------------|
| Description: | | This test case verifies that t station to manually activate BackUp timer and clear sys | the ASC allows the manage BackUp mode regardless tem control parameters. | ement of the |
| Requirement(| s): | • 3.5.2.3.1.1 Enable/D | isable Manual Backup | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure Resul | | |
| 1 | PRE-CONDTIC | -CONDTION 'Manual Backup Mode is disabled.' | | |
| 2 | ASSIGN unitManualBackup EQUALS 1. | | | |
| 3 | SET the following objects: unitManualBackup. | | Pass/Fail | |
| 4 | GET the follow | ing objects: unitManualBackup | | Pass/Fail |
| 5 | VERIFY unitMa | anualBackup IS_EQUAL_TO 1 | | Pass/Fail |
| 6 | DELAY 3 seco | nds | | |
| 7 | ASSIGN unitMa | anualBackup EQUALS 0. | | |
| 8 | SET the following objects: unitManualBackup. | | Pass/Fail | |
| 9 | GET the following objects: unitManualBackup. | | Pass/Fail | |
| 10 | 10 VERIFY unitControlV4 IS_EQUAL_TO 0. Pass/ | | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.12.2 Control Global Minimum Recall

Control Global Minimum Recall

| Test Procedure: | | | | | |
|---------------------|--|--|---|-------------|--|
| Description: | | This test case verifies that station to place calls for mi | the ASC allows the manage nimum recall on all phases | ement S. | |
| Requirement(| s): | • 3.5.2.3.1.2 Control G | Global Minimum Recall | | |
| Variable(s): | | OriginalUnitControl TestUnitControl | unitControlV4 unitControlV4 | | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | | ep in this | |
| Test Step Number | | Test Procedure Results | | | |
| 1 | GET the follow | GET the following objects: unitControlV4. | | Pass/Fail | |
| 1.1 | RECORD this information as OriginalUnitControl. | | | | |
| 2 | ASSIGN TestUnitControl EQUALS OriginalUnitControl. | | | | |
| 3 | ASSIGN TestUnitControl[0] EQUALS TestUnitControl XOR 128. | | | | |
| 4 | ASSIGN unitCo | ASSIGN unitControlV4 EQUALS TestUnitControl. | | | |
| 5 | SET the follow | ing objects: unitControlV4. | | Pass/Fail | |
| 6 | GET the follow | ing objects: unitControlV4. | | Pass/Fail | |
| 7 | VERIFY unitCo | ontrolV4 IS_EQUAL_TO TestU | nitControl. | Pass/Fail | |
| 8 | ASSIGN unitCo | ontrolV4 EQUALS OriginalUnit | Control. | | |
| 9 | SET the following objects: unitControlV4. | | Pass/Fail | | |
| 10 | GET the following objects: unitControlV4. | | Pass/Fail | | |
| 11 | VERIFY unitCo | ontrolV4 IS_EQUAL_TO Origin | alUnitControl. | Pass/Fail | |
| | | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | | |

C.3.12.3 Control to Non-Actuated 1

| Test Procedure: | Control Call to Non-Actuated 1 | | |
|--------------------|---|--|--|
| Description: | This test case verifies that the ASC allows the management station to artificially activate the call to non-actuated 1 input. | | |
| Requirement(s): | • 3.5.2.3.1.3 Control Call to Non-Actuated 1 | | |
| Variable(s): | OriginalUnitControl unitControlV4 TestUnitControl unitControlV4 | | |

| Pass/Fail Criteria: The device under test shall pass every test case to pass the test case. | | pass every verification ase. | step in this | |
|---|----------------|---|-------------------|-----------|
| Test Step Number | | Test Procedure Results | | |
| 1 | GET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 1.1 | RECO | RD this information as Original | UnitControl. | |
| 2 | ASSIGN Test | InitControl EQUALS OriginalUr | nitControl. | |
| 3 | ASSIGN Test | InitControl[0] EQUALS TestUn | itControl XOR 64. | |
| 4 | ASSIGN unitCo | ASSIGN unitControlV4 EQUALS TestUnitControl. | | |
| 5 | SET the follow | SET the following objects: unitControlV4. Pass/Fa | | |
| 6 | GET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 7 | VERIFY unitCo | VERIFY unitControlV4 IS_EQUAL_TO TestUnitControl. Pass/Fa | | |
| 8 | ASSIGN unitCo | ASSIGN unitControlV4 EQUALS OriginalUnitControl. | | |
| 9 | SET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 10 | GET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 11 | VERIFY unitCo | VERIFY unitControlV4 IS_EQUAL_TO OriginalUnitControl. Pass/Fail | | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.12.4 Control to Non-Actuated 2

| Test Procedure: | | Control Call to Non-Actuated 2 | | |
|---------------------|--|---|-----------|--|
| Description: | | This test case verifies that the ASC allows the management station to artificially activate the call to non-actuated 2 input. | | |
| Requirement(| s): | • 3.5.2.3.1.4 Control Call to Non-Actuated 2 | | |
| Variable(s): | | OriginalUnitControl unitControlV4 TestUnitControl unitControlV4 | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure Results | | |
| 1 | GET the follow | ing objects: unitControlV4. | Pass/Fail | |
| 1.1 | RECORD this information as OriginalUnitControl. | | | |
| 2 | ASSIGN TestUnitControl EQUALS OriginalUnitControl. | | | |
| 3 | ASSIGN TestUnitControl[0] EQUALS TestUnitControl XOR 32. | | | |
| 4 | ASSIGN unitControlV4 EQUALS TestUnitControl. | | | |
| 5 | SET the following objects: unitControlV4. Pass/Fail | | | |
| 6 | GET the follow | ing objects: unitControlV4. | Pass/Fail | |
| 7 | VERIFY unitCo | ontroIV4 IS_EQUAL_TO TestUnitControl. | Pass/Fail | |

| 8 | ASSIGN unitControlV4 EQUALS OriginalUnitControl. | | |
|------------------------|---|--|-----------|
| 9 | SET the following objects: unitControlV4. | | Pass/Fail |
| 10 | GET the following objects: unitControlV4. | | Pass/Fail |
| 11 | VERIFY unitControlV4 IS_EQUAL_TO OriginalUnitControl. | | |
| Test Procedure Results | | | |
| Tested By: Date Pase | | | |
| Test Procedure | Notes: | | |

| C 3 12 5 | Control | Walk | Rost | Modifier |
|----------|---------|-------|------|----------|
| 0.3.12.5 | CONTROL | vvain | resi | wounter |

| Test Procedure: | | Control Walk Rest Modifier | | |
|---------------------|--|--|--------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows the management station to set phases configured for non-actuated mode to re in-walk. | | |
| Requirement(| s): | • 3.5.2.3.1.5 Control Walk Rest Modifier | | |
| Variable(s): | | OriginalUnitControl TestUnitControl | unitControlV4 unitControlV4 | |
| Pass/Fail Crite | S/Fail Criteria: The device under test shall pass every verification step i test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedure Result | | | Results |
| 1 | GET the following objects: unitControlV4. Pas | | Pass/Fail | |
| 1.1 | RECORD this information as OriginalUnitControl. | | UnitControl. | |
| 2 | ASSIGN TestU | InitControl EQUALS OriginalUr | nitControl. | |
| 3 | ASSIGN TestU | InitControl[0] EQUALS TestUni | tControl XOR 16. | |
| 4 | ASSIGN unitCo | ontrolV4 EQUALS TestUnitCor | ntrol. | |
| 5 | SET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 6 | GET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 7 | VERIFY unitCo | ontrolV4 IS_EQUAL_TO TestU | nitControl. | Pass/Fail |
| 8 | ASSIGN unitCo | ontrolV4 EQUALS OriginalUnit | Control. | |
| 9 | SET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 10 | GET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 11 | VERIFY unitControlV4 IS_EQUAL_TO OriginalUnitControl. | | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.12.6 Control Interconnect

Control Interconnect

| Test Procedure: | | | | |
|---------------------|---|--|--------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows the management station to store interconnect inputs operate at a higher priority than the timebase control | | |
| Requirement(| s): | • 3.5.2.3.1.6 Control I | nterconnect | |
| Variable(s): | | OriginalUnitControl TestUnitControl | unitControlV4 unitControlV4 | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure Resu | | | Results |
| 1 | GET the following objects: unitControlV4. | | Pass/Fail | |
| 1.1 | RECORD this information as OriginalUnitControl. | | | |
| 2 | ASSIGN TestUnitControl EQUALS OriginalUnitControl. | | nitControl. | |
| 3 | ASSIGN TestU | nitControl[0] EQUALS TestUni | tControl XOR 8. | |
| 4 | ASSIGN unitCo | ontrolV4 EQUALS TestUnitCor | ntrol. | |
| 5 | SET the followi | ng objects: unitControlV4. | | Pass/Fail |
| 6 | GET the follow | ing objects: unitControlV4. | | Pass/Fail |
| 7 | VERIFY unitCo | ontroIV4 IS_EQUAL_TO TestU | nitControl. | Pass/Fail |
| 8 | ASSIGN unitControlV4 EQUALS OriginalUnitControl. | | | |
| 9 | SET the following objects: unitControlV4. Pas | | | Pass/Fail |
| 10 | GET the following objects: unitControlV4. | | | Pass/Fail |
| 11 | VERIFY unitControlV4 IS_EQUAL_TO OriginalUnitControl. Pass/Fail | | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.12.7 Activate System Timing Pattern Remotely

| Test Procedure: | Activate System Timing Patte | ern Remotely | |
|--------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to activate a timing pattern remotely. | | |
| Requirement(s): | • 3.5.2.3.2.1 Activate Sy | ystem Timing Pattern Remotely | |
| Variable(s): | OriginalPatternControl TestPatternControl | systemPatternControl systemPatternControl | |

| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
|------------------------|---|---|-------------------------|-----------|
| Test Step Number | Test Procedure Result | | | Results |
| 1 | GET the follow | ing objects: systemPatternCon | trol. | Pass/Fail |
| 1.1 | RECO | RD this information as Original | PatternControl. | |
| 2 | ASSIGN TestP | atternControl EQUALS RAND | OM (0 TO 255). | |
| 2.1 | IF Test | tPatternControl IS_EQUAL_TC | OriginalPatternControl. | |
| 2.1.1 | | GOTO step 2. | | |
| 3 | ASSIGN system | ASSIGN systemPatternControl EQUALS TestPatternControl. | | |
| 4 | SET the follow | SET the following objects: systemPatternControl. Pass/F | | |
| 5 | GET the following objects: systemPatternControl. Pass/Fa | | | Pass/Fail |
| 6 | VERIFY system | VERIFY systemPatternControl IS_EQUAL_TO TestPatternControl. Pass/Fa | | |
| 7 | ASSIGN systemPatternControl EQUALS OriginalPatternControl. | | | |
| 8 | SET the following objects: systemPatternControl. Pas | | Pass/Fail | |
| 9 | GET the following objects: systemPatternControl. | | Pass/Fail | |
| 10 | VERIFY systemPatternControl IS_EQUAL_TO OriginalPatternControl. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.12.8 Control System Reference Point

| Test Procedure: | Control System Reference Point | | |
|---|---|----------------------------------|--|
| This test case verifies that the ASC allows a managemDescription:station to configure a reference point for the called sy pattern. | | | allows a management int for the called system |
| Requirement | t(s): • 3.5.2.3.2.2 Control System Reference Point | | |
| Variable(s): | OriginalSyncControl systemSyncControl TestSyncControl systemSyncControl | | |
| Pass/Fail Crit | Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | |
| Test Step Number | Test Procedure Resul | | Results |
| 1 | GET the following objects: systemSyncControl. Pass/Fa | | Pass/Fail |
| 1.1 | RECORD this information as OriginalSyncControl. | | ntrol. |
| 2 | ASSIGN TestSyncControl EQUALS RANDOM (0 TO 255). | | |
| 2.1 | IF OriginalSyncControl IS_EQUAL_TO TestSyncControl. | | |
| 2.1.1 | | GOTO step 2. | |
| 3 | ASSIGN syste | mSyncControl EQUALS TestSyncCont | rol. |

| 4 | SET the following objects: systemSyncControl. | | | | |
|----------------|---|---|-----------|--|--|
| 5 | GET the following objects: systemSyncContro | bl. | Pass/Fail | | |
| 6 | VERIFY systemSyncControl IS_EQUAL_TO | TestSyncControl. | Pass/Fail | | |
| 7 | ASSIGN systemSyncControl EQUALS Origina | alSyncControl. | | | |
| 8 | SET the following objects: systemSyncContro | SET the following objects: systemSyncControl. | | | |
| 9 | GET the following objects: systemSyncControl. | | | | |
| 10 | VERIFY systemSyncControl IS_EQUAL_TO OriginalSyncControl. | | | | |
| | Test Procedure Results | ; | - | | |
| Tested By: | | Date Tested: | Pass/Fail | | |
| Test Procedure | Test Procedure Notes: | | | | |

C.3.12.9 Control Phase Omits

| Test Procedure: | | Control Phase Omits | | |
|---------------------|---|---|---------------------|--------------------|
| Description: | | This test case verifies that the ASC allows a management station to omit phases from being serviced. | | nent |
| Requirement(| s): | • 3.5.2.3.3.1 Control Ph | ase Omits | |
| Variable(s): | | MaxRowsmaxPhaseGroupsTable_RowIntOriginalPhaseOmitphaseControlGroupPhTestPhaseOmitphaseControlGroupPh | | PhaseOm PhaseOm |
| Pass/Fail Crite | /Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this | |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPhaseGroups. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO | MaxRows). | |
| 3 | GET the follow | ing objects: phaseControlGroupF | haseOmit.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalPt | aseOmit. | |
| 4 | ASSIGN TestF | PhaseOmit EQUALS RANDOM (0 | TO 255). | |
| 4.1 | IF Tes | tPhaseOmit IS_EQUAL_TO Orig | nalPhaseOmit. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phaseControlGroupPhaseOmit.Table_Row EQUALS TestPhaseOmit. | | | |
| 6 | SET the following objects: phaseControlGroupPhaseOmit.Table_Row. Pass/Fail | | | Pass/Fail |
| 7 | GET the follow | ing objects: phaseControlGroupF | haseOmit.Table_Row. | Pass/Fail |
| 8 | VERIFY phase TestPhaseOmi | ControlGroupPhaseOmit.Table_ it. | Row IS_EQUAL_TO | Pass/Fail |

| 9 | ASSIGN phaseControlGroupPhaseOmit.Table_Row EQUALS OriginalPhaseOmit. | | | | |
|----------------------------|---|-----------------------|-----------|--|--|
| 10 | SET the following objects: phaseControlGroup | oPhaseOmit.Table_Row. | Pass/Fail | | |
| 11 | GET the following objects: phaseControlGrou | pPhaseOmit.Table_Row. | Pass/Fail | | |
| 12 | VERIFY phaseControlGroupPhaseOmit.Table_Row IS_EQUAL_TO OriginalPhaseOmit. | | | | |
| | Test Procedure Results | | | | |
| Tested By: Date Tested: Pa | | | | | |
| Test Procedure | Notes: | | | | |

C.3.12.10 Control Pedestrian Phase Omits

| Test Procedure: | | Control Pedestrian Phase Omits | | | |
|---------------------|--|--|-------------------------------------|--------------------|--|
| Description: | | This test case verifies that the ASC allows a management station to omit pedestrian movements from being served. | | | |
| Requirement(| t(s): • 3.5.2.3.3.2 Control Pedestrian Phase Omits | | | | |
| Variable(s): | | MaxRowsmaxPhaseGroupsTable_RowIntOriginalPedOmitphaseControlGroupPhaseOmitphaseControlGroupPodOmit | | PhaseOm PedOmit | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this | |
| Test Step Number | Test Procedure | | | Results | |
| 1 | GET the following objects: maxPhaseGroups. Pas | | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO 255). | | | | |
| 3 | GET the following objects: phaseControlGroupPedOmit.Table_Row. Pas | | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPedOmit. | | | | |
| 4 | ASSIGN TestPedOmit EQUALS RANDOM (0 TO 255). | | | | |
| 4.1 | IF Test | tPedOmit IS_EQUAL_TO Origi | nalPedOmit. | | |
| 4.1.1 | | GOTO step 4. | | | |
| 5 | ASSIGN phase TestPedOmit. | eControlGroupPedOmit.Table_ | Row EQUALS | | |
| 6 | SET the follow | ing objects: phaseControlGrou | pPedOmit.Table_Row. | Pass/Fail | |
| 7 | GET the follow | ing objects: phaseControlGrou | pPedOmit.Table_Row. | Pass/Fail | |
| 8 | VERIFY phase TestPedOmit. | ControlGroupPedOmit.Table_I | Row IS_EQUAL_TO | Pass/Fail | |
| 9 | ASSIGN phase OriginalPedOm | eControlGroupPedOmit.Table_ nit. | Row EQUALS | | |
| 10 | SET the follow | ing objects: phaseControlGrou | pPedOmit.Table_Row. | Pass/Fail | |
| 11 | GET the follow | ing objects: phaseControlGrou | pPedOmit.Table_Row. | Pass/Fail | |
| 12 | VERIFY phase OriginalPedOm | eControlGroupPedOmit.Table_Row IS_EQUAL_TO Pass/Fail mit. | | Pass/Fail | |
| | | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | | |

C.3.12.11 Control Phase Holds

| Test Procedure: | | Control Phase Hold | | |
|--|--|--|---|--------------|
| Description: | This test case verifies that the ASC allows a management station to place holds on phases. | | nent | |
| Requirement(| s): | • 3.5.2.3.3.3 Control F | Phase Holds | |
| Variable(s): | | MaxRows Table_Row OriginalHold TestHold | maxPhaseGroups Int phaseControlGroupl phaseControlGroupl | Hold Hold |
| Pass/Fail Criteria: The device under test shall pass every verification test case to pass the test case. | | pass every verification ste ase. | p in this | |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxPhaseGroups. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: phaseControlGroupHold.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalHold. | | | |
| 4 | ASSIGN TestH | old EQUALS RANDOM (0 TO | 255). | |
| 4.1 | IF Test | Hold IS_EQUAL_TO Original | lold. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phase | ControlGroupHold.Table_Row | / EQUALS TestHold. | |
| 6 | SET the followi | ng objects: phaseControlGrou | pHold.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: phaseControlGrou | pHold.Table_Row. | Pass/Fail |
| 8 | VERIFY phase | ControlGroupHold.Table_Row | IS_EQUAL_TO TestHold. | Pass/Fail |
| 9 | ASSIGN phaseControlGroupHold.Table_Row EQUALS OriginalHold. | | | _ / |
| 10 | SET the following objects: phaseControlGroupHold.Table_Row. Pass/Fa | | | Pass/Fail |
| 11 | GET the following objects: phaseControlGroupHold.Table_Row. | | Pass/Fail | |
| 12 | VERIFY phase OriginalHold. | aseControlGroupHold.Table_Row IS_EQUAL_TO Pass/Fai | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | - | |

C.3.12.12 Control Phase Force Offs

E

| Description: | | This test case verifies that t station to force off phases. | the ASC allows a managen | nent |
|---------------------|---|--|---|------------|
| Requirement(s): | | • 3.5.2.3.3.4 Control Phase Force Offs | | |
| Variable(s): | | MaxRows Table_Row OriginalForceOff TestForceOff | maxPhaseGroups Int phaseControlGroupForceOff phaseControlGroupForceOff | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPhaseGroups. | | Pass/Fail |
| 1.1 | RECO | RD this information as Original | ForceOff. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: phaseControlGrou | pForceOff.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as OriginalForceOff. | | | |
| 4 | ASSIGN TestForceOff EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF Test | ForceOff IS_EQUAL_TO Origi | nalForceOff. | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN phase TestForceOff. | eControlGroupForceOff.Table_I | Row EQUALS | |
| 6 | SET the following objects: phaseControlGroupForceOff.Table_Row. Pa | | | Pass/Fail |
| 7 | GET the follow | ing objects: phaseControlGrou | pForceOff.Table_Row. | Pass/Fail |
| 8 | VERIFY phaseControlGroupForceOff.Table_Row IS_EQUAL_TO TestForceOff. | | | Pass/Fail |
| 9 | ASSIGN phaseControlGroupForceOff.Table_Row EQUALS OriginalForceOff. | | | |
| 10 | SET the following objects: phaseControlGroupForceOff.Table_Row. | | | Pass/Fail |
| 11 | GET the follow | ing objects: phaseControlGrou | pForceOff.Table_Row. | Pass/Fail |
| 12 | VERIFY phaseControlGroupForceOff.Table_Row IS_EQUAL_TO OriginalForceOff. | | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | Date Tested: | | | Pass/Fail |
| Test Procedure N | Test Procedure Notes: | | | |

C.3.12.13 Control Phase Vehicle Calls

| Test Procedure: | Control Phase Vehicle Calls |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to place calls for service on phases remotely. |
| Requirement(s): | • 3.5.2.3.3.5 Control Phase Vehicle Calls |

| Variable(s): | | MaxRows Table_Row OriginalCall TestCall | maxPhaseGroups Int phaseControlGroup phaseControlGroup) | VehCall VehCall |
|---------------------|--|---|--|--------------------|
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPhaseGroups. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: phaseControlGrou | pVehCall.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Call. | |
| 4 | ASSIGN TestC | ASSIGN TestCall EQUALS RANDOM (0 TO 255). | | |
| 4.1 | IF TestCall IS_EQUAL_TO OriginalCall. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN phaseControlGroupVehCall.Table_Row EQUALS TestCall. | | | |
| 6 | SET the following objects: phaseControlGroupVehCall.Table_Row. Pass/ | | | Pass/Fail |
| 7 | GET the following objects: phaseControlGroupVehCall.Table_Row. Pase | | | Pass/Fail |
| 8 | VERIFY phaseControlGroupVehCall.Table_Row IS_EQUAL_TO TestCall. Pas | | | Pass/Fail |
| 9 | ASSIGN phase | eControlGroupVehCall.Table_R | low EQUALS OriginalCall. | |
| 10 | SET the follow | ing objects: phaseControlGroup | oVehCall.Table_Row. | Pass/Fail |
| 11 | GET the following objects: phaseControlGroupVehCall.Table_Row. Pass | | | Pass/Fail |
| 12 | VERIFY phaseControlGroupVehCall.Table_Row IS_EQUAL_TO OriginalCall. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Test Procedure Notes: | | | |

C.3.12.14 Control Phase Pedestrian Calls

| Test Procedure: | Control Phase Pedestriar | Control Phase Pedestrian Calls | | |
|--------------------|---|--|--|--|
| Description: | This test case verifies that station to place calls for place | This test case verifies that the ASC allows a management station to place calls for pedestrian service on phases remotely. | | |
| Requirement(s): | • 3.5.2.3.3.6 Contro | • 3.5.2.3.3.6 Control Phase Pedestrian Calls | | |
| Variable(s): | MaxRows Table_Row OriginalPedCall TestPedCall | maxPhaseGroups Int phaseControlGroupPedCall phaseControlGroupPedCall | | |

| Pass/Fail Criteria: The device under test shall pass every verification ste test case to pass the test case. | | p in this | | |
|--|---|---------------------------------|-------------------------|-----------|
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxPhaseGroups. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: phaseControlGrou | pPedCall.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | PedCall. | |
| 4 | ASSIGN TestP | edCall EQUALS RANDOM (0 | TO 255). | |
| 4.1 | IF Test | PedCall IS_EQUAL_TO Origin | nalPedCall. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN phase | ControlGroupPedCall.Table_F | Row EQUALS TestPedCall. | |
| 6 | SET the following objects: phaseControlGroupPedCall.Table_Row. | | | Pass/Fail |
| 7 | GET the following objects: phaseControlGroupPedCall.Table_Row. | | Pass/Fail | |
| 8 | VERIFY phaseControlGroupPedCall.Table_Row IS_EQUAL_TO TestPedCall. | | Pass/Fail | |
| 9 | ASSIGN phaseControlGroupPedCall.Table_Row EQUALS OriginalPedCall. | | | |
| 10 | SET the followi | ng objects: phaseControlGrou | pPedCall.Table_Row. | Pass/Fail |
| 11 | GET the following objects: phaseControlGroupPedCall.Table Row. | | | Pass/Fail |
| 12 | VERIFY phaseControlGroupPedCall.Table_Row IS_EQUAL_TO OriginalPedCall. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

| C.3.12.15 | Activate | Preempt | Remotely |
|-----------|----------|----------|------------|
| 0.0.12.10 | Activate | riccinpt | recinotory |

| Test Procedure: | Activate Preempt Remotely | | |
|---------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station activate preempts remotely. | | |
| Requirement(s): | 3.5.2.3.4 Activate Preempt Remotely | | |
| Variable(s): | MaxRows Table_Row OriginalState TestState | maxPreempts Int preemptControlState preemptControlState | |
| Pass/Fail Criteria: | The device under test shall pass every verification step in this test case to pass the test case. | | |

| Test Step Number | Test Procedure | | |
|-----------------------|---|------------------|-----------|
| 1 | GET the following objects: maxPreempts. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRov | VS. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: preemptControlSta | ate.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | State. | |
| 4 | IF OriginalState IS_EQUAL_TO 0. | | |
| 4.1 | ASSIGN TestState EQUALS 1. | | |
| 4.2 | GOTO step 6. | | |
| 5 | ASSIGN TestState EQUALS 0. | | |
| 6 | ASSIGN preemptControlState.Table_Row EC | QUALS TestState. | |
| 7 | SET the following objects: preemptControlState.Table_Row. | | |
| 8 | GET the following objects: preemptControlState.Table_Row. | | |
| 9 | VERIFY preemptControlState.Table_Row IS_EQUAL_TO TestState. | | |
| 10 | ASSIGN preemptControlState.Table_Row EQUALS OriginalState. | | |
| 11 | SET the following objects: preemptControlState.Table_Row. F | | |
| 12 | GET the following objects: preemptControlState.Table_Row. Pass | | |
| 13 | VERIFY preemptControlState.Table_Row IS_EQUAL_TO OriginalState. | | Pass/Fail |
| - | Test Procedure Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | | |

C.3.12.16 Control Ring Stop Time

| Test Procedure: | | Control Ring Stop Time | | |
|--|--|---|-----------|-------------------------|
| Description: | | This test case verifies that the ASC allows a management station to stop timing by ring. | | |
| Requirement(| s): | 3.5.2.3.5.1 Control Ring Stop Time | | |
| Variable(s): | | MaxRowsmaxRingControlGroupsTable_RowIntOriginalStopTimeringControlGroupStopTinTestStopTimeringControlGroupStopTin | | ıps opTime opTime |
| Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | p in this | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxRingControlGroups. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the follow | GET the following objects: ringControlGroupStopTime.Table Row. | | Pass/Fail |

| 3.1 | RECORD this information as OriginalStopTime. | | |
|------------------------|--|--------------------|-----------|
| 4 | ASSIGN TestStopTime EQUALS RANDOM (0 TO 255). | | |
| 4.1 | IF TestStopTime IS_EQUAL_TO Orig | jinalStopTime. | |
| 4.1.1 | GOTO step 4. | | |
| 5 | ASSIGN ringControlGroupStopTime.Table_R TestStopTime. | ow EQUALS | |
| 6 | SET the following objects: ringControlGroupS | topTime.Table_Row. | Pass/Fail |
| 7 | GET the following objects: ringControlGroupS | topTime.Table_Row. | Pass/Fail |
| 8 | VERIFY ringControlGroupStopTime.Table_Row IS_EQUAL_TO TestStopTime. | | Pass/Fail |
| 9 | ASSIGN ringControlGroupStopTime.Table_R OriginalStopTime. | ow EQUALS | |
| 10 | SET the following objects: ringControlGroupS | topTime.Table_Row. | Pass/Fail |
| 11 | GET the following objects: ringControlGroupS | topTime.Table_Row. | Pass/Fail |
| 12 | VERIFY ringControlGroupStopTime.Table_Row IS_EQUAL_TO OriginalStopTime. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | | |

C.3.12.17 Control Ring Force Offs

| Test Procedure: | Test Procedure: Control Ring Force Offs | | | |
|---|--|---|------------------------|-------------------------|
| Description: This test case verifies that the ASC allows a mana station to force off phases remotely using ring set | | C allows a managem ely using ring setting | ient s. | |
| Requirement(| (s): | • 3.5.2.3.5.2 Control Ring Force Offs | | |
| Variable(s): | | MaxRowsmaxRingControlGroupsTable_RowIntOriginalForceOffringControlGroupForceOffTestForceOffringControlGroupForceOff | | ups rceOff rceOff |
| Pass/Fail Criteria: | | The device under test shall pass test case to pass the test case. | every verification ste | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ng objects: maxRingControlGroups. | | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: ringControlGroupForceOff.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalForceOff. | | | |
| 4 | ASSIGN TestForceOff EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF Tes | ForceOff IS_EQUAL_TO OriginalFor | ceOff. | |
| 4.1.1 | | GOTO step 4. | | |

| Test Procedure Notes: | | | | |
|-----------------------|--|--|-----------|--|
| Tested By: | D T | oate fested: | Pass/Fail | |
| | Test Procedure Results | | | |
| 12 | VERIFY ringControlGroupForceOff.Table_Row IS_EQUAL_TO OriginalForceOff. | | Pass/Fail | |
| 11 | GET the following objects: ringControlGroupForceOff.Table_Row. | | Pass/Fail | |
| 10 | SET the following objects: ringControlGroupForceOff.Table_Row. | | Pass/Fail | |
| 9 | ASSIGN ringControlGroupForceOff.Table_Row EQUALS OriginalForceOff. | | | |
| 8 | VERIFY ringControlGroupForceOff.Table_Row I TestForceOff. | VERIFY ringControlGroupForceOff.Table_Row IS_EQUAL_TO TestForceOff. | | |
| 7 | GET the following objects: ringControlGroupFord | ceOff.Table_Row. | Pass/Fail | |
| 6 | SET the following objects: ringControlGroupForceOff.Table_Row. | | | |
| 5 | ASSIGN ringControlGroupForceOff.Table_Row EQUALS TestForceOff. | | | |

C.3.12.18 Control Ring Maximum 2 Time Settings

| Test Procedure: | | Control Ring Maximum 2 Tim | e Settings | |
|--|--|--|--|---|
| Description: | | This test case verifies that th station to activate the phase | e ASC allows a managen maximum2 settings usin | nent g rings. |
| Requirement(| s): | • 3.5.2.3.5.3 Control Rin | ng Maximum 2 Time Setti | ngs |
| Variable(s): | | MaxRows OriginalGroupMax2 TestGroupMax2 Table_Row | maxRingControlGro ringControlGroupMa ringControlGroupMa Int | ups ax2 ax2 |
| Pass/Fail Crite | eria: | The device under test shall p test case to pass the test case | ass every verification ste e. | ep in this |
| - | | - | | |
| Test Step Number | | Test Procedure | | Results |
| Test Step Number 1 | GET the follow | Test Procedure | ups. | Results Pass/Fail |
| Test Step Number 1 1.1 | GET the follow RECO | Test Procedure ing objects: maxRingControlGrou RD this information as MaxRows | ups. | Results Pass/Fail |
| Test Step Number 1 1.1 2 | GET the follow RECO ASSIGN Table | Test Procedure ing objects: maxRingControlGrou RD this information as MaxRows _Row EQUALS RANDOM (1 TO | ups. MaxRows). | Results Pass/Fail |
| Test Step Number11.123 | GET the follow RECO ASSIGN Table GET the follow | Test Procedure ring objects: maxRingControlGrou RD this information as MaxRows -Row EQUALS RANDOM (1 TO ring objects: ringControlGroupMa | ups. MaxRows). x2.Table_Row. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 | GET the follow RECO ASSIGN Table GET the follow RECO | Test Procedure ing objects: maxRingControlGrou RD this information as MaxRows Row EQUALS RANDOM (1 TO ing objects: ringControlGroupMa RD this information as OriginalG | ups. MaxRows). x2.Table_Row. roupMax2. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN Test | Test Procedure ring objects: maxRingControlGrou RD this information as MaxRows -Row EQUALS RANDOM (1 TO ring objects: ringControlGroupMa RD this information as OriginalG GroupMax2 EQUALS RANDOM (| ups. MaxRows). x2.Table_Row. roupMax2. 0 TO 255). | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN Test IF Test | Test Procedure ring objects: maxRingControlGrou RD this information as MaxRows a_Row EQUALS RANDOM (1 TO ring objects: ringControlGroupMa RD this information as OriginalG GroupMax2 EQUALS RANDOM (tGroupMax2 IS_EQUAL_TO Original | ups. MaxRows). x2.Table_Row. roupMax2. 0 TO 255). ginalGroupMax2. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN Testo IF Test | Test Procedure ing objects: maxRingControlGrou RD this information as MaxRows Row EQUALS RANDOM (1 TO ing objects: ringControlGroupMa RD this information as OriginalG GroupMax2 EQUALS RANDOM (tGroupMax2 IS_EQUAL_TO Orig GOTO step 4. | ups. MaxRows). x2.Table_Row. roupMax2. 0 TO 255). ginalGroupMax2. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN Testo IF Test ASSIGN ringC | Test Procedure ring objects: maxRingControlGrou RD this information as MaxRows a_Row EQUALS RANDOM (1 TO ring objects: ringControlGroupMa RD this information as OriginalG GroupMax2 EQUALS RANDOM (tGroupMax2 IS_EQUAL_TO OriginalG GOTO step 4. ontrolGroupMax2.Table_Row EQUAL CONTONE CONTINUES (CONTINUES CONTINUES CO | ups. MaxRows). x2.Table_Row. roupMax2. 0 TO 255). ginalGroupMax2. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestO IF Test ASSIGN ringC SET the follow | Test Procedure ing objects: maxRingControlGrou RD this information as MaxRows a Row EQUALS RANDOM (1 TO ring objects: ringControlGroupMa RD this information as OriginalG GroupMax2 EQUALS RANDOM (tGroupMax2 IS_EQUAL_TO Orig GOTO step 4. ontrolGroupMax2.Table_Row EQ ing objects: ringControlGroupMa | ups. MaxRows). x2.Table_Row. roupMax2. 0 TO 255). ginalGroupMax2. QUALS TestGroupMax2. x2.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 4.1 4.1.1 5 6 7 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN Testo IF Testo ASSIGN ringC SET the follow GET the follow | Test Procedure ing objects: maxRingControlGrou RD this information as MaxRows Row EQUALS RANDOM (1 TO ing objects: ringControlGroupMa RD this information as OriginalG GroupMax2 EQUALS RANDOM (tGroupMax2 IS_EQUAL_TO Original GOTO step 4. ontrolGroupMax2.Table_Row EQ ing objects: ringControlGroupMa ring objects: ringControlGroupMa | ups. MaxRows). x2.Table_Row. roupMax2. 0 TO 255). ginalGroupMax2. QUALS TestGroupMax2. x2.Table_Row. x2.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 9 | ASSIGN ringControlGroupMax2.Table_Row E OriginalGroupMax2. | EQUALS | |
|----------------|---|-----------------|-----------|
| 10 | SET the following objects: ringControlGroupM | lax2.Table_Row. | Pass/Fail |
| 11 | GET the following objects: ringControlGroupM | lax2.Table_Row. | Pass/Fail |
| 12 | VERIFY ringControlGroupMax2.Table_Row IS OriginalGroupMax2. | S_EQUAL_TO | Pass/Fail |
| - | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.12.19 Control Ring Maximum 3 Time Settings

| Test Procedure: | | Control Ring Maximum 3 Time | Settings | |
|---------------------|--|---|--|------------------|
| Description: | | This test case verifies that the station to activate the phase m | ASC allows a managen aximum3 settings usin | nent g rings. |
| Requirement(| s): | • 3.5.2.3.5.4 Control Ring | Maximum 3 Time Setti | ngs |
| Variable(s): | | MaxRows OriginalGroupMax3 TestGroupMax3 Table_Row | maxRingControlGro ringControlGroupMa ringControlGroupMa Int | ups x3 x3 |
| Pass/Fail Crite | eria: | The device under test shall past test case to pass the test case. | s every verification ste | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxRingControlGroup | S. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO M | laxRows). | |
| 3 | GET the follow | ing objects: ringControlGroupMax3 | .Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalGro | upMax3. | |
| 4 | ASSIGN Test | oroupMax3 EQUALS RANDOM (0 | TO 255). | |
| 4.1 | IF Test | tGroupMax3 IS_EQUAL_TO Origin | alGroupMax3. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN ringC | ontrolGroupMax3.Table_Row EQU | ALS TestGroupMax3. | |
| 6 | SET the follow | ng objects: ringControlGroupMax3 | .Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: ringControlGroupMax3 | .Table_Row. | Pass/Fail |
| 8 | VERIFY ringCo TestGroupMax | ontrolGroupMax3.Table_Row IS_E 3. | QUAL_TO | Pass/Fail |
| 9 | ASSIGN ringContrigues of the contribution of t | ontrolGroupMax3.Table_Row EQU //ax3. | ALS | |
| 10 | SET the follow | ng objects: ringControlGroupMax3 | .Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: ringControlGroupMax3 | .Table_Row. | Pass/Fail |

| 12 | VERIFY ringControlGroupMax3.Table_Row IS OriginalGroupMax3. | S_EQUAL_TO | Pass/Fail |
|------------------|--|-----------------|-----------|
| | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.12.20 Control Ring Maximum Inhibit Settings

| Test Procedure: | | Control Ring Maximum Inhibit S | ettings | |
|---------------------|--------------------------------|--|---|-----------------------------|
| Description: | | This test case verifies that the A station to activate the phase ma | SC allows a managen xInhibit settings using | nent g rings. |
| Requirement(| s): | • 3.5.2.3.5.5 Control Ring I | Maximum Inhibit Settii | ngs |
| Variable(s): | | MaxRows OriginalGroupMaxInhibit TestGroupControlInhibit Table_Row | maxRingControlGrou ringControlGroupMa ringControlGroupMa Int | ups xInhibit xInhibit |
| Pass/Fail Crite | eria: | The device under test shall pass test case to pass the test case. | every verification ste | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxRingControlGroups | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO Ma | axRows). | |
| 3 | GET the follow | ing objects: ringControlGroupMaxIn | hibit.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalGrou | oMaxInhibit. | |
| 4 | ASSIGN Test | GroupControlInhibit EQUALS RAND | DM (0 TO 255). | |
| 4.1 | IF Test OriginalGroup | tGroupControlInhibit IS_EQUAL_TO //axInhibit. | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN ringCo TestGroupCon | ontrolGroupMaxInhibit.Table_Row E trolInhibit. | QUALS | |
| 6 | SET the follow | ing objects: ringControlGroupMaxInI | nibit.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: ringControlGroupMaxIn | hibit.Table_Row. | Pass/Fail |
| 8 | VERIFY ringCo TestGroupCon | ontrolGroupMaxInhibit.Table_Row IS trolInhibit. | S_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN ringCo OriginalGroup | ontrolGroupMaxInhibit.Table_Row E //axInhibit. | QUALS | |
| 10 | SET the follow | ing objects: ringControlGroupMaxInl | nibit.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: ringControlGroupMaxIn | hibit.Table_Row. | Pass/Fail |
| 12 | VERIFY ringCo OriginalGroup | ontrolGroupMaxInhibit.Table_Row IS //axInhibit. | S_EQUAL_TO | Pass/Fail |
| | | Test Procedure Results | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.12.21 Control Ring Pedestrian Recycle Settings

| Test Procedure: | | Control Ring Pedestrian Red | cycle Settings | |
|---------------------|--------------------------------|---|---|-----------------------------|
| Description: | | This test case verifies that the station to enable pedestrian settings. | he ASC allows a managem recycle on phases using | ient ring |
| Requirement(| s): | • 3.5.2.3.5.6 Control R | ing Pedestrian Recycle Se | ttings |
| Variable(s): | | MaxRows Table_Row OriginalPedRecycle TestPedRecycle | maxRingControlGrou Int ringControlGroupPe ringControlGroupPe | ups dRecycle dRecycle |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste se. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxRingControlGro | oups. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRow | S. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: ringControlGroupPe | edRecycle.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalF | PedRecycle. | |
| 4 | ASSIGN TestF | PedRecycle EQUALS RANDOM | (0 TO 255). | |
| 4.1 | IF Tes | tPedRecycle IS_EQUAL_TO Or | iginalPedRecycle. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN ringC TestPedRecyc | ontrolGroupPedRecycle.Table_ le. | Row EQUALS | |
| 6 | SET the follow | ing objects: ringControlGroupPe | edRecycle.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: ringControlGroupPe | edRecycle.Table_Row. | Pass/Fail |
| 8 | VERIFY ringCo TestPedRecyc | ontrolGroupPedRecycle.Table_I le. | Row IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN ringC OriginalPedRe | ontrolGroupPedRecycle.Table_ cycle. | Row EQUALS | |
| 10 | SET the follow | ing objects: ringControlGroupPe | edRecycle.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: ringControlGroupPe | edRecycle.Table_Row. | Pass/Fail |
| 12 | VERIFY ringCo OriginalPedRe | ontrolGroupPedRecycle.Table_I cycle. | Row IS_EQUAL_TO | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.12.22 Control Ring Red Rest Settings

| Test Procedure: | | Control Ring Red Rest Setti | ings | |
|---------------------|--------------------------------|---|---|-----------------------|
| Description: | | This test case verifies that t station to configure phases | the ASC allows a managen to rest in red using ring so | ient ettings. |
| Requirement(| s): | • 3.5.2.3.5.7 Control R | ing Red Rest Settings | |
| Variable(s): | | MaxRows Table_Row OriginalRedRest TestRedRest | maxRingControlGrou Int ringControlGroupRe ringControlGroupRe | ups dRest dRest |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxRingControlGr | oups. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRow | /S. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: ringControlGroupR | edRest.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Originall | RedRest. | |
| 4 | ASSIGN TestR | RedRest EQUALS RANDOM (0 | TO 255). | |
| 4.1 | IF Test | tRedRest IS_EQUAL_TO Origi | nalRedRest. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN ringCo | ontrolGroupRedRest.Table_Ro | w EQUALS TestRedRest. | |
| 6 | SET the followi | ing objects: ringControlGroupR | edRest.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: ringControlGroupR | edRest.Table_Row. | Pass/Fail |
| 8 | VERIFY ringCo TestRedRest. | ontrolGroupRedRest.Table_Rov | w IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN ringCo OriginalRedRe | ontrolGroupRedRest.Table_Ro st. | w EQUALS | |
| 10 | SET the followi | ing objects: ringControlGroupR | edRest.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: ringControlGroupR | edRest.Table_Row. | Pass/Fail |
| 12 | VERIFY ringCo OriginalRedRe | ontrolGroupRedRest.Table_Rov st. | w IS_EQUAL_TO | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.12.23 Control Ring Pedestrian Recycle Settings

| Procedure: 8.12 Control Ring Red Clearance Omit Settings |
|--|
|--|
| Description: | | This test case verifies that the ASC allows a management station to omit red clearance intervals. | | |
|---------------------|--|--|---------------------------|-------------------------------|
| Requirement(| (s): | • 3.5.2.3.5.8 Control F | Ring Red Clearance Omit S | ettings |
| Variable(s): | | MaxRows maxRingControlGroups OriginalOmitRedClear ringControlGroupOmitRedCle TestOmitRedClear ringControlGroupOmitRedCle Tablo Row | | ups nitRedCle nitRedCle |
| Pass/Fail Crit | Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxRingControlGr | oups. | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: ringControlGroupOmitRedClear.Table_Row. | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalOmitRedClear. | | | |
| 4 | ASSIGN TestOmitRedClear EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestOmitRedClear IS_EQUAL_TO OriginalOmitRedClear. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN ringControlGroupOmitRedClear.Table_Row EQUALS RANDOM (0 TO 255). | | | |
| 6 | SET the follow | ing objects: ringControlGroupO | mitRedClear.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: ringControlGroupC | mitRedClear.Table_Row. | Pass/Fail |
| 8 | VERIFY ringCo TestOmitRedC | VERIFY ringControlGroupOmitRedClear.Table_Row IS_EQUAL_TO TestOmitRedClear. | | |
| 9 | ASSIGN ringControlGroupOmitRedClear.Table_Row EQUALS OriginalOmitRedClear. | | | |
| 10 | SET the follow | SET the following objects: ringControlGroupOmitRedClear.Table Row. Pass/Fai | | |
| 11 | GET the follow | ing objects: ringControlGroupC | mitRedClear.Table_Row. | Pass/Fail |
| 12 | VERIFY ringControlGroupOmitRedClear.Table_Row IS_EQUAL_TO OriginalOmitRedClear. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.12.24 Activate Special Function Remotely

| Description: | | This test case verifies that the ASC allows a management station to enable or disable special function outputs. | | |
|---------------------|---|--|-------------------------------------|-------------------------------|
| Requirement(s): | | • 3.5.2.3.6 Activate Sp | pecial Function Remotely | |
| Variable(s): | | MaxRowsmaxSpecialFunctionOutputsTable_RowIntOriginalOutputControlspecialFunctionOutputControlTestOutputControlspecialFunctionOutputControl | | Outputs Control Control |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxSpecialFunctic | onOutputs. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: specialFunctionOu | tputControl.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as OriginalOutputControl. | | | |
| 4 | IF OriginalOutputControl IS_EQUAL_TO 0. | | | |
| 4.1 | ASSIGN TestOutputControl EQUALS 1. | | | |
| 4.2 | GOTO step 6. | | | |
| 5 | ASSIGN TestOutputControl EQUALS 0. | | | |
| 6 | ASSIGN specia TestOutputCor | ASSIGN specialFunctionOutputControl.Table_Row EQUALS TestOutputControl. | | |
| 7 | SET the followi | ing objects: specialFunctionOu | tputControl.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: specialFunctionOu | tputControl.Table_Row. | Pass/Fail |
| 9 | VERIFY specia TestOutputCor | VERIFY specialFunctionOutputControl.Table_Row IS_EQUAL_TO TestOutputControl. | | |
| 10 | ASSIGN specialFunctionOutputControl.Table_Row EQUALS OriginalOutputControl. | | | |
| 11 | SET the follow | ing objects: specialFunctionOu | tputControl.Table_Row. | Pass/Fail |
| 12 | GET the follow | ing objects: specialFunctionOu | tputControl.Table_Row. | Pass/Fail |
| 13 | VERIFY specialFunctionOutputControl.Table_Row IS_EQUAL_TO OriginalOutputControl. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.12.25 Remote Manual Control

| Test Procedure: | Remote Manual Control |
|--------------------|--|
| Description: | This test case verifies that the ASC allows a management station to command the signal controller to advance to the next interval. |

| Requirement(s): | | 3.5.2.3.7.1 Enable Manual Control 3.5.2.3.7.2 Remote Manual Control Advance 3.5.2.3.7.3 Configure Manual Control Timeout | | | |
|---------------------|--|--|-------------------------------------|------------|--|
| Variable(s): | | TestTimeOut | unitMCETimeout | | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this | |
| Test Step Number | Test Procedure | | | Results | |
| 1 | GET the follow | ing objects: unitMCETimeout. | | Pass/Fail | |
| 2 | ASSIGN TestT | imeOut EQUALS RANDOM (1) | 0 TO 65535) | | |
| 3 | ASSIGN unitM | ASSIGN unitMCETimeout EQUALS TestTimeOut | | | |
| 4 | SET the followi | SET the following objects: unitMCETimeout. Pass/Fail | | | |
| 5 | DELAY 3 Seconds | | | | |
| 6 | GET the following objects: unitMCETimeout. Pass/Fail | | | | |
| 7 | VERIFY unitMCETimeout IS_NOT_LESS_THAN (TestTimeOut – 3) Pass/Fail | | | | |
| 8 | GET the following objects: unitMCEIntAdv. Pass/Fail | | | | |
| 9 | ASSIGN unitMCEIntAdv EQUALS 1. | | | | |
| 10 | SET the following objects: unitMCEIntAdv. Pass/Fail | | | Pass/Fail | |
| 11 | DELAY 1 Second | | | | |
| 12 | GET the following objects: unitMCEIntAdv. Pass/Fail | | | Pass/Fail | |
| 13 | VERIFY unitMO | CEIntAdv IS_EQUAL_TO 0. | | Pass/Fail | |
| 14 | ASSIGN unitMCETimeout EQUALS 0. | | | | |
| 15 | SET the followi | SET the following objects: unitMCETimeOut. Pass/Fail | | | |
| 16 | GET the following objects: unitMCETimeout. Pass/Fai | | | Pass/Fail | |
| 17 | VERIFY unitMCETimeout IS_EQUAL_TO 0. | | | Pass/Fail | |
| | | Test Procedure Results | | | |
| Tested By: | Date Pass/Fai Tested: | | | Pass/Fail | |
| Test Procedure N | lotes: | | | | |

C.3.12.26 Enable/Disable Automatic Pedestrian Clearance Setting

| Test Procedure: | | Enable/Disable Automatic Pedestrian Clearance Setting | | |
|---------------------|---|--|-----------|--|
| Description: | | This test case verifies that the ASC allows a management station to protect the Pedestrian Clearance Interval from being terminated by an Internal Advance input when Manual Control is active. | | |
| Requirement(| s): | 3.5.2.3.7.4 Enable/Disable Automatic Pedestrian Clearance Setting | | |
| Variable(s): | | OriginalClear unitAutoPedestrianClear TestClear unitAutoPedestrianClear | | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure Results | | |

| 1 | GET the following objects: unitAutoPedestriar | nClear. | Pass/Fail | |
|------------------------|---|-------------------|-----------|--|
| 1.1 | RECORD this information as Original | Clear. | | |
| 2 | IF OriginalClear IS_EQUAL_TO 1. | | | |
| 2.1 | ASSIGN TestClear EQUALS 2. | | | |
| 3 | IF OriginalClear IS_EQUAL_TO 2. | | | |
| 3.1 | ASSIGN TestClear EQUALS 1. | | | |
| 4 | ASSIGN unitAutoPedestrianClear EQUALS T | estClear. | | |
| 5 | SET the following objects: unitAutoPedestriar | nClear. | Pass/Fail | |
| 6 | GET the following objects: unitAutoPedestrian | nClear. | Pass/Fail | |
| 7 | VERIFY unitAutoPedestrianClear IS_EQUAL_TO TestClear. | | | |
| 8 | ASSIGN unitAutoPedestrianClear EQUALS OriginalClear. | | | |
| 9 | SET the following objects: unitAutoPedestrianClear. Pass/Fail | | | |
| 10 | GET the following objects: unitAutoPedestrian | Pass/Fail | | |
| 11 | VERIFY unitAutoPedestrianClear IS_EQUAL | TO OriginalClear. | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure | Notes: | | | |

C.3.13 Manage Detector Configuration

C.3.13.1 Configure Vehicle Travel Mode

| Test Procedure: | 3.05 Configure Vehicle Travel Mode | | | |
|---------------------|--|--|--|-----------------------|
| Description: | | This test case verifies that the station to configure the trave | the ASC allows a managen /el mode for a detector. | nent |
| Requirement(| s): | • 3.5.3.1.1.1 Configure | e Vehicle Travel Mode | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalTravelModevehicleDetectorTravelModeTestTravelModevehicleDetectorTravelMode | | s elMode elMode |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxVehicleDetectors. | | | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorTravelMode.Table_Row. Pas | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalTravelMode. | | | |
| 4 | ASSIGN TestTravelMode EQUALS RANDOM (2 TO 4). | | | |
| 4.1 | IF TestTravelMode IS_EQUAL_TO OriginalTravelMode. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN vehicl TestTravelMod | ASSIGN vehicleDetectorTravelMode.Table_Row EQUALS TestTravelMode. | | |
| 6 | SET the follow | ing objects: vehicleDetectorTra | velMode.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorTra | velMode.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicle TestTravelMod | eDetectorTravelMode.Table_R le | ow IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN vehicl OriginalTravel | eDetectorTravelMode.Table_R Mode. | low EQUALS | |
| 10 | SET the following objects: vehicleDetectorTravelMode.Table Row. | | | Pass/Fail |
| 11 | GET the follow | ing objects: vehicleDetectorTra | velMode.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorTravelMode.Table_Row IS_EQUAL_TO OriginalTravelMode. | | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | - | - |

C.3.13.2 Configure Vehicle Detector Description

| Test Procedure: | | Configure Vehicle Detector Description | | | |
|-------------------------|---|---|--|--------------|--|
| Description: | | This test case verifies that the ASC allows a management station to configure descriptions for vehicle detectors. | | | |
| Requirement(s): | | • 3.5.3.1.1.2 Configure | • 3.5.3.1.1.2 Configure Vehicle Detector Description | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalDescriptionvehicleDetectorDescription | | s ription | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this | |
| Test Step Number | Test Procedure | | Results | | |
| 1 | GET the follow | ing objects: maxVehicleDetecto | ors. | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | | |
| 3 | GET the following objects: vehicleDetectorDescription.Table_Row. | | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalDescription | | | | |
| 4 | ASSIGN vehicleDetectorDescription.Table_Row EQUALS 'Test'. | | | | |
| 5 | SET the following objects: vehicleDetectorDescription.Table_Row. | | | Pass/Fail | |
| 6 | GET the follow | ing objects: vehicleDetectorDe | scription.Table_Row. | Pass/Fail | |
| 7 | VERIFY vehicle | eDetectorDescription.Table_Rc | w IS_EQUAL_TO 'Test'. | Pass/Fail | |
| 8 | ASSIGN vehicl OrignalDescrip | eDetectorDescription.Table_Ro tion. | bw EQUALS | | |
| 9 | SET the followi | ng objects: vehicleDetectorDes | scription.Table_Row. | Pass/Fail | |
| 10 | GET the following objects: vehicleDetectorDescription.Table_Row. | | | Pass/Fail | |
| 11 | VERIFY vehicleDetectorDescription.Table_Row IS_EQUAL_TO OriginalDescription. | | Pass/Fail | | |
| Test Procedure | e Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | | |

C.3.13.3 Configure Vehicle Detector Yellow Lock Call Enabled

| Test Procedure: | Configure Vehicle Detect | Configure Vehicle Detector Yellow Lock Call Enabled | | |
|--------------------|--|---|--|--|
| Description: | This test case verifies that station to set a vehicle de a call to the assigned phat Yellow Change Interval | This test case verifies that the ASC allows a management station to set a vehicle detector to set a vehicle detector to lock a call to the assigned phase if an actuation occurs during the Yellow Change Interval | | |
| Requirement(s): | • 3.5.3.1.1.3 Configu Enabled | 3.5.3.1.1.3 Configure Vehicle Detector Yellow Lock Call Enabled | | |
| Variable(s): | MaxRows Table_Row OriginalOptions | maxVehicleDetectors Int vehicleDetectorOptions | | |

| | | TestOptions | | vehicleDetectorOptic | ons |
|---------------------|---|--|--------------------|-------------------------------------|------------|
| Pass/Fail Criteria: | | The device under tes test case to pass the | t shall test ca | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxVehicle | Detecto | ors. | Pass/Fail |
| 1.1 | RECO | RD this information as N | /laxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDO | DM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: vehicleDete | ctorOp | tions.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as C | Driginal | Options. | |
| 4 | ASSIGN Test | Options EQUALS Origina | alOptior | ns XOR 4. | |
| 5 | ASSIGN vehic | ASSIGN vehicleDetectorOptions.Table_Row EQUALS TestOptions. | | | |
| 6 | SET the follow | SET the following objects: vehicleDetectorOptions.Table_Row. Pass/ | | | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDete | ctorOp | tions.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicleDetectorOptions.Table_Row IS_EQUAL_TO TestOptions. | | | Pass/Fail | |
| 9 | ASSIGN vehic | leDetectorOptions.Table | _Row I | EQUALS OriginalOptions. | |
| 10 | SET the follow | ing objects: vehicleDete | ctorOpt | tions.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: vehicleDete | ctorOp | tions.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorOptions.Table_Row IS_EQUAL_TO OriginalOptions. | | | Pass/Fail | |
| | Test Procedure Results | | | | |
| Tested By: | | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | | |

C.3.13.4 Configure Vehicle Detector Red Lock Call Enabled

| Test Procedure: | | Configure Vehicle Detector Red Lock Call Enabled | | |
|---------------------|--|---|--|-----------------|
| Description: | | This test case verifies that the ASC allows a management station to set a vehicle detector to lock a call to the assigned phase if an actuation occurs while the phase is not timing Green or Yellow Change intervals. | | |
| Requirement(s): | | 3.5.3.1.1.4 Configure Vel Enabled | hicle Detector Red Loc | k Call |
| Variable(s): | | MaxRows Table_Row OriginalOptions TestOptions | maxVehicleDetectors Int vehicleDetectorOptic vehicleDetectorOptic | s ons ons |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure Results | | |

| 1 | GET the following objects: maxVehicleDetected | ors. | Pass/Fail |
|----------------|---|---------------------|-----------|
| 1.1 | RECORD this information as MaxRow | VS. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: vehicleDetectorOp | tions.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | Options. | |
| 4 | ASSIGN TestOptions EQUALS OriginalOption | ns XOR 8. | |
| 5 | ASSIGN vehicleDetectorOptions.Table_Row | EQUALS TestOptions. | |
| 6 | SET the following objects: vehicleDetectorOp | tions.Table_Row. | Pass/Fail |
| 7 | GET the following objects: vehicleDetectorOp | tions.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicleDetectorOptions.Table_Row I TestOptions. | Pass/Fail | |
| 9 | ASSIGN vehicleDetectorOptions.Table_Row EQUALS OriginalOptions. | | |
| 10 | SET the following objects: vehicleDetectorOp | Pass/Fail | |
| 11 | GET the following objects: vehicleDetectorOp | Pass/Fail | |
| 12 | VERIFY vehicleDetectorOptions.Table_Row IS_EQUAL_TO OriginalOptions. | | |
| - | Test Procedure Results | | - |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.13.5 Configure Vehicle Detector Passage Enabled

| Test Procedure: | Configure Vehicle Detector Passage Enabled | | | |
|--|---|--|-----------|--|
| Description: This test case verifies that the ASC allows a management station to set the associated phase passage timer to rese the duration of a vehicle detector actuation if the phase is Green interval. | | nent reset for se is in the | | |
| Requirement(| s): | • 3.5.3.1.1.5 Configure Vehicle Detector Passage Enabled | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalOptionsvehicleDetectorOptionsTestOptionsvehicleDetectorOptions | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure Results | | | |
| 1 | GET the following objects: maxVehicleDetectors. Pass/Fail | | Pass/Fail | |
| 1.1 | RECO | RD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the follow | ing objects: vehicleDetectorOptions.Table_Row. | Pass/Fail | |
| 3.1 | RECO | RD this information as OriginalOptions. | | |

| 4 | ASSIGN TestOptions EQUALS OriginalOption | ns XOR 16. | |
|----------------|---|---------------------|-----------|
| 5 | ASSIGN vehicleDetectorOptions.Table_Row | EQUALS TestOptions. | |
| 6 | SET the following objects: vehicleDetectorOpt | tions.Table_Row. | Pass/Fail |
| 7 | GET the following objects: vehicleDetectorOp | tions.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicleDetectorOptions.Table_Row IS_EQUAL_TO TestOptions. | | |
| 9 | ASSIGN vehicleDetectorOptions.Table_Row EQUALS OriginalOptions. | | |
| 10 | SET the following objects: vehicleDetectorOptions.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: vehicleDetectorOptions.Table_Row. | | Pass/Fail |
| 12 | VERIFY vehicleDetectorOptions.Table_Row IS_EQUAL_TO OriginalOptions. | | Pass/Fail |
| | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.13.6 Configure Vehicle Detector Added Initial Time Enabled

| Test Procedure: | | Configure Vehicle Detector Added Initial Time Enabled | |
|---------------------------------|---|---|---|
| Description: | | This test case verifies that the ASC allows a manage station to set the ASC to use vehicle detector actuat for added initial calculations. | ment on counts |
| Requirement(| s): | 3.5.3.1.1.6 Configure Vehicle Detector Added Initial Time Enabled | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalOptionsvehicleDetectorOptionsTestOptionsvehicleDetectorOptions | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | ep in this |
| Test Step Number | | Test Procedure | |
| 1 | GET the follow | ing objects: maxVehicleDetectors | |
| 1 1 | RECORD this information as MaxRows. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | Pass/Fail |
| 2 | RECO ASSIGN Table | RD this information as MaxRows. Row EQUALS RANDOM (1 TO MaxRows). | Pass/Fail |
| 2 3 | RECO ASSIGN Table GET the follow | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: vehicleDetectorOptions.Table_Row. | Pass/Fail Pass/Fail |
| 2 3 4 | RECO ASSIGN Table GET the follow ASSIGN TestC | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: vehicleDetectorOptions.Table_Row. Options EQUALS OriginalOptions XOR 32. | Pass/Fail Pass/Fail |
| 2 3 4 5 | RECO ASSIGN Table GET the follow ASSIGN TestO ASSIGN vehicl | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: vehicleDetectorOptions.Table_Row. Options EQUALS OriginalOptions XOR 32. eDetectorOptions.Table_Row EQUALS TestOptions. | Pass/Fail Pass/Fail |
| 2 3 4 5 6 | RECO ASSIGN Table GET the follow ASSIGN TestC ASSIGN vehicl SET the follow | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: vehicleDetectorOptions.Table_Row. Options EQUALS OriginalOptions XOR 32. eDetectorOptions.Table_Row EQUALS TestOptions. ing objects: vehicleDetectorOptions.Table_Row. | Pass/Fail Pass/Fail Pass/Fail |
| 2 3 4 5 6 7 | RECO ASSIGN Table GET the follow ASSIGN TestO ASSIGN vehicl SET the follow GET the follow | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: vehicleDetectorOptions.Table_Row. Options EQUALS OriginalOptions XOR 32. eDetectorOptions.Table_Row EQUALS TestOptions. ing objects: vehicleDetectorOptions.Table_Row. ing objects: vehicleDetectorOptions.Table_Row. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 2 3 4 5 6 7 8 | RECO ASSIGN Table GET the follow ASSIGN TestO ASSIGN vehicl SET the follow GET the follow VERIFY vehicle TestOptions. | RD this information as MaxRows. _Row EQUALS RANDOM (1 TO MaxRows). ing objects: vehicleDetectorOptions.Table_Row. Options EQUALS OriginalOptions XOR 32. eDetectorOptions.Table_Row EQUALS TestOptions. ing objects: vehicleDetectorOptions.Table_Row. ing objects: vehicleDetectorOptions.Table_Row. eDetectorOptions.Table_Row IS_EQUAL_TO | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 10 | SET the following objects: vehicleDetectorOptions.Table_Row. | | | |
|-----------------------------------|---|------------------|-----------|--|
| 11 | GET the following objects: vehicleDetectorOp | tions.Table_Row. | Pass/Fail | |
| 12 | VERIFY vehicleDetectorOptions.Table_Row IS_EQUAL_TO OriginalOptions. | | | |
| Test Procedure Results | | | | |
| Tested By: Date Tested: Pass/Fail | | | | |
| Test Procedure Notes: | | | | |

C.3.13.7 Configure Vehicle Detector Queue Enabled

| Test Procedure: | | Configure Vehicle Detector Que | le Enabled | |
|---|---|--|--|-----------------|
| Description: | | This test case verifies that the A station to set a vehicle detector queues. | SC allows a managem to detect the presence | ient e of |
| Requirement(| s): | • 3.5.3.1.1.7 Configure Veh | icle Detector Queue E | nabled |
| Variable(s): | | MaxRows Table_Row OriginalOptions TestOptions | maxVehicleDetectors Int vehicleDetectorOptic vehicleDetectorOptic | s ons ons |
| Pass/Fail Criteria: The device under test shall pass every verification sterest case to pass the test case. | | p in this | | |
| Test Step Number | | Test Procedure Results | | |
| 1 | GET the follow | ng objects: maxVehicleDetectors. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the follow | ng objects: vehicleDetectorOptions. | Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalOptio | ns. | |
| 4 | ASSIGN Test | ptions EQUALS OriginalOptions XC | R 64. | |
| 5 | ASSIGN vehic | <pre>eDetectorOptions.Table_Row EQUA</pre> | LS TestOptions. | |
| 6 | SET the follow | ng objects: vehicleDetectorOptions. | Table_Row. | Pass/Fail |
| 7 | GET the follow | ng objects: vehicleDetectorOptions. | Table_Row. | Pass/Fail |
| 8 | VERIFY vehicleDetectorOptions.Table_Row IS_EQUAL_TO TestOptions. Pass/Fail | | Pass/Fail | |
| 9 | ASSIGN vehic | DetectorOptions.Table_Row EQUA | LS OriginalOptions. | |
| 10 | SET the follow | ng objects: vehicleDetectorOptions. | Table_Row. | Pass/Fail |
| 11 | GET the follow | ng objects: vehicleDetectorOptions. | Table_Row. | Pass/Fail |
| 12 | VERIFY vehicle OriginalOption | DetectorOptions.Table_Row IS_EC | QUAL_TO | Pass/Fail |
| | | lest Procedure Results | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.13.8 Configure Vehicle Detector Call Enabled

| Test Procedure: | Configure Vehicle Detector Call Enabled Procedure: | | | |
|---------------------|---|---|--|-----------|
| Description: | | This test case verifies that the ASC allows a management station to store if a call is placed for vehicle service upon actuation of a vehicle detector while the phase is not timing the Green interval. | | |
| Requirement(| s): | • 3.5.3.1.1.8 Configure | e Vehicle Detector Call Ena | ıbled |
| Variable(s): | | MaxRows Table_Row OriginalOptions TestOptions | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalOptionsvehicleDetectorOptionsTestOptionsvehicleDetectorOptions | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | Test Procedure Result | | | Results |
| 1 | GET the following objects: maxVehicleDetectors. Pass/Fai | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorOptions.Table_Row. | | Pass/Fail | |
| 3.1 | RECO | RD this information as Original | Options. | |
| 4 | ASSIGN TestC | ptions EQUALS OriginalOption | ıs XOR 128. | |
| 5 | ASSIGN vehicl | eDetectorOptions.Table_Row | EQUALS TestOptions. | |
| 6 | SET the followi | ing objects: vehicleDetectorOpt | tions.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorOp | tions.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicle TestOptions. | eDetectorOptions.Table_Row I | S_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN vehicl | eDetectorOptions.Table_Row | EQUALS OriginalOptions. | |
| 10 | SET the followi | ing objects: vehicleDetectorOpt | tions.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: vehicleDetectorOp | tions.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorOptions.Table_Row IS_EQUAL_TO OriginalOptions. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | | |

C.3.13.9 Configure Vehicle Detector Call Phase

| Test Procedure: | | Configure Vehicle Detector Call Phase | | |
|---------------------|---|--|--|---------------------|
| Description: | | This test case verifies that t station to assign call phase | the ASC allows a managemes to vehicle detectors. | ient |
| Requirement(| s): | • 3.5.3.1.1.9 Configure | e Vehicle Detector Call Pha | se |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalCallPhasevehicleDetectorCallPhaseTestCallPhasevehicleDetectorCallPhaseMaxPhasesInt | | s Phase Phase |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | PRE-CONDITI | ON 'The user shall know the m orts' | aximum number of phases | |
| 1.1 | RECORD this information as MaxPhases. | | | |
| 2 | GET the following objects: maxVehicleDetectors. | | Pass/Fail | |
| 2.1 | RECORD this information as MaxRows. | | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 4 | GET the following objects: vehicleDetectorCallPhase.Table_Row. | | Pass/Fail | |
| 4.1 | RECORD this information as OriginalCallPhase. | | | |
| 5 | ASSIGN TestCallPhase EQUALS RANDOM (1 TO MaxPhases). | | | |
| 5.1 | IF Tes | tCallPhase IS_EQUAL_TO Ori | ginalCallPhase. | |
| 5.1.1 | | GOTO step 5. | | |
| 6 | ASSIGN vehicl TestCallPhase | leDetectorCallPhase.Table_Ro | w EQUALS | |
| 7 | SET the follow | ing objects: vehicleDetectorCal | IPhase.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: vehicleDetectorCa | llPhase.Table_Row. | Pass/Fail |
| 9 | VERIFY vehicle TestCallPhase | eDetectorCallPhase.Table_Rov | w IS_EQUAL_TO | Pass/Fail |
| 10 | ASSIGN vehicl OriginalCallPha | leDetectorCallPhase.Table_Ro ase. | w EQUALS | |
| 11 | SET the follow | ing objects: vehicleDetectorCal | IPhase.Table_Row. | Pass/Fail |
| 12 | GET the follow | ing objects: vehicleDetectorCa | llPhase.Table_Row. | Pass/Fail |
| 13 | VERIFY vehicleDetectorCallPhase.Table_Row IS_EQUAL_TO OriginalCallPhase. | | | |
| - | - | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.13.10 Configure Vehicle Detector Switch Phase

| Test Procedure: | | Configure Vehicle Detector Switch Phase | | |
|---------------------|--|---|---|-------------------------|
| Description: | | This test case verifies that the ASC allows a management station to assign a switch phase to a vehicle detector to be switched upon turning green to when its assigned phase is YELLOW or RED. | | ient to be ase is |
| Requirement(| s): | • 3.5.3.1.1.10 Configu | re Vehicle Detector Switch | Phase |
| Variable(s): | | MaxRows OriginalSwitchPhase TestSwitchPhase Table_Row MaxPhases | maxVehicleDetectors vehicleDetectorSwite vehicleDetectorSwite Int Int | s chPhase chPhase |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | PRE-CONDITION 'The user shall know the maximum number of phases | | | |
| 1 1 | RECORD this information as MaxPhases | | | |
| 2 | GET the following objects: maxVehicleDetectors. | | | Pass/Fail |
| 2.1 | RECORD this information as MaxRows. | | | |
| 3 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 4 | GET the following objects: vehicleDetectorSwitchPhase.Table_Row. | | Pass/Fail | |
| 4.1 | RECORD this information as OriginalSwitchPhase. | | | |
| 5 | ASSIGN TestSwitchPhase EQUALS RANDOM (1 TO MaxPhases). | | | |
| 5.1 | IF TestSwitchPhase IS_EQUAL_TO OriginalSwitchPhase. | | | |
| 5.1.1 | | GOTO step 4. | | |
| 6 | ASSIGN vehicl TestSwitchPha | eDetectorSwitchPhase.Table_I se. | Row EQUALS | |
| 7 | SET the followi | ng objects: vehicleDetectorSwi | tchPhase.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: vehicleDetectorSw | itchPhase.Table_Row. | Pass/Fail |
| 9 | VERIFY vehicle TestSwitchPha | eDetectorSwitchPhase.Table_F se. | Row IS_EQUAL_TO | Pass/Fail |
| 10 | ASSIGN vehicl OriginalSwitch | eDetectorSwitchPhase.Table_I ^{>} hase. | Row EQUALS | |
| 11 | SET the followi | ng objects: vehicleDetectorSwi | tchPhase.Table_Row. | Pass/Fail |
| 12 | GET the follow | ing objects: vehicleDetectorSw | itchPhase.Table_Row. | Pass/Fail |
| 13 | VERIFY vehicle OriginalSwitch | eDetectorSwitchPhase.Table_F Phase. | Row IS_EQUAL_TO | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |

Test Procedure Notes:

C.3.13.11 Configure Vehicle Detector Delay Time

| Test Procedure: | Configure Vehicle Detector Delay Time | | | |
|---------------------|---|--|-----------------------------|-------------|
| Description: | ption: This test case verifies that the ASC allows a management station the period a detector actuation shall be delayed when the phase is not GREEN. | | nent d when | |
| Requirement(| s): | • 3.5.3.1.1.11 Configu | re Vehicle Detector Delay T | ſime |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalDetectorDelayvehicleDetectorDelayTestDetectorDelayvehicleDetectorDelay | | s y y |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxVehicleDetectors. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorDelay.Table_Row. Pass/F | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalDetectorDelay. | | | |
| 4 | ASSIGN TestDetectorDelay EQUALS RANDOM (0 TO 2550). | | | |
| 4.1 | IF TestDetectorDelay IS_EQUAL_TO OriginalDetectorDelay. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN Venici | eDetectorDelay. I able_Row EC | QUALS TestDetectorDelay. | Deee/Feil |
| 6 | SET the follow | ing objects: vehicleDetectorDel | ay. Lable_Row. | Pass/Fall |
| 1 | | any objects: venicleDetectorDe | | Pass/Fall |
| 8 | TestDetectorD | elay. | | Pass/Fail |
| 9 | ASSIGN vehicl OriginalDetecto | eDetectorDelay.Table_Row EC prDelay. | QUALS | |
| 10 | SET the followi | ing objects: vehicleDetectorDel | ay.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: vehicleDetectorDe | lay.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorDelay.Table_Row IS_EQUAL_TO OriginalDetectorDelay. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.13.12 Configure Vehicle Detector Extend Time

| Test Procedure: | Configure Vehicle Detector Extend Time | | | |
|---------------------|---|--|---|---------------|
| Description: | | This test case verifies that the station to configure the time detector is extended from the test of the statement of the sta | the ASC allows a managen e that actuation for a vehic he point of termination | nent :le |
| Requirement(| s): | • 3.5.3.1.1.12 Configu | re Vehicle Detector Extend | l Time |
| Variable(s): | | MaxRows Table_Row OriginalDetectorExtend TestDetectorExtend | maxVehicleDetector Int vehicleDetectorExte vehicleDetectorExte | s nd nd |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxVehicleDetectors. Pass | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorExtend.Table_Row. Pass | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalDetectorExtend. | | | |
| 4 | ASSIGN TestDetectorExtend EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestDetectorExtend IS_NOT_EQUAL_TO OriginalDetectorExtend. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN vehicl TestDetectorEx | eDetectorExtend.Table_Row E ktend. | QUALS | |
| 6 | SET the follow | ing objects: vehicleDetectorExt | end.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorExt | end.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicle TestDetectorEx | eDetectorExtend.Table_Row IS xtend. | S_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN vehicl OriginalDetecto | eDetectorExtend.Table_Row E prExtend. | QUALS | |
| 10 | SET the follow | ing objects: vehicleDetectorExt | end.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: vehicleDetectorExt | end.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorExtend.Table_Row IS_EQUAL_TO OriginalDetectorExtend. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

| C.3.13.13 | Configure | Vehicle | Detector | Queue | Limit | Time |
|-----------|-----------|---------|----------|-------|-------|------|
|-----------|-----------|---------|----------|-------|-------|------|

| Test Procedure: | | Configure Vehicle Detector Queue Limit Time | | |
|---------------------|---|--|--|-------------------------|
| Description: | | This test case verifies that station to configure the leng queue detector may continu | the ASC allows a managen gth of time that an actuatio ue into the phase green. | nent In from a |
| Requirement(| s): | • 3.5.3.1.1.13 Configu Time | re Vehicle Detector Queue | Limit |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalQueueLimitvehicleDetectorQueueLimitTestQueueLimitvehicleDetectorQueueLimit | | s ueLimit ueLimit |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxVehicleDetectors. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorQueueLimit.Table Row. Pass/ | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalQueueLimit. | | | |
| 4 | ASSIGN TestQueueLimit EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF vehicleDetectorQueueLimit.Table_Row IS_EQUAL_TO OriginalQueueLimit. | | | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN vehicl TestQueueLim | eDetectorQueueLimit.Table_R it. | ow EQUALS | |
| 7 | SET the follow | ing objects: vehicleDetectorQue | eueLimit.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: vehicleDetectorQu | eueLimit.Table_Row. | Pass/Fail |
| 9 | VERIFY vehicle TestQueueLim | eDetectorQueueLimit.Table_Ro it. | ow IS_EQUAL_TO | Pass/Fail |
| 10 | ASSIGN vehicleDetectorQueueLimit.Table_Row EQUALS OriginalQueueLimit. | | | |
| 11 | SET the following objects: vehicleDetectorQueueLimit.Table Row. Pass/ | | | Pass/Fail |
| 12 | GET the follow | ing objects: vehicleDetectorQu | eueLimit.Table_Row. | Pass/Fail |
| 13 | VERIFY vehicleDetectorQueueLimit.Table_Row IS_EQUAL_TO OriginalQueueLimit. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

| C.3.13.14 | Configure | Vehicle | Detector | No Activ | vity Time |
|-----------|-----------|---------|----------|----------|-----------|
|-----------|-----------|---------|----------|----------|-----------|

| Test Procedure: | t Cedure: | | | |
|---------------------|---|---|--|-------------------------|
| Description: | | This test case verifies that the station to configure the times a vehicle detector that does considered Failed. | the ASC allows a managen e in minutes that must elap s not exhibit an actuation is | nent ose before S |
| Requirement(| s): | • 3.5.3.1.1.14 Configu | re Vehicle Detector No Act | ivity Time |
| Variable(s): | | MaxRows Table_Row OriginalNoActivity TestNoActivity | maxVehicleDetector Int vehicleDetectorNoA vehicleDetectorNoA | s ctivity ctivity |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxVehicleDetectors. Pas | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorNoActivity.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalNoActivity. | | | |
| 4 | ASSIGN TestN | loActivity EQUALS RANDOM (| 0 TO 65535). | |
| 4.1 | IF Test | tNoActivity IS_NOT_EQUAL_T | O OriginalNoActivity. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN vehicl | eDetectorNoActivity.Table_Rov | w EQUALS TestNoActivity. | |
| 6 | SET the followi | ing objects: vehicleDetectorNo/ | Activity.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorNo. | Activity.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicle TestNoActivity. | eDetectorNoActivity.Table_Rov | w IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN vehicleDetectorNoActivity.Table_Row EQUALS OriginalNoActivity. | | | |
| 10 | SET the following objects: vehicleDetectorNoActivity.Table_Row. | | Pass/Fail | |
| 11 | GET the follow | ing objects: vehicleDetectorNo. | Activity.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorNoActivity.Table_Row IS_EQUAL_TO OriginalNoActivity. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

| C.3.13.15 Configure Vehicle Detector Maximum Presence | Time |
|---|------|
|---|------|

| Test Procedure: | | Configure Vehicle Detector Maximum Presence Time | | | |
|---------------------|---|---|--|---------------------------|--|
| Description: | | This test case verifies that the station to configure the main detector actuation may exhibite the detector is considered by the statement of | the ASC allows a managen ximum time in minutes tha ibited continuously before red Failed. | nent t a the | |
| Requirement(| s): | 3.5.3.1.1.15 Configu Presence Time | re Vehicle Detector Maxim | um | |
| Variable(s): | | MaxRows Table_Row OriginalMaxPresence TestMaxPresence | maxVehicleDetector Int vehicleDetectorMaxI vehicleDetectorMaxI | s Presence Presence | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this | |
| Test Step Number | Test Procedure | | | Results | |
| 1 | GET the following objects: maxVehicleDetectors. | | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | | |
| 3 | GET the following objects: vehicleDetectorMaxPresence.Table_Row. | | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalMaxPresence. | | | | |
| 4 | ASSIGN TestMaxPresence EQUALS RANDOM (0 TO 255). | | | | |
| 4.1 | IF Test | MaxPresence IS_EQUAL_TO | OriginalMaxPresence. | | |
| 4.1.1 | | GOTO step 4. | | | |
| 5 | ASSIGN vehicleDetectorMaxPresence.Table_Row EQUALS TestMaxPresence. | | | | |
| 6 | SET the followi | ing objects: vehicleDetectorMa | xPresence.Table_Row. | Pass/Fail | |
| 7 | GET the follow | ing objects: vehicleDetectorMa | xPresence.Table_Row. | Pass/Fail | |
| 8 | VERIFY vehicle TestMaxPrese | VERIFY vehicleDetectorMaxPresence.Table_Row IS_EQUAL_TO TestMaxPresence. | | | |
| 9 | ASSIGN vehicleDetectorMaxPresence.Table_Row EQUALS OriginalMaxPresence. | | | | |
| 10 | SET the following objects: vehicleDetectorMaxPresence.Table_Row. | | | Pass/Fail | |
| 11 | GET the follow | ing objects: vehicleDetectorMa | xPresence.Table_Row. | Pass/Fail | |
| 12 | VERIFY vehicleDetectorMaxPresence.Table_Row IS_EQUAL_TO OriginalMaxPresence. | | Pass/Fail | | |
| | | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | | |

C.3.13.16 Configure Vehicle Detector Erratic Counts

| Test Procedure: | Configure Vehicle Detector Erratic Counts | | | |
|---------------------|--|---|---|---------------------------|
| Description: | | This test case verifies that the station to configure the maximum vehicle detector may exhibit considered Failed. | the ASC allows a managem ximum number of actuation it within a minute before be | nent ns a eing |
| Requirement(| s): | • 3.5.3.1.1.16 Configu | re Vehicle Detector Erratic | Counts |
| Variable(s): | | MaxRows Table_Row OriginalErraticCounts TestErraticCounts | maxVehicleDetector Int vehicleDetectorErrat vehicleDetectorErrat | s icCounts icCounts |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification s test case to pass the test case. | | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the following objects: maxVehicleDetectors. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorErraticCounts.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalErraticCounts. | | | |
| 4 | ASSIGN TestErraticCounts EQUALS RANDOM (0 TO 255). | | M (0 TO 255). | |
| 4.1 | IF Test OriginalErratic0 | ErraticCounts IS_NOT_EQUA | L_TO | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN vehicl TestErraticCou | eDetectorErraticCounts.Table_ nts. | Row EQUALS | |
| 6 | SET the followi | ng objects: vehicleDetectorErra | aticCounts.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorErr | aticCounts.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicle TestErraticCou | eDetectorErraticCounts.Table_ nts. | Row IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN vehicleDetectorErraticCounts.Table_Row EQUALS OriginalErraticCounts. | | | |
| 10 | SET the following objects: vehicleDetectorErraticCounts.Table Row. | | Pass/Fail | |
| 11 | GET the follow | ing objects: vehicleDetectorErr | aticCounts.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorErraticCounts.Table_Row IS_EQUAL_TO OriginalErraticCounts. | | Row IS_EQUAL_TO | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | Date Tested: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.13.17 Configure Vehicle Detector Fail Time

| Test Procedure: | Configure Vehicle Detector Fail Time | | | |
|---------------------|---|--|---|-----------------|
| Description: | | This test case verifies that t station to configure the veh | the ASC allows a managen icle detector fail time. | nent |
| Requirement(| s): | • 3.5.3.1.1.17 Configu | re Vehicle Detector Fail Tir | ne |
| Variable(s): | | MaxRows Table_Row OriginalFailTime TestFailTime | maxVehicleDetector Int vehicleDetectorFailT vehicleDetectorFailT | s ime ime |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxVehicleDetectors. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: vehicleDetectorFailTime.Table_Row. Pass/Fa | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalFailTime. | | | |
| 4 | ASSIGN TestFailTime EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF Test | FailTime IS_EQUAL_TO Origi | nalFailTime. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN vehicl | eDetectorFailTime.Table_Row | EQUALS TestFailTime. | |
| 6 | SET the followi | ng objects: vehicleDetectorFai | ITime.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorFai | ITime.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicle TestFailTime. | eDetectorFailTime.Table_Row | IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN vehicleDetectorFailTime.Table_Row EQUALS OriginalFailTime. | | | |
| 10 | SET the following objects: vehicleDetectorFailTime.Table_Row. Pass/Fail | | | Pass/Fail |
| 11 | GET the following objects: vehicleDetectorFailTime.Table_Row. Pass/Fail | | | Pass/Fail |
| 12 | VERIFY vehicleDetectorFailTime.Table_Row IS_EQUAL_TO OriginalFailTime. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | | |

C.3.13.18 Configure Pedestrian Detector Description

| Test Procedure: | Configure Pedestrian Detector Description |
|--------------------|---|
|--------------------|---|

| Description: | This test case verifies that the ASC allows a management station to configure descriptions for pedestrian detectors. | | | nent tors. |
|---|--|--|-------------------------------------|---------------------|
| Requirement(s): • 3.5.3.1.3.1 Configure Pedestrian Detect | | | e Pedestrian Detector Des | cription |
| Variable(s): | | MaxRowsmaxPedestrianDetectorsTable_RowIntOriginalDescriptionpedestrianDetectorDescriptionn | | ctors Descriptio |
| Pass/Fail Crit | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedu | edure | | Results |
| 1 | GET the follow | ing objects: maxPedestrianDet | ectors. | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: pedestrianDetectorDescription.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalDescription | | | |
| 4 | ASSIGN pedestrianDetectorDescription.Table_Row EQUALS 'Test'. | | | |
| 5 | SET the follow | ing objects: pedestrianDetector | Description.Table_Row. | Pass/Fail |
| 6 | GET the follow | GET the following objects: pedestrianDetectorDescription.Table_Row. | | |
| 7 | VERIFY pedes 'Test'. | trianDetectorDescription.Table | _Row IS_EQUAL_TO | Pass/Fail |
| 8 | ASSIGN pedes OrignalDescrip | strianDetectorDescription.Table tion. | e_Row EQUALS | |
| 9 | SET the following objects: pedestrianDetectorDescription.Table_Row. | | Pass/Fail | |
| 10 | GET the following objects: pedestrianDetectorDescription.Table_Row. | | Pass/Fail | |
| 11 | VERIFY pedestrianDetectorDescription.Table_Row IS_EQUAL_TO OriginalDescription. | | | Pass/Fail |
| Test Procedur | e Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.13.19 Configure Pedestrian Detector Call Phase

| Test Procedure: | Configure Pedestrian Detector Call Phase | | |
|--------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to assign call phases for pedestrian detector inputs. | | |
| Requirement(s): | 3.5.2.1.13.1.2 Configure APS Push Button to Phase Association 3.5.3.1.3.2 Configure Pedestrian Detector Call Phase | | |
| Variable(s): | MaxRows Table_Row OriginalCallPhase MaxPhases TestCallPhase | maxPedestrianDetectors Int pedestrianDetectorCallPhase Int pedestrianDetectorCallPhase | |

| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification stores test case to pass the test case. | | p in this | |
|---------------------|---|--|-------------------------|-----------|
| Test Step Number | | Test Procedure | | Results |
| 1 | PRE-CONDITI | ON 'The user shall know the m ne ASC'. | aximum number of phases | |
| 1.1 | RECO | RD this information as MaxPha | ises. | |
| 2 | GET the follow | ing objects: maxPedestrianDet | ectors. | Pass/Fail |
| 2.1 | RECO | RD this information as MaxRov | VS. | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 4 | GET the follow | ing objects: pedestrianDetecto | rCallPhase.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as Original | CallPhase. | |
| 5 | ASSIGN TestC | ASSIGN TestCallPhase EQUALS RANDOM (0 TO MaxPhases). | | |
| 5.1 | IF TestCallPhase IS_EQUAL_TO OriginalCallPhase. | | | |
| 5.1.1 | GOTO step 5. | | | |
| 6 | ASSIGN pedestrianDetectorCallPhase.Table_Row EQUALS TestCallPhase. | | | |
| 7 | SET the followi | ing objects: pedestrianDetector | CallPhase.Table_Row. | Pass/Fail |
| 8 | GET the following objects: pedestrianDetectorCallPhase.Table_Row. | | Pass/Fail | |
| 9 | VERIFY pedestrianDetectorCallPhase.Table_Row IS_EQUAL_TO TestCallPhase. | | Pass/Fail | |
| 10 | ASSIGN pedestrianDetectorCallPhase.Table_Row EQUALS OriginalCallPhase. | | | |
| 11 | SET the following objects: pedestrianDetectorCallPhase.Table_Row. Pas | | | Pass/Fail |
| 12 | GET the follow | ing objects: pedestrianDetecto | rCallPhase.Table_Row. | Pass/Fail |
| 13 | VERIFY pedestrianDetectorCallPhase.Table_Row IS_EQUAL_TO OriginalCallPhase. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.13.20 Configure Pedestrian Detector No Activity Time

| Test Procedure: | Configure Pedestrian Dete | Configure Pedestrian Detector No Activity Time | |
|--------------------|--|---|--|
| Description: | This test case verifies that station to configure the tin a pedestrian detector that considered Failed. | This test case verifies that the ASC allows a management station to configure the time in minutes that must elapse before a pedestrian detector that does not exhibit an actuation is considered Failed. | |
| Requirement(s): | • 3.5.3.1.3.3 Configu Time | re Pedestrian Detector No Activity | |
| Variable(s): | MaxRows Table_Row OriginalNoActivity | maxPedestrianDetectors Int pedestrianDetectorNoActivity | |

| | | TestNoActivity | pedestrianDetectorN | oActivity |
|---------------------|---|---------------------------------|------------------------|-----------|
| Pass/Fail Crite | The device under test shall pass every verification step in this test case to pass the test case. | | p in this | |
| Test Step Number | Test Procedure Results | | | Results |
| 1 | GET the follow | ing objects: maxPedestrianDet | ectors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: pedestrianDetecto | rNoActivity.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | NoActivity. | |
| 4 | ASSIGN Table | _Row EQUALS RANDOM (0 T | O 65535). | |
| 4.1 | IF Test | NoActivity IS_EQUAL_TO Orig | ginalNoActivity. | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN pedestrianDetectorNoActivity.Table_Row EQUALS TestNoActivity. | | | |
| 6 | SET the following objects: pedestrianDetectorNoActivity.Table_Row. Pass/Fa | | Pass/Fail | |
| 7 | GET the following objects: pedestrianDetectorNoActivity.Table_Row. Pass/Fail | | Pass/Fail | |
| 8 | VERIFY pedestrianDetectorNoActivity.Table_Row IS_EQUAL_TO TestNoActivity. | | Pass/Fail | |
| 9 | ASSIGN pedestrianDetectorNoActivity.Table_Row EQUALS OriginalNoActivity. | | | |
| 10 | SET the followi | ng objects: pedestrianDetector | NoActivity.Table_Row. | Pass/Fail |
| 11 | GET the following objects: pedestrianDetectorNoActivity.Table_Row. Pass/ | | Pass/Fail | |
| 12 | VERIFY pedestrianDetectorNoActivity.Table_Row IS_EQUAL_TO OriginalNoActivity. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.13.21 Configure Pedestrian Detector Maximum Presence Fault Time

| Test Procedure: | Configure Pedestrian Detec | Configure Pedestrian Detector Maximum Presence Time | | |
|--------------------|---|---|--|--|
| Description: | This test case verifies that t station to configure the may actuation may exhibited cor detector is considered Faile | This test case verifies that the ASC allows a management station to configure the maximum time in minutes a detector actuation may exhibited continuously before the pedestrian detector is considered Failed. | | |
| Requirement(s): | 3.5.3.1.3.4 Configure Presence Time | 3.5.3.1.3.4 Configure Pedestrian Detector Maximum Presence Time | | |
| Variable(s): | MaxRows Table_Row OriginalMaxPresence | maxPedestrianDetectors Int pedestrianDetectorMaxPrese nce | | |

| | TestMaxPresence pedestrianDetectorMaxPresence nce | | laxPrese | |
|---------------------|--|---------------------------------|-------------------------|-----------|
| Pass/Fail Crite | Pass/Fail Criteria:The device under test shall pass every verification ste test case to pass the test case. | | p in this | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxPedestrianDet | ectors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: pedestrianDetector | rMaxPresence.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | MaxPresence. | |
| 4 | ASSIGN TestM | laxPresence EQUALS RANDC | OM (0 TO 255). | |
| 4.1 | IF TestMaxPresence IS_EQUAL_TO OriginalMaxPresence. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN maxPedestrianDetectors.Table_Row EQUALS TestMaxPresence. | | | |
| 6 | SET the followi | ng objects: pedestrianDetector | MaxPresence.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: pedestrianDetector | rMaxPresence.Table_Row. | Pass/Fail |
| 8 | VERIFY pedestrianDetectorMaxPresence.Table_Row IS_EQUAL_TO TestMaxPresence. | | Pass/Fail | |
| 9 | ASSIGN maxPedestrianDetectors.Table_Row EQUALS OriginalMaxPresence. | | | |
| 10 | SET the following objects: pedestrianDetectorMaxPresence.Table_Row. Pas | | Pass/Fail | |
| 11 | GET the following objects: pedestrianDetectorMaxPresence.Table_Row. | | Pass/Fail | |
| 12 | VERIFY pedestrianDetectorMaxPresence.Table_Row IS_EQUAL_TO OriginalMaxPresence. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.13.22 Configure Pedestrian Detector Erratic Counts

| Test Procedure: | Configure Pedestrian Detec | Configure Pedestrian Detector Erratic Counts | |
|--------------------|---|--|--|
| Description: | This test case verifies that t station to configure the may pedestrian detector must ex considered Failed. | This test case verifies that the ASC allows a management station to configure the maximum number of actuations a pedestrian detector must exhibit within a minute before being considered Failed. | |
| Requirement(s): | • 3.5.3.1.3.5 Configure | Pedestrian Detector Erratic Counts | |
| Variable(s): | MaxRows Table_Row OriginalErraticCounts | maxPedestrianDetectors Int pedestrianDetectorErraticCo unts | |

| | TestErraticCounts pedestrianDetectorErraticCo unts | | rraticCo | |
|---------------------|--|---------------------------------|---------------------------|-----------|
| Pass/Fail Crite | Pass/Fail Criteria:The device under test shall pass every verification ste test case to pass the test case. | | p in this | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxPedestrianDet | ectors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: pedestrianDetector | rErraticCounts.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | ErraticCounts. | |
| 4 | ASSIGN TestE | rraticCounts EQUALS RANDC | M (0 TO 255). | |
| 4.1 | IF Tes | tErraticCounts IS_EQUAL_TO | OriginalErraticCounts. | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN pedestrianDetectorErraticCounts.Table_Row EQUALS TestErraticCounts. | | | |
| 6 | SET the following objects: pedestrianDetectorErraticCounts.Table_Row. | | Pass/Fail | |
| 7 | GET the follow | ing objects: pedestrianDetector | rErraticCounts.Table_Row. | Pass/Fail |
| 8 | VERIFY pedestrianDetectorErraticCounts.Table_Row IS_EQUAL_TO TestErraticCounts. | | Pass/Fail | |
| 9 | ASSIGN pedestrianDetectorErraticCounts.Table_Row EQUALS OriginalErraticCounts. | | | |
| 10 | SET the follow | ing objects: pedestrianDetector | ErraticCounts.Table_Row. | Pass/Fail |
| 11 | GET the following objects: pedestrianDetectorErraticCounts.Table_Row. | | Pass/Fail | |
| 12 | VERIFY pedestrianDetectorErraticCounts.Table_Row IS_EQUAL_TO OriginalErraticCounts. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.13.23 Configure Pedestrian Detector Non-Lock Calls

| Test Procedure: | Configure Pedestrian Det | Configure Pedestrian Detector Non-Lock Calls | | |
|--------------------|---|--|--|--|
| Description: | This test case verifies that station to configure a peo for pedestrian timings. | This test case verifies that the ASC allows a management station to configure a pedestrian detector for non-locked calls for pedestrian timings. | | |
| Requirement(s): | • 3.5.3.1.3.6 Configu Calls | 3.5.3.1.3.6 Configure Pedestrian Detector Non-Lock Calls | | |
| Variable(s): | MaxRows Table_Row OriginalOptions TestOptions | maxVehicleDetectors Int pedestrianDetectorOptions pedestrianDetectorOptions | | |

| Pass/Fail Crit | Criteria: The device under test shall pass every verification step in this test case to pass the test case. | | | p in this |
|---------------------|---|---------------------------------|------------------------|-----------|
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPedestrianDet | ectors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the follow | ing objects: pedestrianDetecto | rOptions.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Options. | |
| 4 | ASSIGN TestO | ptions EQUALS OriginalOption | ns XOR 4. | |
| 5 | ASSIGN pedes | strianDetectorOptions.Table_R | ow EQUALS TestOptions. | |
| 6 | SET the followi | ng objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: pedestrianDetecto | rOptions.Table_Row. | Pass/Fail |
| 8 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO TestOptions. | | Pass/Fail | |
| 9 | ASSIGN pedestrianDetectorOptions.Table_Row EQUALS OriginalOptions. | | | |
| 10 | SET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail | |
| 11 | GET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail | |
| 12 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO OriginalOptions. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.13.24 Configure Pedestrian Detector for Alternate Pedestrian Timing

| Test Procedure: | Configure Pedestrian Detector for Alternate Pedestrian Timing | | |
|---------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to configure a pedestrian detector to place calls for alternate pedestrian timing instead of normal pedestrian timing. | | |
| Requirement(s): | 3.5.2.1.13.1.4 Configure Pedestrian Detector for Alternate Pedestrian Timing | | |
| Variable(s): | MaxRows Table_Row OriginalOptions TestOptions | maxVehicleDetectors Int pedestrianDetectorOptions pedestrianDetectorOptions | |
| Pass/Fail Criteria: | The device under test shall pa test case to pass the test case | ss every verification step in this | |

| Test Step Number | Test Procedure | | Results |
|---------------------|--|------------------------|-----------|
| 1 | GET the following objects: maxPedestrianDete | ectors. | Pass/Fail |
| 1.1 | RECORD this information as MaxRow | /S. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | Options. | |
| 4 | ASSIGN TestOptions EQUALS OriginalOption | is XOR 2. | |
| 5 | ASSIGN pedestrianDetectorOptions.Table_Ro | ow EQUALS TestOptions. | |
| 6 | SET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 7 | GET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 8 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO TestOptions. | | Pass/Fail |
| 9 | ASSIGN pedestrianDetectorOptions.Table_Row EQUALS OriginalOptions. | | |
| 10 | SET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail |
| 12 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO OriginalOptions. | | Pass/Fail |
| | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.13.25 Configure Pedestrian Detector for Presence Detection

| Test Procedure: | | Configure Pedestrian Detector for Presence Detection | | |
|---------------------|----------------|--|--|-----------------------|
| Description: | | This test case verifies that the ASC allows a management station to set a pedestrian detector to detect the presence of a pedestrian in a crosswalk instead of detecting a pedestrian call for service. | | |
| Requirement(| s): | 3.5.3.1.3.7 Configure Pedestrian Detector for Presence Detection | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalOptionspedestrianDetectorOptionsTestOptionspedestrianDetectorOptions | | s ptions ptions |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure Results | | |
| 1 | GET the follow | ing objects: maxPedestrianDetectors. | | Pass/Fail |

| 1.1 | RECORD this information as MaxRov | VS. | |
|----------------|--|------------------------|-----------|
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as Original | Options. | |
| 4 | ASSIGN TestOptions EQUALS OriginalOption | ns XOR 1. | |
| 5 | ASSIGN pedestrianDetectorOptions.Table_Re | ow EQUALS TestOptions. | |
| 6 | SET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 7 | GET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 8 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS EQUAL TO TestOptions. | | Pass/Fail |
| 9 | ASSIGN pedestrianDetectorOptions.Table_Row EQUALS OriginalOptions. | | |
| 10 | SET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail |
| 12 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO OriginalOptions. | | Pass/Fail |
| | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.13.26 Configure Pedestrian Detector for Delayed Walk

| Test Procedure: | | Configure Pedestrian Detector for Delayed Walk | | |
|---------------------|---|--|----------------------|----------------|
| Description: | | This test case verifies that the ASC allows a management station to set a pedestrian detector to activate delayed walk settings for pedestrian phases. | | ient I walk |
| Requirement(| s): | 3.5.3.1.3.8 Configure Pedestrian Detector for Delayed Walk | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalOptionspedestrianDetectorOptionsTestOptionspedestrianDetectorOptions | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure Res | | Results |
| 1 | GET the following objects: maxPedestrianDetectors. Pass/F | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the follow | ing objects: pedestrianDetecto | orOptions.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Origina | IOptions. | |

| 4 | ASSIGN TestChannelFlash EQUALS Original | Options AND 11. | |
|------------------------|---|------------------------|-----------|
| 4.1 | NOTE 'Toggle Bit 4 LOW because Bits 3 & 4 of pedestrianDetectorOptions cannot both be TRUE'. | | |
| 5 | ASSIGN pedestrianDetectorOptions.Table_Ro | ow EQUALS TestOptions. | |
| 6 | SET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 7 | GET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 8 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO TestOptions. | | Pass/Fail |
| 9 | ASSIGN pedestrianDetectorOptions.Table_Row EQUALS OriginalOptions. | | |
| 10 | SET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: pedestrianDetectorOptions.Table_Row. | | Pass/Fail |
| 12 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO OriginalOptions. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.13.27 Configure Pedestrian Detector for Advanced Walk

| Test Procedure: | | Configure Pedestrian Detector for Advanced Walk | | |
|---|---|--|---------------------|-----------|
| This test case verifies that the ASC allows a managDescription:station to set a pedestrian detector to activate advasettings for pedestrian phases. | | nat the ASC allows a managem an detector to activate advanc phases. | ent ed walk | |
| Requirement(| s): | 3.5.3.1.3.9 Configure Pedestrian Detector for Advanced Walk | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalOptionspedestrianDetectorOptionsTestOptionspedestrianDetectorOptions | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure Re | | Results |
| 1 | GET the following objects: maxPedestrianDetectors. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: pedestrianDetectorOptions.Table_Row. Pass/Fail | | Pass/Fail | |
| 3.1 | RECO | RD this information as Orig | inalOptions. | |
| 4 | ASSIGN TestC | hannelFlash EQUALS Orig | jinalOptions AND 7. | |

| | - | | |
|------------------------|--|------------------------|-----------|
| 4.1 | NOTE 'Toggle Bit 3 LOW because Bits 3 & 4 of pedestrianDetectorOptions cannot both be TRUE'. | | |
| 5 | ASSIGN pedestrianDetectorOptions.Table_Ro | ow EQUALS TestOptions. | |
| 6 | SET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 7 | GET the following objects: pedestrianDetector | Options.Table_Row. | Pass/Fail |
| 8 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO TestOptions. | | Pass/Fail |
| 9 | ASSIGN pedestrianDetectorOptions.Table_Row EQUALS OriginalOptions. | | |
| 10 | SET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail |
| 11 | GET the following objects: pedestrianDetectorOptions.Table_Row.Table_Row. | | Pass/Fail |
| 12 | VERIFY pedestrianDetectorOptions.Table_Row.Table_Row IS_EQUAL_TO OriginalOptions. | | Pass/Fail |
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.13.28 Determine Maximum Number of Vehicle Detectors

| Test Procedure: | | Determine Maximum Number of Vehicle Detectors | | |
|---|--|---|-----------|--|
| Description: This test case verifies that the ASC allows a management station to determine the maximum number of vehicle dete supported by ASC and verifies that the ASC supports the minimum number of vehicle detectors required by the use | | ent detectors the user. | | |
| Requirement(| s): • 3.5.3.1.5.1 Determine Maximum Number of Vehicle Detectors | | | |
| Variable(s): | UserMinVehicleDetectors maxVehicleDetectors | | S | |
| Pass/Fail Crite | Fail Criteria: The device under test shall pass every verification step in this test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure | Results | |
| 1 | USER-ACTION 'Determine the number of vehicle detectors required as specified in FR ID 3.5.3.1.5.1 of the PRL.' | | | |
| 1.1 | RECORD this information as UserMinVehicleDetectors. | | | |
| 2 | GET the following objects: maxVehicleDetectors. Pass/Fail | | Pass/Fail | |
| 3 | VERIFY maxVehicleDetectors IS_NOT_LESS_THAN Pass/Fail UserMinVehicleDetectors. | | | |
| | | Test Procedure Results | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

| C.3.13.29 Determine Maximum Number of Venicle Detector Se | C.3.13.29 | Determine Maximum | Number of | Vehicle Detector S | ets |
|---|-----------|-------------------|-----------|--------------------|-----|
|---|-----------|-------------------|-----------|--------------------|-----|

| Test Procedure: | | Determine Maximum Number of Vehicle Detector Sets | | |
|---------------------|--|--|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of vehicle detector sets supported by ASC and verifies that the ASC supports the minimum number of vehicle detector sets required by the user. | | |
| Requirement(| • 3.5.3.1.5.2 Determine Maximum Number of Vehic Detector Sets | | nicle | |
| Variable(s): | UserMinVehicleDetectorSets maxVehicleDetectorSets | | | Sets |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | Results | |
| 1 | USER-ACTION 'Determine the number of vehicle detector sets required as specified in FR ID 3.5.3.1.5.2 of the PRL.' | | | |
| 1.1 | RECORD this information as UserMinVehicleDetectorSets. | | | |
| 2 | GET the following objects: maxVehicleDetectorSets. | | Pass/Fail | |
| 3 | VERIFY maxVehicleDetectorSets IS_NOT_LESS_THAN UserMinVehicleDetectorSets. | | Pass/Fail | |
| | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.13.30 Determine Maximum Number of Pedestrian Detectors

| Test Procedure: | Determine Maximum Number of Pedestrian Detectors | | |
|--------------------|---|--|--|
| Description: | This test case verifies that the ASC allows a management station to determine the maximum number of pedestrian detectors supported by ASC and verifies that the ASC supports the minimum number of pedestrian detectors required by the user. | | |
| Requirement(s): | 3.5.3.1.5.3 Determine Maximum Number of Pedestrian Detectors | | |
| Variable(s): | UserMinPedestrianDetectors maxPedestrianDetectors | | |

| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
|-------------------------|---|---|-------------------------------------|-----------|
| Test Step Number | Test Procedure | | | Results |
| 1 | USER-ACTION 'Determine the number of pedestrian detectors required as specified in FR ID 3.5.3.1.5.3 of the PRL.' | | | |
| 1.1 | RECORD this information as UserMinPedestrianDetectors. | | | |
| 2 | GET the following objects: maxPedestrianDetectors. | | Pass/Fail | |
| 3 | VERIFY maxPedestrianDetectors IS_NOT_LESS_THAN UserMinPedestrianDetectors. | | | Pass/Fail |
| Test Procedure Results | | | | |
| Tested By: Date Tested: | | Pass/Fail | | |
| Test Procedure N | lotes: | | | |

C.3.13.31 Determine Maximum Number of Pedestrian Detector Sets

| Test Procedure: | | Determine Maximum Number of Pedestrian Detector Sets | | |
|---|---|---|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the maximum number of pedestrian detector sets supported by ASC and verifies that the ASC supports the minimum number of pedestrian detector sets required by the user. | | |
| Requirement(s): • 3.5.3.1.5.4 Determine Maximu Detector Sets | | e Maximum Number of Peo | lestrian | |
| Variable(s): | UserMinPedestrianDetectorSets maxPedestrianDetectorSet | | rSets | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure | | | Results |
| 1 | USER-ACTION 'Determine the number of pedestrian detector sets required as specified in FR ID 3.5.3.1.5.4 of the PRL.' | | | |
| 1.1 | RECORD this information as UserMinPedestrianDetectorSets. | | | |
| 2 | GET the following objects: maxPedestrianDetectorSets. Pass/ | | | Pass/Fail |
| 3 VERIFY maxPedes UserMinPedestriar | | edestrianDetectorSets IS_NOT_LESS_THAN Pass/F trianDetectorSets. | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

| C.3.13.32 | Monitor Active | Vehicle | Detector | Actuations |
|-----------|----------------|---------|----------|------------|
|-----------|----------------|---------|----------|------------|

| Test Procedure: | | Monitor Active Vehicle Detector Actuations | | | |
|---------------------|--|---|-------------------------------------|------------|--|
| Description: | | This test case verifies that the ASC allows a management station to identify vehicle detectors that are actuated. | | | |
| Requirement(s): | | • 3.5.3.2.1 Monitor Active Vehicle Detector Actuations | | | |
| Variable(s): | | MaxRows Table_Row | maxVehicleDetectorGroups Int | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this | |
| Test Step Number | Test Procedure | | | Results | |
| 1 | GET the following objects: maxVehicleDetectorGroups. | | orGroups. | Pass/Fail | |
| 1.1 | RECORD this information as Table_Row. | | | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | | |
| 2.1 | GET th vehicleDetecto | e following objects: StatusGroupActive.Table_Row. | | Pass/Fail | |
| | Test Procedure Results | | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | | |

C.3.13.33 Monitor Active Pedestrian Detector Actuations

| Test Procedure: | | Monitor Active Pedestrian Detector Actuations | | | |
|---------------------|--|--|--------------------|----------|--|
| Description: | | This test case verifies that the ASC allows a management station to identify pedestrian detectors that are actuated. | | | |
| Requirement(s): | | • 3.5.3.2.2 Monitor Active Pedestrian Detector Actuations | | | |
| | | MaxRows | maxPedestrianDetec | torGroup | |
| Variable(s): | | | S | | |
| | | Table_Row | Int | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | | |
| Test Step Number | | Test Proce | dure | Results | |

| 1 | GET the following objects: maxPedestrianDet | ectorGroups. | Pass/Fail |
|------------|--|-----------------|-----------|
| 1.1 | RECORD this information as Table_R | Row. | |
| 2 | FOR Table_Row from 1 to MaxRows. | | |
| 2.1 | GET the following objects: pedestrianDetectorStatusGroupActive.Table_ | Row. | Pass/Fail |
| | | | |
| | Test Procedure Results | | |
| Tested By: | Test Procedure Results | Date Tested: | Pass/Fail |

C.3.13.34 Monitor Vehicle Detector Alarm Status

| Test Procedure: | | Monitor Vehicle Detector Alarm Status | | |
|---------------------|---|--|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to identify vehicle detectors with active alarms. | | |
| Requirement(s): | | • 3.5.3.3.1.1 Monitor Vehicle Detector Alarm Status | | |
| Variable(s): | | MaxRows Table_Row | maxVehicleDetector Int | Groups |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxVehicleDetect | orGroups. | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: vehicleDetectorStatusGroupAlarms.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | ; ; | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.13.35 Monitor Vehicle Detector Faults from Controller

| Test Procedure: | Monitor Vehicle Detector Faults from Controller |
|--------------------|---|
|--------------------|---|

E

| Description: | | This test case verifies that the ASC allows a management station to identify vehicle detector faults as identified by the ASC. | | | |
|--|--|--|--|----------------------|---------------------------------------|
| Requirement(s): | | 3.5.3.3.1.2 Monitor Vehicle Detector Faults from Controller | | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowInt | | 6 | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | | |
| | Test Procedure | | | | |
| Test Step Number | | Test Proce | dure | | Results |
| Test Step Number 1 | GET the follow | Test Proce ing objects: maxVehicleD | dure Detectors | s. | Results Pass/Fail |
| Test Step Number 1 1.1 | GET the follow RECO | Test Proce ing objects: maxVehicleD RD this information as Ma | dure Detectors axRows. | 3. | Results Pass/Fail |
| Test Step Number 1 1.1 2 | GET the follow RECO FOR Table_Ro | Test Proce ing objects: maxVehicleD RD this information as Ma ow from 1 to MaxRows. | dure Detectors axRows. | 5. | Results Pass/Fail |
| Test Step Number 1 1.1 2 2.1 | GET the follow RECO FOR Table_Ro GET th | Test Proce ing objects: maxVehicleD RD this information as Ma ow from 1 to MaxRows. ne following objects: vehic | dure Detectors axRows. | torAlarms.Table_Row. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 2.1 | GET the follow RECO FOR Table_Rc GET th | Test Proce ing objects: maxVehicleD RD this information as Ma ow from 1 to MaxRows. ne following objects: vehic Test Procedure Re | dure Detectors axRows. CleDetect esults | torAlarms.Table_Row. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 2.1 Tested By: | GET the follow RECO FOR Table_Rc GET th | Test Proce ing objects: maxVehicleD RD this information as Ma ow from 1 to MaxRows. ne following objects: vehic Test Procedure Re | dure Detectors axRows. cleDetect esults T | torAlarms.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail |

C.3.13.36 Monitor Vehicle Detector Faults from Detector

| Test Procedure: | | Monitor Vehicle Detector Faults from Detector | | |
|---------------------|--|---|---|------------|
| Description: | | This test case verifies that the ASC allows a management station to identify vehicle detector faults as identified by the detector. | | |
| Requirement(| ement(s): • 3.5.3.3.1.3 Monitor Vehicle Detector Faults from Detector | | | m Detector |
| Variable(s): | | MaxRows maxVehicleDetectors Table_Row Int | | 'S |
| Pass/Fail Criteria: | | The device under tes test case to pass the | t shall pass every verification ste test case. | ep in this |
| Test Step Number | Test Procedure Res | | Results | |
| 1 | GET the following objects: maxVehicleDetectors. Pass/ | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: vehicleReportedDetectorAlarms.Table_Row. | | Pass/Fail | |
| | | Test Procedure R | esults | |
| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.13.37 Monitor Pedestrian Detector Alarm Status

| Test Procedure: | | Monitor Pedestrian Detector Alarm Status | | |
|---------------------|--|---|---------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify pedestrian detectors with active alarms. | | |
| Requirement(| s): | • 3.5.3.3.2.1 Monitor Pedestrian Detector Alarm Status | | |
| Variable(s): | | MaxRows maxPedestrianDetectorG s Table_Row Int | | torGroup |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPedestrianDe | tectorGroups. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRo | WS. | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1.1 | GET the following objects: pedestrianDetectorStatusGroupAlarms.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | 6 | |
| Tested By: | | Date Tested: | | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.13.38 Monitor Pedestrian Detector Faults

| Test Procedure: | Monitor Pedestrian Detector Faults | | |
|--------------------|---|----------------------------|--|
| Description: | This test case verifies that the ASC allows a management station to identify pedestrian detector faults as identified by the ASC. | | |
| Requirement(s): | 3.5.3.3.2.2 Monitor Pedestrian Detector Faults from Controller | | |
| Variable(s): | MaxRows Table_Row | maxVehicleDetectors Int | |

| Pass/Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | p in this | | |
|--|---|---|-----------|--|
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | GET the following objects: maxVehicleDetectors. | | |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: vehicleDetectorAlarms.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: Date Tested: | | Pass/Fail | | |
| Test Procedure Notes: | | | | |

C.3.13.39 Control Vehicle Detector Reset

| Test Procedure: | | Control Vehicle Detector Reset | | |
|---|--|---|-----------------|-----------------|
| Description: This test case verifies that the ASC allows a managen station to reset a vehicle detector. | | nagement | | |
| Requirement(| s): | 3.5.3.4.1 Control Vehicle Detector Reset | | |
| Variable(s): | | MaxRows maxVehicleDetectors Table_Row Int | | tectors |
| Pass/Fail Crite | eria: | ria: The device under test shall pass every verification step in t test case to pass the test case. | | on step in this |
| Test Step Number | | Test Procedure | 9 | Results |
| 1 | GET the follow | ing objects: maxVehicleDeteo | ctors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRo | OWS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 | TO MaxRows). | |
| 3 | ASSIGN vehic | SSIGN vehicleDetectorReset.Table_Row EQUALS 1. | | |
| 4 | SET the follow | SET the following objects: vehicleDetectorReset.Table_Row. Pass/ | | Pass/Fail |
| 5 | USER-ACTION 'Verify the vehicle detector has reset.' | | | |
| | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.13.40 Control Pedestrian Detector Reset

Control Pedestrian Detector Reset

| Test Procedure: | | | | | |
|---------------------|---|---|-----------|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to reset a pedestrian detector. | | ment | |
| Requirement(| s): | • 3.5.3.4.2 Control Pedestrian Detector Reset | | | |
| Variable(s): | | MaxRows maxPedestrianDetectors Table_Row Int | | | ectors |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | tep in this | |
| Test Step Number | | Test Procedure R | | Results | |
| 1 | GET the follow | ing objects: maxPedes | strianDet | ectors. | Pass/Fail |
| 1.1 | RECO | RD this information as | MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RAND | OM (1 T | O MaxRows). | |
| 3 | ASSIGN pedes | strianDetectorReset.Ta | ble_Rov | v EQUALS 1. | |
| 4 | SET the following objects: pedestrianDetectorReset.Table_Row. Pass/FactorReset.Table_Row. | | Pass/Fail | | |
| 5 | USER-ACTION 'Verify the pedestrian detector has reset.' | | | | |
| | | Test Procedure | Results | | |
| Tested By: | | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | | |

C.3.13.41 Control Detector Diagnostic Reset

| Test Procedure: | | Control Detector Diagnostic Reset | | |
|---------------------|---|--|--|--|
| Description: | | This test case verifies that the ASC allows a management station to reset detector diagnostic counters | | |
| Requirement(| s): | 3.5.3.4.3 Control Detector Diagnostic Reset | | |
| Variable(s): | | | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure Results | | |
| 1 | ASSIGN unitDe | IGN unitDetectorDiagnosticReset_Row EQUALS 1. | | |
| 2 | SET the following objects: unitDetectorDiagnosticReset. Pass/Fail | | | |
| | | Test Procedure Results | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.13.42 Control Vehicle Detector Actuation

| Test Procedure: | | Control Vehicle Detector Actuation | | |
|---------------------|---|---|----------------------------------|--|
| Description: | | This test case verifies that the ASC allows a managen station to place or remove artificial actuations on veh detectors. | nent icle | |
| Requirement(| s): | • 3.5.3.4.4 Control Vehicle Detector Actuation | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorTable_RowIntOriginalActuationvehicleDetectorContTestActuationvehicleDetectorContActuationvehicleDetectorContActuationvehicleDetectorCont | Groups trolGroup trolGroup | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification ste test case to pass the test case. | ∍p in this | |
| Test Step Number | Test Procedure Results | | Results | |
| 1 | GET the following objects: maxVehicleDetectorGroups. Pass/Fail | | | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorControlGroupActuation.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalActuation. | | | |
| 4 | ASSIGN TestActuation EQUALS RANDOM (0 TO 255). | | | |
| 4.1 | IF TestActuation IS_EQUAL_TO OriginalActuation. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN vehicl TestActuation. | eDetectorControlGroupActuation.Table_Row EQUALS | | |
| 6 | SET the following objects: vehicleDetectorControlGroupActuation.Table Row. | | Pass/Fail | |
| 7 | GET the following objects: Pass/Fa | | Pass/Fail | |
| 8 | VERIFY vehicleDetectorControlGroupActuation.Table_Row IS_EQUAL_TO TestActuation. | | Pass/Fail | |
| 9 | ASSIGN vehicl OriginalActuati | eDetectorControlGroupActuation.Table_Row EQUALS on. | | |
| 10 | SET the follow vehicleDetecto | ng objects: rControlGroupActuation.Table_Row. | Pass/Fail | |
| 11 | GET the follow vehicleDetecto | ing objects: rControlGroupActuation.Table_Row. | Pass/Fail | |

| 12 | 12 VERIFY vehicleDetectorControlGroupActuation.Table_Row IS_EQUAL_TO OriginalActuation. | | |
|------------------------|--|-----------------|-----------|
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |

C.3.13.43 Control Pedestrian Detector Actuation

| Test Procedure: | | Control Pedestrian Detector Actuation | | |
|---------------------|--|--|---|--------------------------------------|
| Description: | | This test case verifies that the ASC allows a management station to place or remove artificial actuations on pedestrian detectors. | | |
| Requirement(| s): | • 3.5.3.4.5 Contro | ol Pedestrian Detector Actuatio | 'n |
| Variable(s): | | MaxRowsmaxPedestrianDetectorGroupsTable_RowIntOriginalActuationpedestrianDetectorControlGrouTestActuationpedestrianDetectorControlGrouActuationActuationActuationpedestrianDetectorControlGrouActuationActuation | | orGroups ntrolGroup ntrolGroup |
| Pass/Fail Crit | eria: | The device under test test case to pass the test | shall pass every verification st est case. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxPedestrianDetectorDetectors. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: pedestrianDetectorControlGroupActuation.Table_Row. | | Pass/Fail | |
| 3.1 | RECO | RD this information as Or | iginalActuation. | |
| 4 | ASSIGN TestA | Actuation EQUALS RAND | OM (0 TO 255). | |
| 4.1 | IF Te | stActuation IS_EQUAL_T | O OriginalActuation. | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN pedes EQUALS Test/ | ASSIGN pedestrianDetectorControlGroupActuation.Table_Row EQUALS TestActuation. | | |
| 6 | SET the following objects: pedestrianDetectorControlGroupActuation.Table_Row. | | | Pass/Fail |
| 7 | GET the following objects: pedestrianDetectorControlGroupActuation.Table_Row. | | Pass/Fail | |
| 8 | VERIFY pedestrianDetectorControlGroupActuation.Table_Row IS_EQUAL_TO TestActuation. | | Pass/Fail | |
| 9 | ASSIGN pedes EQUALS Origi | strianDetectorControlGrounalActuation. | pActuation.Table_Row | |
| 10 | SET the follow pedestrianDete | ing objects: ectorControlGroupActuation | on.Table_Row. | Pass/Fail |

| 11 GET the following objects: pedestrianDetectorControlGroupActuation.Table_Row. | | | | |
|---|------------------------|--|--|--|
| 12 VERIFY pedestrianDetectorControlGroupActuation.Table_Row IS_EQUAL_TO OriginalActuation. | | | | |
| _ | Test Procedure Results | | | |
| Tested By: Date Tested: | | | | |
| Test Procedure Notes: | | | | |

C.3.14 Detector Data Collection

C.3.14.1 Monitor Detector Data Sequence

| Test Procedure: Monitor De | | Monitor Detector Data Sequ | ience | |
|---|---|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station determine the sequence number for volume/occupancy data collection that is used to detect duplicate or missing reports. | | |
| Requirement(| s): | 3.5.3.5.1.1 Monitor Detector Data Sequence | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: The device test case to | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: volumeOccupancySequence. | | Sequence. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.14.2 Monitor Vehicle Volume Data

| Test Procedure: | Monitor Vehicle Volume Data | |
|--------------------|--|---|
| Description: | This test case verifies that the A station to retrieve volume data r | SC allows a management recorded by the ASC. |
| Requirement(s): | • 3.5.3.5.1.2 Monitor Vehic | le Volume Data |
| Variable(s): | MaxRows Table_Row | maxVehicleDetectors Int |

.

| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification sto ase. | ep in this |
|---------------------|--|--|-------------------------------------|------------|
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | GET the following objects: maxVehicleDetectors. | | |
| 1.1 | RECO | RECORD this information as MaxRows. | | |
| 2 | FOR Table_Ro | ow from 1 to MaxRows. | | |
| 2.1 | GET the following objects: detectorVolume.Table_Row. | | Pass/Fail | |
| | - | Test Procedure Results | ; | |
| Tested By: | /: Date Tested: | | Pass/Fail | |
| Test Procedure | Notes: | | | |

C.3.14.3 Monitor Vehicle Occupancy Data

| Test Procedure: | est rocedure: Monitor Vehicle Occupancy Data | | | |
|--|---|--|---|------------|
| Description: | | This test case verifies that the ASC allows a management station to retrieve occupancy data recorded by the ASC. | | |
| Requirement(| s): | • 3.5.3.5.1.3 Monitor Vehicle Occupancy Data | | |
| Variable(s): | MaxRows maxVehicleDetectors Table_Row Int | | 'S | |
| Pass/Fail Criteria: The device und test case to pa | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure Resu | | Results | |
| 1 | GET the follow | ing objects: maxVehicleDetect | ors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 2.1 | GET th | e following objects: detectorO | ccupancy.Table_Row. | Pass/Fail |
| | - | Test Procedure Results | i de la companya de l | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | - | - |

C.3.14.4 Monitor Vehicle Average Speed

| Test Procedure: | Monitor Vehicle Average Speed |
|--------------------|-------------------------------|
|--------------------|-------------------------------|

| Description: | | This test case verifies that the ASC allows a management station to retrieve average speed data recorded by the ASC. | | |
|--|--|---|---|---------------------------------------|
| Requirement(s): | | • 3.5.3.5.1.3 Monitor Vehicle Average Speed | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowInt | | s |
| Pass/Fail Criteria: | | The device under test shal test case to pass the test c | l pass every verification ste ase. | ep in this |
| | Test Procedure | | | |
| Test Step Number | | Test Procedure | | Results |
| Test Step Number 1 | GET the follow | Test Procedure ing objects: maxVehicleDetect | tors. | Results Pass/Fail |
| Test Step Number 1 1.1 | GET the follow RECO | Test Procedure ing objects: maxVehicleDetect RD this information as MaxRo | tors. ws. | Results Pass/Fail |
| Test Step Number11.12 | GET the follow RECO FOR Table_Ro | Test Procedure ing objects: maxVehicleDetect RD this information as MaxRo ow from 1 to MaxRows. | tors. WS. | Results Pass/Fail |
| Test Step Number 1 1.1 2 2.1 | GET the follow RECO FOR Table_Ro GET th | Test Procedure ing objects: maxVehicleDetect RD this information as MaxRo ow from 1 to MaxRows. ne following objects: detectorA | ors. ws. vgSpeed.Table_Row | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 | GET the follow RECO FOR Table_Ro GET th | Test Procedure ing objects: maxVehicleDetect RD this information as MaxRo ow from 1 to MaxRows. ne following objects: detectorA Test Procedure Results | tors. ws. vgSpeed.Table_Row. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 2 2.1 Tested By: | GET the follow RECO FOR Table_Ro GET th | Test Procedure ing objects: maxVehicleDetect RD this information as MaxRo ow from 1 to MaxRows. ne following objects: detectorA Test Procedure Results | tors. ws. vgSpeed.Table_Row. Date Tested: | Results Pass/Fail Pass/Fail Pass/Fail |

C.3.14.5 Monitor Vehicle Detector Sample Time

| Test Procedure: | | Monitor Vehicle Detector D | ata Sample Time | |
|------------------------|--|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to determine the time the sample period of the detector ends. | | |
| Requirement(| ment(s): • 3.5.3.5.1.5 Monitor Vehicle Detector Data Sample Time | | ole Time | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: detectorSampleTir | ne. | Pass/Fail |
| Test Procedure Results | | | | |
| Tested By: | Tested By: Date Tes | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.14.6 Monitor Vehicle Detector Sample Duration

Monitor Vehicle Detector Data Sample Duration

| Test Procedure: | | | | |
|---------------------|--|---|------------------|-----------|
| Description: | Description:This test case verifies that the ASC allows a management station to determine the duration of the data collection per for the vehicle detectors. | | nent n period | |
| Requirement(| s): | 3.5.3.5.1.6 Monitor Vehicle Detector Data Sample Duration | | ble |
| Variable(s): | | | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: detectorSampleDuration. | | iration. | Pass/Fail |
| | | Test Procedure Results | j | |
| Tested By: | ested By: Date Tested: | | Date Tested: | Pass/Fail |
| Test Procedure | lotes: | | | |

C.3.14.7 Monitor Vehicle Detector Sample Duration

| Test Procedure: | | Monitor Pedestrian Detecto | or Data Sequence | |
|---------------------|---|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify the current sequence number for pedestrian detector data collection. | | |
| Requirement(| s): | • 3.5.3.5.2.1 Monitor Pedestrian Detector Data Sequence | | equence |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: pedestrianDetect | | rSequence. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.14.8 Monitor Pedestrian Counts

| Test Procedure: | | Monitor Pedestrian Counts | | |
|---------------------|---|---|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to view the number of pedestrians currently detected within a detection zone during a defined sample period. | | |
| Requirement(| s): | • 3.5.3.5.2.2 Monitor Pedestrian Counts | | |
| Variable(s): | | MaxRows Table_Row | maxPedestrianDetectors Int | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxPedestrianDe | tectors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | FOR Table_Row from 1 to MaxRows. | | | |
| 2.1 | GET the following objects: pedestrianDetectorVolume.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.14.9 Monitor Pedestrian Detector Actuations

| Test Procedure: | | Monitor Pedestrian Detector Actuations | |
|---------------------|--|---|---------------------|
| Description: | This test case verifies that the ASC allows a managementDescription:station to identify the number of pedestrian actuations collo during a sample period. | | nent s collected |
| Requirement(| s): | 3.5.3.5.2.3 Monitor Pedestrian Detector Actuations | |
| Variable(s): | | MaxRows maxPedestrianDetectors Table_Row Int | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | |
| Test Step Number | | Test Procedure | |
| 1 | GET the follow | lowing objects: maxPedestrianDetectors. | |
| 1.1 | RECO | RECORD this information as MaxRows. | |

| 2 | FOR Table_Row from 1 to MaxRows. | | |
|----------------|---|-----------------|-----------|
| 2.1 | GET the following objects: pedestrianDetectorActuations.Table_Row. | | Pass/Fail |
| | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.14.10 Monitor Pedestrian Services

| Test Procedure: Monitor Pedestrian Services | | | | |
|---|---|--|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify the number of pedestrian services (number of times the ped transitioned from don't walk to walk) during the defined sample period. | | |
| Requirement(| s): | • 3.5.3.5.2.3 Monitor Pedestrian Services | | |
| Variable(s): | s): MaxRows maxPedestrianDetectors Table_Row Int | | ctors | |
| Pass/Fail Criteria: The device under test shall pass every verificatest case to pass the test case. | | l pass every verification ste ase. | ep in this | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxPedestrianDe | tectors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRo | WS. | |
| 2 | FOR Table_Ro | w from 1 to MaxRows. | | |
| 2.1 | GET the following objects: pedestrianDetectorServices.Table_Row. | | | Pass/Fail |
| | | Test Procedure Results | 6 | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | Notes: | | | |

C.3.14.11 Monitor Pedestrian Detector Data Sample Time

| Test Procedure: | Monitor Pedestrian Detector Data Sample Time |
|--------------------|---|
| Description: | This test case verifies that the ASC allows a management station to identify the time of the pedestrian detector data collection period ends. |
| Requirement(s): | 3.5.3.5.2.5 Monitor Pedestrian Detector Data Sample Time |

| Variable(s): | | | | |
|-------------------------|--|---|-------------------------------------|-----------|
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the following objects: pedestrianDetectorSar | | rSampleTime. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: Date Tested: | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.14.12 Monitor Pedestrian Detector Data Sample Duration

| Test Procedure: | | Monitor Pedestrian Detecto | or Data Sample Duration | |
|---------------------|-----------------|---|---|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify the duration of the pedestrian data collection period. | | |
| Requirement(s): | | 3.5.3.5.2.6 Monitor Pedestrian Detector Data Sample Duration | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: pedestrianDetecto | rSampleDuration. | Pass/Fail |
| | | Test Procedure Results | i de la companya de l | |
| Tested By: | Date Tested: | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.14.13 Configure Vehicle Detector Data Sample Period

| Test Procedure: | Configure Vehicle Detector Data Sample Period |
|--------------------|--|
| Description: | This test case verifies that that ASC allows a management station to configure the volume/occupancy/speed collection period. |
| Requirement(s): | • 3.5.3.5.3.1 Configure Detector Data Sample Period |

.

| Variable(s): | | OriginalPeriod TestPeriod | volumeOccup volumeOccup | ancyPeriod ancyPeriod |
|---------------------|--|---|---|--------------------------|
| Pass/Fail Criteria: | | The device under test test case to pass the t | shall pass every verificat est case. | ion step in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: volumeOccu | bancyPeriod. | Pass/Fail |
| 1.1 | RECORD this information as OriginalPeriod. | | | |
| 2 | ASSIGN TestPeriod EQUALS RANDOM (0 TO 3600). | | | |
| 2.1 | IF TestPeriod IS_EQUAL_TO OriginalPeriod. | | | |
| 2.1.1 | GOTO step 2. | | | |
| 3 | ASSIGN volumeOccupancyPeriod EQUALS TestPeriod. | | | |
| 4 | SET the following objects: volumeOccupancyPeriod. Pass | | Pass/Fail | |
| 5 | GET the following objects: volumeOccupancyPeriod. Pass/Factorial P | | Pass/Fail | |
| 6 | VERIFY volumeOccupancyPeriod IS_EQUAL_TO TestPeriod. Pass/F | | Pass/Fail | |
| 7 | ASSIGN volumeOccupancyPeriod EQUALS OriginalPeriod. | | | |
| 8 | SET the following objects: volumeOccupancyPeriod. Pass/Fa | | Pass/Fail | |
| 9 | GET the following objects: volumeOccupancyPeriod. Pass/F | | Pass/Fail | |
| 10 | VERIFY volum | eOccupancyPeriod IS_E | QUAL_TO OriginalPeriod. | Pass/Fail |
| | - | Test Procedure Re | sults | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.14.14 Configure Pedestrian Detector Data Sample Period

| Test Configure Pedestrian Data Collection Sample Period Procedure: Configure Pedestrian Data Collection Sample Period | | | | |
|---|---|---|--------------|----------------|
| Description: | | This test case verifies that that ASC allows a management station to configure the sample period for collecting pedestrian detector data. | | |
| Requirement(s): • 3.5.3.5.3.2 Configure Pedestrian Data Collection S Period | | on Sample | | |
| Variable(s): | | OriginalPeriod pedestrianDetectorPeriod TestPeriod pedestrianDetectorPeriod | | eriod eriod |
| Pass/Fail Criteria: The device under test shall pass every verification ste test case to pass the test case. | | p in this | | |
| Test Step Number | | Test Procedure Result | | Results |
| 1 | GET the following objects: pedestrianDetectorPeriod. Pass/F | | Pass/Fail | |
| 1.1 | RECO | RD this information as Ori | ginalPeriod. | |
| 2 | ASSIGN TestP | eriod EQUALS RANDOM | (0 TO 3600). | |

| Test Procedure | Notes: | | | |
|----------------|---|--|-----------|--|
| Tested By: | | Date Tested: | Pass/Fail | |
| - | Test Procedure Results | | | |
| 10 | VERIFY pedestrianDetectorPeriod IS_EQUAL_TO OriginalPeriod. | | Pass/Fail | |
| 9 | GET the following objects: pedestrianDetectorPeriod. Pass/Fai | | | |
| 8 | SET the following objects: pedestrianDetectorPeriod. Pass/ | | Pass/Fail | |
| 7 | ASSIGN pedestrianDetectorPeriod EQUALS OriginalPeriod. | | | |
| 6 | VERIFY pedestrianDetectorPeriod IS_EQUA | VERIFY pedestrianDetectorPeriod IS_EQUAL_TO TestPeriod. Pass/F | | |
| 5 | GET the following objects: pedestrianDetecto | rPeriod. | Pass/Fail | |
| 4 | SET the following objects: pedestrianDetector | Period. | Pass/Fail | |
| 3 | ASSIGN v pedestrianDetectorPeriod EQUALS | S TestPeriod. | | |
| 2.1.1 | GOTO step 2. | GOTO step 2. | | |
| 2.1 | IF TestPeriod IS_EQUAL_TO Origina | lPeriod. | | |

C.3.14.15 Configure Vehicle Speed Detectors

| Test Procedure: | | tors | | |
|--|---|--|---|---|
| Description: | | This test case verifies that the A station to enable/disable vehicle data. | ASC allows a managen e detectors collecting s | nent speed |
| Requirement(| s): | • 3.5.3.5.3.3 Configure Vehicle Speed Detectors | | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalOptions2vehicleDetectorOptions2TestOptions2vehicleDetectorOptions2 | | s ons2 ons2 |
| Pass/Fail Criteria: | | The device under test shall past test case to pass the test case. | s every verification ste | ep in this |
| | | | | |
| Test Step Number | | Test Procedure | | Results |
| Test Step Number 1 | GET the follow | Test Procedure ing objects: maxVehicleDetectors. | | Results Pass/Fail |
| Test Step Number 1 1.1 | GET the follow RECO | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. | | Results Pass/Fail |
| Test Step Number 1 1.1 2 | GET the follow RECO ASSIGN Table | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. _Row EQUALS RANDOM (1 TO M | axRows). | Results Pass/Fail |
| Test Step Number11.123 | GET the follow RECO ASSIGN Table GET the follow | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. Row EQUALS RANDOM (1 TO M ing objects: vehicleDetectorOptions | axRows). 2.Table_Row. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 | GET the follow RECO ASSIGN Table GET the follow RECO | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. Row EQUALS RANDOM (1 TO M ing objects: vehicleDetectorOptions RD this information as OriginalOptic | axRows). 2.Table_Row. ons2. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestC | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. Row EQUALS RANDOM (1 TO M ing objects: vehicleDetectorOptions RD this information as OriginalOptio Options2 EQUALS OriginalOptions2 | axRows). 2.Table_Row. ons2. XOR 1. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 5 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestO ASSIGN vehicl | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. Row EQUALS RANDOM (1 TO M ing objects: vehicleDetectorOptions RD this information as OriginalOptio options2 EQUALS OriginalOptions2 eDetectorOptions2.Table_Row EQ | axRows). 2.Table_Row. ons2. XOR 1. UALS TestOptions2. | Results Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 5 6 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestO ASSIGN vehicl SET the follow | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. Row EQUALS RANDOM (1 TO M ing objects: vehicleDetectorOptions RD this information as OriginalOption options2 EQUALS OriginalOptions2 eDetectorOptions2.Table_Row EQ ing objects: vehicleDetectorOptions | axRows). 2.Table_Row. ons2. XOR 1. UALS TestOptions2. 2.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 5 6 7 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestO ASSIGN vehicl SET the follow GET the follow | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. Row EQUALS RANDOM (1 TO M ing objects: vehicleDetectorOptions RD this information as OriginalOption ptions2 EQUALS OriginalOptions2 eDetectorOptions2.Table_Row EQ ing objects: vehicleDetectorOptions ing objects: vehicleDetectorOptions | axRows). 2.Table_Row. ons2. XOR 1. UALS TestOptions2. 2.Table_Row. 2.Table_Row. | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Test Step Number 1 1.1 2 3 3.1 4 5 6 7 8 | GET the follow RECO ASSIGN Table GET the follow RECO ASSIGN TestO ASSIGN vehicl SET the follow GET the follow VERIFY vehicl TestOptions2. | Test Procedure ing objects: maxVehicleDetectors. RD this information as MaxRows. Row EQUALS RANDOM (1 TO M ing objects: vehicleDetectorOptions RD this information as OriginalOption options2 EQUALS OriginalOptions2 eDetectorOptions2.Table_Row EQU ing objects: vehicleDetectorOptions ing objects: vehicleDetectorOptions eDetectorOptions2.Table_Row IS_F | axRows). 2.Table_Row. ons2. XOR 1. UALS TestOptions2. 2.Table_Row. 2.Table_Row. EQUAL_TO | Results Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 10 | SET the following objects: vehicleDetectorOptions2.Table_Row. Pass/Fa | | | |
|------------------------|---|---|-----------|--|
| 11 | GET the following objects: vehicleDetectorOp | GET the following objects: vehicleDetectorOptions2.Table_Row. | | |
| 12 | VERIFY vehicleDetectorOptions2.Table_Row IS_EQUAL_TO OriginalOptions2. | | | |
| Test Procedure Results | | | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure | Notes: | | | |

C.3.14.16 Configure Single Detector Speed Mode

| Test Procedure: | | Configure Single Detecto | or Speed Mode | | |
|------------------------|--|---|--|-------------------|--|
| Description: | | This test case verifies th station to configure a de paired detector. | This test case verifies that the ASC allows a management station to configure a detector to calculate speed without a paired detector. | | |
| Requirement(s): | | • 3.5.3.5.3.4 Config | ure Single Detector Speed M | ode | |
| Variable(s): | | MaxRows Table_Row OriginalOptions2 TestOptions2 | maxVehicleDetector Int vehicleDetectorOptic vehicleDetectorOptic | s ons2 ons2 | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | p in this | | |
| Test Step Number | | Test Procedu | re | Results | |
| 1 | GET the following objects: maxVehicleDetectors. | | Pass/Fail | | |
| 1.1 | RECORD this information as MaxRows. | | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | | |
| 3 | GET the following objects: vehicleDetectorOptions2.Table_Row. | | Pass/Fail | | |
| 3.1 | RECORD this information as OriginalOptions2. | | | | |
| 4 | ASSIGN TestOptions2 EQUALS OriginalOptions2 XOR 4. | | | | |
| 5 | ASSIGN vehic | leDetectorOptions2.Table_F | Row EQUALS TestOptions2. | | |
| 6 | SET the follow | ing objects: vehicleDetector | Options2.Table_Row. | Pass/Fail | |
| 7 | GET the follow | ing objects: vehicleDetector | Options2.Table_Row. | Pass/Fail | |
| 8 | VERIFY vehicleDetectorOptions2.Table_Row IS_EQUAL_TO TestOptions2. | | | Pass/Fail | |
| 9 | ASSIGN vehicleDetectorOptions2.Table_Row EQUALS OriginalOptions2. | | | | |
| 10 | SET the follow | ing objects: vehicleDetector | Options2.Table_Row. | Pass/Fail | |
| 11 | GET the follow | ing objects: vehicleDetector | Options2.Table_Row. | Pass/Fail | |
| 12 | VERIFY vehicl OriginalOption | eDetectorOptions2.Table_R s2. | ow IS_EQUAL_TO | Pass/Fail | |
| Test Procedure Results | | | | | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.14.17 Configure Paired Detector

| Test Procedure: | | Configure Paired Detector | |
|---------------------|--|--|----------------------|
| Description: | | This test case verifies that the ASC allows a manager station to assign a detector to be paired to another de | nent etector. |
| Requirement(| s): | • 3.5.3.5.3.5 Configure Paired Detector | |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalDetectorvehicleDetectorPairedTestDetectorvehicleDetectorPaired | Detector Detector |
| Pass/Fail Crite | eria: | The device under test shall pass every verification storest case to pass the test case. | ep in this |
| Test Step Number | | Test Procedure | Results |
| 1 | GET the following objects: maxVehicleDetectors. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | |
| 3 | GET the following objects: vehicleDetectorPairedDetector.Table_Row. | | Pass/Fail |
| 3.1 | RECORD this information as OriginalDetector. | | |
| 4 | ASSIGN TestDetector EQUALS RANDOM (0 TO MaxRows). | | |
| 4.1 | IF TestDetector IS_EQUAL_TO OriginalDetector OR TestDetector IS_EQUAL_TO Table_Row. | | |
| 4.1.1 | | GOTO step '4'. | |
| 5 | ASSIGN vehic TestDetector. | eDetectorPairedDetector.Table_Row EQUALS | |
| 6 | SET the follow | ing objects: vehicleDetectorPairedDetector.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorPairedDetector.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicleDetectorPairedDetector.Table_Row IS_EQUAL_TO TestDetector. | | |
| 9 | ASSIGN vehicleDetectorPairedDetector.Table_Row EQUALS OriginalDetector. | | |
| 10 | SET the following objects: vehicleDetectorPairedDetector.Table Row. Pass/Fai | | |
| 11 | GET the follow | ing objects: vehicleDetectorPairedDetector.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicl OriginalDetect | eDetectorPairedDetector.Table_Row IS_EQUAL_TO pr. | Pass/Fail |
| | | Test Procedure Results | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.14.18 Configure Paired Detector Placement

| Test Procedure: | | Configure Paired Detector I | Placement | |
|---------------------|---|--|--|---------------------|
| Description: | | This test case verifies that t station to specify if a detect a paired placement. | the ASC allows a managen tor is a leading or trailing o | nent Jetector in |
| Requirement(| s): | • 3.5.3.5.3.6 Configure | e Paired Detector Placeme | nt |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalOptions2vehicleDetectorOptions2TestOptions2vehicleDetectorOptions2 | | s ons2 ons2 |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ≱p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxVehicleDetectors. | | Pass/Fail | |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorOptions2.Table_Row. | | Pass/Fail | |
| 3.1 | RECO | RD this information as Original | Options2. | |
| 4 | ASSIGN Test | Options2 EQUALS OriginalOptic | ons2 XOR 2. | |
| 5 | ASSIGN vehic | leDetectorOptions2.Table_Row | / EQUALS TestOptions2. | |
| 6 | SET the follow | ing objects: vehicleDetectorOp | tions2.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorOp | tions2.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicl TestOptions2. | eDetectorOptions2.Table_Row | IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN vehicl OriginalOption | leDetectorOptions2.Table_Row s2. | EQUALS | |
| 10 | SET the follow | ing objects: vehicleDetectorOp | tions2.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: vehicleDetectorOp | tions2.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorOptions2.Table_Row IS_EQUAL_TO OriginalOptions2. | | | Pass/Fail |
| | | Test Procedure Results | i | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.14.19 Configure Paired Detector Spacing

| Test Configure Paired Detector Spacing | | | | |
|--|--|---|--|------------------------|
| Description: | | This test case verifies that station to configure a dista paired placement. | the ASC allows a managen nce between two detectors | in a |
| Requirement(| s): | • 3.5.3.5.3.7 Configure | e Paired Detector Spacing | |
| Variable(s): | | MaxRows Table_Row OriginalSpacing TestSpacing | maxVehicleDetectors Int vehicleDetectorPairedDetect vehicleDetectorPairedDetect | orSpacing orSpacing |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxVehicleDetecte | ors. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | VS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 3 | GET the following objects: vehicleDetectorPairedDetectorSpacing.Table_Row. | | | Pass/Fail |
| 3.1 | RECORD this information as OriginalSpacing. | | | |
| 4 | ASSIGN TestSpacing EQUALS RANDOM (0 TO 65535). | | | |
| | IF Te | stSpacing IS_EQUAL_TO Orig | inalSpacing. | |
| | GUTU Step 4. ASSIGN vehicleDetectorPairedDetectorSpacing Table, Row FOLIALS | | | |
| 5 | TestSpacing. | | | |
| 6 | SET the follow | ing objects: rPairedDetectorSpacing.Table_ | _Row. | Pass/Fail |
| 7 | GET the follow vehicleDetecto | ing objects: rPairedDetectorSpacing.Table __ | Row. | Pass/Fail |
| 8 | VERIFY vehicle IS_EQUAL_TO | eDetectorPairedDetectorSpaci) TestSpacing. | ng.Table_Row | Pass/Fail |
| 9 | ASSIGN vehicl OriginalSpacin | eDetectorPairedDetectorSpaci g. | ng.Table_Row EQUALS | |
| 10 | SET the following vehicleDetector | ng objects: rPairedDetectorSpacing.Table | Row. | Pass/Fail |
| 11 | GET the follow vehicleDetector | ing objects: rPairedDetectorSpacing.Table | Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorPairedDetectorSpacing.Table_Row Pass/Fai | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.14.20 Configure Average Vehicle Length

| Test Procedure: | | Configure Average Vehicle Length | | | |
|---------------------|--|--|--|--------------------------|--|
| Description: | | This test case verifies that t station to store the average | the ASC allows a managen vehicle length for a detect | ient tion zone. | |
| Requirement(| s): | • 3.5.3.5.3.8 Configure | e Average Vehicle Length | | |
| Variable(s): | | MaxRows Table_Row OriginalLength TestLength | maxVehicleDetectors Int vehicleDetectorAvgVeh vehicleDetectorAvgVeh | icleLength icleLength | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in t test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxVehicleDetecto | ors. | Pass/Fail | |
| 1.1 | RECO | RD this information as MaxRov | VS. | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | | |
| 3 | GET the follow vehicleDetecto | ing objects: rAvgVehicleLength.Table_Row | l. | Pass/Fail | |
| 3.1 | RECORD this information as OriginalLength. | | | | |
| 4 | ASSIGN TestLength EQUALS RANDOM (1 TO 4000). | | | | |
| 4.1 | IF TestLength IS_EQUAL_TO OriginalLength. | | | | |
| 4.1.1 | GOTO step 4. | | | | |
| 5 | ASSIGN vehicl TestLength. | eDetectorAvgVehicleLength.Ta | able_Row EQUALS | | |
| 6 | SET the follow vehicleDetecto | ing objects: rAvgVehicleLength.Table_Row | <i>.</i> | Pass/Fail | |
| 7 | GET the follow vehicleDetecto | ing objects: rAvgVehicleLength.Table_Row | · | Pass/Fail | |
| 8 | VERIFY vehicle TestLength. | eDetectorAvgVehicleLength.Ta | ble_Row IS_EQUAL_TO | Pass/Fail | |
| 9 | ASSIGN vehicl OriginalLength | eDetectorAvgVehicleLength.Ta | able_Row EQUALS | | |
| 10 | SET the follow vehicleDetecto | ing objects: rAvgVehicleLength.Table_Row | · | Pass/Fail | |
| 11 | GET the follow vehicleDetecto | ing objects: rAvgVehicleLength.Table_Row | ·. · | Pass/Fail | |
| 12 | VERIFY vehicl OriginalLength | Y vehicleDetectorAvgVehicleLength.Table_Row IS_EQUAL_TO Pass/l | | Pass/Fail | |
| - | | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | - | | |

C.3.14.21 Configure Average Vehicle Length

| Test Procedure: | | Configure Vehicle Detection | n Zone Length | |
|---------------------|---|--|--|-----------------|
| Description: | | This test case verifies that station to configure the len | the ASC allows a managen gth of a vehicle detection z | nent cone. |
| Requirement(| s): | • 3.5.3.5.3.9 Configure | e Vehicle Detection Zone L | ength |
| Variable(s): | | MaxRowsmaxVehicleDetectorsTable_RowIntOriginalLengthvehicleDetectorLengthTestlengthvehicleDetectorLength | | s jth jth |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxVehicleDetecto | ors. | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxRows | | |
| 2 | ASSIGN Table Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: vehicleDetectorLength.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalLength. | | | |
| 4 | ASSIGN TestLength EQUALS RANDOM (1 TO 65535). | | | |
| 4.1 | IF TestLength IS_EQUAL_TO OriginalLength. | | | |
| 4.1.1 | GOTO step 4. | | | |
| 5 | ASSIGN vehicl | eDetectorLength.Table_Row E | QUALS TestLength. | |
| 6 | SET the followi | ing objects: vehicleDetectorLer | ngth.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: vehicleDetectorLer | ngth.Table_Row. | Pass/Fail |
| 8 | VERIFY vehicle | eDetectorLength.Table_Row IS | S_EQUAL_TO TestLength. | Pass/Fail |
| 9 | ASSIGN vehicl | eDetectorLength.Table_Row E | QUALS OriginalLength. | |
| 10 | SET the followi | ing objects: vehicleDetectorLer | ngth.Table_Row. | Pass/Fail |
| 11 | GET the follow | ing objects: vehicleDetectorLer | ngth.Table_Row. | Pass/Fail |
| 12 | VERIFY vehicleDetectorLength.Table_Row IS_EQUAL_TO OriginalLength. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15 Connected Vehicle Interfaces

C.3.15.1 Configure ASC Communications Port for RSU

| Test Procedure: | | Configure ASC Communications Port for RSU | | |
|---------------------|--|--|---|---------------|
| Description: | | This test case verifies that the station to configure the phy exchanging data with an RS | that ASC allows a manager rsical communications por SU. | nent t for |
| Requirement(| s): | • 3.5.4.1.1 Configure / | ASC Communications Port | for RSU |
| Variable(s): | | OriginalPort TestPort MaxPorts | rsuCommPort rsuCommPort Int | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | PRE-CONDITION 'The user shall know the maximum number of | | aximum number of | |
| 1 1 | PECORD this information as MayPorts | | | |
| 2 | GET the following objects: rsuCommPort. | | Pass/Fail | |
| 2.1 | RECORD this information as OriginalPort. | | | |
| 3 | ASSIGN TestP | ort EQUALS RANDOM (0 TO | MaxPorts). | |
| 3.1 | IF TestPort IS EQUAL TO OriginalPort. | | | |
| 3.1.1 | | GOTO step 3. | | |
| 4 | ASSIGN rsuCo | mmPort EQUALS TestPort. | | |
| 5 | SET the followi | ing objects: rsuCommPort. | | Pass/Fail |
| 6 | GET the follow | ing objects: rsuCommPort. | | Pass/Fail |
| 7 | VERIFY rsuCo | mmPort IS_EQUAL_TO TestP | ort. | Pass/Fail |
| 8 | ASSIGN rsuCo | ommPort EQUALS OriginalPort | | |
| 9 | SET the followi | SET the following objects: rsuCommPort. | | Pass/Fail |
| 10 | GET the follow | ing objects: rsuCommPort. | | Pass/Fail |
| 11 | VERIFY rsuCo | mmPort IS_EQUAL_TO Origin | alPort. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.2 Configure Logical RSU Ports and Address

| Test Procedure: | Configure Logical RSU Ports and Address |
|--------------------|---|
|--------------------|---|

| Description: | | This test case verifies that the ASC allows a management station to configure an address and port on the RSU for the ASC to send data to. | |
|---------------------|--|---|------------|
| Requirement(s): | | • 3.5.4.1.2 Configure Logica RSU Ports and Address | |
| Variable(s): | | MaxRowsmaxRsuPortsTable_RowIntOrignalPortNamersuPortNameOriginalPortNumberrsuPortNumberTestPortNumberrsuPortNumberOriginalAddressrsuPortAddress | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | | Test Procedure | Results |
| 1 | GET the follow | ing objects: maxRsuPorts. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO MaxRows). | |
| 3 | GET the follow | ing objects: rsuPortName.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as OriginalPortName. | |
| 4 | GET the following objects: rsuPortNumber.Table Row. Pass/F | | |
| 4.1 | RECORD this information as OriginalPortNumber. | | |
| 5 | GET the following objects: rsuPortAddress.Table_Row. Pass/Fa | | |
| 5.1 | RECORD this information as OriginalAddress. | | |
| 6 | ASSIGN TestPortNumber EQUALS RANDOM (0 TO 65535) | | |
| 6.1 | IF Te | stPortNumber IS_EQUAL_TO OriginalPortNumber | |
| 6.1.1 | | GOTO step 6. | |
| 7 | ASSIGN rsuPortName.Table_Row EQUALS 'Test'. | | |
| 8 | ASSIGN rsuPortNumber.Table_Row EQUALS TestPortNumber | | |
| 9 | ASSIGN rsuPortAddress.Table_Row EQUALS '0.0.0'. | | |
| 10 | SET the followi rsuPortNumber | ng objects: rsuPortName.Table_Row, .Table_Row. rsuPortAddress.Table_Row. | Pass/Fail |
| 11 | GET the follow rsuPortNumber | ing objects: rsuPortName.Table_Row, .Table_Row. rsuPortAddress.Table_Row. | Pass/Fail |
| 12 | VERIFY rsuPo | rtName.Table_Row IS_EQUAL_TO 'Test'. | Pass/Fail |
| 13 | VERIFY rsuPo | rtNumber.Table_Row IS_EQUAL_TO TestPortNumber | Pass/Fail |
| 14 | VERIFY rsuPo | rtAddress.Table_Row IS_EQUAL_TO '0.0.0.0'. | Pass/Fail |
| 15 | ASSIGN rsuPo | rtName.Table_Row EQUALS OriginalPortName. | |
| 16 | ASSIGN rsuPo | rtNumber.Table_Row EQUALS OriginalPortName | |
| 17 | ASSIGN rsuPo | rtAddress.Table_Row EQUALS OriginalAddress. | |
| 18 | SET the followi rsuPortNumber | ng objects: rsuPortName.Table_Row, .Table_Row. rsuPortAddress.Table_Row. | Pass/Fail |
| 19 | GET the follow rsuPortNumber | ing objects: rsuPortName.Table_Row, .Table_Row. rsuPortAddress.Table_Row. | Pass/Fail |
| 20 | VERIFY rsuPo | rtName.Table_Row IS_EQUAL_TO OriginalPortName. | Pass/Fail |
| 21 | VERIFY rsuPo | rtNumber.Table_Row IS_EQUAL_TO OriginalPortName | Pass/Fail |
| 22 | VERIFY rsuPo | rtAddress.Table_Row IS_EQUAL_TO OriginalAddress. | Pass/Fail |
| | | Test Procedure Results | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.15.3 Configure RSU Interface Polling Period

| Test Procedure: | | Configure RSU Interface Po | Illing Period | |
|---------------------|--|--|--|------------|
| Description: | | This test case verifies that station to configure the tim | the ASC allows a managen e between polls to an RSU | nent |
| Requirement(| s): | • 3.5.4.1.3 Configure | RSU Interface Polling Perio | od |
| Variable(s): | | MaxRows Table_Row OriginalPeriod TestPeriod | maxRsuPorts Int rsuPortPollingPeriod rsuPortPollingPeriod | d d |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | |
| 1 | GET the following objects: maxRsuPorts. | | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: rsuPortPollingPeriod.Table_Row. | | Pass/Fail | |
| 3.1 | RECORD this information as OriginalPeriod. | | | |
| 4 | ASSIGN TestPeriod EQUALS RANDOM (0 TO 65535) | | | |
| 4.1 | IF Te | stPeriod IS_EQUAL_TO Origin | alPeriod | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | ASSIGN rsuPc | ortPollingPeriod.Table_Row EC | UALS TestPeriod. | |
| 6 | SET the follow | ing objects: rsuPortPollingPeric | od.Table_Row. | Pass/Fail |
| 7 | GET the follow | ing objects: rsuPortPollingPerio | od.Table_Row. | Pass/Fail |
| 8 | VERIFY rsuPo | rtPollingPeriod.Table_Row IS_ | EQUAL_TO TestPeriod. | Pass/Fail |
| 9 | ASSIGN rsuPc | ortPollingPeriod.Table_Row EC | UALS OriginalPeriod. | |
| 10 | SET the follow | ing objects: rsuPortPollingPeric | od.Table_Row. | Pass/Fail |
| 11 | GET the following objects: rsuPortPollingPeriod.Table_Row. | | Pass/Fail | |
| 12 | VERIFY rsuPo OriginalPeriod. | ERIFY rsuPortPollingPeriod.Table_Row IS_EQUAL_TO Pass/FiginalPeriod. | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.4 Configure RSU Interface Watchdog

| Test Procedure: | Configure RSU Interface Watchdog | | | | |
|---------------------|---|--|--|-----------|--|
| Description: | | This test case verifies that t station to configure the wat not activity is detected on a reported. | This test case verifies that the ASC allows a management station to configure the watchdog time that must elapse whole not activity is detected on a port before a No Activity Fault is reported. | | |
| Requirement(| s): | • 3.5.4.1.4 Configure F | RSU Interface Watchdog | | |
| Variable(s): | | MaxRows Table_Row OriginalTime TestTime | maxRsuPorts Int rsuPortWatchdogTin rsuPortWatchdogTin | ne | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste se. | p in this | |
| Test Step Number | | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxRsuPorts. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | | |
| 3 | GET the following objects: rsuPortWatchdogTime.Table_Row. Pas | | Pass/Fail | | |
| 3.1 | RECORD this information as OriginalTime. | | | | |
| 4 | ASSIGN TestTime EQUALS RANDOM (0 TO 65535). | | | | |
| 4.1 | IF Te | stTime IS_EQUAL_TO Original | Time | | |
| 4.1.1 | | GOTO step 4. | | | |
| 5 | ASSIGN rsuPo | rtWatchdogTime.Table_Row E | QUALS TestTime. | | |
| 6 | SET the followi | ng objects: rsuPortWatchdogTi | me.Table_Row. | Pass/Fail | |
| 7 | GET the follow | ing objects: rsuPortWatchdogT | ime.Table_Row. | Pass/Fail | |
| 8 | VERIFY rsuPo | rtWatchdogTime.Table_Row IS | _EQUAL_TO TestTime. | Pass/Fail | |
| 9 | ASSIGN rsuPo | rtWatchdogTime.Table_Row E | QUALS OriginalTime. | | |
| 10 | SET the followi | ng objects: rsuPortWatchdogTi | me.Table_Row. | Pass/Fail | |
| 11 | GET the follow | ing objects: rsuPortWatchdogT | ime.Table_Row. | Pass/Fail | |
| 12 | VERIFY rsuPo OriginalTime. | prtWatchdogTime.Table_Row IS_EQUAL_TO Pass/Fai | | Pass/Fail | |
| | | Test Procedure Results | | | |
| Tested By: | | | Date Tested: | Pass/Fail | |
| Test Procedure N | lotes: | | | | |

C.3.15.5 Monitor RSU Interface Watchdog Timer

| Description: | | This test case verifies that station to view how much to detected on each port. | the ASC allows a managen ime has elapsed since acti | nent vity was |
|---|--|--|--|---------------------------------------|
| Requirement(s): • 3.5.4.1.5 Monitor RSU Interface Wa | | SU Interface Watchdog Tim | er | |
| Variable(s): MaxRows maxRsuPorts Table_Row Int | | maxRsuPorts Int | | |
| Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | ep in this | | |
| Test Sten | Test Procedure | | | |
| Number | | Test Procedure | | Results |
| Number 1 | GET the follow | Test Procedure ing objects: maxRsuPorts. | | Results Pass/Fail |
| Test Step Number 1 1.1 | GET the follow RECO | Test Procedure ing objects: maxRsuPorts. RD this information as MaxRov | VS. | Results Pass/Fail |
| 1 1 1.1 2 | GET the follow RECO FOR Table_Ro | Test Procedure ing objects: maxRsuPorts. RD this information as MaxRov ow from 1 to MaxRows. | VS. | Results Pass/Fail |
| 1 1 1.1 2 2.1 | GET the follow RECO FOR Table_Ro GET th | Test Procedure ing objects: maxRsuPorts. RD this information as MaxRov ow from 1 to MaxRows. ne following objects: rsuPortWa | vs. itchdogTimer.Table_Row | Results Pass/Fail Pass/Fail |
| 1 1.1 2 2.1 | GET the follow RECO FOR Table_Ro GET th | Test Procedure ing objects: maxRsuPorts. RD this information as MaxRov ow from 1 to MaxRows. ne following objects: rsuPortWa Test Procedure Results | vs. itchdogTimer.Table_Row | Results Pass/Fail Pass/Fail |
| Test step Number 1 1.1 2 2.1 Tested By: | GET the follow RECO FOR Table_Ro GET th | Test Procedure ing objects: maxRsuPorts. RD this information as MaxRov ow from 1 to MaxRows. ne following objects: rsuPortWa Test Procedure Results | vs. ttchdogTimer.Table_Row Date Tested: | Results Pass/Fail Pass/Fail Pass/Fail |

C.3.15.6 Enable Signal Phase and Timing Data

| Test Procedure: | e: | | |
|---------------------|--|--|-----------|
| Description: | escription: This test case verifies that that ASC allows a management station to enable or disable SPaT functions. | | ment |
| Requirement(| s): | • 3.5.4.2.1.1 Enable Signal Phase and Timing Data | |
| Variable(s): | | OriginalSpatOptions spatOptions TestSpatOptions spatOptions | |
| Pass/Fail Crite | il Criteria: The device under test shall pass every verification step in th test case to pass the test case. | | p in this |
| Test Step Number | Test Procedure Resu | | Results |
| 1 | GET the following objects: spatOptions. Pas | | Pass/Fail |
| 1.1 | RECORD this information as OriginalSpatOptions. | | |
| 2 | ASSIGN TestSpatOptions EQUALS OriginalSpatOptions XOR 1. | | |
| 3 | ASSIGN spatOptions EQUALS TestSpatOptions. | | |
| 4 | SET the follow | ng objects: spatOptions. | Pass/Fail |
| 5 | GET the follow | ing objects: spatOptions. | Pass/Fail |
| 6 | VERIFY spatO | ptions IS_EQUAL_TO TestSpatOptions. | Pass/Fail |
| 7 | ASSIGN spatC | ptions EQUALS OriginalSpatOptions. | |

| 8 | 8 SET the following objects: spatOptions. | | Pass/Fail | | |
|---------------------------|--|--|-----------|--|--|
| 9 | 9 GET the following objects: spatOptions. | | Pass/Fail | | |
| 10 | 10 VERIFY spatOptions IS_EQUAL_TO OriginalSpatOptions. | | Pass/Fail | | |
| Test Procedure Results | | | | | |
| Tested By: Date Pass/Fail | | | | | |
| Test Procedure | Notes: | | | | |

C.3.15.7 Retrieve Signal Phase and Timing and Generation Time

| Test Procedure: | | Retrieve Signal Phase and Timing Generation Time | | |
|---------------------|---|---|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the most recent time SPaT data was generated by the ASC. | | |
| Requirement(s): | | 3.5.4.2.1.2 Retrieve Signal Phase and Timing Generation Time | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: spatTimestamp. | | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.8 Monitor CV Movement Minimum End Time

| Test Procedure: | Monitor CV Movement Mir | Monitor CV Movement Minimum End Time | |
|---------------------|---|--|--|
| Description: | This test case verifies that station to view the earlies movement state in a conn | This test case verifies that the ASC allows a management station to view the earliest possible end time for the current movement state in a connected vehicle environment. | |
| Requirement(s): | • 3.5.4.2.1.3.1.1 Mon | • 3.5.4.2.1.3.1.1 Monitor Movement Minimum End Time | |
| Variable(s): | MaxSignalGroups Table_Row | maxSignalGroups Int | |
| Pass/Fail Criteria: | The device under test sha test case to pass the test | The device under test shall pass every verification step in this test case to pass the test case. | |

| Test Step Number | Test Procedure Re | | | |
|---------------------|--|---|-----------|--|
| 1 | GET the following objects: maxSignalGr | roups. | Pass/Fail | |
| 1.1 | RECORD this information as Ma | RECORD this information as MaxSignalGroups. | | |
| 2 | FOR Table_Row from 1 to MaxSignalGroups. | | | |
| 2.1 | GET the following objects: signalStateMinEndTick2.Table_Row.1. | Pass/Fail | | |
| - | Test Procedure Re | esults | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure | Notes: | | | |

C.3.15.9 Monitor CV Movement Maximum End Time

| Test Procedure: | edure: | | | |
|---------------------|---|--|-----------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the latest possible end time for the current movement state in a connected vehicle environment. | | |
| Requirement(| s): | • 3.5.4.2.1.3.1.2 Monitor Movement Maximum End Time | | nd Time |
| Variable(s): | s): MaxSignalGroups maxSignalGroups Table_Row Int | | | |
| Pass/Fail Crit | iteria: The device under test shall pass every verification step in this test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxSignalGroups. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxSignalGroups. | | | |
| 2 | FOR Table_Row from 1 to MaxSignalGroups. | | | |
| 2.1 | GET the following objects: signalStateMaxEndTick2.Table_Row.1. | | Pass/Fail | |
| | | Test Procedure Results | ; ; | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.15.10 Monitor CV Movement Expected End Time

| Description: | | This test case verifies that the ASC allows a management station to view time the current movement state in a connected vehicle environment in expected to end. | | |
|------------------------------------|---|--|------------------------|-------------------------------|
| Requirement(s): | | 3.5.4.2.1.3.1.3 Monitor Movement Likely End Time | | |
| Variable(s): | | MaxSignalGroups Table_Row | maxSignalGroups Int | |
| Pass/Fail Crite | Pass/Fail Criteria: The device under test shall pass every verification step test case to pass the test case. | | p in this | |
| Test Step | Test Procedure | | Results | |
| Number | | | | |
| 1 | GET the follow | ing objects: maxSignalGroups. | | Pass/Fail |
| 1 1.1 | GET the follow RECO | ing objects: maxSignalGroups. RD this information as MaxSigr | nalGroups. | Pass/Fail |
| 1 1.1 2 | GET the follow RECO FOR Table_Ro | ing objects: maxSignalGroups. RD this information as MaxSigr w from 1 to MaxSignalGroups. | nalGroups. | Pass/Fail |
| 1 1.1 2 2.1 | GET the follow RECO FOR Table_Ro GET the signalStateLike | ing objects: maxSignalGroups. RD this information as MaxSign w from 1 to MaxSignalGroups. ne following objects: elyEndTick2.Table_Row.1. | nalGroups. | Pass/Fail Pass/Fail |
| 1 1.1 2 2.1 | GET the follow RECO FOR Table_Ro GET the signalStateLike | ing objects: maxSignalGroups. RD this information as MaxSign w from 1 to MaxSignalGroups. the following objects: elyEndTick2.Table_Row.1. Test Procedure Results | nalGroups. | Pass/Fail Pass/Fail |
| 1 1.1 2 2.1 Tested By: | GET the follow RECO FOR Table_Ro GET th signalStateLike | ing objects: maxSignalGroups. RD this information as MaxSign ow from 1 to MaxSignalGroups. ne following objects: elyEndTick2.Table_Row.1. Test Procedure Results | Date Tested: | Pass/Fail Pass/Fail Pass/Fail |

C.3.15.11 Monitor CV Movement Likely End Time Confidence

| Test Procedure: | Monitor CV Movement Likely End Time Confidence | | |
|---------------------|---|------------------------------|-----------------------|
| Description: | on: This test case verifies that the ASC allows a management station to view the statistical confidence of the expected end time the current movement state in a connected vehicle environment. | | nent ted end le |
| Requirement(| (s): • 3.5.4.2.1.3.1.4 Monitor Movement Likely End Time Confidence | | ime |
| Variable(s): | MaxSignalGroups maxSignalGroups Table_Row Int | | |
| Pass/Fail Crite | ail Criteria: The device under test shall pass every verification step in this test case to pass the test case. | | ep in this |
| Test Step Number | Test Procedure Resul | | Results |
| 1 | GET the following objects: maxSignalGroups. Pass/ | | Pass/Fail |
| 1.1 | RECORD this information as MaxSignalGroups. | | |
| 2 | FOR Table_Ro | w from 1 to MaxSignalGroups. | |
| 2.1 | GET the following objects: signalStateTickConfidence2.Table_Row.1. | | Pass/Fail |
| | - | Test Procedure Results | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.15.12 Monitor CV Movement Next Occurrence

| Test Procedure: | | Monitor CV Movement Nex | t Occurrence | |
|---------------------|--|---|-------------------------------------|------------|
| Description: | | This test case verifies that the ASC allows a management station to view the expected time the current movement state in a connected vehicle environment will be allowed to proceed again after in ends. | | |
| Requirement(| Requirement(s): • 3.5.4.2.1.3.1.5 Monitor Movement Next Occurrence | | ence | |
| Variable(s): | Variable(s): MaxSignalGroups maxSignalGro Table_Row Int | | maxSignalGroups Int | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxSig | nalGroups. | |
| 2 | FOR Table_Ro | w from 1 to MaxSignalGroups. | | |
| 2.1 | GET the following objects: signalStateNextTick2.Table_Row.1. | | eNextTick2.Table_Row.1. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.13 Monitor Next CV Movement Minimum End Time

| Test Procedure: | Monitor Next CV Movement Minimum End Time | | |
|--------------------|--|--|--|
| Description: | This test case verifies that the A station to view the earliest poss state after the current movemen environment. | ASC allows a management ible end time for movement at state in a connected vehicle | |
| Requirement(s): | • 3.5.4.2.1.3.1.7 Monitor No Time | ext Movement Minimum End | |
| Variable(s): | MaxSignalGroups Table_Row | maxSignalGroups Int | |

| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
|---------------------|--------------------------|---|-------------------------------------|-----------|
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | GET the following objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxSignalGroups. | | |
| 2 | FOR Table_Ro | FOR Table_Row from 1 to MaxSignalGroups. | | |
| 2.1 | GET tl signalStateMir | GET the following objects: signalStateMinEndTick2.Table Row.2. | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | Date Tested: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.15.14 Monitor Next CV Movement Maximum End Time

| Test Procedure: | edure: | | | |
|---------------------|--|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the latest possible end time for movement state after the current movement state in a connected vehicle environment. | | |
| Requirement(| Requirement(s): • 3.5.4.2.1.3.1.8 Monitor Next Movement Maximu Time | | m End | |
| Variable(s): | MaxSignalGroups maxSignalGroups Table_Row Int | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test case | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxSig | nalGroups. | |
| 2 | FOR Table_Row from 1 to MaxSignalGroups. | | | |
| 2.1 | GET the following objects: signalStateMaxEndTick2.Table_Row.2. | | Pass/Fail | |
| | - | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.15 Monitor Next CV Movement Maximum Start Time

Monitor Next CV Movement Maximum Start Time

| Test Procedure: | | | | |
|---|---|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the start time of the movement state that will follow the current movement state in a connected vehicle environment. | | |
| • 3.5.4.2.1.3.1.9 Monitor Next Movement Maximum Time | | m Start | | |
| Variable(s): | MaxSignalGroups maxSignalGroups Table_Row Int | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECO | RECORD this information as MaxSignalGroups. | | |
| 2 | FOR Table_Row from 1 to MaxSignalGroups. | | | |
| 2.1 | GET the following objects: signalStateStartTick2.Table_Row.2. | | Pass/Fail | |
| | | Test Procedure Results | | - |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.16 Determine Maximum Number of CV Movement Events

| Test Procedure: | | Determine Maximum Number of CV Movement Events | | |
|---------------------|---|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to retrieve the maximum number of movement events in a connected vehicle environment. | | |
| Requirement(s): | | • 3.5.4.2.1.3.1.10 Dete Movement Events | rmine Maximum Number o | f |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxMovementEvents. | | nts. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |

Г

Test Procedure Notes:

C.3.15.17 Configure Queue Detectors for CV Movement Assistance

| Test Procedure: | | Configure Queue Detectors for CV Movement Assistance | | |
|---------------------|---|--|---|-----------------|
| Description: | | This test case verifies the station to configure the movement in a connected | at the ASC allows a manage queue detectors for a specif ed vehicle environment. | ement ïc |
| Requirement(| s): | • 3.5.4.1.2.1.3.2.1 C Movement Assis | Configure Queue Detectors 1 tance | or |
| Variable(s): | | MaxRows Table_Row OriginalValue | maxMovementManeuvers Int movementManeuverQueu | 2 eDetector2 |
| Pass/Fail Crite | eria: | The device under test sh test case to pass the tes | nall pass every verification s t case. | tep in this |
| Test Step Number | | Test Procedu | re | Results |
| 1 | GET the follow | /ing objects: maxMovement | Vaneuvers2. | Pass/Fail |
| 1.1 | RECO | RD this information as Max | Rows. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: movementManeuverSignalGroupEntryNumber.Table_Row. | | Pass/Fail | |
| 4 | GET the following objects: movementQueueDetector2.Table_Row. Pas | | Pass/Fail | |
| 4.1 | RECORD this information as OriginalValue | | | |
| 5 | ASSIGN movementManeuverQueueDetector2.Table_Row EQUALS '01'. | | | |
| 6 | SET the follow movementMan | ing objects: ieuverQueueDetector2.Tabl | e_Row. | Pass/Fail |
| 7 | GET the follow movementMan | ving objects: neuverQueueDetector2.Tabl | e_Row. | Pass/Fail |
| 8 | VERIFY mover | mentManeuverQueueDetec) '01'. | tor2.Table_Row | Pass/Fail |
| 9 | ASSIGN move OriginalValue. | mentManeuverQueueDetec | tor2.Table_Row EQUALS | |
| 10 | SET the following objects: movementManeuverQueueDetector2.Table Row. | | Pass/Fail | |
| 11 | GET the follow movementMan | ring objects: neuverQueueDetector2.Tabl | e_Row. | Pass/Fail |
| 12 | VERIFY move IS_EQUAL_TO | ERIFY movementManeuverQueueDetector2.Table_Row Pass/Fail | | Pass/Fail |
| | | Test Procedure Resu | ılts | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

| Test Procedure: | Configure Pedestrian Detectors for CV Movement Conflict Assistance | | | |
|---------------------|--|---|--|-------------------------------|
| Description: | | This test case verifies tha station to configure the per may conflict with specific environment. | t the ASC allows a managen edestrian detectors for cross movement in a connected v | nent sings that rehicle |
| Requirement(| s): | • 3.5.4.1.2.1.3.2.2 Co Movement Conflic | onfigure Pedestrian Detector t Assistance | rs for |
| Variable(s): | | MaxRows Table_Row OriginalDetector | maxMovementManeuver Int movementManeuverPedPre | s2 esense2 |
| Pass/Fail Crit | S/Fail Criteria: The device under test shall pass every verification step i test case to pass the test case. | | ep in this | |
| Test Step Number | | Test Procedure | 9 | Results |
| 1 | GET the following objects: maxMovementManeuvers2. | | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRe | ows. | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: movementManeuverSignalGroupEntryNumber.Table_Row. | | Pass/Fail | |
| 4 | GET the following objects: movementManeuverPedPresense2.Table Row | | Pass/Fail | |
| 4.1 | RECO | ORD this information as Origir | alDetector. | |
| 5 | ASSIGN move | mentManeuverPedPresense2 | 2.Table_Row EQUALS '01'. | |
| 6 | SET the follow movementMan | ing objects: neuverPedPresense2.Table_F | Row. | Pass/Fail |
| 7 | GET the follow movementMan | /ing objects: ieuverPedPresense2.Table_F | Row. | Pass/Fail |
| 8 | VERIFY mover '01'. | mentManeuverPedPresense2 | 2.Table_Row IS_EQUAL_TO | Pass/Fail |
| 9 | ASSIGN move OriginalDetected | mentManeuverPedPresense2 or. | 2.Table_Row EQUALS | |
| 10 | SET the follow movementMan | ing objects: neuverPedPresense2.Table_F | Row. | Pass/Fail |
| 11 | GET the follow movementMan | ring objects: neuverPedPresense2.Table_F | Row. | Pass/Fail |
| 12 | VERIFY move OriginalDetect | ementManeuverPedPresense2.Table_Row IS_EQUAL_TO Pass/Fail | | Pass/Fail |
| | | Test Procedure Result | ts | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | - | <u>.</u> |

C.3.15.18 Configure Pedestrian Detectors for CV Movement Conflict Assistance

| C.3.15.19 | Configure Bicycle Detectors for CV Movement Conflict Ass | istance |
|-----------|--|---------|
|-----------|--|---------|

| Test Procedure: | 15.18 Configure Bicycle Detectors for CV Movement Conflict Assistance | | | |
|---------------------|---|---|--|-------------------------------------|
| Description: | | This test case verifies that the station to configure the bicycle may conflict with a specific more environment. Bicycle detectors detectors. | ASC allows a managem e detectors for crossing ovement in a connected s may be configured as | ent s that vehicle vehicle |
| Requirement(| s): | • 3.5.4.1.3.6.3.6 Configur Assistance V2 | e Bicycle Detectors for | Movement |
| Variable(s): | | MaxRows max Table_Row Int OriginalDetector move | MovementManeuvers2 ementManeuverBicyclePre | sense2 |
| Pass/Fail Crit | eria: | The device under test shall past test case to pass the test case | ss every verification ste | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxMovementManeuv | vers2. | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRows. | | |
| 2 | ASSIGN Table Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: movementManeuverSignalGroupEntryNumber.Table Row. | | | Pass/Fail |
| 4 | GET the following objects: movementManeuverBicyclePresense2.Table Row | | Pass/Fail | |
| 4.1 | RECC | RD this information as OriginalDe | etector. | |
| 5 | ASSIGN move '01'. | mentManeuverBicyclePresense2.7 | Table_Row EQUALS | |
| 6 | SET the follow movementMan | ng objects: euverBicyclePresense2.Table_Ro | ow. | Pass/Fail |
| 7 | GET the follow movementMan | ing objects: euverBicyclePresense2.Table_Ro | DW. | Pass/Fail |
| 8 | VERIFY mover IS_EQUAL_TO | nentManeuverBicyclePresense2.1) '01'. | Table_Row | Pass/Fail |
| 9 | ASSIGN move OriginalDetecto | mentManeuverBicyclePresense2. or. | Table_Row EQUALS | |
| 10 | SET the follow movementMan | ng objects: euverBicyclePresense2.Table_Ro | | Pass/Fail |
| 11 | GET the follow movementMan | ing objects: euverBicyclePresense2.Table_Ro | DW. | Pass/Fail |
| 12 | VERIFY mover IS_EQUAL_TO | nentManeuverBicyclePresense2.T) OriginalDetector. | Table_Row | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | Da Te | ate ested: | Pass/Fail |
| Test Procedure | Notes: | | | |

| C.3.15.20 | Configure C | / Lane Connection | Queue Length |
|-----------|-------------|-------------------|---------------------|
|-----------|-------------|-------------------|---------------------|

| Test Procedure: | re: | | on Queue Length | |
|---------------------|--|---|--|------------|
| Description: | | This test case verifies that station to view the queue le maneuver in a connected v | This test case verifies that the ASC allows a management station to view the queue length for a specific movement maneuver in a connected vehicle environment. | |
| Requirement(| s): | • 3.5.4.2.1.3.3.1 Monitor Lane Connection Queue Length | | e Length |
| Variable(s): | | MaxSignalGroups Table_Row | maxSignalGroups Int | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test c | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the follow | ing objects: maxMovementMa | neuvers2. | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | FOR Table_Ro | ow from 1 to MaxRows. | | |
| 2.1 | GET the following objects: movementManeuverSignalGroupEntryNumber.Table_Row, movementManeuverQueue2.Table_Row. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.21 Configure CV Lane Connection Vulnerable Road User Detection

| Test Procedure: | | Monitor CV Lane Connection Vulnerable Road User D | etection | |
|---------------------|--|---|-----------|--|
| Description: | | This test case verifies that the ASC allows a management station to view if any pedestrian or bicycle is detected in a connected vehicle environment for a specific movement. | | |
| Requirement(s): | | 3.5.4.2.1.3.3.2 Monitor Lane Connection Vulnerable Road User Detection | | |
| Variable(s): | | MaxSignalGroups maxSignalGroups Table_Row Int | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | | Test Procedure Results | | |
| 1 | GET the following objects: maxMovementManeuvers2. Pa | | Pass/Fail | |

| 1.1 | RECORD this information as MaxRo | | | | | | |
|------------------------|--|-----------------|-----------|--|--|--|--|
| 2 | FOR Table_Row from 1 to MaxRows. | | | | | | |
| 2.1 | GET the following objects: movementManeuverSignalGroupEntryNumb movementManeuverStatus2.Table_Row. | Pass/Fail | | | | | |
| Test Procedure Results | | | | | | | |
| Tested By: | | Date Tested: | Pass/Fail | | | | |
| Test Procedure Notes: | | | | | | | |

C.3.15.22 Configure Advisory Speed Type

| Test Procedure: | | Configure Advisory Speed Type | | |
|---------------------|--|---|-----------|--|
| Description: | | This test case verifies that the ASC allows a management station to configure the type of speed advisory for a specific movement traversing a connected intersection. | | |
| Requirement(s): | | • 3.5.4.2.1.3.4.1 Configure Advisory Speed Type | | |
| Variable(s): | | MaxRowsmaxAdvisorySpeeds2Table_RowIntOriginalSpeedTypeadvisorySpeedType2TestSpeedTypeadvisorySpeedType2 | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | p in this | |
| Test Step Number | | Test Procedure | Results | |
| 1 | GET the following objects: maxAdvisorySpeeds2. Pass | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: advisorySpeedSignalGroupEntryNumber.Table_Row. | | Pass/Fail | |
| 4 | GET the following objects: advisorySpeedType2.Table_Row. Pass/Fail | | | |
| 4.1 | RECORD this information as OriginalSpeedType. | | | |
| 5 | ASSIGN TestSpeedType EQUALS RANDOM (1 TO 4). | | | |
| 5.1 | IF TestSpeedType IS_EQUAL_TO OriginalSpeedType | | | |
| 5.1.1 | GOTO step 5. | | | |
| 6 | ASSIGN advisorySpeedType2.Table_Row EQUALS TestSpeedType. | | | |
| 7 | SET the following objects: advisorySpeedType2.Table_Row. Pass/Fail | | | |
| 8 | GET the follow | GET the following objects: advisorySpeedType2.Table_Row. Pass/Fail | | |
| 9 | VERIFY advisorySpeedType2.Table_Row IS_EQUAL_TO TestSpeedType. Pass/Fail | | | |
| 10 | ASSIGN advisorySpeedType2.Table_Row EQUALS OriginalSpeedType. | | | |
| 11 | SET the followi | ng objects: advisorySpeedType2.Table_Row. | Pass/Fail | |
| 12 | GET the follow | ing objects: advisorySpeedType2.Table_Row. | Pass/Fail | |
| 13 VERIFY advisorySpeedType2.Table_Row IS_EQUAL_TO OriginalSpeedType. | | | Pass/Fail |
|--|--------|--|-----------|
| Test Procedure Results | | | |
| Tested By: Date Tested: | | | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.15.23 Configure Advisory Speed

| Test Procedure: | Configure Advisory Speed | | | |
|---------------------|--|--|-------------------------------------|---------------|
| Description: | | This test case verifies that the ASC allows a management station to configure an advisory speed for a specific movement traversing a connected intersection. | | |
| Requirement(| s): | • 3.5.4.2.1.3.4.2 Config | gure Advisory Speed | |
| Variable(s): | | MaxRowsmaxAdvisorySpeeds2Table_RowIntOriginalSpeedadvisorySpeedAdvice2TestSpeedadvisorySpeedAdvice2 | | 2 e2 e2 |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxAdvisorySpeeds2. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: advisorySpeedSignalGroupEntryNumber.Table Row. | | Pass/Fail | |
| 4 | GET the follow | ing objects: advisorySpeedAdv | ice2.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as Original | Speed. | |
| 5 | ASSIGN TestS | peed EQUALS RANDOM (0 T | O 500). | |
| 5.1 | IF Te | stSpeed IS_EQUAL_TO Origin | alSpeed | |
| 5.1.1 | | GOTO step 5. | | |
| 6 | ASSIGN advise | prySpeedAdvice2.Table_Row E | EQUALS TestSpeed. | |
| 7 | SET the followi | ng objects: advisorySpeedAdv | ice2.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: advisorySpeedAdv | ice2.Table_Row. | Pass/Fail |
| 9 | VERIFY adviso | orySpeedAdvice2.Table_Row IS | S_EQUAL_TO TestSpeed. | Pass/Fail |
| 10 | ASSIGN advisorySpeedAdvice2.Table_Row EQUALS OriginalSpeed. | | | |
| 11 | SET the followi | ng objects: advisorySpeedAdv | ice2.Table_Row. | Pass/Fail |
| 12 | GET the follow | ing objects: advisorySpeedAdv | ice2.Table_Row. | Pass/Fail |
| 13 | VERIFY adviso OriginalSpeed. | IFY advisorySpeedAdvice2.Table_Row IS_EQUAL_TO | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |

Test Procedure Notes:

C.3.15.24 Configure Advisory Speed Zone

| Test Procedure: | | Configure Advisory Speed Zone | | |
|---------------------|--|---|--|--------------------------|
| Description: | | This test case verifies that the ASC allows a management station to configure the length of an advisory speed length for a specific movement traversing a connected intersection. | | |
| Requirement(| s): | • 3.5.4.2.1.3.4.3 Config | ure Advisory Speed Zone |) |
| Variable(s): | | MaxRows Table_Row OriginalZoneLength TestZoneLength | maxAdvisorySpeeds Int advisorySpeedZone advisorySpeedZone | s2 Length2 Length2 |
| Pass/Fail Crit | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxAdvisorySpeeds2. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the following objects: advisorySpeedSignalGroupEntryNumber.Table_Row. | | Pass/Fail | |
| 4 | GET the following objects: advisorySpeedZoneLength2.Table_Row. | | Pass/Fail | |
| 4.1 | RECO | RD this information as OriginalZ | ZoneLength. | |
| 5 | ASSIGN TestZ | ConeLength EQUALS RANDOM | (0 TO 10000). | |
| 5.1 | IF Te | stZoneLength IS_EQUAL_TO C | DriginalZoneLength | |
| 5.1.1 | | GOTO step 5. | | |
| 6 | ASSIGN advise TestZoneLeng | orySpeedZoneLength2.Table_R th. | Row EQUALS | |
| 7 | SET the follow | ing objects: advisorySpeedZone | Length2.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: advisorySpeedZone | eLength2.Table_Row. | Pass/Fail |
| 9 | VERIFY adviso TestZoneLeng | prySpeedAdvice2.Table_Row IS th. | EQUAL_TO | Pass/Fail |
| 10 | ASSIGN advise OriginalZoneLe | orySpeedAdvice2.Table_Row E ength | QUALS | |
| 11 | SET the follow | ing objects: advisorySpeedZone | Length2.Table_Row. | Pass/Fail |
| 12 | GET the follow | ing objects: advisorySpeedZone | eLength2.Table_Row. | Pass/Fail |
| 13 | VERIFY adviso OriginalZoneLe | /isorySpeedAdvice2.Table_Row IS_EQUAL_TO Pass/Fail eLength. | | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.15.25 Configure Advisory Speed Vehicle Type

| Test Procedure: | | Configure Advisory Speed Vehicle Type | | |
|---------------------|---|---|---|-------------------------|
| Description: | | This test case verifies that the station to configure the vehi a specific movement travers applies to. | ne ASC allows a managem cle type that an advisory s ing a connected intersect | ent speed for ion |
| Requirement(| s): | • 3.5.4.2.1.3.4.4 Config | ure Advisory Speed Vehic | le Type |
| Variable(s): | | MaxRows Table_Row OriginalVehicleClass TestVehicleClass | maxAdvisorySpeeds Int advisorySpeedClass advisorySpeedClass | 2 2 2 |
| Pass/Fail Crite | ass/Fail Criteria: The device under test shall pass every verification step in test case to pass the test case. | | p in this | |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxAdvisorySpeeds2. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows). | | | |
| 3 | GET the follow advisorySpeed | ing objects: SignalGroupEntryNumber.Table | e_Row. | Pass/Fail |
| 4 | GET the follow | ing objects: advisorySpeedClas | s2.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as OriginalV | /ehicleClass. | |
| 5 | ASSIGN TestV | ehicleClass EQUALS RANDOM | 1 (0 TO 255). | |
| 5.1 | IF Te | stVehicleClass IS_EQUAL_TO | OriginalVehicleClass. | |
| 5.1.1 | | GOTO step 5. | | |
| 6 | ASSIGN advise | orySpeedClass2.Table_Row EC | UALS TestVehicleClass. | |
| 7 | SET the follow | ng objects: advisorySpeedClass | s2.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: advisorySpeedClas | s2.Table_Row. | Pass/Fail |
| 9 | VERIFY adviso TestVehicleCla | orySpeedClass2.Table_Row IS_ ss. | EQUAL_TO | Pass/Fail |
| 10 | ASSIGN advisorySpeedClass2.Table_Row EQUALS OriginalVehicleClass. | | | |
| 11 | SET the following objects: advisorySpeedClass2.Table_Row. | | Pass/Fail | |
| 12 | GET the follow | ne following objects: advisorySpeedClass2.Table_Row. Pass/F | | Pass/Fail |
| 13 | VERIFY adviso OriginalVehicle | dvisorySpeedClass2.Table_Row IS_EQUAL_TO Pass/Fail Pass/Fail | | |
| | | Test Procedure Results | | |
| Tested By: | : Date Pass | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.15.26 Configure CV Movement States

| Test Procedure: | | Monitor CV Movement State | es | |
|---------------------|--|--|---|------------|
| Description: | | This test case verifies that the ASC allows a management station to view the current and next movement state in a connected vehicle environment. | | |
| Requirement(| s): | 3.5.4.2.1.3.5 Monitor 3.5.4.2.1.3.6 Monitor | 3.5.4.2.1.3.5 Monitor Movement State 3.5.4.2.1.3.6 Monitor Next Movement State | |
| Variable(s): | | MaxSignalGroups maxSignalGroups Table_Row Int | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxSig | nalGroups. | |
| 2 | FOR Table_Ro | w from 1 to MaxSignalGroups. | | |
| 2.1 | GET the following objects: signalState2.Table_Row.1, signalState2.Table_Row.2. | | e2.Table_Row.1, | Pass/Fail |
| | - | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.27 Configure CV Movement Status

| Test Procedure: | | Monitor CV Movement Status | |
|---------------------|------------------------|---|--|
| Description: | | This test case verifies that the ASC allows a management station to view what movements are permitted and when at an intersection with the signal status block in a connected vehicle environment. | |
| Requirement(s): | | • 3.5.4.2.1.3.7 Monitor Movement Status | |
| Variable(s): | | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | |
| Test Step Number | Test Procedure Results | | |

| 1 | 1 GET the following objects: signalStatusBlock2. | | Pass/Fail |
|------------------------|--|-----------------|-----------|
| Test Procedure Results | | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | |

C.3.15.28 Configure Concurrent Enabled Lanes

| Test Procedure: Configur | | Configure Concurrent Enal | oled Lanes | |
|--|---|--|--|-------------------|
| Description: | | This test case verifies that an ASC allows a management station to configure what lanes in a CV environment may be enabled together. | | |
| Requirement(| s): | • 3.5.4.2.1.4.1 Config | 3.5.4.2.1.4.1 Configure Concurrent Enabled Lanes | |
| Variable(s): | | MaxLanes Table_Row OriginalLanes | maxEnabledLanesCon Int enabledLaneConcur | currency rency |
| Pass/Fail Criteria: The device under test shall pass every veri test case to pass the test case. | | pass every verification ste ase. | p in this | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxEnabledLanesConcurrency. | | Pass/Fail | |
| 1.1 | RECORD this information as MaxLanes. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxLanes). | | | |
| 3 | GET the follow | ing objects: enabledLaneConc | urrency.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | Lanes. | |
| 4 | ASSIGN enable | edLanesConcurrency.Table_R | low EQUALS '01'. | |
| 5 | SET the followi | ing objects: enabledLaneConc | urrency.Table_Row. | Pass/Fail |
| 6 | GET the follow | ing objects: enabledLaneConc | urrency.Table_Row. | Pass/Fail |
| 7 | VERIFY enable | edLaneConcurrency.Table_Ro | w IS_EQUAL_TO '01'. | Pass/Fail |
| 8 | ASSIGN enable | edLaneConcurrency.Table_Rc | w EQUALS OriginalLanes. | |
| 9 | SET the followi | ing objects: enabledLaneConc | urrency.Table_Row. | Pass/Fail |
| 10 | GET the following objects: enabledLaneConcurrency.Table_Row. | | Pass/Fail | |
| 11 | VERIFY enabledLaneConcurrency.Table_Row IS_EQUAL_TO OriginalLanes. | | Pass/Fail | |
| | | Test Procedure Results | | |
| Tested By: | Date Pas | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.15.29 Configure Enabled Lanes by Time of Day

Configure Enabled Lanes by Time of Day

| Test Procedure: | | | | |
|---------------------|--|---|---|----------------|
| Description: | This test case verifies that an ASC allows a management station to configure what lanes in a CV environment may be enabled together. | | ent station enabled | |
| Requirement(| s): | • 3.5.4.2.1.4.2 Configu | ire Concurrent Enabled La | nes |
| Variable(s): | | MaxRows MaxLanes Table_Row OriginalLanes TestLanes | maxPatterns Int Int patternSpatEnabledI patternSpatEnabledI | _anes _anes |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | PRE-CONDITI | ON 'The user shall know the mable lane plans supported by th | aximum number of e ASC.' | |
| 1.1 | RECORD this information as MaxLanes | | | |
| 2 | GET the following objects: maxPatterns. | | Pass/Fail | |
| 2.1 | RECORD this information as MaxRows. | | | |
| 3 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows). | |
| 4 | GET the follow | ing objects: patternSpatEnable | dLanes.Table_Row. | Pass/Fail |
| 4.1 | RECO | RD this information as Originall | _anes. | |
| 5 | ASSIGN TestL | anes EQUALS RANDOM (1 TO |) MaxLanes). | |
| 5.1 | IF Te | stLanes IS_EQUAL_TO Origna | ILanes. | |
| 5.1.1 | | GOTO step 5. | | |
| 6 | ASSIGN patter | nSpatEnabledLanes.Table_Ro | w EQUALS TestLanes. | |
| 7 | SET the followi | ng objects: patternSpatEnable | dLanes.Table_Row. | Pass/Fail |
| 8 | GET the follow | ing objects: patternSpatEnable | dLanes. Lable_Row. | Pass/Fail |
| 9 | VERIFY pattern TestLanes. | nSpatEnabledLanes. I able_Rov | WIS_EQUAL_TO | Pass/Fail |
| 10 | ASSIGN patter | ASSIGN patternSpatEnabledLanes.Table_Row EQUALS OriginalLanes. | | |
| 11 | SET the followi | SET the following objects: patternSpatEnabledLanes.Table_Row. Pass/Fail | | Pass/Fail |
| 12 | GET the follow | ing objects: patternSpatEnable | dLanes.Table_Row. | Pass/Fail |
| 13 | VERIFY pattern OriginalLanes. | nSpatEnabledLanes.Table_Row IS_EQUAL_TO Pass/Fail | | |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.30 Determine CV Lanes Enabled

Determine CV Lanes Enabled

| Test Procedure: | | | | |
|---------------------|--|---|-----------|--|
| Description: | | This test case verifies that the ASC allows a management station to view the revocable lanes that are currently active. | | |
| Requirement(| s): |): • 3.5.4.2.1.4.3 Determine Lanes Enabled | | |
| Variable(s): | | | | |
| Pass/Fail Crite | teria: The device under test shall pass every verification step in this test case to pass the test case. | | p in this | |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: spatEnabledLanesStatus. | | Pass/Fail | |
| | | Test Procedure Results | ; | |
| Tested By: | y: Date Tested: | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.15.31 Enabled Signal Phase and Timing Data Exchange by Port

| Test Procedure: | | Enable Signal Phase and Timing Data Exchange by Port | | |
|---------------------|--|--|-------------------|--|
| Description: | Description: This test case verifies that that ASC allows a management station to enable/disable RSU ports for exchanging SPaT | | ment PaT data. | |
| Requirement(| s): | 3.5.4.2.1.5 Enable Signal Phase and Timing Data Exchange | | |
| Variable(s): | | MaxRowsmaxRsuPortsTable_RowIntOriginalSpatPortOptionsspatPortOptionsTestSpatPortOptionsspatPortOptions | | |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | |
| Test Step Number | Test Procedure Res | | Results | |
| 1 | GET the follow | ing objects: maxRsuPorts. | Pass/Fail | |
| 1.1 | RECORD this information as MaxRows. | | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows) | | | |
| 3 | GET the following objects: spatPortOptions.Table_Row. Pass/Fail | | Pass/Fail | |
| 3.1 | RECO | RECORD this information as OriginalSpatPortOptions. | | |
| 4 | ASSIGN TestS 1. | patPortOptions EQUALS OriginalSpatPortOptions XOR | | |

| 5 | ASSIGN spatPortOptions.Table_Row EQUALS Tes | stSpatPortOptions. | |
|----------------|--|---|-----------|
| 6 | SET the following objects: spatPortOptions.Table_F | Row. | Pass/Fail |
| 7 | GET the following objects: spatPortOptions.Table_ | Row. | Pass/Fail |
| 8 | VERIFY spatPortOptions.Table_Row IS_EQUAL_T TestSpatPortOptions. | 0 | Pass/Fail |
| 9 | ASSIGN spatPortOptions.Table_Row EQUALS OriginalSpatPortOptions. | ASSIGN spatPortOptions.Table_Row EQUALS OriginalSpatPortOptions. | |
| 10 | SET the following objects: spatPortOptionsTable_Row. | | Pass/Fail |
| 11 | GET the following objects: spatPortOptions.Table_I | GET the following objects: spatPortOptions.Table_Row. | |
| 12 | VERIFY spatPortOptions.Table_Row IS_EQUAL_TO OriginalSpatPortOptions. | | Pass/Fail |
| | Test Procedure Results | | |
| Tested By: | Date Test | e ted: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.15.32 Configure Road Authority Identifier

| Test Procedure: | | Configure Road Authority | dentifier | |
|---------------------|---|--|---|---------------|
| Description: | | This test case verifies tha station to configure the age messages. | t that ASC allows a manage gency indicated in SPaT and | ment I MAP |
| Requirement(| s): | • 3.5.4.2.1.6 Configu | ire Road Authority | |
| Variable(s): | | OriginalAuthority spatRoadAuthorityID TestAuthority spatRoadAuthorityID | | |
| Pass/Fail Criteria: | | The device under test sha test case to pass the test | III pass every verification ste case. | ep in this |
| Test Step Number | Test Procedure Re: | | Results | |
| 1 | PRE-CONDITION 'The management station shall be aware of a valid OBJECT IDENTIFIER for the agency'. | | | |
| 1.1 | RECO | RD this information as TestA | uthority. | |
| 2 | GET the following objects: spatRoadAuthorityID. Pass/F | | Pass/Fail | |
| 2.1 | RECORD this information as OriginalAuthority. | | | |
| 3 | ASSIGN spatRoadAuthorityID EQUALS TestAuthority. | | | |
| 4 | SET the following objects: spatRoadAuthorityID. Pass/Fail | | Pass/Fail | |
| 5 | GET the following objects: spatRoadAuthorityID. Pass/Fai | | Pass/Fail | |
| 6 | VERIFY spatRoadAuthorityID IS_EQUAL_TO TestAuthority. Pass/Fail | | Pass/Fail | |
| 7 | ASSIGN spatR | oadAuthorityID EQUALS Ori | ginalAuthority. | |
| 8 | SET the following objects: spatRoadAuthorityID. Pass/Fail | | Pass/Fail | |
| 9 | GET the follow | ing objects: spatRoadAuthori | tyID. | Pass/Fail |
| 10 | VERIFY spatR | oadAuthorityID IS_EQUAL_T | O OriginalAuthority. | Pass/Fail |
| | | Test Procedure Resul | ts | |

| Tested By: | Date Tested: | Pass/Fail |
|-----------------------|-----------------|-----------|
| Test Procedure Notes: | | |

C.3.15.33 Retrieve Signal Phase and Timing Intersection Status

| Test Procedure: | | Retrieve Signal Phase and Timing Intersection Status | | |
|-------------------------|---|--|-----------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to identify intersection status alarm indications at a connected intersection. The indications include • Manual Control Active • Stop Time Active • Failure Flash Active • Preempt Active • Priority Active • Fixed Time Active • Actuation Active • Soft Flash Active • Controller Failure • Invalid MAP Message | | |
| Requirement(s): | | 3.5.4.2.1.7.1 Monitor Manual Control Indication 3.5.4.2.1.7.2 Monitor Stop Indication 3.5.4.2.1.7.3 Monitor Failure Flash Indication 3.5.4.2.1.7.4 Monitor Preemption Operation Indication 3.5.4.2.1.7.5 Monitor Priority Operation Indication 3.5.4.2.1.7.6 Monitor Fixed Time Control 3.5.4.2.1.7.7 Monitor Non-Fixed Time Control 3.5.4.2.1.7.8 Monitor Standby Operation Indication 3.5.4.2.1.7.9 Monitor Controller Failure 3.5.4.2.1.7.10 Monitor Map Message Validity 3.5.4.2.1.7.11 Monitor SPaT Data Validity | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | Test Procedure Res | | Results | |
| 1 | GET the following objects: spatStatus2. Pass/Fa | | Pass/Fail | |
| I EST Procedure Results | | | | |
| Tested By: | | | Tested: | Pass/Fail |
| Test Procedure Notes: | | | | |

C.3.15.34 Mark SPaT Invalid - Controller

| Test Procedure: | | Mark SPaT Invalid - Control | ler | |
|-----------------------|--|---|-------------------------------------|------------|
| Description: | | This test case verifies that that ASC allows a management station to mark SPaT data as invalid for the ASC. | | |
| Requirement(s): | | • 3.5.4.2.1.8 Mark SPaT Invalid - Controller | | |
| Variable(s): | OriginalSpatOptions spatOptions TestSpatOptions spatOptions | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure Re | | Results | |
| 1 | GET the following objects: spatOptions. | | Pass/Fail | |
| 1.1 | RECORD this information as OriginalSpatOptions. | | | |
| 2 | ASSIGN TestS | patOptions EQUALS OriginalS | patOptions XOR 2. | |
| 3 | ASSIGN spatOptions EQUALS TestSpatOptions. | | | |
| 4 | SET the followi | ing objects: spatOptions. | | Pass/Fail |
| 5 | GET the follow | ing objects: spatOptions. | | Pass/Fail |
| 6 | VERIFY spatOptions IS_EQUAL_TO TestSpatOptions. Pass/Fai | | Pass/Fail | |
| 7 | ASSIGN spatC | ptions EQUALS OriginalSpatC | ptions. | |
| 8 | SET the followi | ing objects: spatOptions. | | Pass/Fail |
| 9 | GET the following objects: spatOptions. Pass/Fa | | Pass/Fail | |
| 10 | VERIFY spatO | ptions IS_EQUAL_TO Original | SpatOptions. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure Notes: | | | | |

C.3.15.35 Mark SPaT Invalid - Port

| Test Procedure: | Mark SPaT Invalid - Port | |
|--------------------|--|---------------------------------------|
| Description: | This test case verifies that that ASC allows a management station to mark SPaT data on an RSU port as invalid. | |
| Requirement(s): | • 3.5.4.2.1.9 Mark SPaT Invalid - Port | |
| Variable(s): | MaxRows Table_Row OriginalSpatPortOptions | maxRsuPorts Int spatPortOptions |

| | | TestSpatPortOptions | spatPortOptions | |
|---------------------|--------------------------------|--|-------------------------------------|------------|
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | | Test Procedure Result | | |
| 1 | GET the follow | ving objects: maxRsuPorts. | | Pass/Fail |
| 1.1 | RECO | ORD this information as MaxRo | WS. | |
| 2 | ASSIGN Table | e_Row EQUALS RANDOM (1 T | O MaxRows) | |
| 3 | GET the follow | ving objects: spatPortOptions.Ta | able_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | SpatPortOptions. | |
| 4 | ASSIGN Tests 2. | ASSIGN TestSpatPortOptions EQUALS OriginalSpatPortOptions XOR 2. | | |
| 5 | ASSIGN spatF | ASSIGN spatPortOptions.Table_Row EQUALS TestSpatPortOptions. | | |
| 6 | SET the follow | SET the following objects: spatPortOptions.Table_Row. Pass/Fail | | |
| 7 | GET the follow | GET the following objects: spatPortOptions.Table_Row. Pass/Fail | | |
| 8 | VERIFY spatP TestSpatPortC | VERIFY spatPortOptions.Table_Row IS_EQUAL_TO TestSpatPortOptions. Pass/Fail | | |
| 9 | ASSIGN spatF OriginalSpatPo | ASSIGN spatPortOptions.Table_Row EQUALS OriginalSpatPortOptions. | | |
| 10 | SET the follow | SET the following objects: spatPortOptionsTable_Row. Pass/Fail | | |
| 11 | GET the follow | GET the following objects: spatPortOptions.Table_Row. Pass/Fail | | |
| 12 | VERIFY spatP OriginalSpatPo | VERIFY spatPortOptions.Table_Row IS_EQUAL_TO OriginalSpatPortOptions. | | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | |

C.3.15.36 Mark MAP Message Invalid - Controller

| Test Procedure: | | Mark MAP Message Invalid - Controller | |
|---------------------|---|--|-----------|
| Description: | | This test case verifies that that ASC allows a management station to mark MAP data as invalid for the ASC. | |
| Requirement(s): | | • 3.5.4.2.1.10 Mark MAP Message Invalid - Controller | |
| Variable(s): | | OriginalSpatOptions spatOptions TestSpatOptions spatOptions | |
| Pass/Fail Criteria: | | The device under test shall pass every verification ste test case to pass the test case. | p in this |
| Test Step Number | Test Procedure | | Results |
| 1 | GET the following objects: spatOptions. Pass/Fa | | Pass/Fail |

| 1.1 | RECORD this information as OriginalSpatOptions. | | | |
|-----------------------|---|-----------|--|--|
| 2 | ASSIGN TestSpatOptions EQUALS OriginalSpatOptions XOR 4 | h. | | |
| 3 | ASSIGN spatOptions EQUALS TestSpatOptions. | | | |
| 4 | SET the following objects: spatOptions. | Pass/Fail | | |
| 5 | GET the following objects: spatOptions. | Pass/Fail | | |
| 6 | VERIFY spatOptions IS_EQUAL_TO TestSpatOptions. | Pass/Fail | | |
| 7 | ASSIGN spatOptions EQUALS OriginalSpatOptions. | | | |
| 8 | SET the following objects: spatOptions. Pas | | | |
| 9 | GET the following objects: spatOptions. Pass/Fa | | | |
| 10 | VERIFY spatOptions IS_EQUAL_TO OriginalSpatOptions. | Pass/Fail | | |
| | Test Procedure Results | | | |
| Tested By: | Date Tested: | Pass/Fail | | |
| Test Procedure Notes: | | | | |

C.3.15.37 Mark MAP Message Invalid – Port

| Test Procedure: | | Mark MAP Message Invalid - P | ort | |
|--|---|--|--|---|
| Description: | | This test case verifies that that station to mark MAP data on a | ASC allows a manager n RSU port as invalid. | nent |
| Requirement(| s): | • 3.5.4.2.1.11 Mark MAP | Message Invalid - Port | |
| Variable(s): | MaxRows maxRsuPorts Table_Row Int OriginalSpatPortOptions spatPortOptions TestSpatPortOptions spatPortOptions | | | |
| Pass/Fail Criteria: | | The device under test shall past test case to pass the test case | ss every verification ste | p in this |
| Test Step | | Test Procedure | | Desculto |
| Number | | Test Procedure | | Results |
| Number 1 | GET the follow | ring objects: maxRsuPorts. | | Results Pass/Fail |
| Number 1 1.1 | GET the follow RECC | ring objects: maxRsuPorts. DRD this information as MaxRows. | | Results Pass/Fail |
| Number 1 1.1 2 | GET the follow RECC ASSIGN Table | ring objects: maxRsuPorts. DRD this information as MaxRows. _Row EQUALS RANDOM (1 TO N | /laxRows) | Pass/Fail |
| Number 1 1.1 2 3 | GET the follow RECC ASSIGN Table GET the follow | ring objects: maxRsuPorts. DRD this information as MaxRows. E_Row EQUALS RANDOM (1 TO Mining objects: spatPortOptions.Table | /laxRows) e_Row. | Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 | GET the follow RECC ASSIGN Table GET the follow RECO | ing objects: maxRsuPorts. DRD this information as MaxRows. Row EQUALS RANDOM (1 TO N ring objects: spatPortOptions.Table RD this information as OriginalSpa | /laxRows) Row. tPortOptions. | Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 | GET the follow RECC ASSIGN Table GET the follow RECO ASSIGN TestS 4. | ing objects: maxRsuPorts. DRD this information as MaxRows. Row EQUALS RANDOM (1 TO N ring objects: spatPortOptions.Table RD this information as OriginalSpa SpatPortOptions EQUALS OriginalS | MaxRows) Row. tPortOptions. SpatPortOptions XOR | Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 5 | GET the follow RECC ASSIGN Table GET the follow RECO ASSIGN TestS 4. ASSIGN spatP | ring objects: maxRsuPorts. DRD this information as MaxRows. E_Row EQUALS RANDOM (1 TO Ming objects: spatPortOptions.Table RD this information as OriginalSpatPortOptions EQUALS OriginalSpatPortOptions.Table_Row EQUALS T | MaxRows) Row. tPortOptions. SpatPortOptions XOR estSpatPortOptions. | Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 5 6 | GET the follow RECC ASSIGN Table GET the follow RECO ASSIGN TestS 4. ASSIGN spatP SET the follow | ring objects: maxRsuPorts. DRD this information as MaxRows. PROW EQUALS RANDOM (1 TO N ring objects: spatPortOptions.Table RD this information as OriginalSpatPortOptions EQUALS OriginalS patPortOptions.Table_Row EQUALS T ing objects: spatPortOptions.Table | MaxRows) e_Row. tPortOptions. SpatPortOptions XOR estSpatPortOptions. _Row. | Pass/Fail Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 5 6 7 | GET the follow RECC ASSIGN Table GET the follow RECO ASSIGN TestS 4. ASSIGN spatP SET the follow GET the follow | ring objects: maxRsuPorts. DRD this information as MaxRows. PROW EQUALS RANDOM (1 TO Main the second seco | MaxRows) _Row. tPortOptions. SpatPortOptions XOR estSpatPortOptions. _Row. _Row. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| Number 1 1.1 2 3 3.1 4 5 6 7 8 | GET the follow RECC ASSIGN Table GET the follow RECO ASSIGN TestS 4. ASSIGN spatP SET the follow GET the follow VERIFY spatP TestSpatPortO | ring objects: maxRsuPorts. DRD this information as MaxRows. Row EQUALS RANDOM (1 TO Mathematical Stress S | MaxRows) ProtOptions. EpatPortOptions XOR EestSpatPortOptions. Row. Prov. TO | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 10 | SET the following objects: spatPortOptionsTable_Row. | | | |
|------------------------|---|-----------------|-----------|--|
| 11 | GET the following objects: spatPortOptions.T | Pass/Fail | | |
| 12 | VERIFY spatPortOptions.Table_Row IS_EQU OriginalSpatPortOptions. | Pass/Fail | | |
| Test Procedure Results | | | | |
| Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure Notes: | | | | |

C.3.15.38 Determine Maximum Number of Signal Groups

| Test Procedure: | | Determine Maximum Number of Signal Groups | | |
|------------------------|---|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the number of signal groups supported by the ASC for a connected vehicle environment. | | |
| Requirement(s): | | 3.5.4.2.1.12.1 Determine Maximum Number of Signal Groups | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | GET the following objects: maxSignalGroups. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: | Date Tested: | | Pass/Fail | |
| Test Procedure N | lotes: | | | |

C.3.15.39 Configure Signal Group Intersection Mapping

| Test Procedure: | Configure Signal Group Intersection Mapping | | |
|--------------------|--|---|--|
| Description: | This test case verifies that that ASC allows a management station to assign signal groups to a connected intersection. | | |
| Requirement(s): | 3.5.4.1.2.1.12.2 Configure Signal Group Intersection Mapping | | |
| Variable(s): | MaxRows Table_Row OriginalGroupIntersection TestGroupIntersection OriginalGroupID | maxSignalGroups Int signalGroupIntersection signalGroupIntersection signalGroupID | |

| | | TestGroupID | signalGroupID | |
|---------------------|--|--|-----------------------|-----------|
| Pass/Fail Crite | riteria: The device under test shall pass every verification step in this test case to pass the test case. | | ep in this | |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the follow | ing objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECC | RD this information as MaxRov | WS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows) | |
| 3 | GET the follow | ing objects: signalGroupInterse | ection.Table_Row. | Pass/Fail |
| 3.1 | RECO | RD this information as Original | GroupIntersection. | |
| 4 | ASSIGN TestG | roupIntersection EQUALS RAM | NDOM (1 TO 65535) | |
| 4.1 | IF Te: OrignalGroupIr | stGroupIntersection IS_EQUAL Itersection. | ТО | |
| 4.1.1 | | GOTO step 4. | | |
| 5 | GET the follow | ing objects: signalGroupID.Tab | le_Row. | Pass/Fail |
| 5.1 | RECO | RD this information as Original | GroupID. | |
| 6 | ASSIGN TestG | iroupID EQUALS RANDOM (1 | TO 255) | |
| 6.1 | IF Te | stGroupID IS_EQUAL_TO Orig | nalGroupID. | |
| 6.1.1 | | GOTO step 6. | | |
| 7 | ASSIGN signal TestGroupInter | GroupIntersection.Table_Row section. | EQUALS | |
| 8 | ASSIGN signal | GroupID.Table_Row EQUALS | TestGroupID. | |
| 9 | SET the followi signalGroupID. | ng objects: signalGroupInterse Table_Row. | ction.Table_Row, | Pass/Fail |
| 10 | GET the follow signalGroupID. | ing objects: signalGroupInterse Table_Row. | ection.Table_Row, | Pass/Fail |
| 11 | VERIFY signal TestGroupInter | VERIFY signalGroupIntersection.Table_Row IS_EQUAL_TO TestGroupIntersection. | | Pass/Fail |
| 12 | VERIFY signal | GroupID.Table_Row IS_EQUA | L_TO TestGroupID. | Pass/Fail |
| 13 | ASSIGN signal OriginalGrouple | GroupIntersection.Table_Row ntersection. | EQUALS | |
| 14 | ASSIGN signal | GroupID.Table_Row EQUALS | OriginalGroupID. | |
| 15 | SET the followi signalGroupID. | ng objects: signalGroupInterse Table_Row. | ction.Table_Row, | Pass/Fail |
| 16 | GET the following objects: signalGroupIntersection.Table_Row, signalGroupID.Table_Row. Pass/Fail | | | |
| 17 | VERIFY signal | GroupIntersection.Table_Row Intersection. | IS_EQUAL_TO | Pass/Fail |
| 18 | VERIFY signal | GroupID.Table_Row IS_EQUA | L_TO OriginalGroupID. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure N | lotes: | | | |

C.3.15.40 Configure Signal Group Control Source

| Test Procedure: | | 15.51 Configure Signal Group Control Source | |
|---------------------|--|--|----------------------------------|
| Description: | | This test case verifies that that ASC allows a manage station to configure the control source (phase or over controls a signal group in a connected intersection. | ment lap) that |
| Requirement(| s): | • 3.5.4.1.3.17.3 Configure Signal Group Control | Source |
| Variable(s): | | MaxRowsmaxSignalGroupsTable_RowIntMaxPhasesmaxPhasesOriginalControlSourcesignalGroupControlTestControlSourcesignalGroupControlOriginalControlTypesignalGroupControlTestControlTypesignalGroupControl | Source Source Type Type |
| Pass/Fail Crite | eria: | The device under test shall pass every verification ste test case to pass the test case. | ep in this |
| Test Step Number | | Test Procedure | Results |
| 1 | GET the following objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECORD this information as MaxRows. | | |
| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows) | | |
| 3 | GET the following objects: signalGroupID.Table_Row. | | Pass/Fail |
| 4 | GET the following objects: maxPhases. | | Pass/Fail |
| 4.1 | RECORD this information as MaxPhases. | | |
| 5 | GET the following objects: signalGroupControlSource.Table_Row. | | Pass/Fail |
| 5.1 | RECORD this information as OriginalControlSource. | | |
| 6 | ASSIGN TestControlSource EQUALS RANDOM (1 TO MaxPhases) | | |
| 6.1 | IF TestControlSource IS_EQUAL_TO OriginalControlSource. | | |
| 6.1.1 | | GOTO step 6. | |
| 7 | GET the follow | ing objects: signalGroupControlType.Table_Row. | Pass/Fail |
| 7.1 | RECO | RD this information as OriginalControlType. | |
| 8 | ASSIGN TestC | controlType EQUALS RANDOM (2 TO 6) | |
| 8.1 | IF Te | stControlType IS_EQUAL_TO OriginalControlType. | |
| 8.1.1 | | GOTO step 8. | |
| 9 | ASSIGN signal TestControlSou | GroupControlSource.Table_Row EQUALS urce. | |
| 10 | ASSIGN signal TestControlTyp | GroupControlType.Table_Row EQUALS be. | |
| 11 | SET the followi signalGroupCo | ng objects: signalGroupControlSource.Table_Row, ntrolType.Table_Row. | Pass/Fail |
| 12 | GET the follow signalGroupCo | ing objects: signalGroupControlSource.Table_Row, ntrolType.Table_Row. | Pass/Fail |
| 13 | VERIFY signal TestControlSou | GroupControlSource.Table_Row IS_EQUAL_TO urce. | Pass/Fail |

| 14 | VERIFY signalGroupControlType.Table_Row IS_EQUAL_TO TestControlType. | | |
|----------------|---|-----------------|-----------|
| 15 | ASSIGN signalGroupControlSource.Table_Row EQUALS OriginalControlSource. | | |
| 16 | ASSIGN signalGroupControlType.Table_Row OriginalControlType. | EQUALS | |
| 17 | SET the following objects: signalGroupControlSource.Table_Row, signalGroupControlType.Table_Row. | | Pass/Fail |
| 18 | GET the following objects: signalGroupControlSource.Table_Row, signalGroupControlType.Table_Row. | | Pass/Fail |
| 19 | VERIFY signalGroupControlSource.Table_Row IS_EQUAL_TO OriginalControlSource. | | Pass/Fail |
| 20 | VERIFY signalGroupControlType.Table_Row IS_EQUAL_TO OriginalControlType. | | Pass/Fail |
| | Test Procedure Results | 1 | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.15.41 Configure Signal Group Indication Types

| Test Procedure: | | Configure Signal Group Indication Types | | |
|---|--|--|------------------------|-----------|
| Description: | | This test case verifies that that ASC allows a management station to configure the type of red and green indications shown in SPaT information messages. | | |
| Requirement(| s): | • 3.5.4.2.1.12.4 Configure \$ | Signal Group Indicatio | on Types |
| MaxRows maxSignalGroups Table_Row Int Variable(s): OriginalGreenType signalGroupGreenType Variable(s): TestRedType signalGroupRedType OriginalRedType signalGroupRedType TestRedType signalGroupRedType | | vpe e | | |
| Pass/Fail Criteria: The device under test shall pass every verification st test case to pass the test case. | | every verification ste | p in this | |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the follow | ing objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECC | RD this information as MaxRows. | | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 TO Ma | axRows) | |
| 3 | GET the following objects: signalGroupID.Table_Row. | | Pass/Fail | |
| 3 | GET the following objects: signalGroupGreenType.Table_Row. | | Pass/Fail | |
| 3.1 | RECO | RD this information as OriginalGree | nType | |
| 4 | ASSIGN TestGreenType EQUALS RANDOM (2 TO 5) | | | |
| 4.1 | IF Te | stGreenType IS_EQUAL_TO Origna | alGreenType. | |
| 4.1.1 | | GOTO step 4. | | |

| 5 | GET the following objects: signalGroupRedTy | pe.Table_Row. | Pass/Fail |
|----------------|--|------------------------|-----------|
| 5.1 | RECORD this information as Original | RedType. | |
| 6 | IF OriginalRedType IS_EQUAL_TO 2 | | |
| 6.1 | ASSIGN TestRedType EQUALS 3 | | |
| 6.1.1 | GOTO step 8. | | |
| 7 | ASSIGN TestRedType EQUALS 2 | | |
| 8 | ASSIGN signalGroupGreenType.Table_Row | EQUALS TestGreenType | |
| 8 | ASSIGN signalGroupRedType.Table_Row EC | QUALS TestRedType. | |
| 9 | SET the following objects: signalGroupGreen signalGroupRedType. | Туре, | Pass/Fail |
| 10 | GET the following objects: signalGroupGreenType, signalGroupRedType. | | Pass/Fail |
| 11 | VERIFY signalGroupGreenType.Table_Row IS_EQUAL_TO TestGreenType. | | Pass/Fail |
| 12 | VERIFY signalGroupRedType.Table_Row IS_EQUAL_TO TestRedType. | | Pass/Fail |
| 13 | ASSIGN signalGroupGreenType.Table_Row OriginalGreenType | EQUALS | |
| 14 | ASSIGN signalGroupRedType.Table_Row EC | QUALS OriginalRedType. | |
| 15 | SET the following objects: signalGroupGreenType, signalGroupRedType. | | Pass/Fail |
| 16 | GET the following objects: signalGroupGreenType, signalGroupRedType. | | Pass/Fail |
| 17 | VERIFY signaGroupGreenType.Table_Row IS_EQUAL_TO OriginalGreenType. | | Pass/Fail |
| 18 | VERIFY signalGroupRedType.Table_Row IS_EQUAL_TO OriginalRedType. | | Pass/Fail |
| | Test Procedure Results | | |
| Tested By: | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | |

C.3.15.42 Configure Signal Group Protected or Permissive State

| Test Procedure: | 15.53 Configure Signal Grou | 15.53 Configure Signal Group Protected or Permissive State | | |
|--------------------|---|--|--|--|
| Description: | This test case verifies that the station to configure the cont has secondary control of a s intersection indicating a period | This test case verifies that that ASC allows a management station to configure the control source (phase or overlap) that has secondary control of a signal group in a connected intersection indicating a permissive movement. | | |
| Requirement(s): | • 3.5.4.2.1.12.5 Configu Permissive State | 3.5.4.2.1.12.5 Configure Signal Group Protected or Permissive State | | |
| Variable(s): | MaxRows Table_Row MaxPhases OriginalControlSource TestControlSource OriginalControlType | maxSignalGroups Int maxPhases signalGroupControlSource signalGroupControlSource signalGroupControlType | | |

| | | TestControlType | signalGroupControl | Гуре |
|---------------------|----------------------------------|---|-------------------------------------|-----------|
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ise. | p in this |
| Test Step Number | Test Procedure | | | Results |
| 1 | GET the followi | ing objects: maxSignalGroups. | | Pass/Fail |
| 1.1 | RECO | RD this information as MaxRov | WS. | |
| 2 | ASSIGN Table | _Row EQUALS RANDOM (1 T | O MaxRows) | |
| 3 | GET the followi | ing objects: signalGroupID.Tab | le_Row. | Pass/Fail |
| 4 | GET the followi | ing objects: maxPhases. | | Pass/Fail |
| 4.1 | RECO | RD this information as MaxPha | ases. | |
| 5 | GET the followi | ing objects: signalGroupContro | ISource.Table_Row. | Pass/Fail |
| 5.1 | RECO | RD this information as Original | ControlSource. | |
| 6 | ASSIGN TestC | ontrolSource EQUALS RANDO | DM (1 TO MaxPhases) | |
| 6.1 | IF Tes | stControlSource IS_EQUAL_T | O OrignalControlSource. | |
| 6.1.1 | | GOTO step 6. | | |
| 7 | GET the followi | ing objects: signalGroupContro | IType.Table_Row. | Pass/Fail |
| 7.1 | RECO | RD this information as Original | ControlType. | |
| 8 | ASSIGN TestC | ontrolType EQUALS RANDOM | 1 (2 TO 7) | |
| 8.1 | IF Tes | stControlType IS_EQUAL_TO | OrignalControlType. | |
| 8.1.1 | | GOTO step 8. | | |
| 9 | ASSIGN signal TestControlSou | GroupControlSource.Table_Rc urce. | w EQUALS | |
| 10 | ASSIGN signal TestControlTyp | GroupControlType.Table_Row be. | EQUALS | |
| 11 | SET the followi signalGroupCo | ng objects: signalGroupContro ntrolType.Table_Row. | Source.Table_Row, | Pass/Fail |
| 12 | GET the followi signalGroupCo | ing objects: signalGroupContro ntrolType.Table_Row. | ISource.Table_Row, | Pass/Fail |
| 13 | VERIFY signal | GroupControlSource.Table_Ro urce. | w IS_EQUAL_TO | Pass/Fail |
| 14 | VERIFY signal TestControlTyp | GroupControlType.Table_Row be. | IS_EQUAL_TO | Pass/Fail |
| 15 | ASSIGN signal OriginalControl | GroupControlSource.Table_Ro Source. | W EQUALS | |
| 16 | ASSIGN signal OriginalControl | GroupControlType.Table_Row Type. | EQUALS | |
| 17 | SET the followi signalGroupCo | ng objects: signalGroupContro ntrolType.Table_Row. | Source.Table_Row, | Pass/Fail |
| 18 | GET the followi signalGroupCo | ng objects: signalGroupContro ntrolType.Table_Row | ISource.Table_Row, | Pass/Fail |
| 19 | VERIFY signal OriginalControl | GroupControlSource.Table_Ro Source. | w IS_EQUAL_TO | Pass/Fail |
| 20 | VERIFY signal OriginalControl | GroupControlType.Table_Row Type | IS_EQUAL_TO | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |

Test Procedure Notes:

C.3.15.43 Configure Signal Group Revocable Lanes

| Test Procedure: | Configure Signal Group Revocable Lanes | | | |
|---------------------|--|--|--|------------------|
| Description: | | This test case verifies that station to configure what si the revocable lanes that are | that ASC allows a manager gnal groups may be active made active. | nent based on |
| Requirement(| s): | • 3.5.4.1.3.17.6 Config | gure Signal Group Permiss | ive Lanes |
| Variable(s): | | MaxRows Table_Row MaxLanes OriginalLanes TestLanes | maxSignalGroups Int Int signalLane signalGroupLane | |
| Pass/Fail Crite | eria: | The device under test shall test case to pass the test ca | pass every verification ste ase. | ep in this |
| Test Step Number | Test Procedure | | Results | |
| 1 | PRE-CONDITION 'The user shall know the maximum number of | | | |
| 11 | RECORD this information as Mayl anes | | | |
| 2 | GET the following objects: maxSignalGroups | | Pass/Fail | |
| 2.1 | RECORD this information as MaxRows. | | 1 400/1 41 | |
| 3 | ASSIGN Table Row EQUALS RANDOM (1 TO MaxRows) | | | |
| 4 | GET the following objects: signalGroupID.Table Row. | | Pass/Fail | |
| 5 | GET the follow | ET the following objects: signalGroupControLane.Table Row. Pa | | Pass/Fail |
| 5.1 | RECO | RD this information as Original | Lanes. | |
| 6 | ASSIGN TestC | controlSource EQUALS RAND | OM (1 TO MaxLanes) | |
| 6.1 | IF Te | stLanes IS_EQUAL_TO Origna | alLanes. | |
| 6.1.1 | | GOTO step 6. | | |
| 7 | ASSIGN signal | GroupLane.Table_Row EQUA | LS TestLanes. | |
| 8 | SET the follow | ing objects: signalGroupLane.T | able_Row. | Pass/Fail |
| 10 | GET the follow | ing objects: signalGroupLane.] | Table_Row. | Pass/Fail |
| 11 | VERIFY signal | GroupLane.Table_Row IS_EQ | UAL_TO TestLanes. | Pass/Fail |
| 12 | ASSIGN signal | lGroupLane.Table_Row EQUA | LS OriginalLane. | |
| 13 | SET the follow | ing objects: signalGroupLane.T | able_Row. | Pass/Fail |
| 14 | GET the follow | ollowing objects: signalGroupLane.Table_Row. Pass/Fail | | Pass/Fail |
| 15 | VERIFY signal | GroupLane.Table_Row IS_EQ | UAL_TO OriginalLanes. | Pass/Fail |
| | | Test Procedure Results | | |
| Tested By: | | | Date Tested: | Pass/Fail |
| Test Procedure | Notes: | | | - |

| Test Procedure: | | Determine Maximum Number of Signal State Entries | | |
|----------------------------|---|--|-------------------------------------|-----------|
| Description: | | This test case verifies that the ASC allows a management station to view the number of signal states that may be configured. | | |
| Requirement(s): | | 3.5.4.1.2.1.12.7 Determine Maximum Number of Signal Groups | | |
| Variable(s): | | | | |
| Pass/Fail Criteria: | | The device under test shall test case to pass the test ca | pass every verification ste ase. | p in this |
| Test Step Number | | Test Procedure | | Results |
| 1 | GET the following objects: maxAgencySignalStates. | | Pass/Fail | |
| Test Procedure Results | | | | |
| Tested By: Date Tested By: | | Date Tested: | Pass/Fail | |
| Test Procedure N | Notes: | | | |

C.3.15.44 Determine Maximum Number of Signal State Entries

C.3.15.45 Determine Maximum Number of Signal State Entries

| Test Procedure: | | Configure Signal State Parameters | | |
|---|----------------|---|-------------------------|------------------------------------|
| Description: This test case verifies that that ASC allows a management station to configure what signal states may be indic SPaT information messages dependent on signal c permissive movements. | | ASC allows a manager states may be indicat pendent on signal colo | nent ed in or and | |
| Requirement(| s): | • 3.5.4.1.3.17.8 Configure Signal State Parameters | | |
| Variable(s): | | MaxRowsmaxSignalGroupsTable_RowIntOriginalColoragencySignalStateColorOriginalControlTypeagencySignalStateControlTypOriginalStateOptionsagencySignalStateOptionsOriginalStateValueagencySignalStateValue | | olor trolType ptions alue |
| Pass/Fail Crite | eria: | The device under test shall pass every verification step in this test case to pass the test case. | | p in this |
| Test Step Number | | Test Procedure R | | Results |
| 1 | GET the follow | ing objects: maxAgencySignalState | S. | Pass/Fail |
| 1.1 | RECC | RD this information as MaxRows. | | |

| 2 | ASSIGN Table_Row EQUALS RANDOM (1 TO MaxRows) | |
|--|--|---|
| 3 | GET the following objects: agencySignalStateColor.Table_Row. | Pass/Fail |
| 3.1 | RECORD this information as OriginalColor | |
| 4 | GET the following objects: agencySignalStateControlType.Table_Row | Pass/Fail |
| 4.1 | RECORD this information as OriginalControlType. | |
| 5 | GET the following objects: agencySignalStateOptions.Table Row. | Pass/Fail |
| 5.1 | RECORD this information as OriginalStateOptions. | |
| 6 | GET the following objects: agencySignalStateValue.Table Row. | Pass/Fail |
| 6.1 | RECORD this information as OriginalStateValue. | |
| 7 | ASSIGN agencySignalStateColor.Table Row EQUALS 2. | |
| 7.1 | NOTE '2 = red'. | |
| 8 | ASSIGN agencyStateControlType.Table Row EQUALS 2. | |
| 8.1 | NOTE '2 = vehicle'. | |
| 9 | ASSIGN agencySignalStateOptions.Table Row EQUALS 272 | |
| 10 | ASSIGN agencySignalStateValue.Table Row EQUALS 5 | |
| 10.1 | NOTE '2 = stopAndRemain'. | |
| | SET the following objects: agencySignalStateColor.Table Row. | |
| 4.4 | agencySignalStateControlType.Table_Row, | |
| 11 | agencySignalStateOptions.Table_Row, | Pass/Fall |
| | agencySignalStateValue.Table_Row. | |
| | GET the following objects: agencySignalStateColor.Table_Row, | |
| 12 | agencySignalStateControlType.Table_Row, | Pass/Fail |
| | agencySignalStateOptions. Lable_Row, | - |
| 4.5 | | |
| 40 | VEDIEV exercise allestate Cales Table Davy IC EQUAL TO 2 | |
| 13 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. | Pass/Fail |
| 13 13 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. | Pass/Fail Pass/Fail |
| 13 13 14 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. | Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions.Table_Row EQUALS | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. | Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. SET the following objects: agencySignalStateColor.Table_Row, | Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateControlType.Table_Row, | Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, | Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 21 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateOptions.Table_Row, agencySignal | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 21 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateOptions.Table_Row, | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 21 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 21 23 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencySignalStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateValue.Table_Row, agencySignalStateOptions.Table_Row. VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO OriginalColor. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 13 14 15 16 17 18 19 20 21 23 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOp | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 14 15 16 17 18 19 20 21 23 24 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 272. VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, AgencySignalStateOptions.Table_Row, Ag | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |
| 13 13 13 14 15 16 17 18 19 20 21 23 24 | VERIFY agencySignalStateColor.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateControlType.Table_Row IS_EQUAL_TO 2. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO 5. ASSIGN agencySignalStateColor.Table_Row EQUALS OriginalColor. ASSIGN agencyStateControlType.Table_Row EQUALS OriginalControlType. ASSIGN agencySignalStateOptions.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateOptions ASSIGN agencySignalStateValue.Table_Row EQUALS OriginalStateValue. SET the following objects: agencySignalStateColor.Table_Row, agencySignalStateOptions.Table_Row IS_EQUAL_TO OriginalColor. VERIFY agencySignalStateOptions.Table_Row IS_EQUAL_TO OriginalControlType. | Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail Pass/Fail |

| 26 VERIFY agencySignalStateValue.Table_Row IS_EQUAL_TO OriginalStateValue. | | | |
|---|--------|--|--|
| Test Procedure Results | | | |
| Tested By: Date Tested: | | | |
| Test Procedure N | lotes: | | |

C.3.15.46 Retrieve Signal Phase and Timing Time Point

| Test Procedure: | | Retrieve Signal Phase and Timing Time Point | | | | |
|-----------------------|---|--|-----------------|-----------|--|--|
| Description: | | This test case verifies that the ASC allows a management station to view movement current time on the ASC used as the reference for SPaT data generation. This time may differ from the time on an RSU. | | | | |
| Requirement(s): | | 3.5.4.1.3.18 Retrieve Signal Phase and Timing Time Point V2 | | | | |
| Variable(s): | | | | | | |
| Pass/Fail Criteria: | | The device under test shall pass every verification step in this test case to pass the test case. | | | | |
| Test Step Number | | Test Procedure | | Results | | |
| 1 | GET the following objects: ascCurrentTick2. | | | Pass/Fail | | |
| | Test Procedure Results | | | | | |
| Tested By: | | | Date Tested: | Pass/Fail | | |
| Test Procedure Notes: | | | | | | |

C.3.15.47 Retrieve MAP Plan in Effect

| Test Procedure: | Retrieve MAP Plan in Effect | |
|---------------------|---|--|
| Description: | This test case verifies that the ASC allows a management station to view the MAP plan currently being broadcasted in a connected vehicle environment. | |
| Requirement(s): | • 3.5.4.3.3.1 Retrieve MAP Plan in Effect | |
| Variable(s): | | |
| Pass/Fail Criteria: | The device under test shall pass every verification step in this test case to pass the test case. | |

| Test Step Number | Test Procedure | | | | | |
|------------------------|---|-----------------|-----------|--|--|--|
| 1 | GET the following objects: mapActivatePlan. | | | | | |
| Test Procedure Results | | | | | | |
| Tested By: | | Date Tested: | Pass/Fail | | | |
| Test Procedure Notes: | | | | | | |