ATC 5201 v06B

This is a connected intersection-ready Advanced Transportation Controller (ATC) Standard family developed by AASHTO, ITE, and NEMA

November 10, 2022

Advanced Transportation Controller (ATC) v06B.01

(Working Group Draft)

Supported/Sponsored By: The United States Department of Transportation (USDOT)

U.S. Department of Transportation

Foreword

The Advanced Transportation Controller Standard (ATC Standard) is one of the four standards of the ATC family of standards. The ATC Standard provides a powerful, on-street, computing platform for numerous ITS applications intersection control, ramp metering, data collection, safety, security, and other applications. It uses a transportation controller architecture where the computational components of the controller reside on a single small printed circuit board (PCB), called the “Engine Board,” with standardized connectors and pinout. It is made up of a central processing unit (CPU), Linux operating system (O/S), memory, external and internal interfaces, and other associated hardware necessary to create an embedded transportation computing platform. The Engine Board plugs into a “Host Module” which supplies power and physical connection to the input/output (I/O) facilities of the controller. While the interface to the Engine Board is completely specified, the Host Module may be of various shapes and sizes to accommodate innumerable transportation controller designs and cabinet architectures. The ATC Standard works together with the Application Programming Interface (API) Standard to allow application programs to share the resources of the controller and the transportation field cabinet system.

More information on this standard effort can be found on the ITE website.

The Standards Development Organizations (SDOs) supporting this standard include the following:

ITE

Siva Narla, snarla@ite.org
Tatiana Richey, trichey@ite.org
Nicola Tavares, standards@ite.org

AASHTO

Venkat Nallamothu, vnallamothu@aashto.org
Gummada Murthy, gmurthy@aashto.org

NEMA

Brian Doherty, brian.doherty@nema.org
Steve Griffith, steve.griffith@nema.org

Controller Working Group Co-chairs

John Thai, City of Anaheim (Public Sector)
Jim Rose, Econolite (Private Sector)

Controller Working Group (WG) Members

Ralph Boaz, Pillar Consulting, Inc.
George Chen, LADOT
Mike Gallagher, Q-Free
Robin Harrison, Oriux
Herasmo Iniguez, McCain
Mohammad Iraki, CalTrans
Dave Miller, Yunex Traffic
Clyde Neel, Cubic
Robert Rausch, TransCore
Peter Skewers, Minnesota DOT
Mohamed Talas, NYCDOT
Derek Vollmer, Florida DOT
April Wire, Maricopa County, AZ DOT

ATC Joint Committee Members

Doug Crawford, Q-Free
Alan Davis, Georgia DOT
Kleinjan Deetlefs, Applied Information
Steve Gault, Pennsylvania DOT
Matt Luker, Utah DOT
Dave Miller, Yunex Traffic
Robert Rausch, TransCore
Edward Seymour, Texas Transportation Institute
Doug Spencer, ODOT
Mohamed Talas, New York City DOT
John Thai, City of Anaheim
Andy Dowie, Eberle Design Inc.
Jon Wyatt, Parsons
Derek Vollmer, Florida DOT

Subject Matter Experts (SMEs)

Ralph Boaz, Pillar Consulting, Inc.
James Kinnard, Adaptive Solutions, Inc.

Copyright Notice

Content and Liability Disclaimer

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document. NEMA, AASHTO and ITE standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and/or seeks out the views of persons who have an interest in the topic covered by this publication. While NEMA, AASHTO and ITE administer the process and establish rules to promote fairness in the development of consensus, they do not write the document and they do not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in their standards and guideline publications. NEMA, AASHTO or ITE disclaim liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly, or indirectly resulting from the publication, use of, application, or reliance on this document. NEMA, AASHTO and ITE disclaim and make no guaranty or warranty, express or implied, as to the accuracy or completeness of any information published herein and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. NEMA, AASHTO and ITE do not undertake to guarantee the performance of any individual manufacturer or seller’s products or services by virtue of this standard or guide.

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

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

Additional Contributors and Reviewers

In addition to the SDOS, Co-Chairs, WG Members and SMEs, there were many others that contributed to the development of this standard and their input and assistance was critical to the final product. The following list includes those volunteers and others who gave their time to help both the consultant and the committee ensure that the resulting standard met their needs. The following is a more complete list of those who volunteered their time and travel to contribute to the input and review during the development of this standard: The Standards Discussion Groups on ITE Community (ATC Community).

CHANGE HISTORY

DATE	VERSION	NOTE
06/26/2006	Version 5.2b	Release version.
07/14/2011	Version 6.0a	Draft for Working Group review. Includes disposition of working group comments submitted for ATC Standard v5.2b.
06/08/2012	Versions 06.01 –06.09	Technical updates and document reformatting.
07/30/2012	Version 06.10	User Comment Draft.
06/15/2015	Version 06.20	Working Group Draft following the incorporation of changes based on the adjudication comments by the ATC WG on UCD ATC 5201 Standard v06.10.
08/10/2015	Version 06.21	Second Working Group Draft to address comments received regarding first Working Group Draft v06.20.
08/24/2015	Version 06.22	Third Working Group Draft to address comments received regarding Working Group Drafts v06.20 and v06.21.
08/31/2015	Version 06.23	Proposed Recommended Standard (pRS); addresses comments received regarding Working Group Drafts v06.20, v06.21 and v06.22.
10/04/2015	Version 06.24	Recommended Standard accepted by the ATC Joint Committee.
01/12/2018	Version 06.25	Joint Standard of AASHTO, ITE and NEMA.
09/30/2017	Version 06.30	Working Group Draft (WGD) to addressing comments received regarding Recommended Standard v06.24. Note that the date of this WGD precedes that of Version 06.25 due to a delay in the approval process.
05/09/2018	Version 06.31	Proposed User Comment Draft (pUCD) sent to the ATC Joint Committee.
06/30/2018	Version 06.32	User Comment Draft (UCD). Administrative advancement of the minor version number.
04/30/2019	Version 06.33	Working Group Draft with comments on the UCD version adjudicated and addressed. Subsequently voted by the Controller WG to send to the ATC Joint Committee as a Proposed Recommended Standard on 5/14/19.
05/24/2019	Version 06A.34	Proposed Recommended Standard (pRS) sent to the ATC Joint Committee for review. Editorial updates made after Controller WG acceptance of v06.33 are included. Subsequently accepted by vote of the ATC JC as a Recommended Standard on 06/20/19.
09/03/2019	Version 06A.35	Editorial updates post ATC Joint Committee acceptance of v06A.34.
01/10/2020	Version 06A.36	Recommended Standard sent to AASHTO, ITE and NEMA for final ballot. Includes additional editorial updates.
07/29/2020	Version 06A.37	ATC 5401 v06A (v06A.37). Joint Standard of AASHTO, ITE and NEMA. Minor editorial corrections and changes made for publication.
11/10/2022	Version 06B.01	Working Group Draft following the incorporation of changes based on the adjudication of comments by the ATC WG on ATC 5201 Standard v06A.37.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This section provides an introduction for this document. It includes sections on “Purpose,” “Scope,” “Definitions, Acronyms, and Abbreviations;” “References,” and “Overview.”

1.1 Purpose

The Advanced Transportation Controller (ATC) standards are intended to provide an open architecture hardware and software platform that can support a wide variety of Intelligent Transportation Systems (ITS) applications including traffic management, safety, security, and other applications. The ATC standards are being developed and maintained under the direction of the ATC Joint Committee (JC) that is made up of representatives from the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE) and the National Electrical Manufacturers Association (NEMA).

This standard defines a transportation field device known as an ATC. It has been prepared by the Controller Working Group (WG), a technical subcommittee of the ATC JC. It establishes a common understanding of the specifications for an ATC for the following:

a) Local, state, and federal transportation agencies who specify and use ATC equipment

b) Manufacturers who produce ATC equipment

c) Software developers who develop application programs for ATC equipment

d) The public who benefit from the application programs that run on ATC equipment and who directly or indirectly pays for these products

1.2 Scope

The ATC Standard defines a minimum required functionality of hardware and software for ATC-conforming transportation controllers. It includes a minimum processing capability, specifies a multi-tasking operating system, and a minimum set of user and communications interfaces. There is no limit on the use of technologies as long as these minimums are met. Its open architecture approach allows software to be purchased independently of the controller hardware and enables the ATC to be used for traditional traffic applications, advanced Intelligent Transportation Systems (ITS) or any other application requiring an on-street computing platform.

Other than the minimum physical interfaces, the ATC 5201 Standard is not specific on most mechanical aspects of the controller. This allows controllers to be built in conformance with this standard that will be compatible with all of the major transportation field cabinet systems and controllers available today including the Model 332 (family), NEMA TS 1, NEMA TS 2, Model 2070, ITS Cabinets and ATC Cabinets. See Section 1.4 for the referenced documents that describe the various field equipment standards and specifications.

1.2.1 Procurement Guidance

The ATC 5201 Standard requires the use of a minimum version of the Linux operating system and the inclusion of a Board Support Package (BSP). Agencies specifying the use of an ATC need to recognize that the ATC 5201 Standard does not prescribe a specific processor, processor clock speed, or a specific version of Linux; as a result, existing software may not be compatible with fully conformant ATCs from all vendors. Agencies should expect that the application software may need to be compiled and linked for the specific hardware, board support package, and version of Linux supplied since the object code may not be compatible for different processors. Further, some software may require modifications for specific hardware/platform [Linux and BSP] configurations. When purchasing ATC hardware separately from the application software, the agency may require software support services from their software vendor to verify compatibility,

test the application, or to modify their application software for the specific platform and vendor chosen.

It is essential to understand that the ATC is a functional standard that identifies minimum requirements with a variety of options. This standard does not attempt to limit the features and future capabilities of the ATC. The intent is to define a transportation control platform that supports source code portability and interchangeability of application software programs. Agencies also need to recognize that it is inappropriate for a procurement specification to require the hardware vendor to ensure compatibility with a software package from another vendor (typically a competitor) or the agency itself, unless the source code is provided to the hardware vendor. Technology today advances rapidly, and it is not unusual for hardware obsolescence to outpace the life of the product in traffic control devices. The ATC 5201 Standard has been structured to allow for advances in technology (hardware and software) and to allow designs to change over time to adopt newer and better technology, which is sometimes necessitated by the obsolescence of key components. Such changes may require modification of application software programs.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Physical Units

Wherever the following units are used, the intent and meaning shall be interpreted as follows:

A	Ampere
b	bit
bps	bits per second
B	byte
˚C	Degrees Celsius
dB	Decibel
dBa	Decibels above reference noise, adjusted
F	Farad
ft	foot
g	gram
G	Earth gravitational constant
Hz	Hertz
in	inches
J	Joule
m	meter
N	Newton
Ω	Ohm
s	second
V	Volt
W	Watt

1.3.2 Modifiers

Wherever the following modifiers are used as a prefix to a physical unit, the intent and meaning shall be interpreted as follows:

k	kilo = 1000
M	Mega = 1 000 000
m	milli = 0.001
µ	micro = 0.000 001
n	nano = 0.000 000 001
p	pico = 0.000 000 000 001

1.3.3 Acronyms and Definitions

AASHTO	American Association of State Highway and Transportation Officials
AC	Alternating Current
AC-	120 VAC, 60 Hz neutral (grounded return to the power source)
AC+	120 VAC, 60 Hz line source (ungrounded)
ANSI	American National Standard Institute
ASCII	American Standard Code for Information Interchange
Assembly	A complete machine, structure, or unit of a machine that was manufactured by fitting together parts and/or modules
ASTM	American Society for Testing and Materials
ASYNC	Asynchronous Serial Communications
ATC	Advanced Transportation Controller
AWG	American Wire Gage
BSP	Board Support Package
Cabinet	An outdoor enclosure generally housing the controller unit and associated equipment
Caltrans	California Department of Transportation
CD	Carrier Detect
Component	Any electrical or electronic device
CPU	Central Processing Unit
CTS	Clear to send (data)
DAT	Design Acceptance Testing
DC	Direct Current
DCD	Data Carrier Detect (receive line signal detector)
DRAM	Dynamic Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
EG	Equipment Ground
EIA	Electronic Industries Association
EIA-694	An EIA standard defining electrical characteristics only for an unbalanced digital interface with data signaling rates up to 512kbps (electrically compatible with EIA-232 but with higher data rates)
EL	Electro-luminescent
EMI	Electromagnetic Interference
ENET	Ethernet

EPROM	Ultraviolet Erasable, Programmable, Read-Only Memory
Equal	Connectors: conform to physical dimensions, contact material, plating, and method of connection. Devices: conform to function, pin out, electrical and operating parameter requirements, access times and interface parameters of the specified device
ETL	Electrical Testing Laboratories, Inc.
FCU	Field Controller Unit
Firmware	A computer program or software stored permanently in PROM, EPROM, ROM, or semi-permanently in EEPROM
FLASH	Long-Term Non-Volatile Memory: a form of EEPROM that allows multiple memory locations to be erased or written in one programming operation. It is solid-state, permanent, and non-volatile memory typically having fast access and read/write cycles
FPA	Front Panel Assembly
FSK	Frequency Shift Keying
GPS	Global Positioning System
HDLC	High-level Data Link Control
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISO	International Standards Organization
ITE	Institute of Transportation Engineers
ITS	Intelligent Transportation Systems (including standards approved by AASHTO/NEMA/ITE)
Jumper	A means of connecting/disconnecting two or more conductors by soldering/desoldering a conductive wire or by PCB post-jumper
Keyed	Means by which like connectors can be physically altered to prevent improper insertion
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LOGIC	Negative logic convention (Ground True) state
logic-level	HCT or equivalent TTL – compatible voltage and fanout interface levels
lsb	Least Significant Bit
LSB	Least Significant Byte
MIPS	Million Instructions Per Second
Model 2070	A traffic control device that meets one of the California Department of Transportation (Caltrans) Transportation Electrical Equipment Specifications for Model 2070 traffic control devices.
Module	A functional unit that plugs into an assembly
msb	Most Significant Bit
MS	Military Specification, Mil-Spec or Mil-Standard
MSB	Most Significant Byte
N/A	Not Applicable
NA	Presently Not Assigned. Cannot be used by the contractor for other purposes.

NEMA	National Electrical Manufacturer's Association
NETA	InterNational Electrical Testing Association
NLSB	Next Least Significant Byte
NMSB	Next Most Significant Byte
NTCIP	National Transportation Communication for ITS Protocol
NTP	Network Time Protocol
NYSDOT	New York State Department of Transportation
O/S	Operating System
Open System	Standardized hardware interfaces in a module
OST	Operating System Time
PCB	Printed Circuit Board
RAM	Random Access Memory
RF	Radio Frequency
RMS	Root mean square
ROM	Read only memory
RTC	Real Time Clock
RTS	Request to send (data)
RX	Abbreviation for “Receive” when used to describe communication signals. Typically, a prefix for other character(s).
RXC	Receive Clock
RXD	Receive Data
SDLC	Synchronous Data Link Control
SP	Serial Port
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SYNC	Synchronous Serial Communications
TEES	Transportation Electrical Equipment Specifications
TMC	Transportation Management Center
TOD	Time Of Day Clock
TTL	Transistor-Transistor Logic
TX	Abbreviation for “Transmit” when used to describe communication signals. Typically a prefix for other character(s).
TXC	Transmit Clock
TXD	Transmit Data
UL	Underwriter's Laboratories, Inc.
USB	Universal Serial Bus
VAC	Volts Alternating Current
VDC	Volts Direct Current
WDT	Watchdog Timer: A monitoring circuit, external to the device watched, which senses an Output Line from the device and reacts

1.4 References

Normative references contain provisions that, through reference in this text, constitute provisions of ATC 5201 v06. Other references in ATC 5201 v06 may provide a better understanding or provide additional information about certain features.

At the time of publication, the editions indicated below were valid. All standards are subject to revision, and parties to agreements based on ATC 5201 v06 are encouraged to investigate the possibility of applying the most recent editions of the standards listed.

"ATC 5201 v06, Advanced Transportation Controller (ATC) Standard Version 6," ATC JC, 29 July 2020. Available from the Institute of Transportation Engineers. https://www.ite.org/technical-resources/standards/

"ATC 5301 v02, Advanced Transportation Controller (ATC) Cabinet Standard Version 02," ATC JC, 18 March 2019. Available from the Institute of Transportation Engineers. https://www.ite.org/technical-resources/standards/

"ATC 5401 v02, Application Programming Interface (API) Standard for the Advanced Transportation Controller (ATC) Version 2," ATC JC, 15 September 2013. Available from the Institute of Transportation Engineers. https://www.ite.org/technical-resources/standards/

"Caltrans Transportation Electrical Equipment Specifications (TEES)," California Department of Transportation, 12 March 2009. Available from the California Department of Transportation. http://www.dot.ca.gov/trafficops/tech/tees.html

IEEE 802.3 Ethernet Specifications. Available from the Institute of Electrical and Electronics Engineers.

"Intelligent Transportation System (ITS) Standard Specification for Roadside Cabinets v01.02.17b," ATC JC, 16 November 2006. Available from the Institute of Transportation Engineers. https://www.ite.org/technical-resources/standards/

"ISO/IEC 13239:2002, Information technology -- Telecommunications and information exchange between systems -- High-level data link control (HDLC) procedures." Available from the International Organization for Standardization (ISO).

"ITU-T X.680 Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation," International Telecommunication Union, August 2015. Available from the International Telecommunication Union (ITU).

"NEMA TS 1-1989, Traffic Control Systems," NEMA, 1989. Available from the National Electrical Manufacturers Association.

"NEMA TS 2-2016, Traffic Controller Assemblies with NTCIP Requirements Version 03.07," NEMA, 2016. Available from the National Electrical Manufacturers Association.

Universal Serial Bus (USB) Specifications. Available from USB Implementers Forum, Inc.

1.4.1 Contact Information

1.4.1.1 ISO/IEC Standards Information

ISO/IEC standards can be purchased on-line in electronic format or printed copy from:

Techstreet
6300 Interfirst Dr.
Ann Arbor, MI 48108
(800) 699-9277
www.techstreet.com

1.4.1.2 NEMA Standards Information

NEMA standards can be purchased on-line in electronic format or printed copy from:

Techstreet
6300 Interfirst Dr.
Ann Arbor, MI 48108
(800) 699-9277
www.techstreet.com

1.5 Overview

This standard is made up of 9 sections and three appendices. Section 1, “Introduction,” provides an overview of the entire document. Section 2, “Overall Description,” provides the background information and context necessary for the requirements. Section 3, “Functional Requirements” identifies the requirements of an ATC based on the “Representative Usage” described in Section 2. Sections 4-9 contain the detailed requirements and specifications for the ATC 5201 Standard.

2 OVERALL DESCRIPTION

This section provides an overall description for the ATC. It includes sections on “Product Perspective,” “Operational Environment,” “Representative Usage,” “Modes of Operation” and “Security.”

2.1 Product Perspective

Transportation controllers are environmentally-ruggedized computational devices used for on-street field applications. They are designed to operate within a field cabinet system to carry out the functions of installed application programs. The most common of the field cabinet systems are those defined by the Model 332 (type) Specifications, the NEMA TS 1 Standard, the NEMA TS 2 Standard, the ITS Cabinet Standard and the ATC Cabinet Standard. Since the 1980’s, the demand for more sophisticated transportation controllers has gone from simple intersection control to advanced Intelligent Transportation Systems (ITS) applications such as adaptive signal control, active traffic management and real-time vehicle-infrastructure systems. This has created a need for traffic controllers that are capable of performing a diversity of tasks, running concurrent application programs, and operating in high-speed communication networks.

There have been two general approaches to transportation controller architecture: 1) “open architecture” controllers where the hardware is so specified that third parties can develop application software for the device and 2) “closed architecture” controllers where the hardware and application software are produced and sold as a package. In open architecture controllers, the controller hardware is so specified that third party software developers can provide application programs for the device. In closed architecture controllers, the manufacturer provides both the hardware and the software for the device. One of the benefits of open architecture controllers are greater competition since any manufacturer can develop the hardware and third parties can provide application programs. One of the disadvantages is that, in the effort to provide portability, the technology used in the specification has been fixed (i.e., processor, commercial off-the-shelf operating systems, memory, etc.). This has led to underpowered equipment and early obsolescence. One of the benefits of closed architecture controllers has been their ability to take advantage of new technologies. Since the computational aspects of the hardware are not explicit, manufacturers have had the ability to apply innovative solutions more expeditiously in their transportation controllers. The drawbacks of the closed architecture approach are less competition and the inability to obtain application programs or custom software from third parties.

The ATC 5201 Standard has been developed to leverage the advantages of both architectural approaches. It specifies an open architecture computational capability intended to augment all of the current transportation equipment standards. The standard also has a built-in methodology for ATC units to grow with technology in order to provide for transportation controllers in the future. Some of the features of the ATC 5201 Standard are as follows:

- Works in Existing Cabinet Architectures. This is so users do not need to replace all of their field equipment and still benefit from transportation controllers built to the ATC 5201 Standard.
- Stated Minimum Processing Capability. This is necessary so that application developers (other than the original equipment manufacturer) understand what resources are guaranteed to be available for all ATC units.
- Standardized Method for Upgrading ATC Units. This mitigates issues with obsolescence and allows the standard to maintain relevancy for the future.
- Multi-Tasking Open Source Operating System. This is so software developers can write innovative application programs that can be made to operate on all ATC units and also

allows these application programs to run concurrently on the ATC. An additional benefit is that runtime license fees may not apply in these circumstances helping to reduce the cost to the user.

- Various I/O Capabilities. This is to allow the ATC to perform internal cabinet system operations and external communications for whatever architecture or applications the ATC is designed for.

The ATC 5201 Standard uses a transportation controller architecture where the computational components of the controller reside on a single small printed circuit board (PCB), called the "Engine Board," with standardized connectors and pinout. It is made up of a central processing unit (CPU), Linux operating system (O/S), memory, external and internal interfaces, and other associated hardware necessary to create an embedded transportation computing platform (see Figure 2-1). The Engine Board plugs into a Host Module which supplies power and physical connection to the input/output (I/O) facilities of the controller. While the interface to the Engine Board is completely specified, the Host Module may be of various shapes and sizes to accommodate innumerable transportation controller designs and cabinet architectures. Figure 2-2 illustrates how the Engine Board and Host Module can be used in different types of transportation controllers. Figure 2-3 shows how a controller conformant with two different existing cabinet standards can utilize the architecture defined by the ATC Standard.

Guidance: In addition to the Linux operating system, software libraries are available to allow concurrently running application programs to share the resources of the ATC unit and its field cabinet system. These libraries and functionality are defined by the Application Programming Interface (API) Standard (see Section 1.4).

Guidance: The inclusion of an Engine Board in a transportation controller is not sufficient for said controller to be considered conformant to the ATC 5201 Standard. All other elements of the standard must also be satisfied.

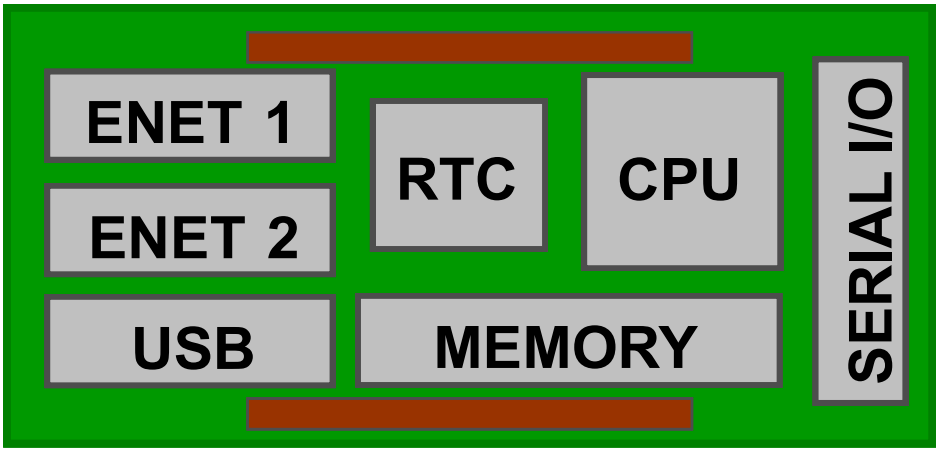


Figure 2-1. Block diagram of the ATC Engine Board.

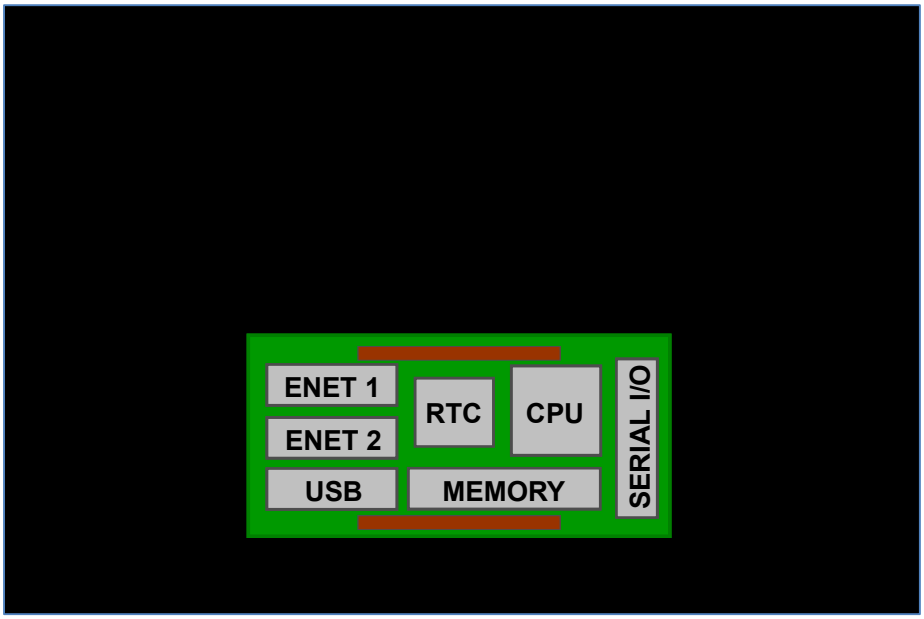


Figure 2-2. ATC Engine Board used in Host Modules for different types of transportation controllers.

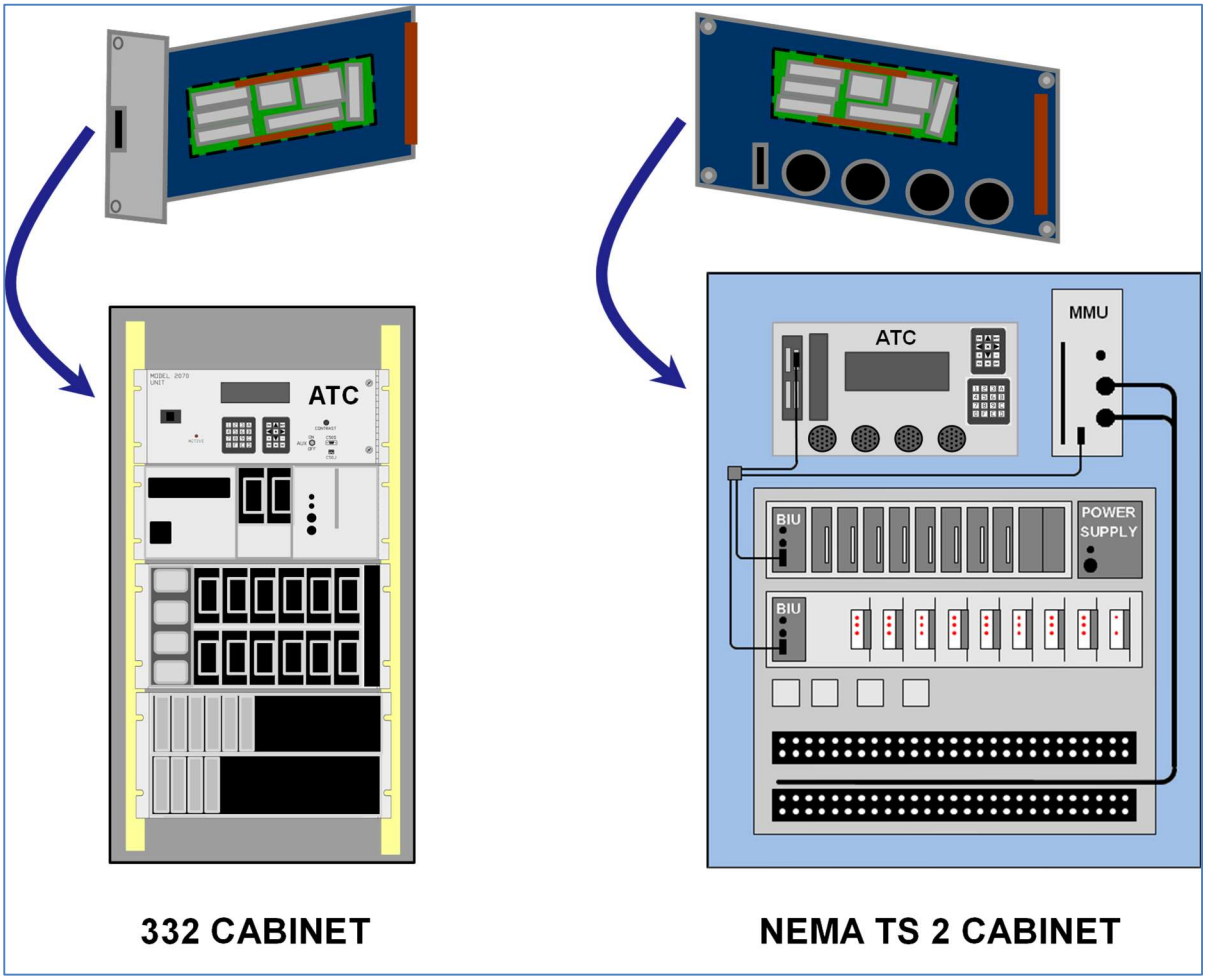


Figure 2-3. ATC units used in existing Cabinet Systems.

2.2 Operational Environment

Typically, an operator interfaces to an ATC through one of the following three mechanisms:

- Remote computer – this type of operation configures and manages ITS applications from a computer located at a traffic management location, such as a Transportation Management Center (TMC) or from a field located computer such as a traffic signal field master controller.
- Local computer – this type of operation performs the same functions as a central computer does, but uses a portable interface device (e.g., laptop, etc.) connected directly to a port of the ATC.
- Locally – this type of operation uses the front panel or portable interface devices (e.g., keyboard, displays, switches) at the ATC to perform the functions of configuring and managing the ITS applications.

The connection between the central computer and the ATC runs over a communications network. This can be either hard-wired (cables) or wireless. The network interface at the ATC can be either a serial communications port or Ethernet port. Figure 2-4 depicts the physical architecture of the key components related to a typical ATC-based system run from a central location.

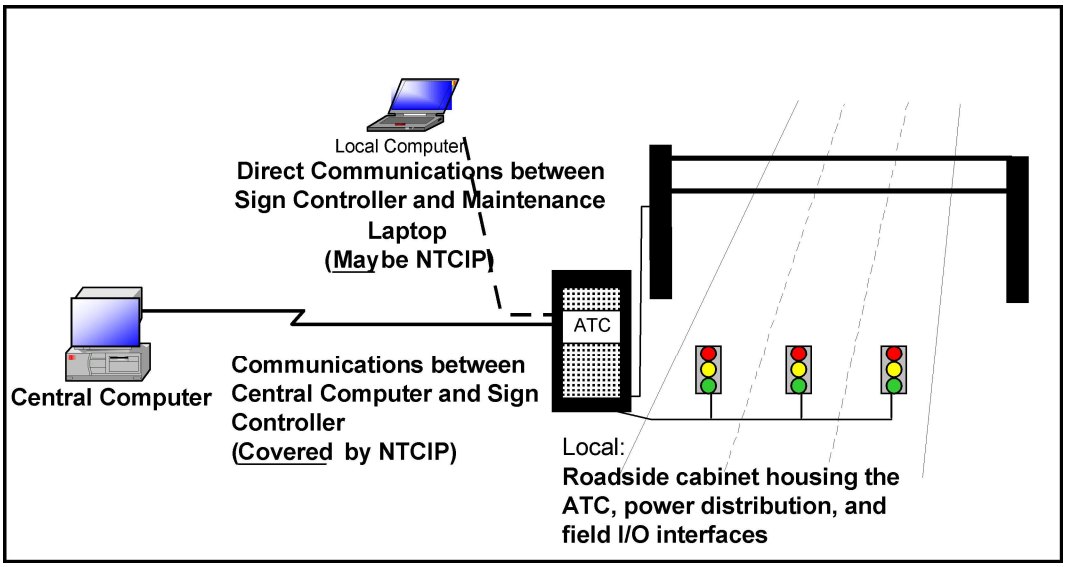


Figure 2-4. View of a Typical ATC System Environment.

The ATC is enclosed in a field-located cabinet. The ATC connects to other cabinet-located input/output devices (i.e., load switches, detector sensors, etc.) through serial and/or parallel connections. Cabinet input/output devices, in turn, connect to field-located elements (i.e., signal head, dynamic message sign, sensors, etc.).

In practice, there are additional components in a field-located cabinet which support the system including power distribution equipment, monitoring devices and terminal facilities. The exact device interfaces and cabinet configuration depends on the particular ITS application and type of equipment being deployed.

2.3 Representative Usage

As previously indicated, the functionality of a deployed ATC will depend on the applications software loaded into it. Some of the anticipated applications to be hosted on the ATC are listed in Table 2-1.

Table 2-1. Some of the anticipated ATC applications.

Traffic Signal
Traffic Surveillance
Lane Use Signals
Communications
Field Masters
Ramp Meter
Variable/Dynamic Message Signs
General ITS Beacons
CCTV Cameras
Highway Rail Intersections
Speed Monitoring
Incident Management
Highway Advisory Radio
Freeway Lane Control
High Occupancy Vehicle Systems
Access Control
Roadway Weather Information Systems
Irrigation Control

Due to its general purpose nature, an ATC may be used for future ITS applications that are not currently anticipated. These expanded functions may, over time, expand the operational user needs for an ATC. Nonetheless, a number of basic operational usage scenarios can be discerned from present day applications.

This section identifies and describes some of the most common “use cases” to be supported by the ATC and its applications software. Figure 2-5 provides a top-level view of the operational features offered by a typical ITS application using an ATC. The definition of each feature is provided after the presentation of the diagram. The features in this diagram are subdivided into more detailed features in the text below. For these use cases, a more detailed use case feature diagram is presented along with corresponding definitions. Section 3 then uses these definitions to organize and define the various functional requirements of an ATC.

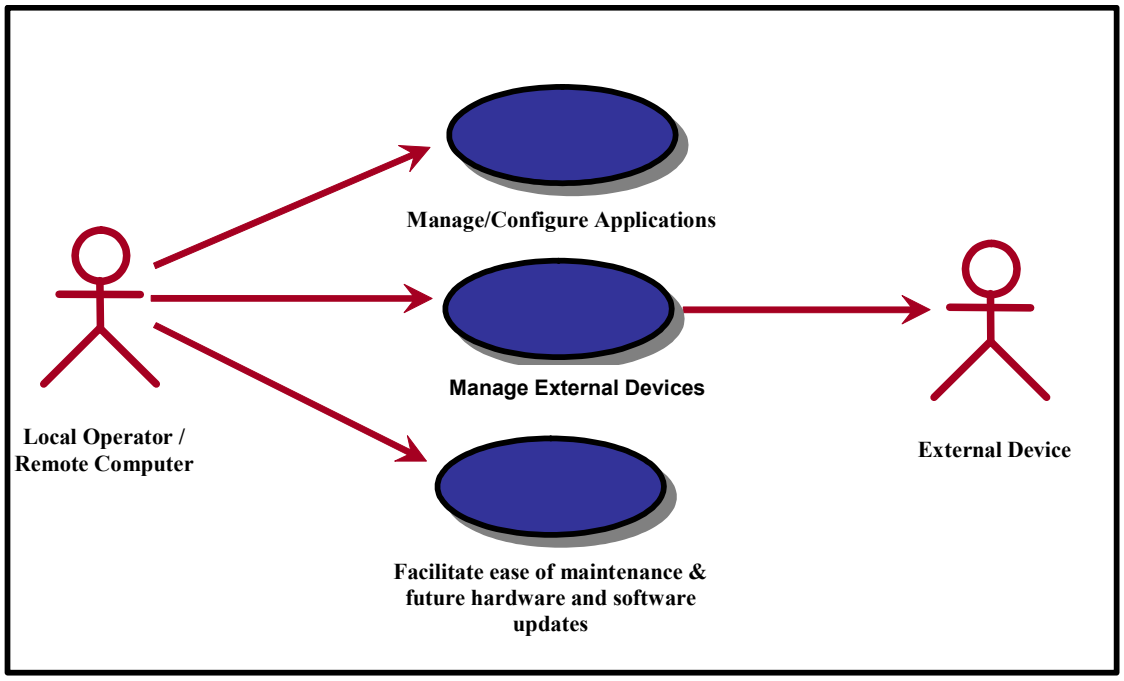


Figure 2-5. Main Maintenance / Support Diagram.

The generalized operational features of an ATC can be categorized into the following three major areas:

- Manage/Configure Applications
- Manage External Devices
- Facilitate Ease of Maintenance and Future Hardware/Software Updates

The maintenance and support function includes features for maintenance and update/enhancement of the controller unit's hardware and/or software.

2.3.1 Manage/Configure Controller Applications

The various sub-features for managing and configuring software applications are shown in the following figure. The subsequent sections detail these sub-features.

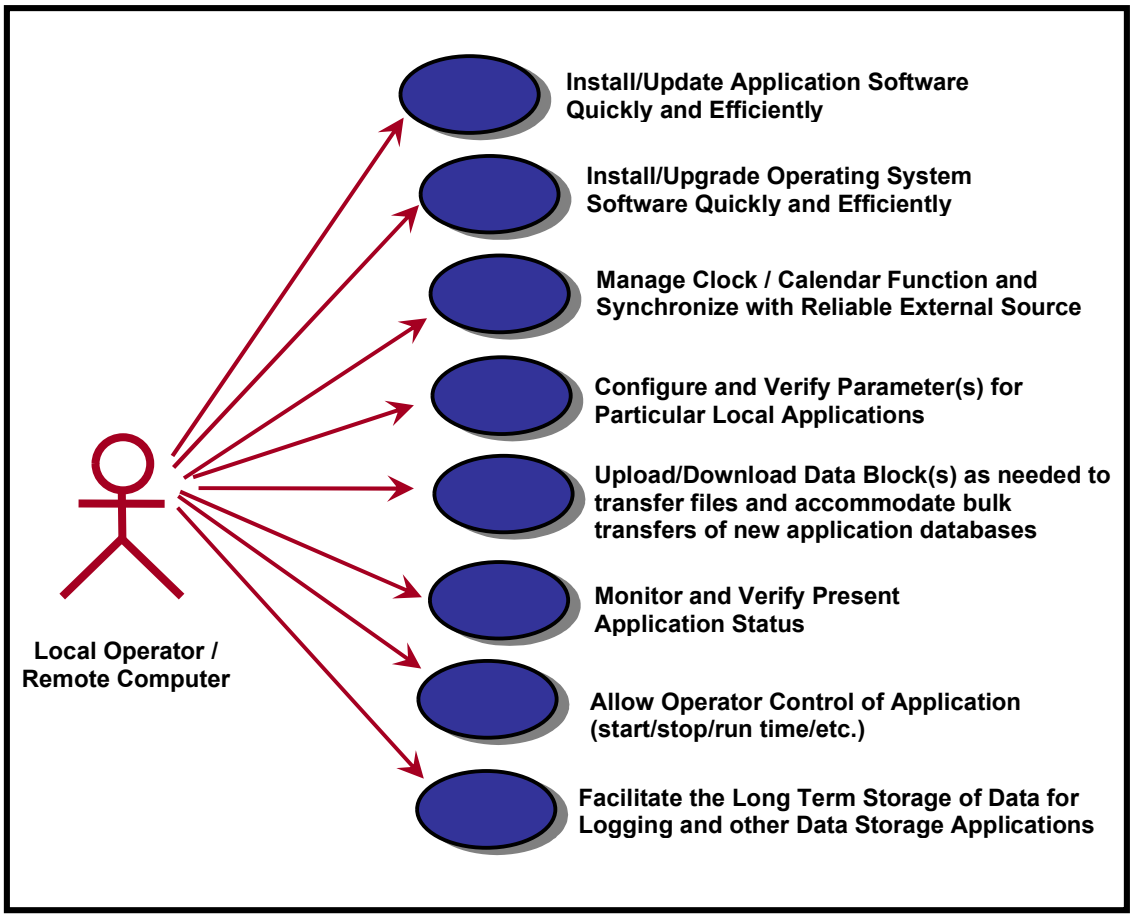


Figure 2-6. Manage/Configure Applications' Sub-feature Areas.

2.3.1.1 Install/Update Applications Software Quickly and Efficiently

This feature allows the local operator to install or update the application software resident on the ATC. With Standard Operating Policies and appropriate safety precautions in place, a User may also permit a remote computer to install or update the application software resident on the ATC.

2.3.1.2 Install/Upgrade O/S Quickly and Efficiently

This feature allows the local operator to install or update the O/S resident on the ATC. With Standard Operating Policies and appropriate safety precautions in place, a User may also allow remote upgrade capability as an optional feature.

2.3.1.3 Manage Clock / Calendar Function and Synchronize with Reliable External Source

This feature is responsible for management of a real-time clock calendar function within the ATC. It allows the operator or a remote computer to interrogate and/or update the current time and date information kept by the ATC. It is responsible for synchronizing the ATC O/S clock to an AC power source or other suitable locally available reference to adjust for internal ATC clock drift.

2.3.1.4 Configure and Verify Parameters for Particular Local Applications

This feature allows the operator or a remote computer to manage and update the currently operational applications data stored in the ATC.

2.3.1.5 Upload/Download Data Block(s) as needed to Transfer Files and Accommodate Bulk Transfers of new Application Databases

This feature allows an operator to remotely or locally download or upload complete data blocks or data files from another computer device. It supports the operator's ability to do bulk transfers of complete application databases to and from the ATC.

2.3.1.6 Monitor and Verify Present Applications Status

This feature allows an operator to remotely or locally view real-time reports of current applications status. The feature, depending on the application, would allow the operator to view status indicators such as operating modes, failure status, event logs, operation algorithm outputs, input and output states, timer countdowns, etc.

2.3.1.7 Allow Operator Control of Application Execution

This feature allows the operator to manage the starting, stopping and scheduling of one or more applications on the ATC.

2.3.1.8 Facilitate the Long Term Storage of Data for Logging and other Data Storage Applications

This feature facilitates the long-term storage of data for logging and other data storage applications.

2.3.2 Manage External Devices

The various sub-features for "managing external devices" are shown in the following figure. The subsequent sections detail these sub-features.

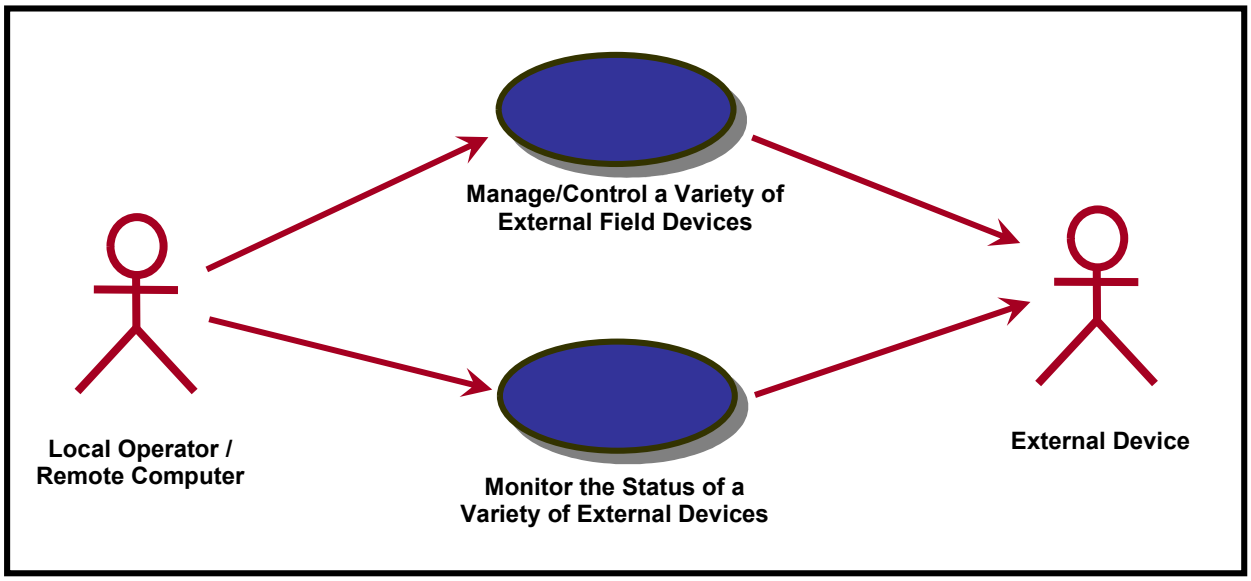


Figure 2-7. Manage External Devices' Sub-feature Areas.

2.3.2.1 Manage/Control a Variety of External Field Devices

This feature addresses the need for external devices to be controlled remotely (through a local controller using commands from a central computer), locally (from a laptop computer connected to the controller), or from an unattended controller.

2.3.2.2 Monitor the Output and Status of a Variety of External Field Devices

This feature provides the capability for the controller to monitor device output and status and to use that status for local control configuration, failure diagnosis, logging and/or reporting to a local operator or remote computer.

2.3.3 Facilitate Ease of Maintenance and Future Hardware/Software Updates

The various sub-features for facilitating ease of maintenance and future hardware and software are shown in the following figure. The subsequent sections detail these sub-features.

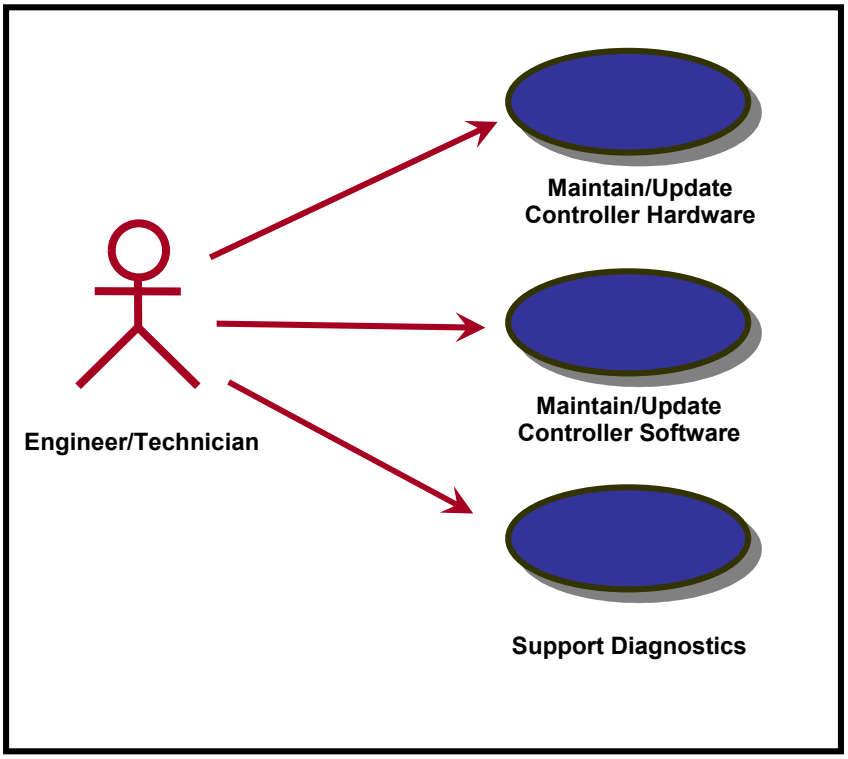


Figure 2-8. Facilitate Ease of Maintenance and Future Hardware/Software Updates Sub-feature Areas.

2.3.3.1 Maintain/Update Controller Hardware

This feature addresses the need for controller unit hardware to be maintained and updated as technology changes and additional functional and performance capabilities are needed.

2.3.3.2 Maintain/Update Controller Software

This feature addresses the need for controller applications software to easily be maintained, updated, or ported between different manufacturers' hardware units.

2.3.3.3 Support Diagnostics

This feature addresses the need for the controller to support diagnostic capabilities.

2.4 Modes of Operation

The features identified above were developed with the following three modes of operation in mind: standalone, direct, and distributed. Each of these is discussed below.

The “standalone” control mode assumes that the ATC is operating in the field without remote monitoring by a central computer or master controller. In this mode, application software is loaded into non-volatile controller memory and used to control and/or monitor externally connected devices such as gates, signals, beacons, signs, etc. Device control is based on locally-stored schedule, predefined control algorithms, or manual operation by a person present at the controller. Device monitoring might include processing of remote sensor inputs and/or monitoring the results of the controller’s control actions. Under this mode, no communications are assumed to exist between the ATC and central computer or remote master. Local operator interactions take place through the ATC front panel interface, laptop computer, or similar portable device.

The direct control mode assumes that a remote control center or master device controls the external device(s) via commands to the ATC. In this mode, commands are sent from control center/master to the ATC via communications network to affect the operation of local device(s) connected to the ATC.

The distributed control mode is a combination of the first two. Here the local ATC applications software exercises normal control, but the operation is managed and synchronized through a communication network connection with a central computer or master. Local control operations may frequently be overridden remotely to meet current needs and situations.

2.5 Security

The standard does not explicitly address security issues as cybersecurity involves technologies, processes, and practices. The cyber vulnerability and threat environment are also highly dynamic and addressing such issues is largely beyond the scope of this standard. In order to facilitate security concerning network communications, all network communication ports on the ATC shall be able to support authentication and encryption. If individual applications require it, security should be addressed either through the software hosted by the ATC or by physically protecting access to the ATC and its interfaces.

3 FUNCTIONAL REQUIREMENTS

This section defines the functional requirements to be supported by the ATC. These functions fall into three major categories:

- Manage/Configure Controller Applications
- Manage External Devices
- Facilitate Ease of Maintenance and Future Hardware/Software Updates

The ATC is fundamentally defined as a general-purpose field computing device supporting many different possible software applications. Therefore, the particular functional and sub-functional requirements applicable to any particular ATC implementation cannot be fully defined here and are left to each end-users' discretion so long as the basic functions described here are supported by the particular ATC.

3.1 Manage/Configure Controller Applications

3.1.1 Install and Update Applications Software

The ATC shall provide hardware to support the installation and update of applications software. If performed locally, this requirement shall be satisfied by the following hardware:

- Ethernet port for interfacing with a laptop computer or similar locally-connected device with software for performing this function
- Front panel portable memory device interface and a minimal front panel user interface for initiating bulk data transfers to and from a portable memory device; satisfied by the following requirements:
    - USB port with support for portable memory device and BSP-described drivers for portable memory device file access
    - Front panel display and keyboard or a serial interface for connection to a laptop computer to serve as an operator interface for initiating file transfers to and from a portable memory device when such a device is connected to USB port per above requirement

If performed remotely, this requirement shall be satisfied by the following hardware:

- Ethernet port for possible use to communicate with a remote device having the necessary software for performing this function

3.1.2 Installing and Upgrading the Operating System Software

The ATC shall provide hardware to support the installation and upgrade of drivers, utilities, etc. This requirement shall be satisfied by the same local and remote requirements given in Section 3.1.1.

3.1.3 Maintain Clock/Calendar Function and Synchronize with External Sources

The ATC shall provide hardware and firmware to support a clock/calendar function. This function shall be capable of synchronizing with external time sources as described in Section 4.3.4.

3.1.4 Configure and Verify Parameters

The ATC shall provide a means to support the configuration and verification of parameters for particular local applications.

If performed locally, this requirement shall be satisfied by at least one of the following hardware interfaces:

- Front panel display and keyboard(s) to support operator configuring/verifying of application parameter(s) and/or
- EIA-694 terminal port, Ethernet port or secure wireless interface for use with a local laptop, tablet or similar device that has appropriate software to support configuration and verification of application parameters by the operator.

If performed remotely, this requirement shall be satisfied by the following hardware:

- Serial communications port
- Ethernet port

This hardware is understood to be matched with applications support and/or BSP support functions supporting NTCIP transfers through remote system interface.

3.1.5 Uploading/Downloading Data Block(s)

The ATC shall provide hardware to support file transfers and bulk transfers of new application databases.

If performed locally, this requirement shall be satisfied by the following hardware:

- Communication port(s) for interface to a locally-connected laptop or similar device with necessary software to support operator configuration and verification of application parameter(s) from this device

If performed remotely, this requirement shall be satisfied by the following:

- Communications port (no provisions for operator data entry)
- Presence of application support for NTCIP transfers through communications port

3.1.6 Monitoring and Verifying Present Application Status

The ATC shall provide hardware to monitor system health overall as well as internal parameters related to particular application such as operating modes, event logs, device failures, algorithm results, etc.

If performed locally, this requirement shall be satisfied by the following hardware:

- Communication port(s) for interface to a locally-connected laptop, or similar device with necessary software to support operator monitoring and verification of present applications status from this device

If performed remotely, this requirement shall be satisfied by the following hardware:

- Requirements listed above
- Presence of applications support for NTCIP transfers through communications port(s)

3.1.7 Allowing Operator Control of Application(s)

The ATC shall provide hardware and software to support the operator control of start/stop/run times of all applications.

This requirement shall be satisfied by:

- Communication port(s) for interface to a locally-connected laptop, or similar device with necessary software to support operator control of applications (start/stop/run times, etc.).
- Controller-resident operator interface software to control other applications tasks (start/stop/run time/etc.).

Remote performance of this function is not supported.

3.1.8 Facilitate the Retention of Data

The ATC shall provide hardware to facilitate data logging and other local data storage applications via:

- Short-Term Non-Volatile Memory for applications to store data
- Long-Term Non-Volatile Memory (FLASH) file management system

3.1.9 Holdup Power

The ATC Power Supply and/or Engine Board shall provide holdup power to maintain the RTC and Short-Term Non-Volatile Memory during power interruptions.

- When service power is not available to the controller, the RTC and Short-Term Non-Volatile Memory shall be maintained for a minimum of 30 days at 25°C
- When standby power from the controller's power supply is not available (either disconnected or depleted), the RTC and Short-Term Non-Volatile Memory shall be maintained for a minimum of seven days at 25°C

3.2 Manage External Devices

The ATC shall support optional hardware to provide control, management, and monitoring of a variety of field devices through conventional parallel I/O as described by the NEMA TS 1 Standard, NEMA TS 2 Standard and the Type 170 and 179 controller interface specifications from the State of California Department of Transportation (Caltrans) Transportation Electrical Equipment Specifications (TEES) and the New York State Department of Transportation (NYSDOT) Management Equipment Specifications. The ATC shall also support optional serial interfaces for control, management, and monitoring of field devices via industry-standard asynchronous and synchronous serial communication connections.

In support of this requirement, this standard provides multiple serial communications ports on the Engine Board for possible interface to external field devices:

- Each port shall support asynchronous serial communication. Three ports shall support synchronous, SDLC communication.
- Each port shall support a range of baud rates as defined in Section 4.4.3
- Ports shall be configurable to the various mechanical field connections defined in Section 5.2 and the respective modulation and demodulation methods defined in Section 5.3.2 of this standard

Sections 5 and 6 of this standard also provide details of packaging and interfaces that allow this controller to be deployed in industry standard cabinet configurations including: NEMA TS 1, NEMA TS 2 Types 1 and 2, ATC, ITS and Model 332 family.

One optional synchronous serial port shall be capable of directly interfacing (select one as appropriate):

- To an ITS or NEMA TS2 Type 1 cabinet
- Via a parallel I/O module to a NEMA TS2 Type 2 or Model 332 cabinet

3.3 Facilitate Ease of Maintenance and Future Hardware/Software Updates

3.3.1 Board Support Package (BSP)

The ATC hardware described here requires an appropriate BSP (supplied by the Engine Board vendor) to support the indicated functions and to facilitate the porting of applications software between different CPU and operating systems combinations. It is implicitly understood throughout this standard that the associated BSP shall support, at a minimum, the following classes of functions:

- Serial communications
- Long-Term Non-Volatile Memory (FLASH) memory file management
- Portable memory devices, as needed
- Applications task control
- Time and date management functions
- User interface support

3.3.2 Provide a Platform that Allows for Hardware Upgrades

This ATC 5201 Standard is intended to provide a general design that readily adapts to newer processors, operating systems and increased memory size and speed. In order to maintain an upgrade path for previously-deployed Model 2070 controller units, the Engine Board form, fit and complement of serial ports of the Standard are defined such that older Model 2070 units can benefit from upgrades to technology defined by the Standard. While the ATC packaging is ultimately left open to allow manufacturers to be responsive to special needs, the Standard describes packaging and interfaces that allow the ATC to be deployed in industry-standard cabinet configurations.

3.3.2.1 Standardize Controller Packaging

The overall ATC physical design shall allow for either rack mount or shelf mount cabinet configurations.

- Controller unit may be capable of being rack-mounted in a cabinet including, but not limited to, those adhering to the ITS Cabinet Standard and the Model 332 cabinet specifications
- If used in standard NEMA TS1 or TS2 cabinet, the controller unit shall be shelf- or rack-mounted

3.3.2.2 Standardize Engine Board Contents

A key design goal of this ATC 5201 Standard is that it provides for easy hardware upgrades to adapt to newer processors as well as increased memory size and speed. It does this by requiring that all primary computational functions be concentrated on an Engine Board within the ATC. To maintain interchangeability, the Engine Board shall conform to a designated specific physical form and pin-out interface. Pins designated as “reserved” allow for future enhancements to the Engine Board and are not to be used for any purpose. They shall be unconnected on both Engine Board and Host Modules. Section 4 of the Standard designates minimum Engine Board requirements on the following:

- CPU and RAM memory
- Long-Term Non-Volatile Memory (FLASH) memory storage
- Operating system software
- Serial ports
- Ethernet interfaces
- Standardized (form, fit, and function) pin-out interface
- Real-time clock

3.3.2.3 Standardize Communication Interfaces

The ATC 5201 Standard includes communication interface slot(s) for optional plug-in internal communication interface module(s) that have a standardized interface (form, fit and function) established so that the communication boards of various manufacturers shall operate properly when installed within another manufacturer's unit.

3.3.3 Facilitate Software Application Portability

The ATC facilitates application portability by abstracting application software from the ATC hardware thereby allowing application programs to be written that can be made to operate on any ATC (regardless of manufacturer). This is accomplished through a layered software architecture and Linux open source operating system as defined in Appendix A Linux Operating System and Minimum Kernel Configuration. In previous controller architectures, source code would require considerable modification and, in some cases, to be completely rewritten to run on a different vendor's platform. The ATC facilitates portability by requiring only modest efforts on the part of the developer such as recompiling source code and linking object modules for a particular processor.

3.3.4 Facilitate Diagnostic Capabilities

The ATC facilitates diagnostics capabilities by providing standardized external physical interfaces for parallel and serial I/O and non-volatile memory to log time/date stamped messages/errors/etc. These capabilities allow both manufacturer and third party diagnostic tools to be developed.

4 ENGINE BOARD DETAILS

4.1 General Information

4.1.1 Engine Board

The Engine Board is the heart of an ATC. The CPU, all memory devices, serial interface devices and processor housekeeping circuits shall be located on the Engine Board, which shall be interchangeable between manufacturers. The plug-in form factor and standardized connectorization of the Engine Board allow it to fit onto the Host Module of any manufacturer's controller to suit any particular application.

The Engine Board is designed as a modular unit with the following features and characteristics:

- Permits uniqueness of overall ATC hardware design while providing a migration path for Model 2070 software applications
- Provides a cost-effective migration path for future capability expansion
- Provides for interchangeability and innovation between manufacturers
- Facilitates customization of an ATC for particular applications

The Engine Board dramatically simplifies future updates of the processor, operating system, memory, and other core elements of the ATC. See Figure 4-1. Note that all features of the Engine Board as shown in the figure may not be available in every ATC.

These specifications for the Engine Board require a minimum level of real-time processing capability. Section 7.1 specifies the set of tests that shall be used to determine Engine Board conformance with this standard. Manufacturers are free to add additional capabilities to their Engine Board designs so long as said functionality does not conflict with this standard in any way.

The following concepts were the fundamental basis upon which the functional and design requirements specified herein for the Engine Board have been established:

- Build on the experiences of the 2070-1A and -1B CPUs
- Encapsulate the CPU-specific elements (CPU, support hardware, and O/S) into a modular form which will provide a reliable migration path for future performance and obsolescence upgrades
- Update existing features of the CPU functionality to make better use of current technology
- Selectively add new features, which may now be available through advancements in technology, only where said features are necessary in order to meet designated functional requirements

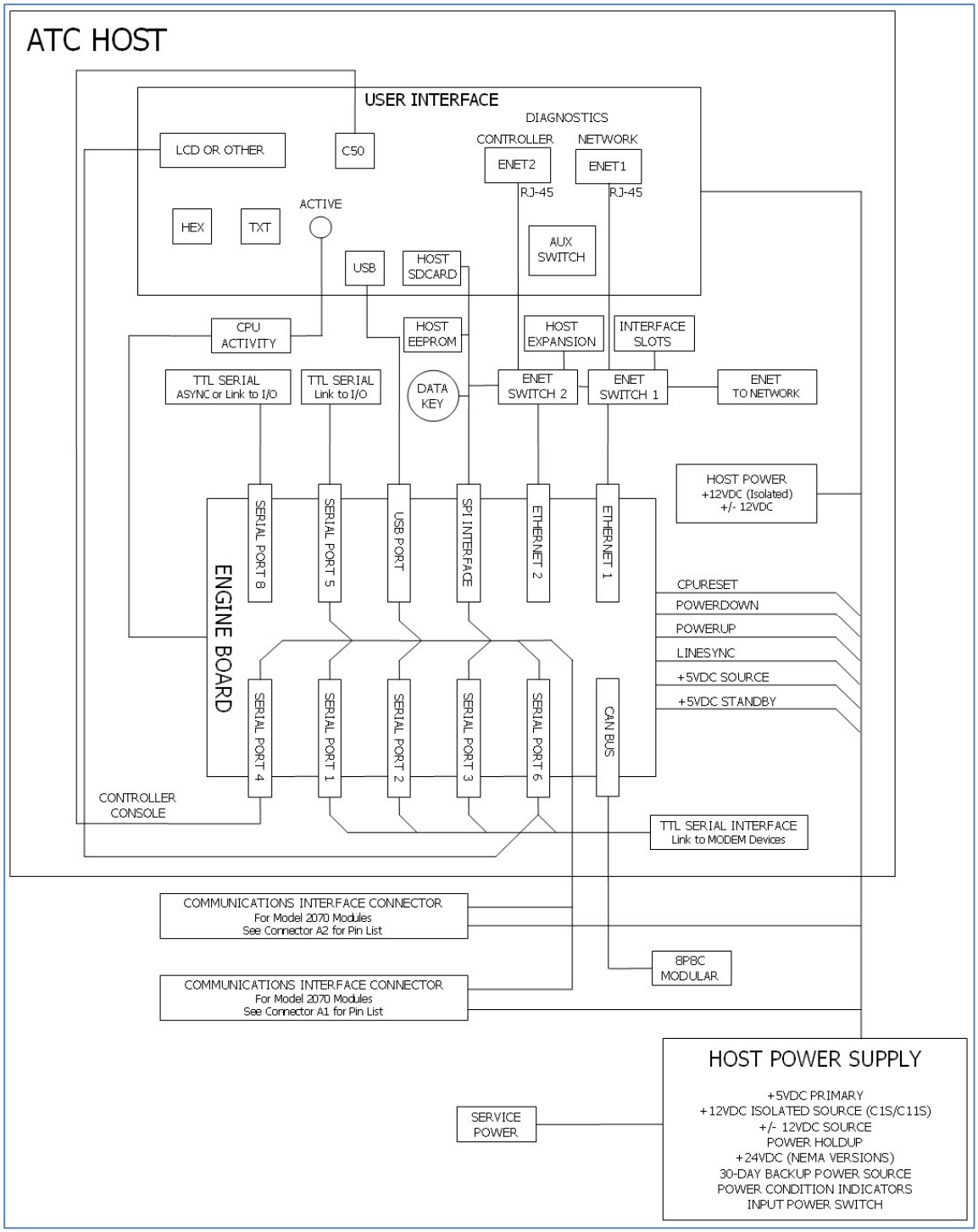


Figure 4-1. Components of the ATC and their connections.

4.1.2 Host Module

The Host Module shall provide the mechanical and electrical interface to the Engine Board and is responsible for providing sufficient power and interface paths as required by this standard. With the exception of the requirements detailed in the Standard, manufacturers are free to construct virtually any type and form of Host Module to meet any specific market need.

4.2 Mechanical and Physical

4.2.1 Board Dimensions and Mechanical Requirements

The maximum horizontal dimensions of the Engine Board shall be 5.000 in L x 3.937 in W. The nominal thickness of the PCB material shall be 0.062 in. Thicker materials may be used as required provided that the resulting Engine Board envelope remains within the overall dimensions specified in this standard.

The Engine Board shall have two interface connectors and four standoff holes, which shall be located as illustrated in Figure 4-2. Each connector shall have fifty pins, numbered 1-50, beginning with pin number 1 as the upper left-hand pin on each connector and with pin numbers increasing left-to-right and top-to-bottom. Pin 1 of each connector shall be clearly marked with the number "1" either in the top layer foil or silkscreen. Standoff holes shall be 0.125 in +0.010 in / -0 in in diameter. A 0.250 in keep-out area for circuit traces and components, concentric with each standoff hole, must be observed. 4-40 hex threaded standoffs and appropriate length 4-40 mating screws are required to be installed between the Engine Board and the Host Module. The assembled distance between the Engine Board and the Host Module shall provide a minimum of 0.100 in of clearance between the Engine Board envelope and any components on the Host Module below the Engine Board (including the actual Host Module PCB). Any additional hardware necessary to meet the environmental and test requirements of Section 7, such as lock- or split-ring washers, shall also be provided. Components may be placed on either side of the PCB. Component height, with the exception of the interface connectors, shall not exceed the overall envelope dimensions as shown in Figure 4-3. The recommended component height shall not exceed 0.742 in on the top including the Engine Board PCB material thickness, nor exceed 0.100 in on the bottom excluding the Engine Board PCB material thickness.

Manufacturers may choose alternative height Engine Board connectors to provide additional bottom-side clearance. Under no condition shall any components extend beyond the overall Engine Board envelope dimensions as shown in Figure 4-3, except for installed SD Card sockets and subject to the constraints as defined in Section 4.4.6 of the Standard. Manufacturers who choose to use alternate height Engine Board connectors shall provide the proper height F-F standoffs for host board mating with any Engine Boards sold separately from an ATC.

The Engine Board shall be labeled with its ATC 5201 Standard revision level.

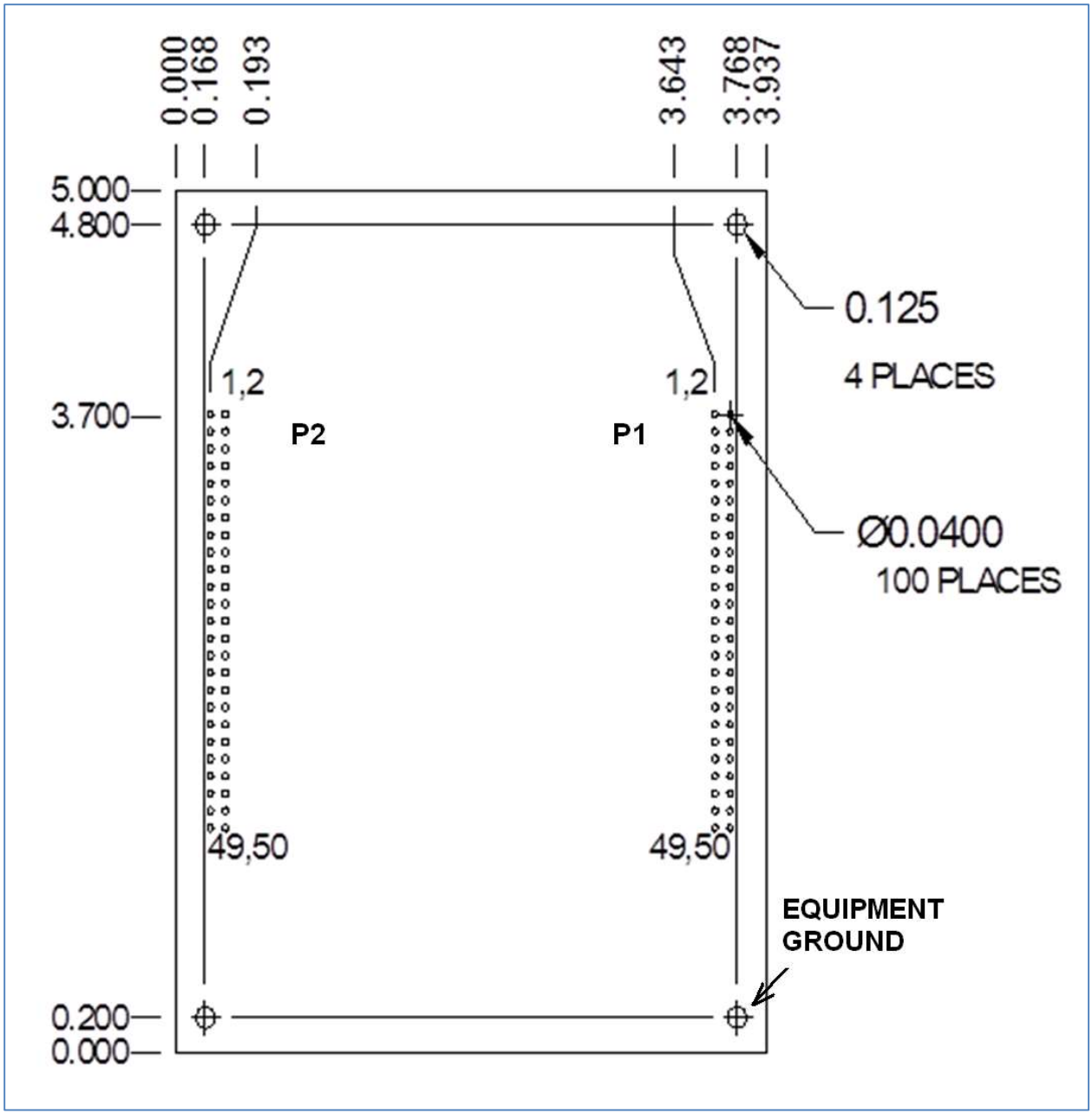


Figure 4-2. Engine Board Top View.

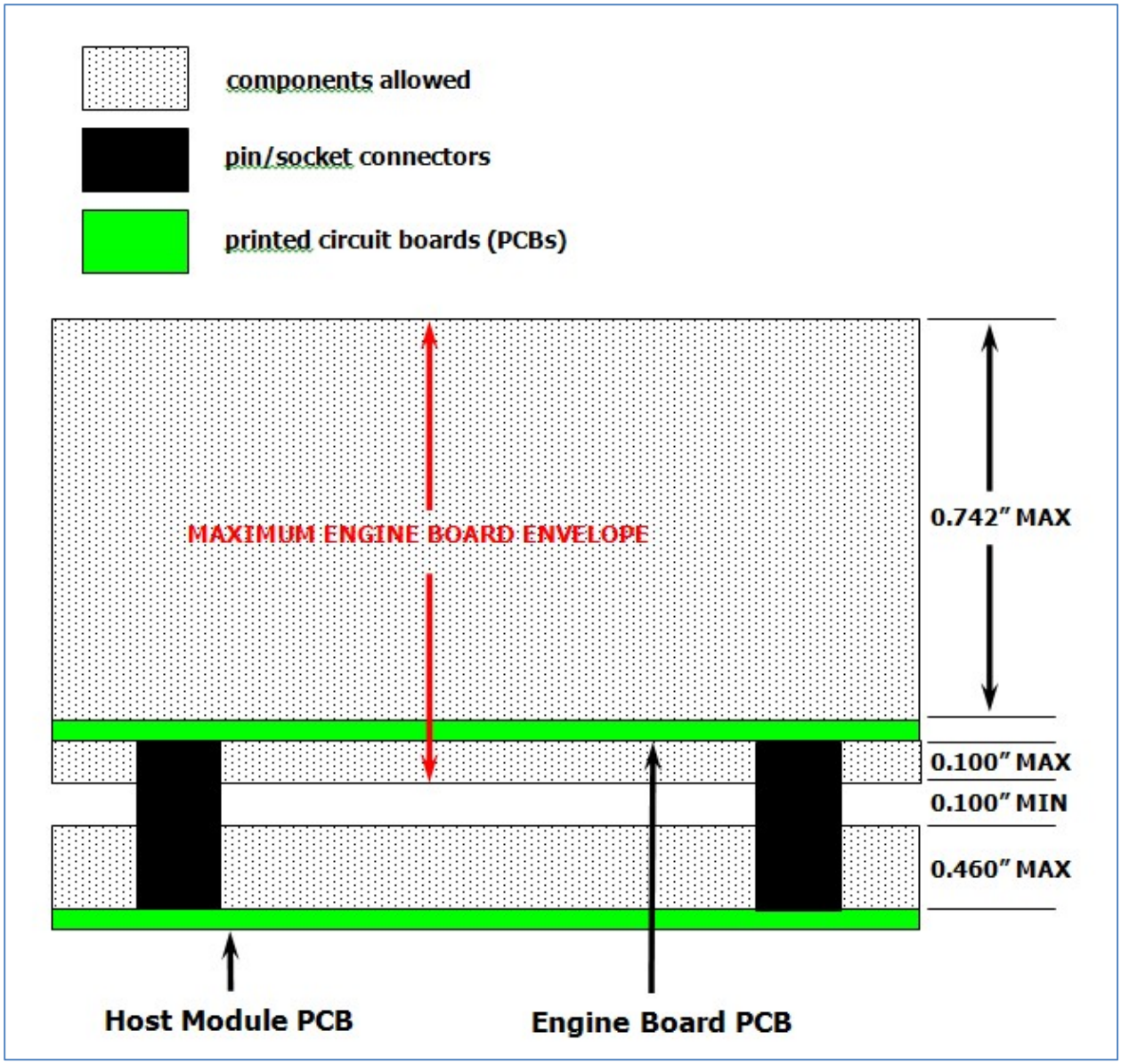


Figure 4-3. Recommended Engine Board / Host Module Stackup (not to scale).

4.2.2 Connector Pinout and Signal Names

The Engine Board shall have two connectors, designated P1 and P2, which are mounted on the bottom of the PCB. Both the Engine Board connectors and their mating host board connectors shall be dual-row, DIN 41612 pin headers with the following specifications:

Distance post-to-post, same row:	0.100 in nominal
Distance post-to-post, between rows:	0.100 in nominal
Representative connector: (or equivalent)	Hirose PCN10-50P-2.54DSA
Host module mating connector: (or equivalent)	Hirose PCN10C-50S-2.54DSA

Table 4-1 lists the connector pinouts and signal names. All name designations are from the perspective of the Engine Board (for example, TXD means data transmitted by the Engine Board).

Table 4-1. Connector P1 Pinout and Signal Names.

Connector P1		Connector P1 (cont.)	
1	VPRIMARY	26	SP4_RXD
2	VPRIMARY	27	SP6_TXD
3	VPRIMARY	28	SP6_RXD
4	VPRIMARY	29	CPU_ACTIVE
5	GROUND	30	ENET1_TX_POS
6	GROUND	31	ENET1_TX_NEG
7	GROUND	32	ENET1_RX_POS
8	GROUND	33	ENET1_RX_NEG
9	SP1_TXD	34	RESERVED
10	SP1_RXD	35	RESERVED
11	SP1_RTS	36	RESERVED
12	SP1_CTS	37	RESERVED
13	SP1_CD	38	ENGINE_PRESENT
14	*SP1_TXC_INT	39	ENET2_TX_POS
15	*SP1_TXC_EXT	40	ENET2_TX_NEG
16	*SP1_RXC_EXT	41	ENET2_RX_POS
17	SP3_TXD	42	ENET2_RX_NEG
18	SP3_RXD	43	RESERVED
19	SP3_RTS	44	RESERVED
20	SP3_CTS	45	RESERVED
21	SP3_CD	46	RESERVED
22	SP3_TXC_INT	47	RESERVED
23	*SP3_TXC_EXT/SPI_IRQ	48	*I2C_IRQ_OC
24	SP3_RXC_EXT	49	*I2C_SCL
25	SP4_TXD	50	*I2C_SDA

(*): optional signals and/or multi-purpose pin; if not used as indicated then RESERVED

Table 4-2. Connector P2 Pinout and Signal Names.

Connector P2		Connector P2 (cont.)	
1	VSTANDBY_5	26	SP8_TXC_INT
2	*CAN_GPIO	27	SP8_RXC_EXT
3	*CAN_TX	28	CPU_RESET
4	*CAN_RX	29	LINESYNC
5	SP2_TXD	30	POWERDOWN
6	SP2_RXD	31	POWERUP
7	SP2_RTS	32	SPI_MOSI
8	SP2_CTS	33	SPI_MISO
9	SP2_CD	34	SPI_CLK
10	*SP2_TXC_INT	35	SPI_SEL_1
11	*SP2_TXC_EXT	36	SPI_SEL_2
12	*SP2_RXC_EXT	37	SPI_SEL_3
13	SP5_TXD	38	SPI_SEL_4
14	SP5_RXD	39	DKEY_PRESENT
15	SP5_TXC_INT	40	PROG_TEST
16	SP5_RXC_EXT	41	PROG_TEST
17	USB_POWER_SWITCH	42	PROG_TEST
18	USB_OVERCURRENT	43	PROG_TEST
19	USB_DPLUS	44	PROG_TEST
20	USB_DMINUS	45	PROG_TEST
21	SP8_TXD	46	PROG_TEST
22	SP8_RXD	47	PROG_TEST
23	SP8_RTS	48	PROG_TEST
24	SP8_CTS	49	PROG_TEST
25	SP8_CD	50	PROG_TEST

(*): optional signals and/or multi-purpose pin; if not used as indicated then RESERVED

4.2.3 Environmental Requirements

Engine boards shall meet all environmental requirements specified in Section 7, Environmental and Test Procedures. All thermal management on the Engine Board must be by convection means only. Testing shall be performed with the Engine Board mounted to a Host Module. The Engine Board shall be mounted using the same construction and retention devices used in the manufacturer's normal production.

4.3 On-Board Resources

4.3.1 Central Processing Unit

The Engine Board shall incorporate a CPU and support circuitry that shall have a minimum computational capability to attain a score of 500 using the CoreMark 1.0 benchmark at 25°C.

Guidance: This benchmark is intended only to specify a minimum level of performance for the CPU. It is understood that this benchmark alone does not completely characterize the overall performance of the ATC in a typical application.

4.3.2 Startup Considerations

The Engine Board low-level hardware and O/S software initialization shall be completed and application software shall be capable of exercising control of all ATC unit hardware within a maximum of 5.5 seconds from the rise of both the POWERUP and POWERDOWN signals to the HIGH state (see Figure 6-1). In order that the startup time requirement may be verified, an

application program shall be provided by the manufacturer, as an independently-loaded software module, which will activate the CPU_ACTIVE signal. Section 4.3.5.1 outlines the typical startup sequence.

4.3.3 Memory

Long-Term Non-Volatile Memory

The Engine Board shall contain a minimum of 64 MB of Long-Term Non-Volatile memory (e.g. FLASH) in which a minimum of 32 MB shall be reserved for application programs. This memory shall meet the following requirements:

- Shall not require power for data retention purposes
- Use a segmented architecture allowing erasing, writing, and reading of individual segments
- Have a data retention period of no less than 10 years
- Provide a minimum of 10,000 program/erase cycles (use of ECC is permitted)
- Provide a sequential read access rate of at least 20 MB/s minimum when tested reading 4KB (aligned) blocks

All Long-Term Non-Volatile memory available for application program storage shall be formatted and mounted by the manufacturer and immediately available for use.

Application software shall be capable of reading from and writing to the Long-Term Non-Volatile Memory without the memory being corrupted by the power fail conditions specified in Section 4.4.1.

Volatile / Dynamic RAM (DRAM)

The Engine Board shall contain a minimum of 128 MB of DRAM or equivalent Volatile memory for application and O/S program execution. This memory shall support data retention under the following conditions:

- The Engine Board processor is not in RESET
- The Engine Board is not currently in a “long power outage”
- VPRIMARY is within the normal operational range as specified in Section 4.4.1

Short-Term Non-Volatile Memory

The Engine Board shall contain a minimum of 1 MB of Short-Term Non-Volatile memory (e.g., SRAM) for parameter storage. This memory shall meet the following requirements:

- Maintain data content free from corruption by short power outages or by the Engine Board processor being held in RESET
- Shall not degrade or decrease in performance over time or due to read/write activity
- Support minimum sequential read/write access rates of 9 MB/s when tested accessing 4KB (aligned) blocks with cache disabled

In the absence of VPRIMARY, the Short-Term Non-Volatile memory shall be supported and maintained by VSTANDBY_5 or by an on-board standby power source. Data shall be retained during the absence of Service Power for at least the time period specified in Section 3.1.9 or while standby power is above 2.0V.

4.3.4 Real-Time Clock (RTC)

A software-settable, hardware RTC shall be provided. The clock shall track, as a minimum, seconds, minutes, hours, day of month, month, and year. The RTC must provide one-second accuracy within 0.1 second resolution. This accuracy and resolution may be provided entirely by the RTC hardware or may be supported by BSP-described driver software as needed. In the absence of VPRIMARY the RTC shall operate from VSTANDBY_5 or by an on-board standby power source.

- RTC drift shall be less than +/- one minute per 30 days at 25˚C
- Maximum RTC drift shall be sufficient to pass the tests specified in Section 7.10.4 when configured with LINESYNC as the clock source (ATC_TIMESRC_LINESYNC) and at temperatures of 25°C, -37°C, and +74°C
- The RTC shall maintain time/date for a minimum of 30 days (with temperature between 10°C and 35°C) without service power applied to the controller
- The RTC shall maintain time/date for a minimum of seven days (at 25°C) when standby power support from the ATC's power supply is not available (either disconnected or depleted)
- When external service power is present, current time/date information shall be maintained by the operating system (OST – Operating System Time) and shall be accessible by the application software through standard Linux/POSIX functions. When the time/date is set by application software, any fractional portion of the time shall be cleared.
- Power transients and short term power outages shall not introduce clock drift
- Under normal service power conditions (as defined in Section 5.6.6.1) or during power failure conditions of less than 500 ms as indicated by the POWERDOWN signal, the Engine Board/BSP shall maintain an accurate time/date in the OST and RTC by following these rules based on the clock source configuration:
    - LINESYNC as clock source (ATC_TIMESRC_LINESYNC): All timing functions and time keeping functions (including OST) shall be referenced to the LINESYNC signal, shall drift with the power line frequency, and shall not attempt to track to a crystal or GPS (or equivalent) reference while external service power is available.

      The controller time keeping functions, including but not limited to the time-of-day clock, shall not be adversely affected by transients, brown-out conditions, dropouts, and noise on the external service power line; the controller shall include internal clock management circuitry such that short power outages (< 2.7 minutes @ 25˚C, < 45 seconds over full temperature range) do not introduce any errors or drift into the controller's clock. The controller's clock shall not deviate by more than +/- .5 AC cycles from the external service power reference; this deviation shall not accumulate. Outages longer than the short periods defined above shall not cause the clock to drift by more than +/-.0025 percent of the duration of the outage @ 25˚C, nor more than +/- 0.020 percent over the full temperature range.

      Guidance: For purposes of testing, the agency/testing lab needs to take into consideration possible deviations in the line frequency (power grid) and the accuracy of the test equipment used to measure time deviations.

Guidance: As used here, "not accumulate" means that the controller time shall be within the required deviation of +/- .5 AC cycles when referenced to the external service power line regardless of the number of short power outages that occur. During power outages, the external service power line frequency is assumed to be its nominal frequency (+/- 0.0000 Hz).

- RTC as the clock source (ATC_TIMESRC_RTCSQWR): This is an optional mode and if supported by the RTC hardware device on the Engine Board then it shall be included and function as follows. All timing functions and time keeping functions including OST shall be referenced to the RTC hardware and may include utilizing a pulsing output or alarm from the RTC device as a clock tick if desired. Note that the OST will always exactly match the RTC in this mode. If this mode is not supported by the Engine Board RTC hardware device, then the ATC shall use the Engine Board processor crystal as its clock source for OST (ATC_TIMESRC_CRYSTAL).
- Processor crystal as clock source (ATC_TIMESRC_CRYSTAL): All timing functions and time keeping functions including OST shall be referenced to the Engine Board processor crystal.

- External time references, such as NTP and GPS, may be used to set controller time but are not directly supported by the TOD driver. When an external time reference is in use, set the TOD driver time source to ATC_TIMESRC_CRYSTAL to allow standard Linux timekeeping functions to operate normally
- The BSP shall ensure that the RTC time is updated sufficiently to satisfy the requirements of this section and the tests specified in Section 7.10.4
- Upon power failure, after reapplication of power and system initialization, the BSP shall copy the RTC time values into the OST registers with accuracy sufficient to satisfy the requirements of this section and the tests specified in Section 7, in particular Section 7.10.4
- Accuracy requirements of the LINESYNC signal are stated in Section 5.6.5.2

Guidance: It is understood that the controller's RTC and OST will need to be periodically resynchronized with an external source, either via system communications or by a local WWV or GPS receiver.

4.3.5 ATC Board Support Package (BSP)

The BSP supplied by the vendor shall provide operating-system-level support for the Engine Board. See Appendix A, Linux Operating System and Minimum Kernel Configuration.

4.3.5.1 Startup Sequence

The BSP performs many steps starting from system reset to configure all low-level hardware, file systems and system level drivers for such items as SP4, SP6, timers, etc.

The bootstrap code:

- Initializes the microprocessor(s) internal registers
- Does low-level memory subsystem and peripheral initialization
- Decompresses the Linux kernel, if compressed, and moves it from FLASH EEPROM to Volatile / Dynamic RAM (DRAM)

- Creates an initial RAM disk (initrd) if needed
- Boots the kernel

The kernel then:

- Parses the kernel command line if provided
- Determines the processor MIPS rating (BogoMIPS)
- Determines available memory and many other things
- Loads compiled-in BSP drivers
- Creates the RAM disk(s) from the FLASH EEPROM image
- Frees up memory used by the initial RAM disk
- Starts processing the startup scripts

The startup scripts, which run before the prompt is displayed on the terminal, should then do the following:

- Install any additional time-critical BSP driver modules

(The steps listed above must be completed within the time period specified in Section 4.3.2.)

- Start time-critical user applications
- Start IP stack protocol modules
- Start other driver modules
- Start other applications

The file /etc/inittab is the main location to put calls to startup scripts

4.4 Electrical Interface

4.4.1 Power

Operating Voltages and Currents

Primary power shall be applied to the Engine Board between the VPRIMARY and GROUND interface pins which shall be connected to +5 VDC and DCGND1 respectively of the power supply. The Engine Board shall be capable of operation from any supply voltage ranging from +4.8VDC to +5.2VDC on VPRIMARY. The Engine Board shall not draw more than 10.0 W from VPRIMARY. Any additional voltages required for normal operation by the Engine Board shall be derived from VPRIMARY with circuitry located on the Engine Board.

In the absence of VPRIMARY, the Short-Term Non-Volatile Memory (SRAM) and RTC components shall be supported and maintained by VSTANDBY_5 provided from the Host Module (+5 VDC Standby Power from the Power Supply) and/or standby sources on the Engine Board. VSTANDBY_5 shall provide standby power to the Engine Board over the voltage range of VPRIMARY down to 2.0 VDC. VSTANDBY_5 is allowed to fall below 2.0 VDC, but in that case it will not be considered to be providing standby power. The maximum combined average current draw from VSTANDBY_5 shall be 8.0 µA over the standby voltage range of 4.5 VDC to 2.0 VDC. Alternatively, a maximum instantaneous current draw of 6.0 µA (sum of Short-Term Non-Volatile Memory (SRAM) and RTC current) measured at the SRAM and RTC devices, at a voltage of 2.5 VDC and at 25˚ C, shall be considered equivalent to the maximum average current draw

requirement stated above. Additional standby power sources physically located on the Engine Board are not subject to the average or instantaneous draw limitations.

Power Interruption and Restoration

The Engine Board must properly interpret and respond to power control signals provided by the Host Module, specifically the POWERUP and POWERDOWN signals. Specifications of these signals are in Section 5.6.5.1 and diagrams of their states under various operational conditions are shown in Figures 6-1, 6-2, and 6-3.

The Engine Board shall provide circuitry to prevent accessing the Short-Term Non-Volatile Memory (SRAM) area and to keep the processor in a RESET state any time that VPRIMARY is less than the minimum-specified operating voltage regardless of the state of the POWERUP and POWERDOWN signals.

POWERUP

POWERUP is a logic-level input signal to the Engine Board. This input signal is specified in Section 5.6.5.1 and is normally in the HIGH state following a controller cold start and during normal operation. A HIGH-to-LOW transition, while the POWERDOWN signal is also in the LOW state (Figure 6-2), indicates to the Engine Board that all software execution is to be halted and that a cold restart is to be performed once controller power has been restored (POWERUP and POWERDOWN are both HIGH) (Figure 6-1). This condition is considered a long power outage. A HIGH-to-LOW transition while the POWERDOWN signal is in the HIGH state shall be ignored.

POWERDOWN

POWERDOWN is a logic-level input signal to the Engine Board. This input signal is specified in Section 5.6.5.1 and is normally in the HIGH state following a controller cold start and during normal operation. A HIGH-to-LOW transition indicates to the Engine Board that AC power to the ATC has been lost (Figures 6-2 and 6-3). This signal serves as an advance warning of an impending power failure and can be used to trigger data storage or other pre-shutdown activities. Should the POWERDOWN signal transition back from LOW-to-HIGH while the POWERUP signal continues to be in the HIGH state (Figure 6-3), the application software shall continue operating normally without a restart. This condition is considered a short power outage.

4.4.2 Synchronization

LINESYNC

The LINESYNC signal is specified in Section 5.6.5.2 and is an input to the Engine Board and provides a 50 percent duty cycle square-wave at the incoming AC line frequency. This signal is at logic-level between VPRIMARY and GROUND and is used to provide a periodic interrupt to the CPU for use as an O/S clock reference when LINESYNC is configured as the time-of-day clock source (ATC_TIMESRC_LINESYNC).

4.4.3 Serial Interface Ports

4.4.3.1 Serial Communications Ports

The Engine Board shall provide seven serial communications ports. These ports are described below. Each port shall be capable of operating at a completely independent bit rate from all other ports. All interface pins shall operate at logic-levels. Input pins are indicated by (I), output pins by (O). All ports are not expected to operate at maximum speed simultaneously. Only two ports are required to operate concurrently using SDLC.

Ports shall meet the test requirements of Section 7.1.1 Engine Board Communication Loading Test.

Serial Port 1 (SP1)

Principal Usage:	general-purpose	
Operating Modes:	ASYNC (required) SYNC / HDLC / SDLC (all optional)	
Asynchronous Rates (bps):	1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k / 57.6k / 115.2k 230.4k (optional)	
Synchronous Rates (bps):	19.2k / 38.4k / 57.6k / 76.8k / 153.6k	
Default Rate (bps):	(none)	
Interface Pins:	SP1_TXD:	Transmit Data (O)
	SP1_RXD:	Receive Data (I)
	SP1_RTS:	Request To Send (O)
	SP1_CTS:	Clear To Send (I)
	SP1_CD:	Carrier Detect (I)
	SP1_TXC_INT:	Transmit Clock Internal (O) (optional)
	SP1_TXC_EXT:	Transmit Clock External (I) (optional)
	SP1_RXC_EXT:	Receive Clock External (I) (optional)

All synchronous protocol support on SP1 shall be optional within this standard. ATC Host Modules that do not provide synchronous protocol support on SP1 shall provide a 10K pullup resistor to +5V or HCT driver output for each SP1 clock input to the Engine Board. Engine Boards which do not include synchronous protocol support on SP1 shall provide a 10K pullup resistor to +5V or HCT driver output for each SP1 clock output from the Engine Board.

Serial Port 2 (SP2)

Principal Usage:	general-purpose	
Operating Modes:	ASYNC (required) SYNC / HDLC / SDLC (all optional)	
Asynchronous Rates (bps):	1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k / 57.6k / 115.2k 230.4k (optional)	
Synchronous Rates (bps):	19.2k / 38.4k / 57.6k / 76.8k / 153.6k	
Default Rate (bps):	(none)	
Interface Pins:	SP2_TXD:	Transmit Data (O)
	SP2_RXD:	Receive Data (I)
	SP2_RTS:	Request To Send (O)
	SP2_CTS:	Clear To Send (I)
	SP2_CD:	Carrier Detect (I)
	SP2_TXC_INT:	Transmit Clock Internal (O) (optional)
	SP2_TXC_EXT:	Transmit Clock External (I) (optional)
	SP2_RXC_EXT:	Receive Clock External (I) (optional)

All synchronous protocol support on SP2 shall be optional within this standard. ATC Host Modules that do not provide synchronous protocol support on SP2 shall provide a 10K pullup resistor to +5V or HCT driver output for each SP2 clock input to the Engine Board. Engine Boards which do not include synchronous protocol support on SP2 shall provide a 10K pullup resistor to +5V or HCT driver output for each SP2 clock output from the Engine Board.

Serial Port 3 (SP3)

Principal Usage:	In-cabinet devices	
Operating Modes:	ASYNC / SDLC (both required) SYNC / HDLC (both optional)	
Asynchronous Rates (bps):	1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k / 57.6k / 115.2k 230.4k (optional)	
Synchronous Rates (bps):	153.6k / 614.4k	
Default Rate (bps):	(none)	
Interface Pins:	SP3_TXD:	Transmit Data (O)
	SP3_RXD:	Receive Data (I)
	SP3_RTS:	Request To Send (O)
	SP3_CTS:	Clear To Send (I)
	SP3_CD:	Carrier Detect (I)
	SP3_TXC_INT:	Transmit Clock Internal (O)
	SP3_TXC_EXT:	Transmit Clock External (I) (optional)
	SP3_RXC_EXT:	Receive Clock External (I)

Guidance: Reference IBM Document GA27-3093-3 for details on SDLC operation.

Serial Port 4 (SP4)

Principal Usage:	external user-interface (console) and general purpose	
Operating Modes:	ASYNC	
Asynchronous Rates (bps):	1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k / 57.6k / 115.2k 230.4k (optional)	
Default Rate (bps):	115.2k	
Interface Pins:	SP4_TXD:	Transmit Data (O)
	SP4_RXD:	Receive Data (I)

Serial Port 5 (SP5)

Principal Usage:	In-cabinet devices	
Operating Modes:	ASYNC / SDLC (both required) SYNC / HDLC (both optional)	
Asynchronous Rates (bps):	1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k / 57.6k / 115.2k 230.4k (optional)	
Synchronous Rates (bps):	153.6k / 614.4k	
Default Rate (bps):	(none)	
Interface Pins:	SP5_TXD:	Transmit Data (O)
	SP5_RXD:	Receive Data (I)
	SP5_TXC_INT:	Transmit Clock Internal (O)
	SP5_RXC_EXT:	Receive Clock External (I)

Guidance: Reference IBM Document GA27-3093-3 for details on SDLC operation.

Serial Port 6 (SP6)

Principal Usage:	Front panel user-interface
Operating Modes:	ASYNC
Asynchronous Rates (bps):	1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k 57.6k / 115.2k 230.4k (optional)

Default Rate (bps):	115.2k	
Interface Pins:	SP6_TXD:	Transmit Data (O)
	SP6_RXD:	Receive Data (I)

Serial Port 8 (SP8)

Principal Usage:	general-purpose	
Operating Modes:	ASYNC / SDLC (both required) SYNC / HDLC (both optional)	
Asynchronous Rates (bps):	1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k / 57.6k / 115.2k 230.4k (optional)	
Synchronous Rates (bps):	153.6k / 614.4k	
Default Rate (bps):	(none)	
Interface Pins:	SP8_TXD:	Transmit Data (O)
	SP8_RXD:	Receive Data (I)
	SP8_RTS:	Request To Send (O)
	SP8_CTS:	Clear To Send (I)
	SP8_CD:	Carrier Detect (I)
	SP8_TXC_INT:	Transmit Clock Internal (O)
	SP8_RXC_EXT:	Receive Clock External (I)

4.4.3.2 Serial Peripheral Interface (SPI) Port

The Engine Board shall provide a Serial Peripheral Interface (SPI) Port. All SPI interface pins shall conform to HCT logic levels where the high-level voltage represents a logic 1 state (de-asserted) and the low-level voltage represents a logic 0 state (asserted).

Four device select lines, SPI_SEL_1 through SPI_SEL_4, shall be used to select the currently active device on the SPI bus. These lines shall only be used to select SPI devices on the Host Module. This standard defines the following two methods for selecting SPI devices:

- Basic Addressing:

  Allows selection of up to four devices by asserting individual select lines

- Expanded Addressing:

  Allows selection of up to 15 devices by asserting combinations of select lines

Support for Expanded Addressing is optional. The SPI device select lines shall be utilized as shown in the following table.

SPI Device Selection

Selected Device	Basic SPI Address	Expanded SPI Address	SPI_SEL_4 Logic State	SPI_SEL_3 Logic State	SPI_SEL_2 Logic State	SPI_SEL_1 Logic State
(none)	N/A	N/A	1	1	1	1
Datakey	1	1	1	1	1	0
Host EEPROM	2	2	1	1	0	1
(reserved)	N/A	3	1	1	0	0
SD Card	4	4	1	0	1	1
(reserved)	N/A	5	1	0	1	0
(reserved)	N/A	6	1	0	0	1
(reserved)	N/A	7	1	0	0	0

Selected Device	Basic SPI Address	Expanded SPI Address	SPI_SEL_4 Logic State	SPI_SEL_3 Logic State	SPI_SEL_2 Logic State	SPI_SEL_1 Logic State
Manufacturer-Specific	8	8	0	1	1	1
Network Switch 1	N/A	9	0	1	1	0
Network Switch 2	N/A	10	0	1	0	1
Manufacturer-Specific	N/A	11	0	1	0	0
Manufacturer-Specific	N/A	12	0	0	1	1
Manufacturer-Specific	N/A	13	0	0	1	0
Manufacturer-Specific	N/A	14	0	0	0	1
Manufacturer-Specific	N/A	15	0	0	0	0

To support Engine Board interchangeability, if the Host Module includes any manufacturer-specific SPI devices, then the manufacturer shall provide documentation to allow third parties to write device drivers which accesses those devices. To support application software portability, if an Engine Board includes any device drivers for manufacturer specific SPI devices which will be accessed directly by application software, then the manufacturer shall provide documentation to allow third parties to write application programs which utilize those device drivers.

For backwards compatibility, drivers that use more than one concurrent SPI_SEL select line to select an expanded address shall query the “Expanded SPI Supported” field in the Host EEPROM to determine if expanded SPI address selection is supported by the host. If the Host EEPROM version is less than 2 or if no Host EEPROM is present, the driver shall assume that expanded SPI address selection is not supported.

Expanded SPI address 1 is allocated to support optional Datakey operations.

Expanded SPI address 2 is allocated to support an optional Host Module serial EEPROM device containing controller configuration information. The content and organization of the information is described in Appendix B.3.1.

Expanded SPI address 4 is allocated to support an optional SD Card device located on the Host Module. If this device is included, it shall be supported for both expanded and non-expanded Host Modules.

Expanded SPI addresses 3, 5, 6 and 7 are currently unimplemented and are allocated for future SPI-related expansion in this standard.

Expanded SPI addresses 9 and 10 are allocated to support possible manufacturer-specific management and configuration operations on the two Host Module Network Switches, if applicable. If a single Network Switch is provided (with second switch functionality provided by VLAN support) it shall utilize Expanded SPI address 9 for management operations if provided.

Expanded SPI addresses 8 and 11-15 are allocated for general manufacturer-specific use.

Principal Usage:		Datakey / serial EEPROM interface
Operating Modes:		SYNC
Synchronous Rates (bps):		(application-specific)
Interface Pins:	SPI_MOSI:	Master-Out-Slave-In (O)

SPI_MISO:	Master-In-Slave-Out (I)
SPI_CLK:	Clock (O)
SPI_SEL_1:	Encoded Select 1 (O)
SPI_SEL_2:	Encoded Select 2 (O)
SPI_SEL_3:	Encoded Select 3 (O)
SPI_SEL_4:	Encoded Select 4 (O)
SPI_IRQ	SPI Interrupt (I) (optional)

SPI_IRQ (optional)

SPI_IRQ is an optional, active-low, logic-level input signal that indicates that a Host Board SPI device is requesting an interrupt to the Engine Board. All Host Board SPI devices shall share the same SPI_IRQ signal which shall remain logic-level low while any connected Host Board SPI device is requesting interrupt service. Host Board SPI devices shall use open-drain IRQ signals configured as a "wired-OR". The SPI_IRQ signal shall utilize Engine Board *SP3_TXC_EXT/SPI_IRQ interface pin as defined in Table 4-1 (dedicated pins on Connector P1).

Support for this signal shall be indicated in the Host EEPROM.

4.4.3.3 Universal Serial Bus (USB) Port

The Engine Board shall provide a USB port. This port shall facilitate the transfer of large data files to and from the controller through the use of USB-based memory devices and is intended to provide a simple alternative to a laptop computer.

The following minimum requirements for this port have been established:

- The USB port shall conform to the appropriate sections of the USB v2.0 specification for both hardware and software operation in order to support bulk transfer operations. A minimum of Full-Speed operation shall be supported.

Specific operational requirements for file transfers via the USB port shall be described by the BSP.

Interface Pins:	USB_DPLUS:	Data Line Positive (I/O)
	USB_DMINUS:	Data Line Negative (I/O)
	USB_POWER_SWITCH	Power Switch (O)
	USB_OVERCURRENT	Over-current (I)

USB_POWER_SWITCH

USB_POWER_SWITCH is an active-low, logic level output signal generated by the Engine Board. This signal shall be provided to indicate that the Host Module shall apply power to the USB connector.

USB_OVERCURRENT

USB_OVERCURRENT is an active-low, logic level input signal to the Engine Board. This signal shall be provided to indicate that the attached USB device is requesting more current than allowed and that power to the USB device should be removed.

4.4.3.4 Ethernet Ports

The Engine Board shall provide two 10BASE-T Ethernet ports which fully conform to the applicable requirements of IEEE 802.3-2002. Each port must have a unique 48-bit MAC address. All components necessary to produce the Ethernet physical layer (PHY) for each port, including the magnetic interface module, shall be located on the Engine Board.

The MAC addresses of the Ethernet ports shall be available to a user without opening/disassembling the ATC unit. This may be accomplished either by the use of external labeling or via the ATC user interface.

Guidance: Independent physical or logical switches for each Ethernet port on the Host Module will provide auto-switching capability in support of both 10BASE-T and 100BASE-T external to the controller.

Ethernet Interface (ENET)

Principal Usage:	Local and network communications	
Operating Mode:	Synchronous, Manchester-encoded, differential	
Synchronous Rates (bps):	10M	
Interface Pins:	ENET1_TX_POS:	Port 1 Transmit Data Positive (O)
	ENET1_TX_NEG:	Port 1 Transmit Data Negative (O)
	ENET1_RX_POS:	Port 1 Receive Data Positive (I)
	ENET1_RX_NEG:	Port 1 Receive Data Negative (I)
	ENET2_TX_POS:	Port 2 Transmit Data Positive (O)
	ENET2_TX_NEG:	Port 2 Transmit Data Negative (O)
	ENET2_RX_POS:	Port 2 Receive Data Positive (I)
	ENET2_RX_NEG:	Port 2 Receive Data Negative (I)

4.4.3.5 CAN Bus

CAN Bus is an optional serial interface feature that may be provided by the Engine Board.

Guidance: If CAN Bus interface is provided, then Host EEPROM must be installed and contain proper CAN configuration information as defined in Section B.3.2.

If CAN Bus hardware/software support is provided, it shall meet the following criteria:

- Utilize the Engine Board CAN* interface pins as defined in Table 4-1 (dedicated pins on Connector P2)
- Support CANBus v2.0B
- Provide the Linux SocketCAN drivers and a CANopen protocol library

Support a bus speed of 1Mbps

4.4.3.6 I2C Bus

I2C Bus is an optional serial interface feature that may be provided by the Engine Board.

If I2C Bus hardware/software support is provided, it shall meet the following criteria:

- Utilize the Engine Board *I2C interface pins as defined in Table 4-1 (dedicated pins on Connector P1).
- Support a bus speed of 400kbps with a Host Board load of up to 380 pF
- Provide a 2.4kohm pullup resistor to VPRIMARY on each of the SCL and SDA lines

Include an I2C Interrupt input signal (I2C_IRQ_OC) from the Host Board to the Engine Board so that Host Board devices can request service from the Engine Board. This signal shall be an open-collector / open-drain signal on the Host Board so that dissimilar VDDIO voltages can be accommodated. The pull-up resistor shall be provided on the Engine Board.

4.4.4 Programming/Test Port

Interface pins are available on the Engine Board for a manufacturer-specific programming and test port. Pins for this purpose are designated PROG_TEST. This optional port (or ports) may be used for programming and testing of any on-board device(s). Examples of this test port (or ports)

include JTAG, BDM, Boundary-Scan, custom CPLD programming, and proprietary In-Circuit FLASH programming. Manufacturers are free to designate these pins for these purposes in any configuration on special Engine Board test adapter hosts, however all mating PROG_TEST pins on production ATC Host Modules shall be no-connects.

Manufacturers are also free to place programming and test connectors directly on the Engine Board, subject to the component placement height restrictions in Section 4.2.1.

4.4.5 Miscellaneous

CPU_RESET

CPU_RESET is an active-low, logic-level output signal generated by the Engine Board. This signal shall be provided to reset other system devices and shall be accessible to application programs as described in the BSP.

CPU_ACTIVE

CPU_ACTIVE is an active-low, logic-level output signal generated by the Engine Board. This signal shall be provided to indicate an active CPU and shall be accessible to application programs as described in the BSP.

A typical use for this signal is to drive a front-panel “active” or “health” LED.

DKEY_PRESENT

DKEY_PRESENT is an active-low, logic-level input signal to the Engine Board. When this signal is active, it indicates the physical presence of a key in the Datakey receptacle as described in the BSP.

ENGINE_PRESENT

ENGINE_PRESENT is an active-low, logic-level output signal from the Engine Board. This signal indicates the physical presence of an Engine Board to the host board. This signal shall not be used to carry power supply current and may be used for manufacturer test purposes only.

EQUIPMENT GROUND

The Engine Board mounting hole nearest P1-50 shall be used to pass Equipment Ground (EG) from the Host Board to the Engine Board.

RESERVED

All pins marked as RESERVED are reserved for future enhancements to the Engine Board and are not to be used for any purpose. They shall be no-connects on both the Engine Board and Host Module.

4.4.6 SD Card

An optional SD Card socket may be provided on the Engine Board. The socket type may be either Standard SD or Micro SD. Any suitable electrical interface may be used. Physical access to this socket external to the controller is not required but is permitted.

The SD Card socket may be mounted on either the top or bottom side of the Engine Board. If the socket and installed SD Card are not completely contained within the Engine Board envelope, the installed card may only extend from a short edge of the Engine Board and by no more than 0.1”. If the socket is mounted on the bottom side of the Engine Board, it may descend into the Engine Board/Host Module clearance area by no more than 0.045” (maximum installed bottom-side height of 0.145”).

5 COMMUNICATION INTERFACE DETAILS

5.1 General Description

The Communications Interface performs the signal conditioning needed to adapt the ATC serial I/O to various transmission media, such as phone lines, radio, and optical fiber.

This Communications Interface Standard Section includes the following:

- Transmission Media
- Modulation and Demodulation
- Mechanical Form Factor

This Communications Interface Standard Section does not include the following:

- Bit Rate Generation
- Data Content
- Error Detection and Indication

This Communications Interface Standard allows the design and manufacture of hundreds of different varieties of communications modules which are interchangeable among vendors. To meet this standard, a Communications Interface shall conform to the following:

- Mechanical dimensions and ATC connector of this standard
- Faceplate connectors of this standard
- Modulation methods of this standard

See Figures 5-1 and 5-2.

Section 5.2.3 provides descriptions for standardized legacy connections. An ATC may contain non-standardized connections provided that they are thoroughly documented by the ATC manufacturer as to their pinout and electrical requirements.

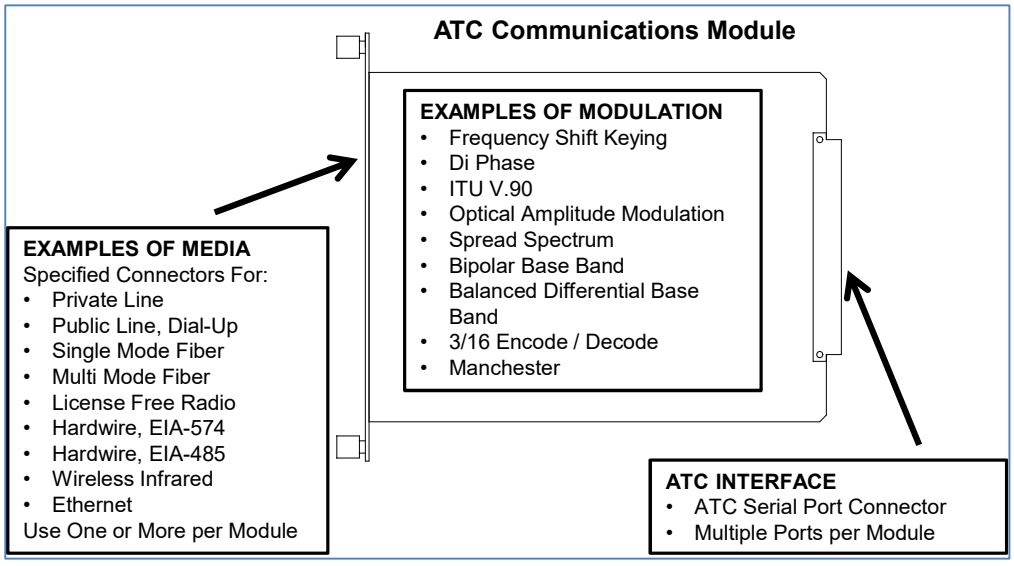


Figure 5-1. ATC Communications Interface Block Diagram & Option Choices.

5.1.1 Interchangeability Control

The ATC may provide zero, one or two communications interface slots.

Installing communications interface modules in the ATC is optional, not required.

When used, all communications interface modules shall conform to this standard, including dimensions and pin assignments.

Communications circuitry may be embedded inside the ATC, provided that the field connectors and pin assignments conform to this standard.

5.1.2 Communication Interface Slot Identification (deprecated)

5.1.3 Serial Port Identification

Each Communications Interface Module may use one or more ATC Serial Ports. The Communication Interface Module manufacturer shall document the relationship between each Module Port and the applicable ATC Serial Port for each communication interface slot. If the front panel connector can be assigned to different serial ports, the front panel legend shall indicate such. For example, if a 9-pin EIA-694 front panel connector can be assigned to Port A or Port B via a program switch, the program switch shall also illuminate either a Port A or Port B LED.

Each embedded ATC Serial Port shall be labeled by either its industry standard function (e.g., Port 1 or C14S), ATC function (e.g., Diagnostics), or Engine Board function (e.g., SP1).

5.2 Mechanical Description

5.2.1 Mechanical Outline Dimensions

The ATC Communications Interface uses the Model 2070 communications slot mechanical form factor and pin configurations. The mechanical dimensions are as follows:

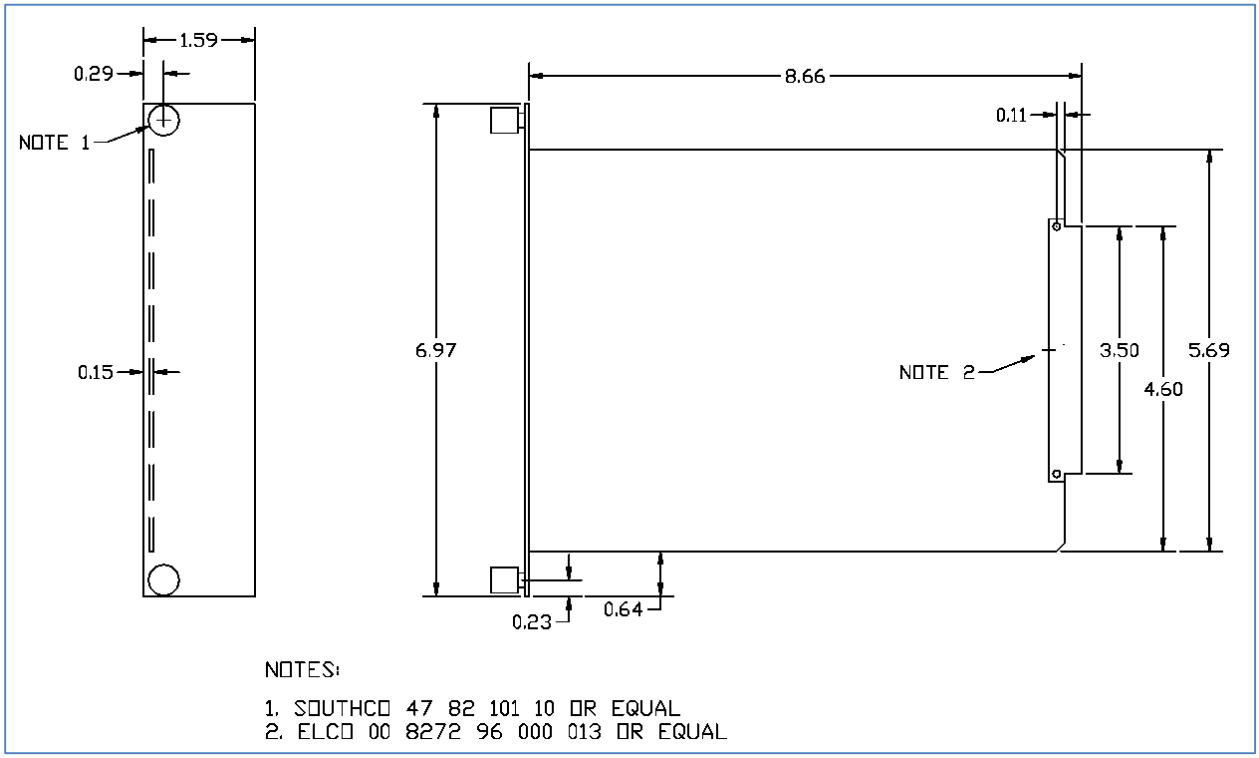


Figure 5-2. Mechanical Dimensions.

5.2.2 ATC Communications Connector Mechanical Pin Assignments

Communications Slot A2 (minimum requirements):

PIN	ROW A	ROW B	ROW C
1	SP1TXD+	(reserved)	(reserved)
2	SP1TXD-	(reserved)	(reserved)
3	SP1RXD+	(reserved)	(reserved)
4	SP1RXD-	(reserved)	(reserved)
5	SP1RTS+	(reserved)	(reserved)
6	SP1RTS-	(reserved)	(reserved)
7	SP1CTS+	(reserved)	(reserved)
8	SP1CTS-	(reserved)	(reserved)
9	SP1DCD+	(reserved)	(reserved)
10	SP1DCD-	(reserved)	(reserved)
11	SP2TXD+	(reserved)	(reserved)
12	SP2TXD-	(reserved)	(reserved)
13	SP2RXD+	(reserved)	(reserved)
14	SP2RXD-	(reserved)	(reserved)
15	SP2RTS+	(reserved)	(reserved)
16	SP2RTS-	(reserved)	(reserved)
17	SP2CTS+	(reserved)	(reserved)
18	SP2CTS-	(reserved)	(reserved)
19	SP2DCD+	(reserved)	(reserved)
20	SP2DCD-	(reserved)	(reserved)
21	DCGND1	(reserved)	(reserved)
22	Ethernet Sw1.P4 Tx+	(reserved)	(reserved)
23	Ethernet Sw1.P4 Tx-	INSTALLED	(reserved)
24	(reserved)	LINESYNC	(reserved)
25	Ethernet Sw1.P4 Rx+	POWERUP	CPU_RESET
26	Ethernet Sw1.P4 Rx-	POWERDOWN	(reserved)
27	DCGND1	DCGND1	DCGND1
28	+12 VDC	-12 VDC	+5 VDC Standby (optional)
29	+5 VDC	+5 VDC	+5 VDC
30	DCGND1	DCGND1	DCGND1
31	(reserved)	(reserved)	(reserved)
32	(reserved)	(reserved)	(reserved)

Communications Slot A1 (minimum requirements):

PIN	ROW A	ROW B	ROW C
1	SP3TXD+	(reserved)	(reserved)
2	SP3TXD-	(reserved)	(reserved)
3	SP3RXD+	(reserved)	(reserved)
4	SP3RXD-	(reserved)	(reserved)
5	SP3RTS+	(reserved)	(reserved)
6	SP3RTS-	(reserved)	(reserved)
7	SP3CTS+	(reserved)	(reserved)
8	SP3CTS-	(reserved)	(reserved)
9	SP3DCD+	(reserved)	(reserved)
10	SP3DCD-	(reserved)	(reserved)
11	SP4TXD+	(reserved)	(reserved)
12	SP4TXD-	(reserved)	(reserved)
13	SP4RXD+	(reserved)	(reserved)
14	SP4RXD-	(reserved)	(reserved)
15	(reserved)	(reserved)	(reserved)
16	(reserved)	(reserved)	(reserved)
17	(reserved)	(reserved)	(reserved)
18	(reserved)	(reserved)	(reserved)
19	(reserved)	(reserved)	(reserved)
20	(reserved)	(reserved)	(reserved)
21	DCGND1	C50 ENABLE	(reserved)
22	Ethernet Sw1.P8 Tx+	(reserved)	(reserved)
23	Ethernet Sw1.P8 Tx-	INSTALLED	(reserved)
24	(reserved)	LINESYNC	(reserved)
25	Ethernet Sw1.P8 Rx+	POWERUP	CPU_RESET
26	Ethernet Sw1.P8 Rx-	POWERDOWN	(reserved)
27	DCGND1	DCGND1	DCGND1
28	+12 VDC	-12 VDC	+5 VDC Standby (optional)
29	+5 VDC	+5 VDC	+5 VDC
30	DCGND1	DCGND1	DCGND1
31	(reserved)	(reserved)	(reserved)
32	(reserved)	(reserved)	(reserved)

Notes:
Signal directions are referenced to the Engine Board, not the communications interface. For example, SP1_TXD (SP1TXD+ and SP1TXD-) is Serial Port 1 data transmitted from the Engine Board to the communications interface. SP1RXD (SP1RXD+ and SP1RXD-) is Serial Port 1 data received by the Engine Board from the communications interface.

5.2.3 Standardized Legacy Field Connections

Guidance:

References to EIA-232 in this Standard have been replaced with references to EIA-694, an electrical-only standard whose signals are interoperable with EIA-232 but support data rates up to 512kbps. As EIA-694 is an electrical-only standard, the specific connectorization requirements for each application are also necessary to provide a complete specification for the connection.

Where standardized legacy field connections are provided by an ATC, either as integral to the controller assembly or via a plug-in module in one of the communications slots, these connections

shall be labeled appropriately and shall meet all applicable requirements of their respective standards as defined below.

An ATC may contain non-standardized field connections. These connections shall be thoroughly documented by the ATC manufacturer as to their pinout and electrical requirements.

5.2.3.1 NEMA Field Connections

The following field connections shall meet all applicable requirements as defined herein and in NEMA TS 2-2016, Traffic Controller Assemblies with NTCIP Requirements, v03.07.

In the case of a conflict between the content within this section and the aforementioned document, this section shall govern.

Port 1 (TS2 Type 1 Port 1, 15-Pin, SDLC)

Port 1 provides an EIA-485 field connection for synchronous operation.

The Port 1 connector shall be a 15-pin metal shell D sub-miniature type connector. The connector shall utilize female contacts and shall be equipped with latching blocks. Pin connections shall be as follows:

Pin	Function
1	Tx Data +
2	DC Ground 2
3	Tx Clock +
4	DC Ground 2
5	Rx Data +
6	DC Ground 2
7	Rx Clock +
8	DC Ground 2
9	Tx Data -
10	Port 1 Disable (0VDC=disable) (*)
11	Tx Clock -
12	Earth Ground
13	Rx Data -
14	Reserved
15	Rx Clock –

*Pin 10 shall be connected to Pin 8 when it is desirable to disable all Port 1 communications activity.

Port 2 (TS2 Type 1 Port 2, 25-Pin, Asynchronous)

Port 2 provides an EIA-694 asynchronous interface for connection to a printer or personal computer.

The Port 2 connector shall be a 25-pin metal shell D sub-miniature type connector. The connector shall utilize female contacts and shall be equipped with latching blocks. Pin connections shall be as follows:

Pin	Function	I/O
1	Earth Ground	[-]
2	Transmitted Data	[O]
3	Received Data	[I]
4	Request To Send	[O]
5	Clear To Send	[I]

6	Not Used	[-]
7	Logic Ground	[-]
8	Received Line Signal Detector	[I]
9-19	Not Used	[-]
20	Reserved	[-]
21-25	Not Used	[-]

Port 3 (TS2 Type 1 Port 3, 9-Pin, Type 3002 FSK Modem)

Port 3 provides a four-wire full-duplex communications interface over an unconditioned Type 3002 voice grade channel utilizing FSK modulation.

The Port 3 connector shall be a nine-pin metal shell D sub-miniature type connector. The connector shall utilize male contacts and shall be equipped with latching blocks. Pin connections shall be as follows:

Pin	Function	I/O
1	Transmit 1	[O]
2	Transmit 2	[O]
3	Reserved	[-]
4	Receive 1	[I]
5	Receive 2	[I]
6	EG	[-]
7	Reserved	[-]
8	Reserved	[-]
9	EG	[-]

5.2.3.2 2070 Field Connections

The following field connections shall meet all applicable requirements as defined herein and in Caltrans Transportation Electrical Equipment Specifications (TEES), dated March 12, 2009, plus all applicable errata.

In the case of a conflict between the content within this section and the aforementioned document, this section shall govern.

C50S

C50S provides a reduced pin count EIA-694 asynchronous interface for connection to a printer or personal computer.

The C50S connector shall be a nine-pin metal shell D sub-miniature type connector. The connector shall utilize female contacts and shall be equipped with jack screws. Pin connections shall be as follows:

Pin	Function	
1	C50 Enable	[I]
2	SP4 Receive Data	[I]
3	SP4 Transmit Data	[O]
4	Reserved	[-]
5	Isolated Interface Ground	[-]
6	Reserved	[-]
7	Reserved	[-]
8	Reserved	[-]
9	Reserved	[-]

C50J

C50J provides an EIA-694-compatible asynchronous interface for connection to a printer or personal computer.

The C50J connector shall be an eight-pin modular RJ45-type connector. Pin connections shall be as follows:

Pin	Function	
1	Reserved	[-]
2	SP4 Receive Data	[I]
3	C50 Enable	[I]
4	Reserved	[-]
5	SP4 Transmit Data	[O]
6	Isolated Interface Ground	[-]
7	Reserved	[-]
8	Reserved	[-]

C60P

C60P provides an EIA-694 asynchronous interface for connection to a remote front panel interface device.

The C60P connector shall be a nine-pin metal shell D sub-miniature type connector. The connector shall utilize male contacts and shall be equipped with jack screws. Pin connections shall be as follows:

Pin	Function	
1	* Power	[I]
2	SP6 Receive Data	[I]
3	SP6 Transmit Data	[O]
4	Reserved	[-]
5	Isolated Interface Ground	[-]
6	Reserved	[-]
7	CPURESET	[O]
8	Reserved	[-]
9	CPULED	[O]

*Pin 1 shall provide +5VDC, 350mA maximum draw, referenced to Isolated Interface Ground. All signals on C60P shall be referenced to Isolated Interface Ground.

C12S

C12S provides an EIA-485 field connection for synchronous operation.

The C12S connector shall be a 25-pin metal shell D sub-miniature type connector. The connector shall utilize female contacts and shall be equipped with jack screws. Pin connections shall be as follows:

Pin	Function
1	SP5 Tx Data +
2	SP5 Rx Data +
3	SP5 Tx Clock +
4	SP5 Rx Clock +
5	SP3 Tx Data +
6	SP3 Rx Data +
7	SP3 Tx Clock +
8	SP3 Rx Clock +

9	LINESYNC +
10	NRESET +
11	POWERDOWN +
12	* Power
13	Isolated Interface Ground
14	SP5 Tx Data -
15	SP5 Rx Data -
16	SP5 Tx Clock -
17	SP5 Rx Clock -
18	SP3 Tx Data -
19	SP3 Rx Data -
20	SP3 Tx Clock -
21	SP3 Rx Clock -
22	LINESYNC -
23	NRESET -
24	POWERDOWN -
25	Equipment Ground

*Pin 12 shall provide up to 50mA at +5VDC referenced to Isolated Interface Ground. All signals on C12S shall be referenced to Isolated Interface Ground.

C13S

C13S provides an EIA-485 field connection for asynchronous/synchronous operation.

The C13S connector shall be a 25-pin metal shell D sub-miniature type connector. The connector shall utilize female contacts and shall be equipped with jack screws. Pin connections shall be as follows:

Pin	Function
1	SP8 Tx Data +
2	SP8 Rx Data +
3	SP8 Tx Clock +
4	SP8 Rx Clock +
5	SP8 Request To Send +
6	SP8 Clear To Send +
7	SP8 Carrier Detect +
8	Reserved
9	LINESYNC +
10	NRESET +
11	POWERDOWN +
12	* Power
13	Isolated Interface Ground
14	SP8 Tx Data -
15	SP8 Rx Data -
16	SP8 Tx Clock -
17	SP8 Rx Clock -
18	SP8 Request To Send -
19	SP8 Clear To Send -
20	SP8 Carrier Detect -
21	Reserved
22	LINESYNC -
23	NRESET -

24	POWERDOWN -
25	Equipment Ground

*Pin 12 shall provide up to 50mA at +5VDC referenced to Isolated Interface Ground. All signals on C13S shall be referenced to Isolated Interface Ground.

C21S and C22S (DE-9S versions)

C21S/22S (DE-9S versions) provide an EIA-694 asynchronous interface for connection to an external communications device or personal computer.

The C21S/22S (DE-9S versions) connector shall be a nine-pin metal shell D sub-miniature type connector. The connector shall utilize female contacts and shall be equipped with jack screws. Pin connections shall be as follows:

Pin	Function	
1	Carrier Detect (DCD)	[I]
2	Receive Data (RXD)	[I]
3	Transmit Data (TXD)	[O]
4	Reserved	[-]
5	* Isolated Interface Ground	[-]
6	Reserved	[-]
7	Request To Send (RTS)	[O]
8	Clear To Send (CTS)	[I]
9	Reserved	[-]

*Pin 5 shall provide an interface ground which is isolated from all internal ground references. All signals on C21S/22S shall be referenced to this isolated ground.

C21S and C22S (DA-15S versions)

C21S/22S (DA-15S versions) provides an EIA-485 field connection for synchronous operation.

The C21S/22S (DA-15S versions) connector shall be a 15-pin metal shell D sub-miniature type connector. The connector shall utilize female contacts and shall be equipped with jack screws. Pin connections shall be as follows:

Pin	Function
1	Tx Data +
2	* Isolated Interface Ground
3	Tx Clock +
4	* Isolated Interface Ground
5	Rx Data +
6	* Isolated Interface Ground
7	Rx Clock +
8	Reserved
9	Tx Data -
10	* Isolated Interface Ground
11	Tx Clock -
12	* Isolated Interface Ground
13	Rx Data -
14	* Isolated Interface Ground
15	Rx Clock –

*Pins 2, 4, 6, 10, 12 and 14 shall provide an interface ground which is isolated from all internal ground references. All signals on C21S/22S shall be referenced to this isolated ground.

C2S and C20S

C2S/20S provide EIA-694 and FSK field connections for asynchronous operation.

The C2S/20S connector shall be a 14-pin AMP M14 type connector. The connector shall utilize female contacts and shall be equipped with spring latch supports. Pin connections shall be as follows:

Pin	Function
A	Audio In 1
B	Audio In 2
C	Audio Out 1
D	* Power
E	Audio Out 2
F	Reserved
H	Carrier Detect (DCD) (EIA-232)
J	Request To Send (RTS) (EIA-232)
K	Receive Data (RXD) (EIA-232)
L	Transmit Data (TXD) (EIA-232)
M	Clear To Send (CTS) (EIA-232)
N	** Isolated Interface Ground
P	Reserved
R	Reserved

*Pin D shall provide up to 100mA at +5VDC referenced to interface ground (Pin N).

**Pin N shall provide an interface ground which is isolated from all internal ground references. All EIA-232 signals on C21S/22S shall be referenced to this isolated ground.

5.2.3.3 Internal Dial-Up Line Modem Connections

Dial-up phone line twisted pair field connections to the communications interface shall be via an RJ-11 connector mounted to the communications interface front panel. The pin assignments are as follows:

Pin	Signal	Description	Direction
1	NC		
2	NC		
3	Ring		
4	Tip		
5	NC		
6	NC		

5.2.3.4 Single Mode Fiber Connections

Single-mode fiber field connections to the communications interface shall be via 1300 nm threaded FC or 1300 nm ST connector for both transmitters and receivers.

5.2.3.5 Multi-Mode Fiber Connections

Multi-mode fiber connections to the communications interface shall be via 820 nm ST connectors for both transmitters and receivers.

5.2.3.6 Wide Area Radio Connections

Wide area radio field connections to the antenna shall be via a TNC coaxial.

5.2.3.7 Ethernet Connections

Ethernet connections to the communications interface shall be via an RJ-45 modular jack, with the following pin configuration. Note that Switch 1 Port 3 must have auto-MDIX capabilities.

Pin	Signal	Description	Direction
1	TXD+	Transmitter Pair +	Out
2	TXD-	Transmitter Pair -	Out
3	RXD+	Receiver Pair +	In
4	NA		
5	NA		
6	RXD-	Receiver Pair -	In
7	NA		
8	NA		

5.2.3.8 CAN Bus Connection

Connection to the optional CAN Bus shall be via an 8p8c modular jack, with the following pin configuration:

Pin	Signal	Description
1	CAN_H	Data +
2	CAN_L	Data -
3	CAN_GND	Ground
4	<reserved>	
5	<reserved>	
6	CAN_SHLD	Shield (optional)
7	CAN_GND	Ground
8	CAN_V+	Power (optional)

5.3 Operational Description

5.3.1 Interface to ATC

5.3.1.1 EIA-485 Signals

Except for ENET*, CPU_RESET, LINESYNC, POWERUP, POWERDOWN and CPU_ACTIVE, all signal lines of the 96-pin ATC connector shall be electrically EIA-485, balanced differential. The electrical specifications and signal definition shall conform to the requirements of EIA-485.

The following EIA-485 signals are biased by the ATC (not the communications interface module):

- A 100 Ω resistor connected from DATA+ to DATA- on each simplex receiver
- No termination resistor on each simplex transmitter
- 100 Ω resistor connected from DATA+ to DATA- on each half duplex transceiver
- A 680 Ω resistor from DATA+ to +5 VDC and a 680 Ω resistor from DATA- to DCGND1 to ensure a stable state when the communications interface module is not installed

The following EIA-485 signals are biased by the communications interface module (and not the ATC):

- A 100 Ω resistor connected from DATA+ to DATA- on each simplex receiver

5.3.1.2 Ethernet Signals

ENET* are 10 Base-T Ethernet signals TX+, TX-, RX+, RX- respectively on the 96-pin ATC connector. Proper selection of circuit board trace width, spacing and shielding shall be observed for correct characteristic impedance and to prevent cross talk to adjacent signals.

5.3.1.3 Power Signals

DCGND1 shall be the common reference for +5 VDC, +12 VDC, -12 VDC and all signals.

DCGND2 shall be the common reference for +12 VDC ISO.

5.3.1.4 Electrical Isolation

DCGND2 and +12 VDC ISO as a group shall be electrically isolated from all other signals and power sources as a group, maintaining the isolation specifications of Section 8.1.4 “Electrical Isolation”. EG shall maintain the isolation specifications of Section 8.1.4 Electrical Isolation.

Communications interface field connections shall be electrically isolated from all ATC signals, power sources, and EG.

Field connections of the EIA-694 and EIA-485 versions of the Communications Interface shall be optically or magnetically isolated using devices capable of at least 1 Mbps.

Field connections of the Ethernet, private line modem and dial-up modem versions of the communications interface shall be magnetically isolated via isolation transformers with the proper characteristic impedance.

Field connections of the single mode fiber, multi-mode fiber and infrared versions of the communications interface are inherently isolated via the non-conductive optical media.

Field connections of the wide area radio version of the communications interface are inherently isolated via the non-conductive radio frequency media.

5.3.1.5 Hot Swap

Communications modules shall be hot-swappable without damage to circuitry or operations. Power-on and hot-swap current surges shall not exceed a 10 ms surge at three times (3x) the maximum rating of each voltage supply used by the module.

5.3.1.6 INSTALLED Signals

INSTALLED signals (at connector position B23 in each communications slot) are active-low signals, driven by the installed module, indicating the installation of a module into the corresponding slot. The ATC may utilize these signals to disable any integrated serial ports which may be in conflict with active ports on the installed module.

Should neither of the INSTALLED signals be active, the ATC may (optionally) do the following:

- Disable all serial port drivers to the communications slots
- Disable ±12VDC (referenced to DCGND1) to the communications slots
- If ±12VDC (referenced to DCGND1) is ONLY used by the communications slots and not by any integrated serial ports within the ATC, those supplies may be completely extinguished

5.3.2 Modulation and Demodulation

5.3.2.1 EIA-694

Description:

The EIA-694 versions of the communications interface shall convert the ATC EIA-485 signals to EIA-694 bipolar simplex, meaning each signal is unidirectional and point-to-point.

If the ATC front panel contains a C50S/C50J interface with an active (LOW) C50_ENABLED signal such that one or more of the EIA-694 interfaces could be in contention, the C50_ENABLED signal may be used to disable the associated transmitter on the communications interface.

Indicators:

EIA-694 versions of the communications interface shall include the following indicators:

Front Panel Legend	Indicator Function
TX	ON= Transmitted Data at Field Wire is Positive V, per EIA-694
RX	ON= Received Data at Field Wire is Positive V, per EIA-694

Specifications:

The electrical specifications and signal definition shall conform to the requirements of EIA-694.

5.3.2.2 EIA-485

Description:

EIA-485 versions of the communications interface shall convert the ATC EIA-485 signals to isolated EIA-485, which is full-duplex.

Indicators:

EIA-485 versions of the communications interface shall include the following indicators:

Front Panel Legend	Indicator Function
TX	ON=DATA+ at Field Wire is 0V, DATA- at Field Wire is Positive V
RX	ON=DATA+ at Field Wire is 0V, DATA- at Field Wire is Positive V

Specifications:

The electrical specifications and signal definition shall conform to the requirements of EIA-485.

5.3.2.3 Private Line Modem

Description:

Private line modem versions of the communications interface shall convert the ATC EIA-485 signals to modulated audio suitable for communications on an unconditioned private phone line pair, meaning the line is direct wire not connected to a phone company.

Modulation schemes used here convert the binary “1” and binary “0” bits of the data stream into audio tones, known as the MARK and SPACE. The demodulation scheme consists of converting each of the tones back to binary “1” and binary “0” bits to replicate the original transmitted data stream at the receiving device.

When RTS is asserted by the ATC, the modem shall transmit the MARK tone for a period of time, allowing the receiving modem to lock on to the tone and assert Carrier Detect (DCD). At the end

of this time period, the transmitting modem asserts CTS, signaling the ATC to begin sending data. At the end of the data packet, the ATC unasserts RTS and the transmitting modem stops sending a tone. DCD is unasserted by the receiving modem.

A method shall be provided on the front panel to select half or full duplex for a channel, in addition to a front panel method to disable/enable a channel.

This scheme shall be capable of operating half-duplex on a single phone line, or full duplex on different phone lines, one line for transmission and another line for reception, allowing simultaneous data transmission in both directions, and to disable the modem transmitter, in the event an ATC malfunctions with its RTS constantly asserted.

A front panel method shall break the power supply current to all channels, allowing the communications interface to be inserted into the ATC without causing a reboot or other ATC malfunction other than a normal recoverable communications error.

A switch, mounted internally, shall implement anti-streaming which shall disable the modem transmitter in the event an ATC malfunctions with its RTS constantly asserted. If RTS is asserted for the specified time, the modem transmitter shall be turned OFF. The anti-streaming timer is reset if RTS is unasserted, or if TXD is active. This switch allows anti-streaming to be disabled for situations where the transmitter is expected to continuously transmit under normal operations.

Indicators:

Front Panel Legend	Indicator Function
TX	ON= SPACE Tone at Field Wire
RX	ON= SPACE Tone at Field Wire
CD	ON= Received Tone Within Specified Sensitivity and Filter Band

Modulation Methods:

Two different modulation methods shall be allowed under this standard as follows:

Frequency Shift Keying (FSK), 300 to 1200 bps, 0 to 9600 bps, 0 to 19,200 bps.

Guidance: This paragraph is intended to represent 2070-6A, -6B, & -6/19.2 modulation methods.

Three different FSK versions shall be available, which are 300 to 1,200 bps, 0 to 9,600 bps, and 0 to 19,200 bps. The three versions differ in the MARK and SPACE tones. The 0 to 9,600 bps and 0 to 19,200 bps versions handle a wider variety of bit rates, but the higher frequency tones travel shorter distances. For example, all versions transmit at the same power level and receive at the same sensitivity, but wire attenuates the higher frequencies of the 0 to 9600 bps, and 0 to 19,200 bps versions more rapidly. (Please refer to wire manufacturer's specifications for decibels (dB) loss per mile.)

Specifications:

The 300 to 1200 bps shall have the following specifications:

MARK Tone	1.2 kHz, ± 1% tolerance
SPACE Tone	2.2 kHz, ± 1% tolerance
Soft Carrier Freq	900 Hz
Modulation:	Phase Coherent Frequency Shift Keying (FSK), Bell Standard 202
Data Format:	Asynchronous, serial by bit
Line:	Type 3002 voice-grade, unconditioned
Transmit Level:	0, -2, -4, -6, and –8 dB at 1.7 kHz, continuous or switch selectable

Receiver Sensitivity:	0 to –40 dB
Receiver Filter:	20 dB/Octave minimum active attenuation for all frequencies outside the operating band (half power, -3 dB) between 1.0 kHz and 2.4 kHz
RTS to CTS Delay:	11 ms ± 3 ms
Carrier Detect:	8 ms ± 2 ms MARK frequency
Receiver Squelch:	6.5 ms ± 1 ms, 0 ms (OUT)
Soft Carrier OFF:	10 ms ± 2 ms
Recovery Time:	22 ms maximum from Transmit to Receive
Error Rate:	Less than 1 bit in 100,000 bits
Signal to Noise:	16 dBm measured with flat-weight over a 300 to 3000 Hz band
Transmit Noise:	-50 dBm maximum into 600 Ω resistive load within frequency spectrum of 300 to 3000 Hz at maximum output
Anti-Stream Time:	6 to 8 seconds

The 0 to 9600 bps shall have the following specifications:

MARK Tone	11.2 kHz, ± 1% tolerance
SPACE Tone	17.6 kHz, ± 1% tolerance
Soft Carrier Freq	7.8 kHz
Modulation:	Phase Coherent Frequency Shift Keying (FSK)
Data Format:	Asynchronous, serial by bit.
Line:	Type 3002 voice-grade, unconditioned.
Transmit Level:	0, -2, -4, -6, and –8 dB at 14.7 kHz, continuous or switch selectable
Receiver Sensitivity:	0 to –40 dB
Receiver Filter:	20 dB/Octave minimum active attenuation for all frequencies outside the operating band (half power, -3 dB) between 9.9 kHz and 18.9 kHz
RTS to CTS Delay:	11 ms ± 3 ms
Carrier Detect:	8 ms ± 2 ms MARK frequency
Receiver Squelch:	6.5 ms ± 1 ms, 0 ms (OUT)
Soft Carrier OFF:	10 ms ± 2 ms
Recovery Time:	22 ms maximum from Transmit to Receive
Error Rate:	Less than 1 bit in 100,000 bits
Signal to Noise:	16 dBm measured with flat-weight over a 300 to Controller Hz band
Transmit Noise:	-50 dBm maximum into 600 Ω resistive load within frequency spectrum of 300 to Controller Hz band at maximum output
Anti-Stream Time:	6 to 8 seconds

The 0 to 19,200 bps shall have the following specifications:

MARK Tone	19.2 kHz, ± 1% tolerance, BW = 19.2 kHz
SPACE Tone	38.4 kHz, ± 1% tolerance, BW = 19.2 kHz
Soft Carrier Freq	13.2 kHz, BW = 7.6 kHz
Modulation:	Phase Coherent Frequency Shift Keying (FSK)
Data Format:	Asynchronous, serial by bit
Line:	Type 3002 voice-grade, unconditioned.
Transmit Level:	0 to –8 dB at Alternate MARK/SPACE (i.e., toggle Oscillator between 19.2 kHz and 38.4 kHz), continuously adjustable
Sensitivity:	0 to –40 dB

Receiver Filter:	20 dB/Octave minimum active attenuation for all frequencies outside the operating band (half power, -3 dB) between 9.6 kHz and 48.0 kHz
RTS to CTS Delay:	11 ms ± 3 ms
Carrier Detect:	8 ms ± 2 ms MARK frequency
Receiver Squelch:	6.5 ms ± 1 ms, 0 ms (OUT)
Soft Carrier OFF:	5 or 10 ms, ± 2 ms (switch selectable)
Recovery Time:	22 ms maximum from Transmit to Receive
Error Rate:	Less than 1 bit in 100,000 bits
Signal to Noise:	16 dBm measured with flat-weight over a 300 to Controller Hz band
Transmit Noise:	-50 dBm maximum into 600 Ω resistive load within frequency spectrum of 300 to Controller Hz band at maximum output
Anti-Stream Time:	6 to 8 seconds

Di-Phase, 2,400 to 19,200 bps

Guidance: This paragraph describes a modulation/demodulation technique to replace legacy 1200 bps FSK modems on existing unconditioned phone lines. Equivalent transmission distances are achieved at 19,200 bps, without software changes. ITU "V" series modems, such as V.90 are not recommended for this application due to the excessive RTS to CTS training time in half-duplex polling applications such as NTCIP.

Di-phase modulation provides two tones as well as two phases, allowing increased bit rates over FSK modulation.

Specifications:

Modulation:	Differential Di-Phase, EUROCOM Standard D1
Data Format:	Asynchronous, serial by bit.
Line:	Type 3002 voice-grade, unconditioned.
Transmit Level:	0 to –8 dB at 1.7 kHz, continuously adjustable
Sensitivity:	0 to –40 dB
Receiver Filter:	20 dB/Octave min. active attenuation outside operating band
RTS to CTS Delay:	8 to 14 ms
Carrier Detect:	6 to 10 ms at MARK frequency
Receiver Squelch:	5.5 to 7.5 ms
Soft Carrier OFF:	NA (no soft carrier)
Recovery Time:	22 ms maximum from Transmit to Receive
Error Rate:	Less than 1 bit in 100,000 bits
Signal to Noise:	16 dB over 300 to 3000 Hz band
Transmit Noise:	-50 dB maximum into 600 Ω, 300 to 3000 Hz band
Anti-Stream Time:	6 to 8 seconds

5.3.2.4 Dial Up Line Modem

Guidance: This paragraph is intended to represent a standard dial-up modem.

Description:

The dial up modem versions of the communications interface shall convert the ATC EIA-485 signals to audio tones attached to public phone lines and switching equipment. The dial up modem shall be capable of data transmission and reception, as well as dialing out and dialing in on a standard analog phone line.

Indicators:

Dial up versions of the communications interface shall include the following indicators:

Front Panel Legend	Indicator Function
TX	ON= Transmitted Data Activity
RX	ON= Received Data Activity
CD	ON= Received Tone Present
RS	ON= RTS Asserted

Specifications:

The electrical specifications and signal definition shall conform to the requirements of ITU V.90. Front panel connector shall be 9-pin “D”.

5.3.2.5 Single Mode Fiber

Description:

The single mode fiber versions of the communications interface shall convert the ATC EIA-485 transmitted data to laser light, and laser light to ATC EIA-485 received data. Modulation method shall be specified by the modem manufacturer.

Danger: Be aware that single-mode laser light is invisible to the human eye but is of sufficient power to cause damage. Never look directly into a laser transmitter. Always cover unused laser transmitters with opaque dust covers.

Indicators:

Single mode fiber versions of the communications interface shall include the following indicators:

Front Panel Legend	Indicator Function
TX	ON= Transmitter MARK state
RX	ON= Receiver MARK state

Specifications:

Optical	1300 nm Single Mode Laser
Transmit Level:	-6 to -15 decibels, Continuously Adjustable
Receiver Sensitivity	-30 decibels
Data Rate	100K bps minimum
Transmitter Compensation	Temperature and aging

5.3.2.6 Multi-Mode Fiber

Description:

The multi-mode fiber versions of the communications interface shall convert the ATC EIA-485 transmitted data to light, and light to ATC EIA-485 received data. Modulation method shall be specified by the modem manufacturer.

Indicators:

Multi-mode fiber versions of the communications interface shall include the following indicators:

Front Panel Legend	Indicator Function
TX	ON= Transmitter MARK state
RX	ON= Receiver MARK state

Specifications:

Optical	820 nm Multi Mode Light Emitting Diode (LED)
Transmit Level:	-6 to -15 dBm, Continuously Adjustable
Receiver Sensitivity	-30 dBm
Data Rate	100K bps minimum
Transmitter Compensation	Uncompensated

5.3.2.7 Wide Area Radio

Guidance: This paragraph is intended to represent a license-free data radio offering a good combination of distance and data integrity.

Description:

The wide area radio version of the communications interface shall convert the ATC EIA-485 transmitted data to RF, and RF to ATC EIA-485 received data. Spread spectrum is employed, meaning that the radio transmits at high power on a range of frequency channels. This ensures that the average power transmitted on any one frequency is below the limit to require an FCC license.

Indicators:

Wide area radio versions of the communications interface shall include the following indicators:

Front Panel Legend	Indicator Function
TX	ON= Transmitted Data Activity
RX	ON= Received Data Activity

Specifications:

Shall be specified by radio manufacturer.

5.3.2.8 Infrared

Guidance: This paragraph is intended to represent an interface to a standard PDA. Agencies with single door cabinets shall ensure, prior to manufacture, that this line of sight device be positioned so that the user does not have to remove the ATC from the cabinet to utilize the infrared feature.

Description:

The infrared versions of the communications interface shall convert the ATC EIA-485 transmitted data to light and light to ATC EIA-485 received data. The light beam is infrared, meaning it is outside the visible color range detected by the human eye. The light transmission is similar to a standard television remote control, meaning that its light emission power is safe to the human eye. As with a TV remote control, the transmitting device must be used within the line of sight, aimed towards the controller red window, and located within approximately six feet of the ATC.

Indicators:

None

Specifications:

Optical:	Shall conform to Infrared Data Association Physical Layer
Modulation	3/16 Encode / Decode
Data Rate	1200 bps to 115.2K bps

5.3.2.9 Ethernet

Description:

The Ethernet version of the communications interface shall adapt the ATC ENET* signals. The Ethernet port may be directly tied to ENET* or buffered as a hub or switch.

Indicators:

Due to the higher event speeds of Ethernet, each indication shall be extended 100 ms. Ethernet versions of the communications interface shall include the following indicators:

Front Panel Legend	Indicator Function
T	ON= Transmitted Data or Received Data is logic “1”
100	ON= 100 MBPS Data Rate

Specifications:

The electrical specifications and signal definition shall conform to the requirements of IEEE 802.3.

5.4 Communications Interface Versions

Each version of the communications interface shall consist of the following:

- A printed circuit board assembly of the size and shape described in Section 5.2.1
- A connection to the ATC serial ports and power, as described in Section 5.2.2
- One or more communications ports, as described in Section 5.2.3
- Modulation / demodulation circuitry for each port, as described in Section 5.3.2

By using different combinations of ports, an unlimited number of communications interface versions may be configured, compliant to this standard.

Please refer to the Model 2070 Controller Standard for detailed specifications of the 2070-6A, 2070-6B, 2070-7A and 2070-7B.

USER INTERFACE, POWER SUPPLY AND MECHANICAL DETAILS

5.5 User Interface General Description

The user interface is the device used by an operator to operate the ATC. The user interface of a controller has traditionally consisted of a keyboard and display, and more recently laptop computers. For example, the user interface of a NEMA controller is normally a keyboard and display, with NEMA Port 2 allocated to a personal computer. The Model 2070 provides its user interface via a keyboard and display mounted in its front panel assembly or a serial port connector for a personal computer.

It is the intent of this specification to accomplish the following:

- Preserve compatibility with existing Model 2070 user interface software
- Create a standard for future advanced user interfaces, such as graphics

- Adhere to the Engine Board vendor-supplied BSP for software compatibility

It is NOT the intent of this specification to do the following:

- Preserve user interface interchangeability among vendors
- Dictate user interface requirements, other than minimum requirements and the specifications for optional legacy interfaces
- Limit the choices of user interfaces

5.5.1 Minimum User Interface

The minimum user interface shall consist of the following components in the designated locations:

Table 5-1. Minimum User Interface Components

Minimum Components	Rack	Shelf
ACTIVE LED for CPU	Front	Front
Ethernet Port 2 • "100" LED (illuminated linked at 100 Mbps, extinguished at all other times) • "ACTIVE" LED (displays Tx and Rx activity) • RJ-45 connector (Network Connection)	Front or Back	Front, Back or Side
Ethernet Port 3 • "100" LED (illuminated linked at 100 Mbps, extinguished at all other times) • "ACTIVE" LED (displays Tx and Rx activity) • RJ-45 connector (Network Diagnostics)	Front or Back	Front, Back or Side
Ethernet Port 6 • "100" LED (illuminated linked at 100 Mbps, extinguished at all other times) • "ACTIVE" LED (displays Tx and Rx activity) • RJ-45 connector (Controller Diagnostics)	Front or Back	Front, Back or Side
Ethernet Port 7 • "100" LED (illuminated linked at 100 Mbps, extinguished at all other times) • "ACTIVE" LED (displays Tx and Rx activity) • RJ-45 connector (Controller Expansion)	Front or Back	Front, Back or Side
Power Supply Overcurrent Device	Front or Back	Front, Back or Side
Serial User Interface to Console (C50S or C50J)	Front	Front
Front Panel User Interface (C60P or Keyboard, LCD, Bell)	Front	Front
USB Port A	Front or Back	Front

5.5.2 Optional User Interfaces

In addition to the minimum user interface, the ATC may include one or more optional user interface components in the designated locations:

Table 5-2. Optional User Interface Components

Optional Components	Rack	Shelf
Aux Switch (only supplied with Keyboard, LCD, Bell)	Front	Front
Serial I/O C13S	Front or Back	Front, Back or Side
Communication Slot 1 with Ethernet Port 4	Front or Back	Front, Back or Side
Communication Slot 2 with Ethernet Port 8	Front or Back	Front, Back or Side
Data Key	Front or Back	Front, Back or Side
SD Card	*Front or Back	*Front, Back or Side
Parallel I/O C11S	Back	Front, Back or Side
Parallel I/O C12S	Back	Front, Back or Side
Parallel I/O C1S	Back	Front, Back or Side
Parallel I/O MSA, MSB, MSC and MSD connectors	Front	Front
Power Indicator LED for each DC Power Source	Front or Back	Front, Back or Side
Power Switch	Front or Back	Front, Back or Side
Serial I/O NEMA Port 1	Front or Back	Front
NEMA TS 2 Type 1 MSA Power Connector	Front or Back	Front
CAN Bus	Front or Rear	Front

**(external access is optional)*

5.5.3 User Interface Pin Connections

SP4 Connector Pinout

Pin	Function
1	C50_ENABLE
2	SP4 RXD
3	SP4 TXD
4	NA
5	DCGND3
6	NA
7	NA
8	NA
9	NA

SP6 Connector Pinout

Pin	Function
1	+5VDC
2	SP6 RXD
3	SP6 TXD
4	NA
5	DCGND3
6	NA
7	CPU_RESET
8	NA
9	CPU_ACTIVE

Guidance:

When the C50_ENABLED signal is asserted (active low), communications to all other SP4 ports is disabled and no other SP4 received data is presented to the Engine Board.

DCGND3 is the ground reference for all externally-accessible signals on front panel SP4 and SP6 connectors. If the front panel is integrated, SP6 signals are not required to be isolated and may use the Engine Board ground reference.

5.5.4 User Interface Operation

5.5.4.1 Keyboard, LCD and Bell Operation

Keyboard

The keyboard, at a minimum, shall be capable of the complete single keystroke functionality of the standard Model 2070 front panel. It shall not be necessary to translate or otherwise affect alternate non-standard keypresses in order to provide the keystrokes available on a standard

Model 2070 front panel. Each key shall have a legend which is permanently attached to the ATC, either on the front panel immediately adjacent to the key; or engraved, embossed, or otherwise permanently affixed to the key cap. Minimum key surface area shall be 0.09 square inches. Minimum key spacing shall be 0.5” on centers. The actual keypad arrangement is not specified here.

CPU_ACTIVE LED Indicator

The CPU ACTIVE LED indicator shall be controlled by the CPU ACTIVE signal. When this signal is asserted low, the LED shall illuminate. The CPU_ACTIVE LED Indicator shall be located on the front panel.

The CPU_ACTIVE LED Indicator shall be turned ON following a power-on reset.

Display

The display shall include a Liquid Crystal Display (LCD) or other dot matrix display. It shall include integrated backlighting and shall provide contrast or brightness control as needed by the display technology. The contrast/brightness control may be implemented by dedicated button(s), a rotating knob, or software control via the standard keyboard. The adjustment shall provide the entire contrast/brightness range of the display. The active contrast/brightness level shall be retained across a power cycle. The lighting shall be turned on by keyboard activity or by command. The lighting circuitry shall consume no power when in the off state.

If contrast/brightness is controlled using a rotating knob, it shall increase with clockwise rotation.

If contrast/brightness is software controlled via the standard keyboard, the control shall be accomplished by pressing the (*) key followed by the (+) or (-) key within 1.0 sec to enable adjustment mode. During adjustment mode, the (+) key shall increase contrast/brightness and the (-) key shall decrease it. Pressing the (*) key again shall disable adjustment mode. Adjustment mode shall automatically be disabled after 10 sec of keyboard inactivity. The display shall not send any keys to the Engine Board during adjustment mode.

Contrast and brightness levels shall be persistent across power outages.

The display shall have a minimum of 8 lines with 40 characters each with minimum character height of 0.122” by 0.087” wide contained within a frame 0.140” high by 0.104” wide. The LCD shall be capable of displaying, at any position on the display, any standard printable ASCII characters as well as the user-defined special characters.

The display may optionally contain a heating element for low-temperature operation. This heating element shall be application-controllable using configuration commands for that purpose found in Table 6-1. The front panel shall ignore commands to turn on the heating element should the ambient temperature be high enough for proper LCD operation.

Cursor

Cursor display shall be turned ON and OFF by command.

- When ON, the cursor shall be displayed at the current cursor position

When OFF, no cursor shall be displayed

All other cursor functions shall remain in effect.

Reset

A user interface RESET shall be generated whenever power is applied, upon activation of a momentary reset switch on the PCB (if it exists), and upon activation of the CPU_RESET line.

Following the user interface RESET being active for a minimum of 25 ms, or receipt of a valid Soft Reset display command, the following shall occur:

1. Auto-repeat, blinking, auto-wrap, and auto-scroll shall be set to OFF.
2. Each special character shall be set to ASCII space (0x20).
3. The tab stops shall be set to columns 9, 17, 25, and 33.
4. The backlight timeout value shall be set to 6 (60 seconds).
5. The display shall be cleared (all ASCII space).
6. The user interface shall transmit a power up string to SP6 RXD once power is applied to the User Interface, or the USER Interface RESET is active. The string shall be “ESC [PU”, (0x1B 0x5B 0x50 0x55).

The cursor shall be turned on.

Key Press

When a key press is detected, the appropriate key code shall be transmitted to SP6 RXD. If two keys are depressed simultaneously, the following sequence shall be transmitted:

Table 5-3. Key Press Command-Response Codes

ESC [K P1 P2	Response: 0x1B 0x5B 0x4B P1 P2	P1: key one pressed; P2: key two pressed (NOTE: key codes for P1 and/or P2 may be multi-byte per Key Codes table)

No key code shall be transmitted upon release of one or both keys following a two keypress sequence.

If more than two keys are pressed simultaneously, no code shall be sent.

Auto Repeat

Auto-repeat shall be turned ON and OFF by command. When ON, the key code shall be repeated at a rate of five times per second starting when the key has been depressed continuously for 0.5 second and shall terminate when the key is released or another key is pressed.

Special Characters

The controller circuitry shall be capable of composing and storing eight special graphical characters on command, and displaying any number of these characters in combination with the standard ASCII characters. Undefined characters shall be ignored. User-composed characters shall be represented in the front panel communication protocol in the Model 2070 specification. P1 represents the special character number (1-8). Pn's represent columns of pixels from left to right. The most significant bit of each Pn represents the top pixel in a column and the least significant bit shall represent the bottom pixel. A logic ‘1’ shall turn the pixel ON. There shall be a minimum of 5 Pn's for 5 columns of pixels in a command code sequence terminated by an “f.” If the number of Pn's is greater than the number of columns available on the LCD, the extra Pn's shall be ignored. P1 and all Pn's shall be in ASCII-coded decimal characters without leading zero.

Character Overwrite

Character overwrite mode shall be the only display mode supported. A displayable character received shall always overwrite the current cursor position on the display. The cursor shall automatically move right one character position on the display after each character write

operation. When the rightmost character on a line (position 40) has been overwritten, the cursor position shall be determined based on the current settings of the auto-wrap mode.

Auto Wrap

Auto-wrap shall be turned ON & OFF by command. When ON, a new line operation shall be performed after writing to the right-most position. When OFF, upon reaching position 40, input characters shall continue to overwrite the right-most position.

Cursor Positioning

Cursor positioning shall be non-destructive. Cursor movement shall not affect the current display, other than blinking the cursor and momentarily hiding the character at that cursor position.

Blinking

Blinking characters shall be supported and shall be turned ON and OFF by command. When ON, all subsequently received displayable characters shall blink at the rate of 1 Hz with a 60 percent ON / 40 percent OFF duty cycle. It shall be possible to display both blinking and non-blinking characters simultaneously.

Tab Stops

Tab stops shall be configurable at all columns. A tab stop shall be set at the current cursor position when a SetTabStop command is received. Tab stop(s) shall be cleared on receipt of a ClearTabStop command. On receipt of the HT (tab) code, the cursor shall move to the next tab stop to the right of the cursor position. If no tab stop is set to the right of the current cursor position, the cursor shall not move.

Auto Scroll

Auto-scroll shall be turned ON and OFF by command. When ON, a linefeed or new line operation from the bottom line shall result in the display moving up one line. When OFF, a linefeed or new line from the bottom line shall result in the top line clearing, and the cursor being positioned on the top line.

Simultaneous Auto Wrap and AutoScroll

If AutoScroll is OFF, nothing should happen. If AutoScroll is ON, the display should scroll down one row (so that row 1 is now row 2), the cursor should go to the right-most column of the "new" row 1 and write a SPACE to that location.

Refresh Rate

Displayable characters that require updating, including those that are blinking, shall be updated within 50 ms of being changed (effective refresh rate of 20 times per second).

Backlight Timeout

The display backlight shall illuminate when any key is pressed and shall illuminate or extinguish by command. The backlight shall extinguish when no key is pressed for a specified time. This time shall be program selected by command, by a number in the range 0 to 63 corresponding to that number of 10-second intervals. A value of 1 shall correspond to a timeout interval of 10 seconds. A value of 0 shall indicate no timeout.

Command Codes

The command codes shall use the following conventions:

Parameters and options: Parameters are depicted in both the ASCII and hexadecimal representations as the letter 'P' followed by a lower-case character or number. These are interpreted as follows:

Pn: Value parameter, to be replaced by a value, using one ASCII character per digit without leading zeros.

P1: Ordered and numbered parameter. One of listed known parameter(s) with a specified order and number (Continues with P2, P3, etc.)

Px: Display column number (1- end), using one ASCII character per digit without leading zeros.

Py: Display line (1- bottom) one ASCII character. Continue the list in the same fashion

Values of 'h' (0x68) and 'l' (0x6C) are used to indicate binary operations. 'h' represents ON (high), 'l' represents OFF (low).

ASCII representation: Individual characters are separated by spaces for clarity; these are not to be interpreted as the ASCII space character.

Hexadecimal representation: Characters are shown as their hexadecimal values and will be in the range 00 to 7F (7 bits).

Communications

The controller circuit shall communicate via a SP6 asynchronous serial interface. The interface shall be configured for 115.2 kbps, 8 data bits, 1 stop bit, and no parity.

Bell

The user interface shall include an electronic bell to signal receipt of ^G (0x07). Receipt of all other characters and ESC codes shall continue during the time the bell sounds. The bell shall be rated at 70 decibels minimum sound pressure at the operating frequency which shall be between 1.5 kHz and 4.5 kHz.

Configuration Command Codes

Table 5-4. Configuration Command Codes

CONFIGURATION COMMAND CODES		
ASCII REPRESENTATION	HEX VALUE	FUNCTION
HT	09	Move cursor to next tab stop
CR	0D	Position cursor at first position on current line
LF	0A	(Line Fee) Move cursor down one line
BS	08	(Backspace) Move cursor one position to the left and write space
ESC [Py ; Px f	1B 5B Py 3B Px 66	Position cursor at (Px, Py)
ESC [Pn C	1B 5B Pn 43	Position cursor Pn positions to right
ESC [Pn D	1B 5B Pn 44	Position cursor Pn positions to left
ESC [Pn A	1B 5B Pn 41	Position cursor Pn positions up

CONFIGURATION COMMAND CODES		
ASCII REPRESENTATION	HEX VALUE	FUNCTION
ESC [Pn B	1B 5B Pn 42	Position cursor Pn positions down
ESC [H	1B 5B 48	Home cursor (move to 1,1)
ESC [2 J	1B 5B 32 4A	Clear screen with spaces without moving cursor
ESC c	1B 63	Soft reset
ESC P P1 [Pn ; Pn…f	1B 50 P1 5B Pn 3B…Pn 66	Compose special character number Pn (1-8) at current cursor position
ESC [< Pn V	1B 5B 3C Pn 56	Display special character number Pn (1-8) at current cursor position
ESC [25 h	1B 5B 32 35 68	Turn Character blink on
ESC [25 l	1B 5B 32 35 6C	Turn Character blink off
ESC [< 5 h	1B 5B 3C 35 68	Illuminate Backlight
ESC [< 5 l	1B 5B 3C 35 6C	Extinguish Backlight
ESC [33 h	1B 5B 33 33 68	Cursor blink on
ESC [33 l	1B 5B 33 33 6C	Cursor blink off
ESC [27 h	1B 5B 32 37 68	Reverse video on – Note 2
ESC [27 l	1B 5B 32 37 6C	Reverse video off – Note 2
ESC [24 h	1B 5B 32 34 68	Underline on – Note 2
ESC [24 l	1B 5B 32 34 6C	Underline off – Note 2
ESC [0 m	1B 5B 30 6D	All attributes off
ESC H	1B 48	Set tab stop at current cursor position
ESC [Pn g	1B 5B Pn 67	Clear tab stop Pn = 0,1,2 at cursor = 3 all tab stops
ESC [? 7 h	1B 5B 3F 37 68	Auto-wrap on
ESC [? 7 l	1B 5B 3F 37 6C	Auto-wrap off
ESC [? 8 h	1B 5B 3F 38 68	Auto-repeat on
ESC [? 8 l	1B 5B 3F 38 6C	Auto-repeat off
ESC [? 25 h	1B 5B 3F 32 35 68	Cursor on
ESC [? 25 l	1B 5B 3F 32 35 6C	Cursor off
ESC [< 47 h	1B 5B 3C 34 37 68	Auto-scroll on
ESC [< 47 l	1B 5B 3C 34 37 6C	Auto-scroll off
ESC [< Pn S	1B 5B 3C Pn 53	Set Backlight timeout value to Pn (0-63)

CONFIGURATION COMMAND CODES		
ASCII REPRESENTATION	HEX VALUE	FUNCTION
ESC [P U	1B 5B 50 55	String sent to ENGINE BOARD when EIPA power up
ESC [? 12 h	1B 5B 3F 31 32 68	Heater on (subject to temperature constraints)
ESC [? 12 l	1B 5B 3F 31 32 6C	Heater off

NOTE: 1. Numerical values have one ASCII character per digit without leading zero.

2. Reverse video and underline NOT required for integrated front panel displays.

Table 5-5. Inquiry Command-Response Codes

INQUIRY COMMAND – RESPONSE CODES				
COMMAND Engine Board to Front Panel Module		RESPONSE Front Panel Module to Engine Board		FUNCTION
ASCII Representation	HEX Value	ASCII Representation	HEX Value	
ESC [6 n	1B 5B 36 6E	ESC [Py; Px R	1B 5B Py 3B Px 52	Inquire Cursor Position
ESC [B n	1B 5B 42 6E	ESC [P1; P2;....P6 R	1B 5B P1 3B P2 3B.....P6 52	Status Cursor Position P1: Auto-wrap (h, 1) P2: Auto-scroll (h, l) P3: Auto-repeat (h, l) P4: Backlight (h, l) P5: Backlight timeout P6: AUX Switch (h, l)
ESC [A n	1B 5B 41 6E	ESC [P1 R	1B 5B P1 52	P1: AUX Switch (h, l)
ESC [h n	1B 5B 68 6E	ESC [P1 R	1B 5B P1 52	P1: Heater (h, l)
ESC [c	1B 5B 63	ESC [P1 R	1B 5B P1 52	P1: Type (A,B,D,H)
ESC [K n	1B 5B 4B 6E	(normal key code)	(normal key code)	Return key code if key is currently pressed, else return NULL (hex 00)

Key Codes

Table 5-6. Key Press Codes

Key	ASCII DATA (TEXT)	ASCII DATA (HEX)
0	0	30
1	1	31
2	2	32
3	3	33
4	4	34
5	5	35
6	6	36
7	7	37
8	8	38
9	9	39
A	A	41
B	B	42
C	C	43
D	D	44
E	E	45
F	F	46
(UP ARROW)	ESC [A	1B 5B 41
(DOWN ARROW)	ESC [B	1B 5B 42
(RIGHT ARROW)	ESC [C	1B 5B 43
(LEFT ARROW)	ESC [D	1B 5B 44
ESC	ESC O S	1B 4F 53
NEXT	ESC O P	1B 4F 50
YES	ESC O Q	1B 4F 51
NO	ESC O R	1B 4F 52
*	*	2A
+	+	2B
-	-	2D
ENTER	CR	0D
Double Key Press (P1; P2)	ESC [K P1 P2	1B 5B 4B P1 P2
Vendor Specific Key 1	ESC O V	1B 4F 56
Vendor Specific Key 2	ESC O W	1B 4F 57
Vendor Specific Key 3	ESC O X	1B 4F 58
Vendor Specific Key 4	ESC O Y	1B 4F 59
Vendor Specific Key 5	ESC O Z	1B 4F 5A
Vendor Specific Key 6	ESC O [	1B 4F 5B
Vendor Specific Key 7	ESC O \	1B 4F 5C
Vendor Specific Key 8	ESC O]	1B 4F 5D
Vendor Specific Key 9	ESC O ^	1B 4F 5E
Vendor Specific Key 10	ESC O _	1B 4F 5F
Vendor Specific Key 11	ESC O ‘	1B 4F 60
Vendor Specific Key 12	ESC O a	1B 4F 61
Vendor Specific Key 13	ESC O b	1B 4F 62
Vendor Specific Key 14	ESC O c	1B 4F 63
Vendor Specific Key 15	ESC O d	1B 4F 64
Vendor Specific Key 16	ESC O e	1B 4F 65

AUX Switch Codes

Table 5-7. AUX Switch Codes

SWITCH POSITION	ASCII DATA (TEXT)	ASCII DATA (HEX)
ON	ESC O T	1B 4F 54
OFF	ESC O U	1B 4F 55

5.5.4.2 Serial Port

The above key codes, configuration command codes and inquiry command-response codes shall be communicated via SP6 in the absence of a front panel display. In lieu of the keyboard and display, an intelligent device may be used.

5.5.4.3 Ethernet Port

10/100 Ethernet ports are used for hardwired communications to external devices. Please refer to the IEEE 802.3 standard for operation.

5.5.4.4 USB Port

USB is used as an interface to memory devices. Refer to USB specification.

5.5.4.5 Datakey

Datakey Keyceptacle™ (KC4210, KC4210PCB or equal).

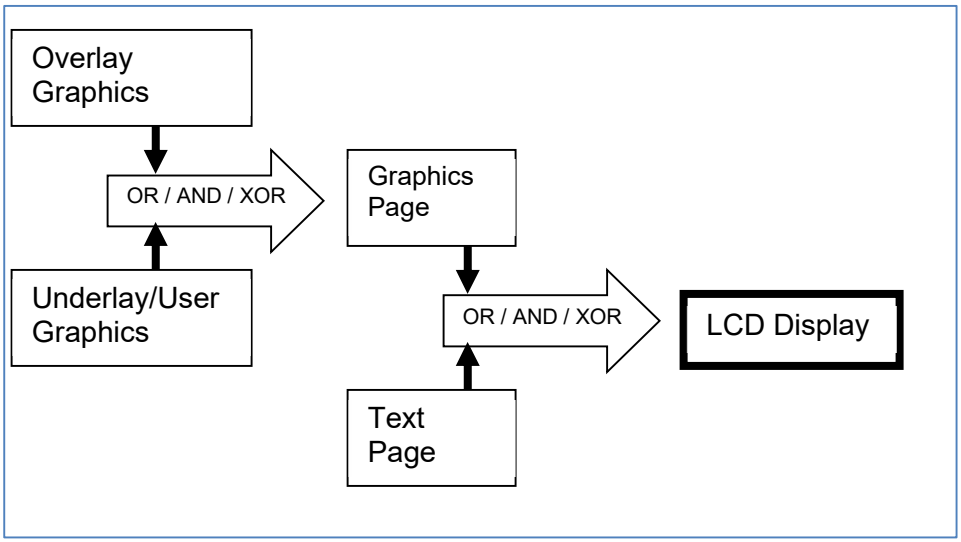
5.5.4.6 Graphics Interface

Support for bit-mapped graphics is optional. If the manufacturer chooses to support bit-mapped graphics, then the following graphics interface shall be supported. The graphic interface commands are supplemental to the normal configuration command codes and functionality specified in Section 5.5.4.1.

Concept

The graphical interface supports bit-mapped graphics and is designed around the concept of layers and pages with the application program having control over how these layers and pages interact. The graphical portion of the interface is composed of two layers: Underlay/User (U) and Overlay (O). These two layers can then be OR'd, AND'd, or XOR'd on a pixel by pixel basis to create the Graphics Page (using the ESC G G _ commands). The graphics page is then further OR'd, AND'd, or XOR'd with the character-based text page (using the ESC G D _ commands) for final display on the LCD.

Pictorially:



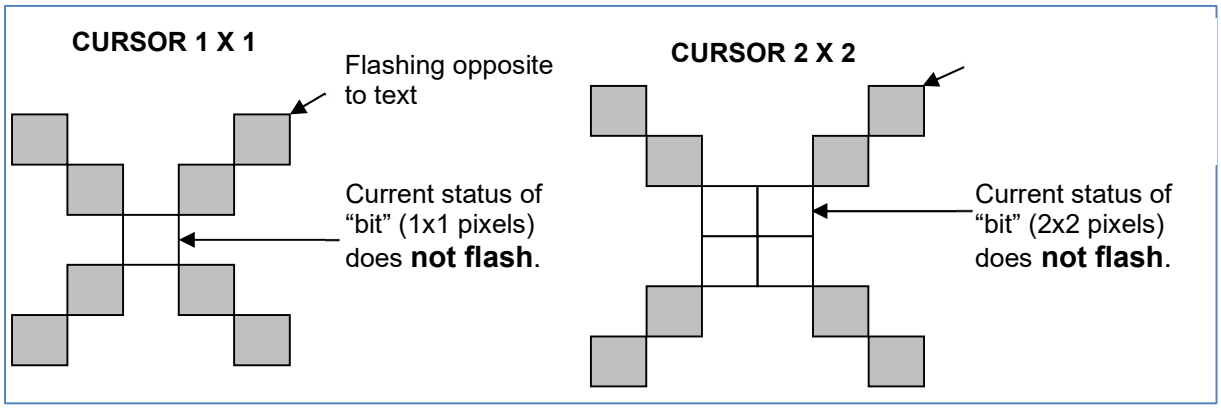
For flexibility purposes, each graphical layer can be individually disabled, enabled with a “bit” represented as a 1 x1 pixel matrix, or enabled with a bit being represented as a 2 x 2 pixel matrix. Note that the two graphic layers can be programmed independently and layer operations such as OR, AND and XOR work on the true pixels and not the bits.

The pixel coordinate of the upper left hand corner is (0,0) and the lower right hand corner is (x_max, y_max). That is, the x-coordinate increases positively to the right with maximum value of x_max and the y-coordinate increases positively in the downward direction with maximum value of y_max (x_max and y_max are specified by the host EEPROM). If any command would result in a coordinate value less than zero, then it zero shall be used for that coordinate. Similarly, if any command would result in a coordinate value greater than x_max or y_max, then x_max or y_max would be used as appropriate for that coordinate.

Graphic Cursor

A graphic cursor shall be supported and under application program control can either be turned OFF or enabled for either the underlay/user layer or overlay layer. When enabled, the graphic cursor shall always flash opposite to the text flash and the center of the cursor shall indicate the current status of the bit. The graphic cursor style is dependent on the bit size of the layer it is enabled for. If the graphic cursor is enabled for a layer that is turned off, then no graphic cursor shall be displayed. The graphic cursor can be positioned either absolutely or relatively by command. Note that a relative position command requires that a 16-bit (signed) value be supplied for the y-coordinate movement.

Pictorially:



Graphic Operations

The graphical interface shall provide independent control for each of the two graphical layers. Graphic operation commands are available to perform the following functions on a block of bits on a layer:

Function	ASCII Representation ("fn")	HEX Value
Clear	C	43
Fill	F	46
AND	&	26
OR	+	2B
XOR	X	58
Write	W	57

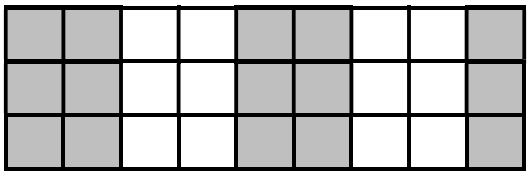
When using some of the commands, one must supply one or more bytes of binary data ([data] in the Graphic Configuration Command Codes table).

- Clear does not have any data bytes included and turns off each bit in the graphic block specified
- Fill has one data byte and uses the binary value repetitively to fill the block of graphic bits specified. A value of zero (0x00) can be used to clear the block of graphic bits, while a value of 255 (0xFF) can be used to turn on all graphic bits within a block. Other values will result in bit patterns being displayed.

AND, OR, XOR and Write require a sufficient number of data bytes to be included with the command to precisely specify each graphic bit in the block specified. AND, OR and XOR perform a bit-wise operation on each bit within the graphic block and its respective bit in the data byte stream. The write command sets each bit in the graphic block to its respective bit in the data stream.

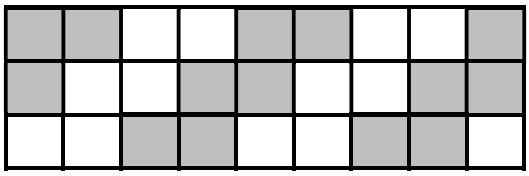
When specifying the data bytes, one starts with the upper left hand bit of the block and places it in the msb position of the first data byte. As one proceeds right on a bit-by-bit basis, the next bit is placed in the next msb position of the data byte and after filling the lsb of the first data byte, one begins by placing the next bit in the msb of the second byte. Upon reaching the right-most bit of the first row of the block, one advances down one row within the block and continues extracting bits beginning from the left side again. This process continues until the last bit of the block has been placed in the last data byte. Any bits of the last data byte that are not required to represent the graphic block will be ignored by the display so that value does not matter. Note that this process is also used when filling a block with a pattern.

For example, the graphic block:



would be represented by the data bytes (hex values) “CC E6 73 20”.

As a second example, the following graphic block



would be represented by the data bytes “0xCC 0xCC 0xCC 0xC0” and it could also be created using a Fill command with the single data value “0xCC.”

It should be noted that one must include exactly the correct number of data bytes as required by the command and the graphic block being modified. If an insufficient number of data bytes are provided, then the bytes of subsequent commands to the display will be used as data bytes. If too many data bytes are provided, then the graphic function will only use the amount of data it requires, and the display will try to process the remaining data bytes as new commands and erratic behavior may be observed.

When performing a graphic operation, one is always working on a block of bits. The block can either be specified completely by providing the bit coordinates of the upper left hand corner (x1, y1) and the lower right hand corner (x2, y2), or relatively by only specifying the lower right hand corner relative to the current cursor position (Δx, Δy).

Reset Condition

Upon reset (power up or soft reset command), the graphical interface shall clear both the underlay/user and overlay layers; disable both the underlay/user and overlay layers; set the graphics page creation mode to OR; and set the LCD Display creation mode to OR.

Graphics Configuration Command Codes

Table 5-8. Graphics Configuration Command Codes

GRAPHIC CONFIGURATION COMMAND CODES		
ASCII REPRESENTATION	**HEX VALUE**	**FUNCTION**
ESC G U 0	1B 47 61 30	Disable Underlay/User graphic layer
ESC G U 1	1B 47 61 31	Enable Underlay/User graphic layer in 1x1 pixel mode
ESC G U 2	1B 47 61 32	Enable Underlay/User graphic layer in 2x2 pixel mode
ESC G O 0	1B 47 5B 30	Disable Overlay graphic layer
ESC G O 1	1B 47 5B 31	Enable Overlay graphic layer in 1x1 pixel mode
ESC G O 2	1B 47 5B 32	Enable Overlay graphic layer in 2x2 pixel mode
ESC G C 0	1B 47 43 30	Disable Graphic Cursor
ESC G C U	1B 47 43 61	Enable Graphic Cursor on Underlay/User graphic layer
ESC G C O	1B 47 43 5B	Enable Graphic Cursor on Overlay graphic layer
ESC G P x y	1B 47 5C XX xx yy	Move cursor position to absolute (x, y) pixel coordinate where x= (unsigned) 16 bit with XX = MSB and xx = LSB, and y = (unsigned) 8 bit.

GRAPHIC CONFIGURATION COMMAND CODES		
ASCII REPRESENTATION	**HEX VALUE**	**FUNCTION**
ESC G p sx sy	1B 47 6F XX xx YY yy	Move cursor relatively to current position by sx pixels horizontally and sy pixels vertically. Note: sx= (signed) 16 bit with XX = MSB and xx = LSB, and sy = (signed) 16 bit with YY=MSB and yy = LSB.
ESC G O fn x1 y1 x2 y2 [data]	1B 47 5B fn XX xx YY yy XX xx YY yy [data]	Perform graphic function fn using [data] on Overlay layer starting at upper left pixel coordinate (x1, y1) and ending at lower right pixel coordinate (x2, y2). Note that there may be zero or more bytes of data as required by the function specified and graphic area involved. Operation does not affect the graphic cursor position.
ESC G U fn x1 y1 x2 y2 [data]	1B 47 61 fn XX xx YY yy XX xx YY yy [data]	Perform graphic function fn using [data] on Underlay/User layer starting at upper left pixel coordinate (x1, y1) and ending at lower right pixel coordinate (x2, y2). Note that there may be zero or more bytes of data as required by the function specified and graphic area involved. Operation does not affect the graphic cursor position.
ESC G o fn Δx Δy [data]	1B 47 6F fn XX xx yy [data]	Perform graphic function fn using [data] on Overlay layer starting at current graphic cursor position (x, y) and ending at (x+Δx, y+Δy) where Δx = (unsigned) 16-bit with XX = MSB and xx = LSB, and Δy = (unsigned) 8-bit. Note that there may be zero or more bytes of data as required by the function specified and graphic area involved. Operation does not affect the graphic cursor position.
ESC G u fn Δx Δy [data]	1B 47 75 fn XX xx yy [data]	Perform graphic function fn using [data] on Underlay/User layer starting at current graphic cursor position (x, y) and ending at (x+Δx, y+Δy) where Δx = (unsigned) 16-bit with XX = MSB and xx = LSB, and Δy = (unsigned) 8-bit. Note that there may be zero or more bytes of data as required by the function specified and graphic area involved. Operation does not affect the graphic cursor position.
ESC G G +	1B 47 47 2B	Create Graphics Page by ORing Underlay/User and Overlay layers together. For each layer that is disabled, assume that the layer is cleared (i.e., all pixels are turned off).
ESC G G &	1B 47 47 26	Create Graphics Page by ANDing Underlay/User and Overlay layers together. For each layer that is disabled, assume that the layer is filled (i.e., all pixels are turned on).
ESC G G X	1B 47 47 58	Create Graphics Page by XORing Underlay/User and Overlay layers together. For each layer that is disabled, assume that the layer is filled (i.e., all pixels are turned on).

GRAPHIC CONFIGURATION COMMAND CODES		
ASCII REPRESENTATION	HEX VALUE	FUNCTION
ESC G D +	1B 47 44 2B	Create LCD Display image by ORing the Graphics and Text Pages together. If the Graphics Page is disabled (i.e. both the Underlay/User and Overlay layers are turned off), assume that the page is cleared (i.e., all pixels are turned off).
ESC G D &	1B 47 44 26	Create LCD Display image by ANDing the Graphics and Text Pages together. If the Graphics Page is disabled (i.e. both the Underlay/User and Overlay layers are turned off), assume that the page is filled (i.e., all pixels are turned on).
ESC G D X	1B 47 44 58	Create LCD Display image by XORing the Graphics and Text Pages together. If the Graphics Page is disabled (i.e. both the Underlay/User and Overlay layers are turned off), assume that the page is filled (i.e., all pixels are turned on).

5.5.4.7 SD Card

An optional SD Card socket may be provided on the Host Module. The socket type may be either Standard SD or Micro SD. This socket must utilize the SPI interface as described in Section 4.4.3.2. Physical access to this socket external to the controller is not required but is permitted.

5.6 Power Supply General Description

The power supply shall consist of one or more subassemblies that can be removed and reinstalled using standard hand tools. All electrical connections to the power supply subassembly(s) shall be through a connector(s) rated to handle required power supply voltage, current and signal levels defined in Section 5.6.6. It shall be cooled by convection only. The power supply shall be capable of supporting the internal ATC circuitry, plus provide power for each optional module. The power supply shall convert service voltage to the proper DC voltages at the power rating needed to support the unit and any external power as described in paragraph 5.6.6.

The power supply must produce all output voltages with the specified tolerances and capacities within 750 ms after the application of external power to the ATC. The power supply must also raise the POWERUP and POWERDOWN signals to a HIGH state, indicating that power is stable and available, within this same 750 ms time period.

5.6.1 "On/Off" Power Switch (optional)

If provided, an "On/Off" Power switch shall disconnect the AC line of the service voltage from the power supply. The “ON” position of the Power Switch shall be clearly labeled on the switch or the panel to which the switch is mounted.

Guidance: The Power Switch is not required to be located on the Power Supply subassembly(s).

5.6.2 LED DC Power Indicators (optional)

If provided, LED DC power indicators shall monitor individual DC voltages generated by the power supply and shall be lit under the following conditions:

a) Primary +5VDC is between +4.875V and +5.125V (referenced to DC Ground 1)

b) +12 VDC is between +11.7V and +12.3V (referenced to DC Ground 1)

c) -12 VDC is between -11.7V and -12.3V (referenced to DC Ground 1)

d) +12 VDC ISO is between +11.7V and +12.3V (332 Parallel I/O versions)

+24VDC is between +23.5V and +24.5V (NEMA versions)

Outside each range but within the acceptable range for each DC voltage (as defined elsewhere in this Standard) the LED condition may be either lit or dark. Outside each acceptable range the LED shall be dark.

Guidance: The LED DC Power Indicators are not required to be located on the Power Supply subassembly(s).

5.6.3 Service Voltage Over-Current Protection Device

An over-current protection device (OCPD) shall be provided to protect from over-current on the service voltage connection to the ATC power supply. The Service Voltage OCPD shall be accessible from the front of the unit.

Guidance: The Service Voltage OCPD is not required to be located on the power supply subassembly(s).

5.6.4 +5 VDC Standby Power

+5 VDC standby power shall be provided to hold up specified circuitry during the power down period. It shall consist of the monitor circuitry, holdup capacitors and charging circuitry. A charging circuit shall be provided, that under normal operation, shall fully charge and float the capacitors consistent with the manufacturers' recommendations. The holdup power requirements shall be a minimum constant drain of 600 µA at a range of +5 to +2 VDC for over 600 minutes. Capacitors shall be fully charged within one hour.

5.6.5 Monitor Circuitry

Monitor circuitry shall be provided to monitor incoming service power for power failure and restoration and LINESYNC generation.

5.6.5.1 Power Down and Power Up

Power Loss and Restoration Calibration for Model 332 and ITS Cabinets:

The POWERDOWN Output signal shall go LOW (ground true) within 50 ± 10 ms after the AC service voltage falls below 92 ±2 VAC ("**Loss of AC Service Power**"). The signal shall transition to HIGH within 50 ms of when both the AC service voltage exceeds 97 ± 2 VAC (**"Restoration of AC Service Power"**) and the +5VDC supply is within the range specified in Section 5.6.6.

The POWERUP output signal shall transition to LOW 525 ± 20 ms after POWERDOWN transitions to LOW. The signal shall transition to HIGH within 750 ms of **Restoration of AC Service Power** and 225 ± 25 ms after the supply is fully recovered (e.g. after +5 VDC is within the range specified in Section 5.6.6).

Power Loss and Restoration Calibration for NEMA Cabinets and ATC Cabinets:

The POWERDOWN Output signal shall go LOW (ground true) within 50 ± 10 ms of when the AC service voltage falls below 85 VAC ("**Loss of AC Service Power**"). The signal shall transition to HIGH within 50 ms of when both the AC service voltage exceeds 89 VAC (**"Restoration of AC Service Power"**) and the +5VDC supply is within the range specified in Section 5.6.6. When AC

service voltage is between the Loss and Restoration values the POWERDOWN signal may be either HIGH or LOW.

The POWERUP Output signal shall transition to LOW 525 ± 20 ms after POWERDOWN transitions to LOW. The signal shall transition to HIGH within 750 ms of **Restoration of AC Service Power** and 225 ± 25 ms after the supply is fully recovered (e.g. after +5 VDC is within the range specified in Section 5.6.6).

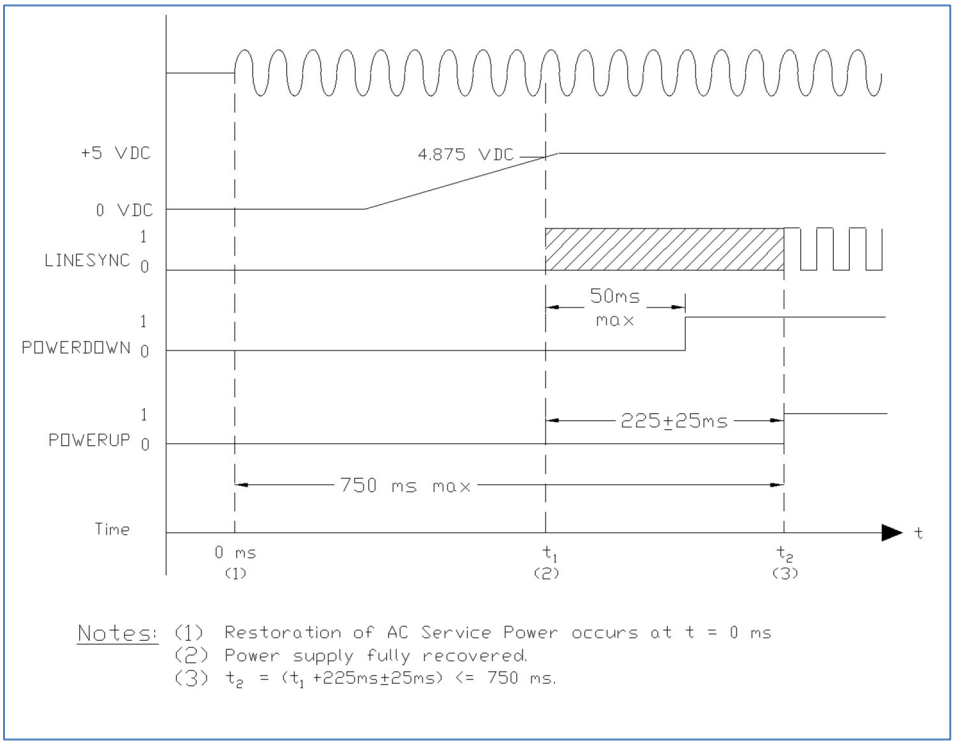


Figure 5-3. Power Supply Signals During Restoration of AC Service Power.

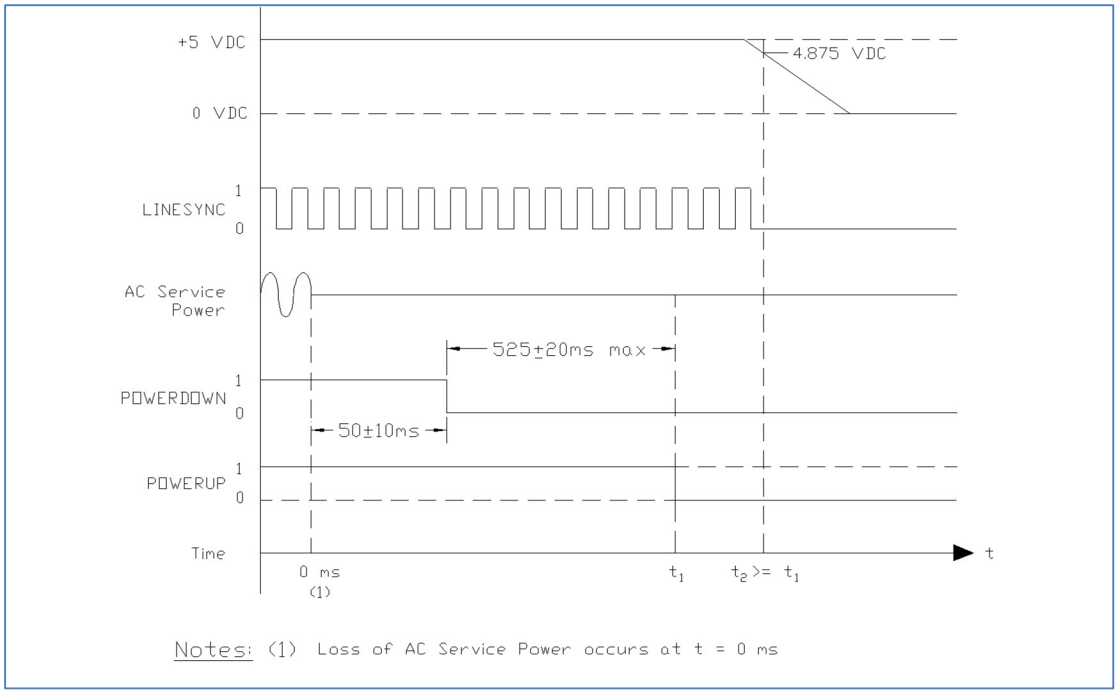


Figure 5-4. Power Supply Signals During Loss of AC Service Power.

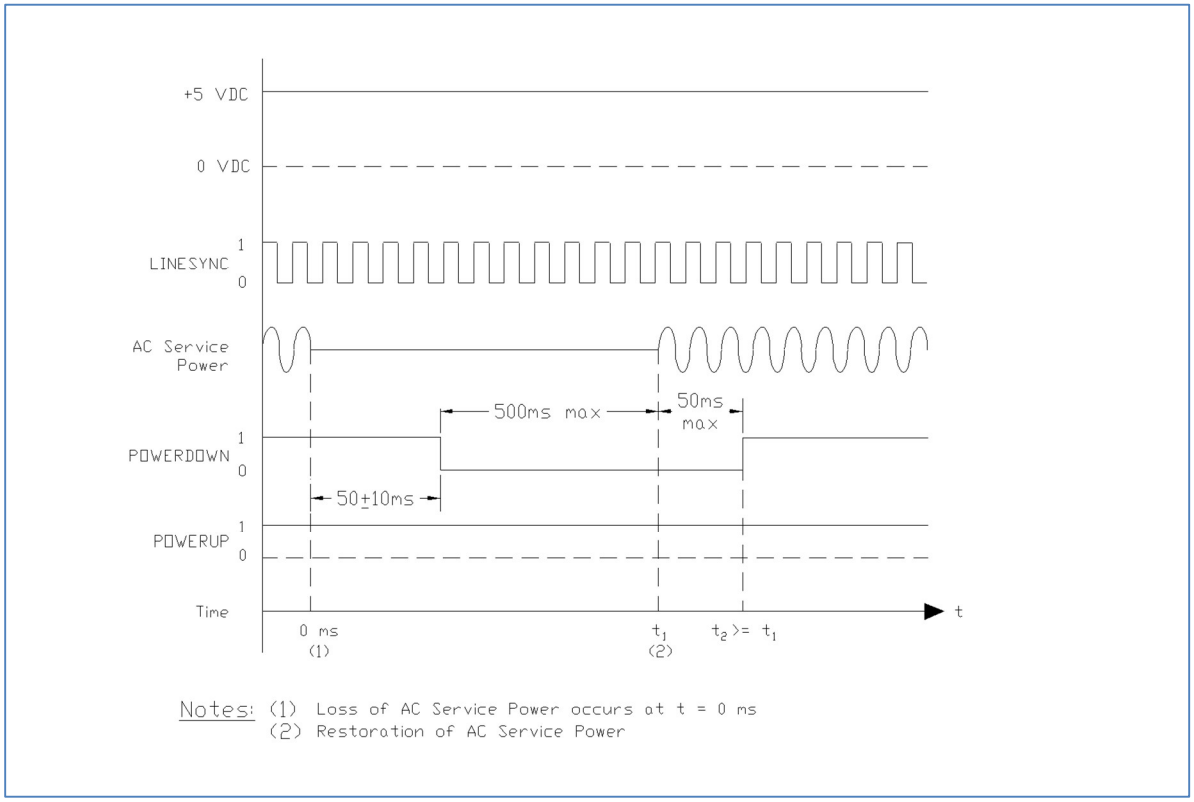


Figure 5-5. Power Supply Signals during Short Loss of AC Service Power.

5.6.5.2 LINESYNC

The LINESYNC signal shall be generated by a crystal oscillator, which shall synchronize to the AC service power line at 120 and 300˚ during normal AC service power conditions (see section 5.6.6.1). During loss of AC service power (holdup time) or when the AC service power line frequency is outside the normal range, the LINESYNC signal shall be a continuous square-wave signal with +5 VDC amplitude, nominal AC line frequency, and 50 ± 1 percent duty cycle. The output driver shall have sinking capability of at least 16 mA. The monitor circuit shall compensate for missing pulses and line noise during normal operation. The circuit shall begin generating the LINESYNC signal no later than POWERUP signal becoming active and shall continue generating the LINESYNC signal during power failure until the +5 VDC power supply drops below its minimum voltage as specified in Section 5.6.6 (hence the LINESYNC signal shall be present whenever the POWERUP signal is HIGH at a minimum). The crystal oscillator used to generate this signal shall have an accuracy of ± 0.005 percent at 25 C. The relationship of AC Service Voltage (top trace) and LINESYNC (bottom trace) is shown in Figure 6-4.

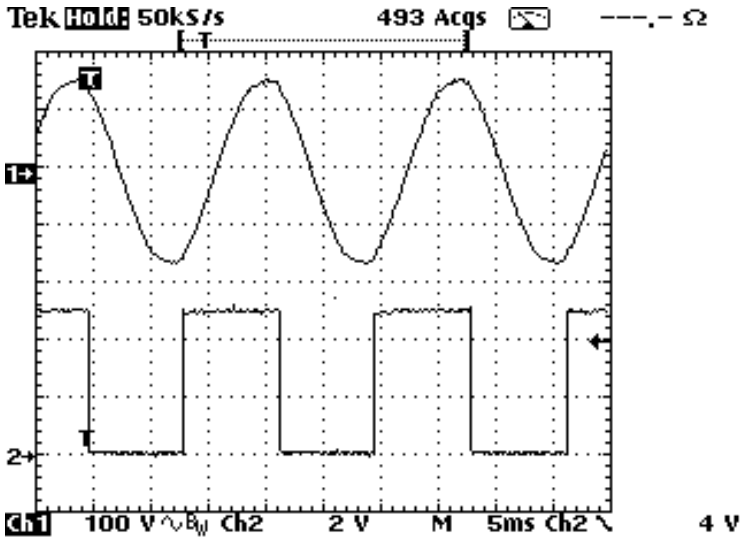


Figure 5-6. Relationship of AC Service Voltage and LINESYNC.

5.6.6 External Power Supply Requirements

The following external voltages shall be within the parameters shown for each of the indicated applications.

For each Optional Communications Interface Module (note that power supply should be sized appropriately based on the number of communication interface slots the ATC provides):

Voltage	Tolerances	I Minimum	I Maximum
+5 VDC	+4.850 to +5.20 VDC	0.050 A	0.500 A
+12 VDC	+11.4 to +12.6 VDC	0.050 A	0.100 A
-12 VDC	-11.4 to –12.6 VDC	0.050 A	0.100 A

For NEMA TS1 and NEMA TS2 Type 2 versions:

Voltage	Tolerances	I Minimum	I Maximum
+24 VDC	+22.0 to +26.0VDC	0.050 A	0.500 A

For 332 Parallel I/O version:

Voltage	Tolerances	I Minimum	I Maximum
+12 VDC ISO	+11.4 to +12.6 VDC	0.050 A	0.750 A

For the Required USB Port:

Voltage	Tolerances	I Minimum	I Maximum
+5 VDC	+4.875 to +5.125 VDC	0.000 A	0.500 A

For the optional Display (LCD) Heater:

Voltage	Tolerances	I Minimum	I Maximum
Up to +15 VDC	N/A	0.000 A	1.000 A

5.6.6.1 Line and Load Regulation

The power supply shall meet the external voltage tolerances for minimum and maximum loads called out. The normal AC service power conditions are defined as follows:

a. 332 Parallel I/O Version: 90 VAC to 135 VAC

b. NEMA: 89 VAC to 135 VAC

5.6.6.2 Ripple and Noise

Less than 0.5% RMS, 2 percent peak to peak, whichever is greater.

5.6.6.3 Over Voltage

The power supply shall clamp at 130 percent of the nominal output voltage for all outputs.

5.6.6.4 Inrush Current

Cold start inrush shall be less than 25A at 115VAC.

5.6.6.5 Holdup Time

The power supply shall supply sufficient current budget for all internal voltages, except for display heater and backlight, for 550 ms after power loss at the minimum line voltage specified in Section 5.6.6.1. The supply shall be capable of holding up the ATC for two 500 ms power loss periods occurring in a 1.5-second period at the minimum line voltage specified in Section 5.6.6.1. The supply shall also be capable of holding up the ATC during the rapid power interruption test as described in Section 7.10.3. All USB port(s) and communication(s) slot(s) shall be able to constantly draw their maximum current for each voltage as specified in Section 5.6.6 during the outages. The ATC power supply shall recover sufficiently such that this power loss test can be repeated continuously once every 60 seconds over the full operating temperature range. Note that the I/O voltage used for internal purposes (e.g., pull-up resistors) must be maintained throughout the outages.

5.6.6.6 Overload Protection

The power supply shall include automatic overload protection circuitry for each output, as well as the USB +5 VDC output. The overload protection circuitry shall limit the output power during overload conditions, including shorted outputs, without blowing the fuse and without exceeding the ratings of any component. When the overload condition is removed, the output shall automatically recover specified regulation. An overload condition on the +5 VDC output may simultaneously limit power of all outputs, as the ATC will be RESET by the drop in +5 VDC voltage. Overload on Heater, +12VDC, -12VDC, +24 VDC and +12VDC ISO shall not limit the output

power of +5 VDC, allowing the Engine Board to log secondary overloads. Overload of the USB +5 VDC output shall not result in overload of any other power supply output.

5.6.7 Internal Power Supply Requirements

The following internal voltages and currents shall be within the following ranges.

Voltage	Tolerances	I Minimum	I Maximum
VPRIMARY (+5VDC)	+4.800 to +5.20 VDC	0.050 A	5.000 A
VSTANDBY (+5VDC)	+5.2 down to +2.0 VDC	0.000 A	8uA @2.5V @25dC

5.7 Mechanical and Physical General Description

The ATC mechanical and physical attributes provide mechanical enclosure and human engineering, including:

- Maximum Size
- Form Factor
- Mounting and Installation Method
- Materials
- Structural Integrity
- Ease of Use
- Cost Effectiveness

It is the intent of this specification to accomplish the following:

- Preserve compatibility with existing cabinet styles
- Reduce the size and complexity of existing controllers
- Improve human engineering for intuitive use of complex control functions
- It is not the intent of this specification to do the following:
- Interchange electronic modules and mechanical assemblies among vendors (except communications interface and Engine Board)
- Dictate mechanical details
- Preserve existing controller sizes and form factors

5.7.1 Chassis

5.7.1.1 Construction Materials

The CHASSIS, including supports, mounting surfaces, power supply enclosures, and front panel, shall be made of 0.063-inch minimum aluminum sheet metal or equivalent strength non-corrosive material. Construction materials shall withstand all environmental standards of this specification.

5.7.1.2 Weight

The total ATC weight with all internally installed components shall not exceed 25 pounds including communication interface slot modules. The weight of a communication interface slot module shall not exceed 1.5 pounds.

5.7.1.3 Mounting Method

As a minimum, the chassis shall be capable of mounting to an EIA-310-B rack using 4U (or smaller standard increment) open-end mounting slots. If not rack-mounted, EIA-310-B does not apply and other chassis mounting methods are allowed, not to exceed overall dimensions specified here.

Mounting method shall withstand all mechanical shock and vibration requirements of this standard.

5.7.1.4 Dimensions (All dimensions are given in inches)

Details of maximum basic dimensions (not restricted to shape shown):

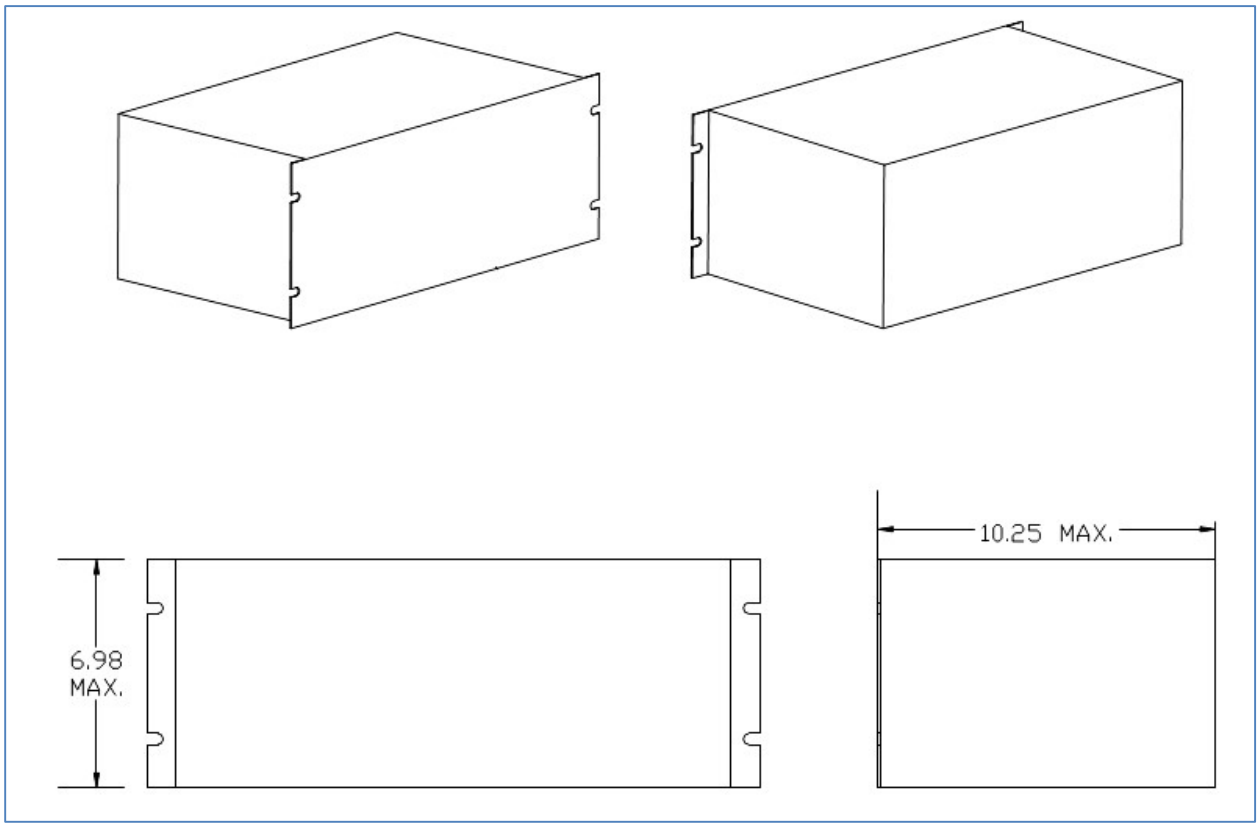


Figure 5-7. Maximum Basic Dimensions, Rack Mount.

Dimensions of shelf-mounted controller unit shall conform to NEMA TS-2 Standard, paragraph 3.2.1.

6 PARALLEL AND SERIAL I/O DETAILS

6.1 General Information

The ATC Input / Output (I/O) provides both serial and parallel connections to field devices connected to the ATC.

6.1.1 Parallel Input / Output Overview

The parallel I/O connects the ATC to transportation cabinets including, but not limited to, the following:

- NEMA TS-1
- NEMA TS-2 Type 1
- NEMA TS-2 Type 2
- NEMA TS-2 Type 2 with custom D connector
- Model 332
- ITS
- Other legacy transportation cabinets

6.1.2 Serial I/O Overview

The serial connections described here provide communications for implementation of existing transportation standards, including but not limited to, the following:

- NEMA TS-1
- NEMA TS-2 Type 1
- NEMA TS-2 Type 2
- Model 170
- Model 2070

6.2 Parallel Input / Output (PI/O)

6.2.1 Parallel Connection to Model 332 Cabinets

The parallel connection to a Model 332 cabinet shall consist of the field controller unit (FCU), parallel input/output ports, connectors C1S, and C11S, C12S and other circuit functions including muzzle jumper.

6.2.1.1 Field Controller Unit (FCU)

The FCU shall include a programmable microprocessor/controller unit together with all required clocking and support circuitry. Operational software necessary to meet housekeeping and functional requirements shall be provided.

Alternatively, the FCU may be separate and distinct from the Field I/O Module as long as all other requirements of this Standard are met, including isolation requirements for parallel I/O.

6.2.1.2 Parallel I/O Ports

Input Ports

The I/O ports shall provide 64 bits of input using ground-true logic. Each input shall be read logic "1" when the input voltage at its field connector input is less than 3.5 VDC, and shall be read logic "0" when either the input current is less than 100 µA or the input voltage exceeds 8.5 VDC. Each input shall have an internal pull-up to the +12 VDC ISO power supply and shall not deliver greater than 20 mA to a short circuit to ground.

Output Ports

The I/O ports shall provide 64 bits of output. Each output written as a logic "1" shall have a voltage at its field connector output of less than 4.0 VDC. Each output written as logic "0" shall provide an open circuit (1 MΩ or more) at its field connector output. Each output shall consist of an open-collector capable of driving 40 VDC minimum and sinking 100 mA minimum. Each output circuit shall be capable of switching from logic "1" to logic "0" within 100 μs when connected to a load of 100 kΩ minimum. Each output circuit shall be protected from transients of 10 ± 2 μs duration, ±300 VDC from a 1 kΩ source, with a maximum rate of 1 pulse per second.

Parallel I/O Port Timing

Each output shall latch the data written and remain stable until either new data is written or the active-low reset signal (NRESET). Upon an active-low reset signal (NRESET), each output shall latch a logic "0" and retain that state until a new writing. The state of all output circuits at the time of Power Up or in Power Down state shall be open. It shall be possible to simultaneously assert all outputs within 100 μs. An output circuit state not changed during a new writing shall not glitch when other output circuits are updated.

6.2.1.3 Other Parallel I/O Functions

Signals and Capacitive Load

A maximum capacitive load of 100 pF shall be presented to the LINESYNC input signal.

Legacy Signal Monitors

An external WDT “muzzle” jumper shall be provided internal to the ATC. With the jumper IN and NRESET transitions HIGH (FCU active), the FCU shall output a state change on O39 every 100 ms for 10 seconds or due to any Engine Board command. When the jumper is missing, the feature shall not apply.

This feature is required to operate with legacy monitors, which requires activity on O39 within 2.0 seconds of application of AC service power to determine that the ATC is functioning. Without the muzzle jumper installed, the ATC boot-up time prevents the application software from performing this task in time. If the controller is truly malfunctioning, the activity on O39 ceases within 10 seconds.

More modern monitors have an adjustable power up time, allowing the controller to boot and the application software to start toggling O39 before the monitor fails.

Watchdog Circuit

A watchdog circuit shall be provided. It shall be enabled by the software during Power Up initialization with a value of 100 ms. Its enabled state shall be machine readable and reported in the status byte. Once enabled, the watchdog timer shall not be disabled without resetting the FCU. Failure of the FCU to reset the watchdog timer within the prescribed timeout shall result in a FCU hardware reset.

One kHz Reference

A synchronizable 1 kHz time reference shall be provided. It shall maintain a frequency accuracy of ± 0.01 percent (±0.1 counts per second).

Millisecond Counter

A 32-bit Millisecond Counter (MC) shall be provided for time stamping. Each 1 kHz reference interrupt shall increment the MC. The MC shall be initialized to zero at power up.

Communications Loss

The FCU shall enter the Communications Loss state following Power Up initialization or hardware reset, or if no valid command message has been received from the Engine Board within the previous 2.0 sec. During the Communications Loss state, all outputs shall be OFF and the communications loss bit “E” shall be set in the Request Module Status response.

The FCU shall exit the communications loss state only after a Request Module Status command has been received with the Communications Loss bit “E” set. All outputs shall remain OFF until a subsequent Set Outputs command has also been received.

Control Signals

LINESYNC and POWERDOWN Lines shall be isolated and routed to FCU for shut down functions. CPU_RESET and POWERUP Line signals shall be isolated and logically ORed to form NRESET. NRESET shall be used to reset FCU and other module devices.

Isolation

Isolation shall be provided between internal +5 VDC / DCGND1 and +12 VDC ISO / DCGND2. +12 VDC ISO shall be used for board power and external logic.

6.2.1.4 Buffers

A transition buffer shall be provided capable of holding a minimum of 1024 recorded entries. The transition buffer shall default to empty. There shall be two entry types: transition and rollover. The inputs shall be monitored for state transition. At each transition (if the input has been configured to report transition), a transition entry shall be added to the transition buffer. The MC shall be monitored for rollover. At each rollover transition (i.e., MC least significant 16 bits with hexadecimal value 0xFFFF transitioning to hexadecimal value 0x0000), a rollover entry shall be added to the transition buffer. Transition buffer blocks are sent to the Engine Board upon command. Upon confirmation of their reception, the blocks shall be removed from the transition buffer. The entry types are depicted as follows:

Input Transition Entry

Description	msb							lsb	Byte Number
Transition Entry Identifier	S	Input Number							1
Timestamp NLSB	X	x	x	x	x	X	x	x	2
Timestamp LSB	X	x	x	x	x	X	x	x	3

Millisecond Counter Rollover Entry

Description	Msb							lsb	Byte Number
Rollover Entry Identifier	1	1	1	1	1	1	1	1	1
Timestamp MSB	X	x	x	x	x	X	x	x	2
Timestamp NMSB	X	x	x	x	x	X	x	x	3

6.2.1.5 I/O Functions

Input Scanning

Input scanning shall begin at input 0 and proceed in ascending order to the highest input. Each complete input scan shall finish within 100 μs. Once sampled, the logic state of input shall be held until the next input scan. Each input shall be sampled 1,000 times per second. The time interval between samples shall be 1 ± 0.1 ms. If configured to report, each input that has transitioned since its last sampling shall be identified by input number, transition state and timestamp (at the time the input scan began) and shall be added as an entry to the transition buffer. If multiple inputs change state during one input sample, these transitions shall be entered into the input transition buffer by increasing input number. The MC shall be sampled within 10 μs of the completion of the input scan.

Data Filtering

If configured, the inputs shall be filtered by the FCU to remove signal bounce. The filtered input signals shall then be monitored for changes as noted. The filtering parameters for each input shall consist of “ignore input flag” and the On and Off filter samples. If the ignore input flag is set, no input transitions shall be recorded. The On and Off filter samples shall determine the number of consecutive samples an input must be on and off, respectively, before a change of state is recognized. If the change of state is shorter than the specified value, the change of state shall be ignored. The On and Off filter values shall be in the range of 0 to 255. A filter value of 0, for either or both values, shall result in no filtering for this input. The default values for input signals after reset shall be as follows:

Filtering Enabled
On and off filter values set to 5
Transition monitoring Disabled (Timestamps are not logged)

Outputs

Simultaneous assertion of all outputs shall occur within 100 μs. Each output shall be capable of being individually configured in state to ON, OFF, or a state synchronized with either phase of LINESYNC. During the Communications Loss state, all outputs shall be OFF.

Each output shall be controlled by the data and control bits in the Engine Board-PI/O frame protocol as follows:

Output Bit Translation

Case	Output Data Bit	Output Control Bit	Function
A	0	0	Output in the OFF state
B	1	1	Output is a square wave, synchronized to the LINESYNC signal. When LINESYNC is ON (1), the output is OFF, and when LINESYNC is OFF (0), the output is ON.
C	0	1	Output is a square wave, synchronized to the LINESYNC signal. When LINESYNC is ON (1), the output is ON, and when LINESYNC is OFF (0), the output is OFF
D	1	0	Output is in the ON state.

Output Stability

In Case A above, the corresponding output shall be turned OFF if previously ON and if previously OFF remain OFF until otherwise configured. For half-cycle switching (cases B and C), all outputs to be changed shall be changed within 50 µs after the corresponding LINESYNC transition and shall remain in the same state during the entire half cycle. In Case D above, the corresponding output shall be turned ON if previously OFF and if previously ON remain ON until otherwise configured. All outputs shall not change state unless commanded to do so.

6.2.1.6 Other Processor Functions

Interrupts

All interrupts shall be capable of asynchronous operation with respect to all processing and all other interrupts. A millisecond interrupt shall be activated by the 1 kHz reference once per ms. A timestamp rollover flag set by MC rollover shall be cleared only on command. A LINESYNC Interrupt shall be generated by both the 0-to-1 and 1-to-0 transitions of the LINESYNC signal. The LINESYNC interrupt shall monitor the MC interrupt and set the MC error flag if there has not been an interrupt from the 1 kHz source for 0.5 s (≥60 consecutive LINESYNC interrupts). The LINESYNC interrupt shall synchronize the 1 kHz time reference with the 0-to-1 transition of the LINESYNC signal once a second. A LINESYNC error flag shall be set if the LINESYNC interrupt has not successfully executed for 0.5 s or longer (≥500 consecutive millisecond interrupts).

Communication Service Routine

A low-level communication service routine shall be provided to handle reception, transmission, and communication faults.

Communication Processing

The task shall be to process the command messages received from the Engine Board, prepare and start response transmission. The response message transmission shall begin within 4 ms of the receipt of the received message. Any message processing shall be completed by the I/O module within 70 ms of a valid message being received from the Engine Board.

Input Processing

This task shall process the raw input data scanned in by the 1 ms interrupt routine, perform all filtering and maintain the transition queue entries.

6.2.1.7 Data Communications Protocols

6.2.1.7.1 Protocols

All communication with the Engine Board shall be via command-response protocol. The Engine Board shall always initiate the communication and should the command frame be incomplete or in error, no PI/O response shall be transmitted. The amount of bytes of a command or response is dependent upon the I/O module identification. The physical interface is not controlled by this standard, and interchangeability among vendors from PI/O to Engine Board is not intended.

Guidance:

For example, communications to PI/O module may be implemented via EIA-485 at 614 K bps, 5V TTL, or via a 1 GHz fiber channel provided all PI/O specifications herein are met, including:

- ***Command and Response Message Content***
- ***Command and Response Timing***
- ***Error Checking***

- ***Electrical Isolation***

Therefore, a frame is merely a field in the data stream, not related to the physical interface between the CPU and the PI/O, in contrast to the Model 2070 which requires serial communications via SP5 to its Field I/O. Other communications media may be used also.

6.2.1.7.1.1 Frame Types

The frame type shall be determined by the value of the first byte of the message. The command frames type values 112 – 127 and associated response frame type values 240 – 255 are allocated to the manufacturer diagnostics. All other frame types not called out are reserved. The command-response frame type values and message times shall be as follows:

Guidance:

The ATC Standard supports a range of I/O devices including the ITS Cabinet I/O components, the ATC Cabinet I/O components, and the 2070 FCU. The following table lists the valid commands, responses, and command recipients. The minimum and maximum message times specified are representative of a protocol using SDLC at 614,400 bps.

Table 7-1. Frame Types

Module Command	I/0 Module Response	Description	Minimum Message Time	Maximum Message Time
0-43	128-171	Reserved for NEMA TS2	N/A	N/A
44-48	172-176	Reserved	N/A	N/A
49	177	Request Module Status	250 μs	275 μs
50	178	MILLISECOND CTR. Mgmt.	222.5 μs	237.5 μs
51	179	Configure Inputs	344.5 μs	6.8750 ms
52	180	Poll Raw Input Data	317.5 μs	320 μs
53	181	Poll Filtered Input Data	317.5 μs	320 μs
54	182	Poll Input Transition Buffer	300 μs	10.25 ms
55	183	Command Outputs	405 μs	410 μs
56	184	Config. Input Tracking Functions	340 μs	10.25 ms
57	185	Configure Complex Output Functions (**deprecated**)	340 μs	6.875 ms
58	186	Configure Watchdog	222.5 μs	222.5 μs
59	187	Controller Identification (**deprecated**)	222.5 μs	222.5 μs
60	188	I/O Module Identification	222.5 μs	222.5 μs
61-62	189	Reserved (see Section 6.2.1.7.1.2)	N/A	N/A
63	191	Poll variable length raw input	317.5μs	320μs
64	192	Variable length command outputs	405 μs	410 μs
65	193	Reserved (see Section 6.2.1.7.1.2)	N/A	N/A
66	194	Reserved	N/A	N/A
67	195	Reserved (see Section 6.2.1.7.1.2)	N/A	N/A
68	196	I/O Input/Output Sizing	222.5 μs	222.5 μs
69	197	I/O Module Extended Description	222.5 μs	222.5 μs
70-80	198-208	Reserved	N/A	N/A
81-83	209-211	Reserved (see Section 6.2.1.7.1.2)	N/A	N/A
84-111	212-239	Reserved	N/A	N/A
112-127	240-255	Manufacturer Diagnostics	N/A	N/A

6.2.1.7.1.2 ATC Cabinet / ITS Cabinet Frames

Messages 61/189, 62/190 and 65/193 are for the ITS cabinet monitor unit. See the ITS Cabinet Standard, Section 4.4.16, Serial Bus #1 Frames, for the cabinet monitor system serial bus #1 command and response frames.

See the ATC 5301v02 Standard, Section 9.1.7, CMU2212 SB#1 Frame Types, for details of messages 62/190, 67/195, 81/209, 82/210 and 83/211.

Message 63 / Message 191 shall be the same as Message 52 / 180 except Byte 2 of Message 180 response shall denote the following number of input bytes. Message 64 / 192 shall be the same as Message 55 / 183 except Byte 2 of the Message 55 command shall denote the number of output data bytes plus the following output data.

6.2.1.7.2 Request Module Status

The command shall be used to request PI/O status information response. Command/response frames are as follows:

Request Module Status Command

Description	msb							lsb	Byte Number
(Type Number = 49)	0	0	1	1	0	0	0	1	1
Reset Status Bits	P	E	K	R	T	M	L	W	2

Request Module Status Response

Description	msb							lsb	Byte Number
(Type Number = 177)	1	0	1	1	0	0	0	1	1
System Status	P	E	K	R	T	M	L	W	2
SCC Receive Error Count	Receive Error Count								3
SCC Transmit Error Count	Transmit Error Count								4
Timestamp MSB	Timestamp MSB								5
Timestamp NMSB	Timestamp NMSB								6
Timestamp NLSB	Timestamp NLSB								7
Timestamp LSB	Timestamp LSB								8

6.2.1.7.2.1 Request Module Status Response

The response status bits are defined as follows:

P -Indicates FCU hardware reset
E -Indicates a communications loss of greater than 2 seconds
M -Indicates an error with the MC interrupt
L -Indicates an error in the LINESYNC
W -Indicates that the FCU has been reset by the watchdog
R -Indicates that the receive error count byte has rolled over
T -Indicates that the transmit error count byte has rolled over
K -Indicates the Datakey has failed or is not present

6.2.1.7.2.2 Bit Information

Each of these bits shall be individually reset by a “1” in the corresponding bit of any subsequent request module status frame, and the response frame shall report the current status bits. The serial communications controller error count bytes shall not be reset. When a count rolls over (255 - 0), its corresponding roll-over flag shall be set.

6.2.1.7.3 MC Management Frame

MC management frame shall be used to set the value of the MC. The “S” bit shall return status “0” on completion or “1” on error. The 32-bit value shall be loaded into the MC at the next 0-1 transition of the LINESYNC signal. The frames are as follows:

Millisecond Counter Management Command

Description	msb							lsb	Byte Number
(Type Number = 50)	0	0	1	1	0	0	1	0	1
New Timestamp MSB	X	x	x	x	x	x	x	x	2
New Timestamp NMSB	X	x	x	x	x	x	x	x	3
New Timestamp NLSB	X	x	x	x	x	x	x	x	4
New Timestamp LSB	X	x	x	x	x	x	x	x	5

Millisecond Counter Management Response

Description	msb							lsb	Byte Number
(Type Number = 178)	1	0	1	1	0	0	1	0	1
Status	0	0	0	0	0	0	0	S	2

6.2.1.7.4 Configure Inputs

The configure inputs command frame shall be used to change input configurations. The command-response frames are as follows:

Configure Inputs Command

Description	msb							lsb	Byte Number
(Type Number = 51)	0	0	1	1	0	0	1	1	1
Number of Items (n)	n	n	n	n	n	n	n	n	2
Item # - Byte 1	E	Input Number							3(I-1)+3
Item # - Byte 2	Leading edge filter (e)								3(I-1)+4
Item # - Byte 3	Trailing edge filter (r)								3(I-1)+5

Configure Inputs Response

Description	msb							lsb	Byte Number
(Type Number = 179)	1	0	1	1	0	0	1	1	1
Status	0	0	0	0	0	0	0	S	2

Block field definitions shall be as follows:

E - Ignore Input Flag.
"1" = do not report transitions for this input,
"0" = report transitions for this input

e - A one-byte leading edge filter specifying the number of consecutive input samples which must be "0" before the input is considered to have entered to "0" state from "1" state (range 1 to 255, 0 = filtering disabled)

r - A one-byte trailing edge filter specifying the number of consecutive input samples which must be "1" before the input is considered to have entered to "1" state from "0" state (range 1 to 255, 0 = filtering disabled)
S - return status S = “0” on completion or “1” on error

6.2.1.7.5 Poll Raw Input Data

The poll raw input data frame shall be used to poll the PI/O for the current unfiltered status of all inputs. The response frame shall contain 8 or 15 bytes of information indicating the current input status. The frames are as follows:

Poll Raw Input Data Command

Description	msb							lsb	Byte Number
(Type Number = 52)	0	0	1	1	0	1	0	0	1

Poll Raw Input Data Response

Description	msb							lsb	Byte Number
(Type Number = 180)	1	0	1	1	0	1	0	0	1
Inputs I0 (lsb) to I7 (msb)	x	x	x	x	x	x	x	x	2
Inputs I8 to I119	x	x	x	x	x	x	x	x	3 to 16
Timestamp MSB	x	x	x	x	x	x	x	x	17
Timestamp NMSB	x	x	x	x	x	x	x	x	18
Timestamp NLSB	x	x	x	x	x	x	x	x	19
Timestamp LSB	x	x	x	x	x	x	x	x	20

6.2.1.7.6 Poll Filtered Input Data

The poll filtered input data frame shall be used to poll the PI/O for the current filtered status of all inputs. The response frame shall contain 8 bytes (-332 cabinet) or 15 bytes (NEMA TS 2 Type 2 cabinet) of information indicating the current filtered status of the inputs. Raw input data shall be provided in the response for inputs that are not configured for filtering. The frames are as follows:

Poll Filter Input Data Command

Description	msb							lsb	Byte Number
(Type Number = 53)	0	0	1	1	0	1	0	1	1

Poll Filter Input Data Response

Description	msb							lsb	Byte Number
(Type Number = 181)	1	0	1	1	0	1	0	1	1
Inputs I0 (lsb) to I7 (msb)	x	x	x	x	x	x	x	x	2
Inputs I8 to I119	x	x	x	x	x	x	x	x	3 to 16
Timestamp MSB	x	x	x	x	x	x	x	x	17
Timestamp NMSB	x	x	x	x	x	x	x	x	18
Timestamp NLSB	x	x	x	x	x	x	x	x	19
Timestamp LSB	x	x	x	x	x	x	x	x	20

6.2.1.7.7 Poll Input Transition Buffer

The poll input transition buffer frame shall poll the PI/O for the contents of the input transition buffer. The response frame shall include a three-byte information field for each of the input changes that have occurred since the last interrogation. The frames are as follows:

Poll Input Transition Buffer Command

Description	msb							lsb	Byte Number
(Type Number = 54)	0	0	1	1	0	1	1	0	1
Block Number	X	x	x	X	x	X	x	x	2

Input Transition Buffer Response

Description	Msb							lsb	Byte Number
(Type Number = 182)	1	0	1	1	0	1	1	0	1
Block Number	x	X	x	X	x	X	x	x	2
Number of Entries (N)	x	X	x	X	x	X	x	x	3
Item I	S	Input Number							3(I-1)+4
Item I Timestamp NLSB	x	X	x	X	x	X	x	x	3(I-1)+5
Item I Timestamp LSB	x	X	x	X	x	X	x	x	3(I-1)+6
Status	0	0	0	0	C	F	E	G	3(N-1)+7
Timestamp MSB	x	X	x	X	x	X	x	X	3(N-1)+8
Timestamp NMSB	x	X	x	X	x	X	x	X	3(N-1)+9
Timestamp NLSB	x	X	x	X	x	X	x	X	3(N-1)+10
Timestamp LSB	x	X	x	X	x	X	x	X	3(N-1)+11

6.2.1.7.7.1 State Transitions

Each detected state transition for each active input (see configuration data) is placed in the queue as it occurs. Bit definitions are as follows:

S - Indicates the state of the input after the transition

C - Indicates the 255 entry buffer limit has been exceeded

F - Indicates the buffer has overflowed

G - Indicates the requested block number is out of monotonic increment sequence

E - Same block number requested, E is set in response

6.2.1.7.7.2 Block Number

The block number byte is a monotonically increasing number incremented after each command issued by the Engine Board. When the PI/O module receives this command, it shall compare the associated block number with the block number of the previously received command. If it is the same, the previous buffer shall be re-sent to the Engine Board and the 'E' flag set in the status response frame. If it is not equal to the previous block number, the old buffer shall be purged and the next block of data sent. If the block number is not incremented by one, the status G bit shall be set. The block number received becomes the current number (even if out of sequence). The block number byte sent in the response block shall be the same as that received in the command block. Counter rollover shall be considered as a normal increment.

6.2.1.7.8 Set Outputs

The set outputs frame shall be used to command the PI/O to set the outputs according to the data in the frame. If there is any error configuring the outputs, the 'E' flag in the response frame shall be set to '1'. If the LINESYNC reference has been lost, the 'L' bit in the response frame shall be set. Loss of LINESYNC reference shall also be indicated in system status information. The output bytes depend upon field I/O module. These command and response frames are as follows:

Set Outputs Command

Description	msb							lsb	Byte Number
(Type Number = 55)	0	0	1	1	0	1	1	1	1
O0 (lsb) to O7 (msb) Data	X	x	x	X	x	X	x	x	2
O8 to O103 Data	X	x	x	X	x	X	x	x	3 to 14
O0 (lsb) to O7 (msb) Control	X	x	x	X	x	X	x	x	15
O8 to O103 Control	X	x	x	X	x	X	x	x	16 to 27

Set Outputs Response

Description	msb							lsb	Byte Number
(Type Number = 183)	1	0	1	1	0	1	1	1	1
Status	0	0	0	0	0	0	L	E	2

6.2.1.7.9 Configure Input Tracking Functions

The configure input tracking functions frame shall be used to configure outputs to respond to transitions on a specified input. Each output number identified by item number shall respond as configured to the corresponding input number identified by the same item number. Input to output mapping shall be one to one. If a command results in more than eight input tracking outputs being configured, the response V bit shall be set to ‘1’ and the command shall not be implemented. The command and response frames are as follows:

Configure Input Tracking Functions Command

Description	msb							lsb	Byte Number
(Type Number = 56)	0	0	1	1	1	0	0	0	1
Number of Items (N)	Number of Items								2
Item I - Byte 1	E	Output Number							2(I-1)+3
Item I - Byte 2	I	Input Number							2(I-1)+4

Configure Input Tracking Functions Response

Description	msb							lsb	Byte Number
(Type Number = 184)	1	0	1	1	1	0	0	0	1
Status	0	0	0	0	0	0	0	V	2
Timestamp MSB	x	x	x	X	x	x	x	x	3
Timestamp NMSB	x	x	x	X	x	x	x	x	4
Timestamp NLSB	x	x	x	X	x	x	x	x	5
Timestamp LSB	x	x	x	X	x	x	x	x	6

6.2.1.7.9.1 Configure Input Tracking Functions Response

Definitions are as follows:

E '1' - Enable input tracking functions for this output

'0' - Disable input tracking functions for this output

I '1' - The output is OFF when input is ON, ON when input OFF

'0' - The output is ON when input is ON, OFF when input is OFF

V '1' - The max. number of 8 configurable outputs exceeded

'0' - No error

Number of Items - The number of entries in the frame. If zero, all outputs currently configured for input tracking shall be disabled.

6.2.1.7.9.2 Timestamp

The timestamp value shall be sampled prior to the response frame.

6.2.1.7.9.3 Output Updates

Outputs which track inputs shall be updated no less than once per ms. Input to output signal propagation delay shall not exceed 2 ms.

6.2.1.7.9.4 Number of Item Field

The “number of item” field is valid from 0 to 16 (the most that is sent at one time is eight enables and eight disables). If processing a command resulting in more than eight input tracking functions being enabled, none of the command shall be implemented and response message “V” bit set to 1. If an invalid output or input number is specified for a function, the FCU software shall not do that function definition. It shall also not be counted toward the maximum of eight input tracking function allowed. The rest of the message shall be processed. When an input tracking function is disabled, the output is set according to the most recently received Set Outputs Command. When

an input tracking function for an output is superseded (redefined as either another input tracking function or as a complex output function) nothing shall be done with the output. The most recent value remains until the new function changes it.

6.2.1.7.10 Configure Complex Output Functions (deprecated)

6.2.1.7.11 Configure Watchdog

Configure watchdog represents a legacy command and response that is not required and is included here for reference. If the command frame is received then it shall be responded to, but the timeout value in the command frame shall be ignored and the manufacturer selected timeout value shall be maintained.

The configure watchdog frames were previously used to change the software watchdog timeout value. The command and response frames were as follows:

Configure Watchdog Command

Description	msb							lsb	Byte Number
(Type Number = 58)	0	0	1	1	1	0	1	0	1
Timeout Value	x	X	x	x	x	x	X	x	2

Configure Watchdog Response

Description	msb							lsb	Byte Number
(Type Number = 186)	1	0	1	1	1	0	1	0	1
Status	0	0	0	0	0	0	0	Y	2

6.2.1.7.11.1 Timeout Value

The timeout value shall be in the range between 10 to 100 ms. If the value is lower than 10, 10 shall be assumed. If the value is greater than 100, 100 shall be assumed.

6.2.1.7.11.2 Time-Out Change

On receipt of this frame, the watchdog time-out value shall not be changed and the “Y” bit shall be set. The response frame bit (Y) shall indicate a '1' if the watchdog had been previously set and a '0' if not.

6.2.1.7.12 Controller Identification (deprecated)

6.2.1.7.13 I/O Module Identification

Guidance:

This is a legacy Model 2070 message command / response for Field I/O modules with Datakey resident.

The I/O Module Identification command frame shall be used to request the I/O identification value response of:

1. Model 332 PI/O
2. NEMA TS-2 Type 2 PI/O per TEES / Model 2070 (includes A, B, C, and D connectors)
3. NEMA TS-2 Type 1 PI/O with standard A connector only

4. NEMA TS-2 Type 2 PI/O with standard A, B, and C connectors only
5. NEMA TS-2 Type 2 PI/O with standard A, B, and C plus custom additional connector(s)
6. Custom PI/O

7-15. Reserved

The remaining I/O identification values (16-255) are set aside for manufacturer or agency specific configurations which are not specified herein. However, in order for the device to be considered conformant to this standard, it must support one of the options listed above (1-4) in addition to the custom I/O connector(s); this has been done to allow agencies to procure ATC units that provide the advanced capabilities and conform to the balance of this standard while still supporting agency specific legacy parallel I/O or D connector(s). Please note that the SDLC address for these I/O modules is 20. The command and response frames are as follows:

I/O Module Identification Command

Description	msb							lsb	Byte Number
(Type Number = 60)	0	0	1	1	1	1	0	0	1

I/O Module Identification Response

Description	msb							lsb	Byte Number
(Type Number = 188)	1	0	1	1	1	1	0	0	1
PI/O I D byte	x	x	x	x	x	x	x	x	2

6.2.1.7.14 I/O Input/Output Sizing

The I/O Input/Output Sizing command frame shall be used to request the number of inputs and outputs supported by the I/O module. Each I/O module is limited to a maximum of 128 inputs (numbered I0-I127) and a maximum of 256 outputs (O0-O255) with only outputs O0-O127 being available for use with input tracking and complex I/O commands. Please note that the SDLC address for these I/O modules is 20. The command and response frames are as follows:

I/O Module Identification Command

Description	msb							lsb	Byte Number
(Type Number = 68)	0	0	1	1	1	1	0	0	1

I/O Module Identification Response

Description	msb							lsb	Byte Number
(Type Number = 196)	1	0	1	1	1	1	0	0	1
Number of Inputs Supported	x	x	X	x	x	x	x	x	2
Number of Outputs Supported	x	x	X	x	x	x	x	x	3

6.2.1.7.15 I/O Module Extended Description

The I/O Module Extended Description command frame shall be used to request details of the I/O module. These details include the Manufacturer NTCIP ID as assigned by the NTCIP Coordinator and a Manufacturer Specific Code (proprietary) which can provide additional information concerning the actual I/O module provided. Use of this command frame is not required by an application if the Module ID return code (frame = 60) is 1, 2, 3, or 4 (i.e. module defined completely by a standard). If this command frame is received by one of these standard modules, then it shall

return the Manufacturer NTCIP ID and 0 (zero) for the Manufacturer Specific Code. Please note that the SDLC address for these I/O modules is 20. The command and response frames are as follows:

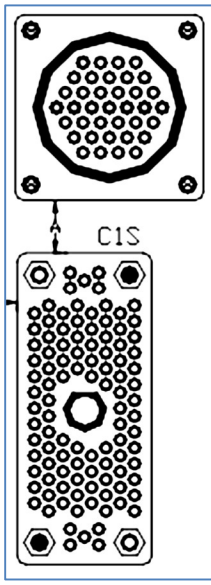
I/O Module Extended Description Command

Description	msb							lsb	Byte Number
(Type Number = 69)	0	0	1	1	1	1	0	0	1

I/O Module Extended Description Response

Description	msb							lsb	Byte Number
(Type Number = 197)	1	0	1	1	1	1	0	0	1
Manufacturer NTCIP ID	x	x	X	x	x	x	x	x	2
Manufacturer Specific Code	x	x	X	x	x	x	x	x	3

6.2.1.8 Mechanical Details



NOTES:

1. C1S Dark Circles denote guide pin locations
2. C1S Open Circles denote guide socket locations
3. Dimension “A” shall be 0.5” minimum
4. C1S shall be M104 type
5. C11S shall be 37 pin circular plastic type
6. C1S & C11S pin assigned per 2070 ATC Standard

Figure 6-1. C1S and C11S Pin Configuration: Refer to Model 2070 Standard.

Connectors C1S and C11S shall be mechanically secured to the chassis or frame of the unit or assembly.

6.2.2 Parallel Connection to NEMA TS-1 or TS-2 Type 2 Cabinets

6.2.2.1 Description

This PI/O shall consist of a microcontroller, parallel input / output ports, field connectors and communications circuits. It is similar in function to the Model 332 PI/O, except it provides more inputs and outputs via different physical connectors.

Alternatively, the microcontroller may be separate and distinct from the Parallel I/O Module as long as all other requirements of this Standard are met, including isolation requirements for parallel I/O.

6.2.2.2 Front Panel

The front panel shall be furnished with the following:

- Incoming VAC fuse holder
- Three NEMA defined connectors: A, B & C
- Optional manufacturer/application specific D connector may be provided

6.2.2.3 Functional Requirements Exceptions

This PI/O shall meet all requirements identified above except that:

- The NEMA-defined inputs and outputs shall be provided
- Optional manufacturer/application-specific D connector inputs and outputs may be provided
- Specification for inputs applies, except the voltage is +24 V in lieu of +12 V, and logical "0", exceeds 16.0 VDC.
- There is no "muzzle jumper" and O39 is used for "phase 8 ped clear" instead of as the toggling watchdog output required for the 332 cabinet.

6.2.2.4 Fault Monitor and Voltage Monitor

The NEMA Fault Monitor and Voltage Monitor circuits shall be implemented using the following mechanism:

Watchdog

- The Engine Board shall change the state of O86 in the Set Outputs Command (Section 6.2.1.7.8).at a 1hz rate. This is the flashing logic output bit.
- The field IO processor shall have a "watchdog strobe" output pin that follows the state of O86.
- The field IO processor's "watchdog strobe" pin shall be connected to the "strobe" input an external Watchdog Timer (WDT) circuit.
- The watchdog timer shall cause its output to go into the FALSE state (logic 0) if the state of O86 has not changed for 2 ± 0.1 s.

Fault Monitor (Per NEMA TS2-2003, Section 3)

- The Fault Monitor output shall be an open collector or open drain output that is maintained TRUE (Low State) as long the following requirements are met:
    - All voltages required for normal operation are at predetermined levels
    - The watchdog timer output is in the TRUE state (logic 1).
- The Fault Monitor output shall meet the electrical characteristics defined in Section 3.3.5.1.4 of NEMA TS2-2003
- Logic ground for the Fault Monitor output shall be DCGND2 of the ATC.
- The Fault Monitor output will also be FALSE (High State) during ATC generated flash modes as defined in Section 3 of NEMA TS2-2003
- O78 in the Set Outputs Command (Section 6.2.1.7.8) shall be used for Engine Board control of Fault Monitor.

Voltage Monitor (Per NEMA TS2-2003, Section 3)

- The Voltage Monitor output shall be an open collector or open drain output that is maintained TRUE (Low State) as long the following requirements are met:
    - Primary +5VDC supply is within +4.775 to +5.225 +- 0.025VDC
    - All other voltages required for normal operation are at predetermined levels
    - The watchdog timer output is in the TRUE state (logic 1).
    - Bit O79 in the Set Outputs Command (Section 6.2.1.7.8) is set to 0.
- O79 in the Set Outputs Command (Section 6.2.1.7.8) shall be used for Engine Board control of Voltage Monitor.
- The Voltage Monitor output shall meet the electrical characteristics defined in Section 3.3.5.1.4 of NEMA TS2-2003
- Logic ground for the Voltage Monitor output shall be DCGND2 of the ATC.
- The Voltage Monitor output shall also be False (High State) during ATC generated flash modes as defined in Section 3 of NEMA TS2-2003
- The Voltage Monitor output shall operate concurrently with Fault Monitor except during periods of No Fault Flash, see Section 3 of NEMA TS2-2003 for further details

Monitor Output Power Up Conditions

Engine Board / field IO processor operation at POWERUP shall be as follows:

- Field IO processor sets its Comm Loss Flag
- Fault & Voltage Monitor outputs set to FALSE (i.e., inactive)
- Engine Board sends the CPU Request Module Status Command with “E” bit set to field IO processor to clear the Comm Loss Flag
- Field IO processor responds to Engine Board with “E” bit reset
- Before the Comm Loss timer expires, the Engine Board must send the Set Outputs Command. Bits O78 and O79, of the Set Outputs Command, shall be set to “0” which shall cause the field IO Processor port pins assigned for Fault Monitor and Voltage Monitor outputs to go to their TRUE (i.e., active) state
- At this point, the signal states defined in the Set Outputs Command message shall be transferred to the output connectors
- Any number of other messages may be sent between the Module Status Command and Set Outputs Command
- If the above message sequence is not followed, Comm Loss Flag shall be set (or remain) and Voltage Monitor & Fault Monitor shall retain the FALSE (inactive) output state

Communications Loss

During the Communication Loss state, all outputs shall be OFF, and the Fault Monitor and Voltage Monitor states shall be set to FALSE (inactive).

6.2.2.5 Mechanical Details

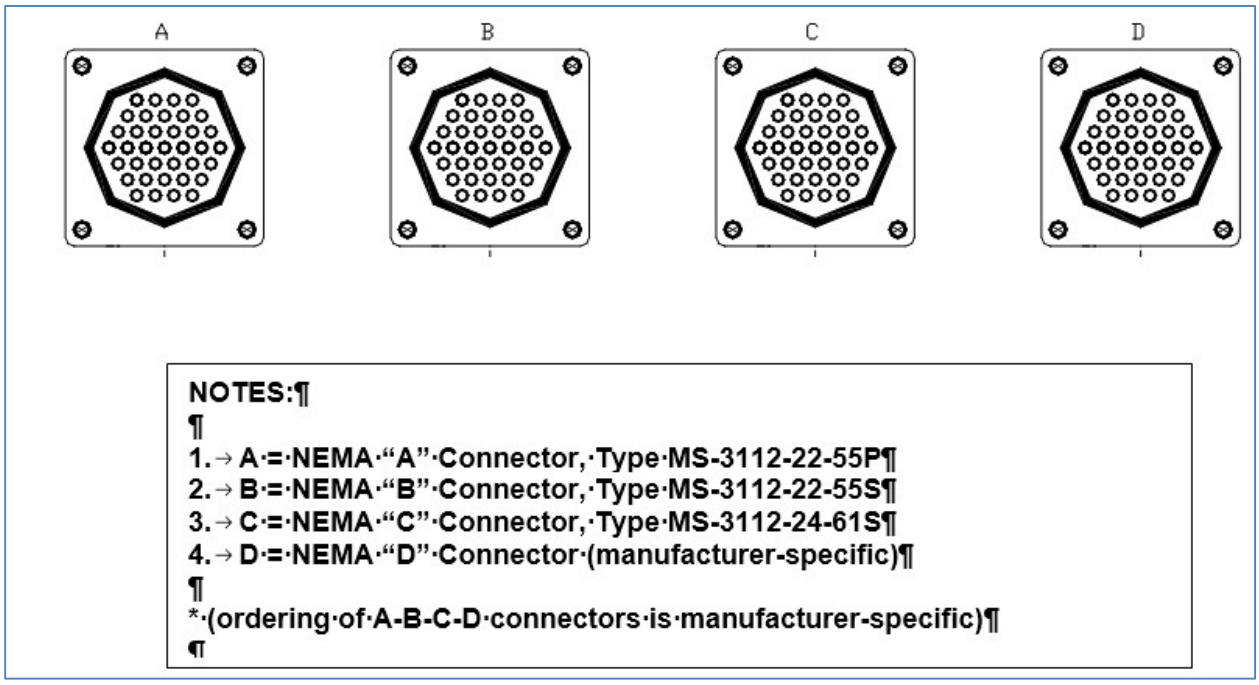


Figure 6-2. Connector Diagram.

Connector A, B, C Pin Configurations: Refer to NEMA TS-2 Specification

Connector D Pin Configuration: (manufacturer-specific)

NEMA A-B-C-D connectors shall be mechanically secured to the chassis or frame of the unit or assembly.

6.2.3 Connection to NEMA TS-2 Type 1 Cabinets

Description

The NEMA TS 2 Type-1 compatible I/O interface provides:

- AC Power to the ATC,
- Fault Monitor output to the NEMA TS 2 Malfunction Monitor Unit (MMU).
- Logic Ground to the cabinet will be DCGND2 of the ATC

NEMA TS 2 Port 1 interface for SDLC communications

Front Panel I/O

As a minimum, the front panel shall be furnished with the following:

- Incoming VAC fuse protection
- One NEMA TS 2 Type-1 Connector, A

One NEMA Port 1 connector

Port 1 Connector

The Port 1 connector shall meet the requirements for NEMA TS 2 Port 1. The DCD signal of the serial port used for the TS 2 Type 1 communications interface shall be allocated to Port 1 disable where 0VDC input equals DCD inactive (False).

Parallel Connector A

The connector, pin assignment and electrical characteristics shall meet the requirements for NEMA TS 2 Type-1 Port A.

Service Power Connection

Incoming AC Power is derived from Connector A, Pin C (AC+), Pin A (AC-) and Pin H (Equipment Ground).

Fault Monitor

FCU output O78 shall drive an open collector transistor whose output shall be routed to Connector A Pin F for use as a FAULT MONITOR Output. The transistor shall be capable of sinking 200 mA at 30 VDC. The Fault Monitor pin shall operate per Section 6.2.2.4 Fault Monitor and Voltage Monitor with the exception that all references to Voltage monitor shall be omitted.

6.3 Serial Input / Output

Guidance: The traditional (Model 2070) use of the serial ports is as follows:

SP1: External communications (NEMA Port 2)

SP2: External communications (NEMA Port 3)

SP3: ITS Cabinet / ATC Cabinet SB #2, or external communication (NEMA Port 1)

SP4: O/S console and asynchronous communications

SP5: Field I/O communications module or ITS Cabinet / ATC Cabinet SB #1

SP6: User Interface

SP8: Communications or ITS Cabinet / ATC Cabinet SB #1 via C13S connector

SPI: Portable Memory Device (Datakey)

ENET1 Ethernet for Network

Guidance: The planned use of the added (not on Model 2070) serial ports is as follows:

USB Removable Memory Device

ENET2 Ethernet for Local Cabinet (Expansion Rack, Upload to Laptop)

Refer to Section 4 for a description of each serial port operation.

The ATC shall provide two internal 100BASE-TX store and forward Ethernet switches per the IEEE 802.3 specification for 10/100 Mbps signaling with CSMA/CD over two pairs of Category 5 UTP or STP wire.

Alternatively, a single Ethernet switch with port-based VLAN capability, configured to provide two independent VLANs (ENET1/ENET2), may be provided.

Ethernet switch logical ports shall be allocated as follows:

Port 1: Internal 10/100 Mbps Port to Engine Board ENET1

Port 2: External 10/100 Mbps Port (Typically to Network Backbone)

Port 3: External 10/100 Mbps Port (Typically for Network Diagnostics)

Port 4: Internal 10 Mbps Port to Communication Interface Slot A2

(not required for ATCs with no Communication Interface Slots)

Port 5: Internal 10/100 Mbps Port to Engine Board ENET2

Port 6: External 10/100 Mbps Spare Port (Typically for Controller Expansion)

Port 7: External 10/100 Mbps Port (Typically for Controller Diagnostics)

Port 8: Internal 10 Mbps Port to Communications Interface Slot A1

(not required for ATCs with a single Communication Interface Slot)

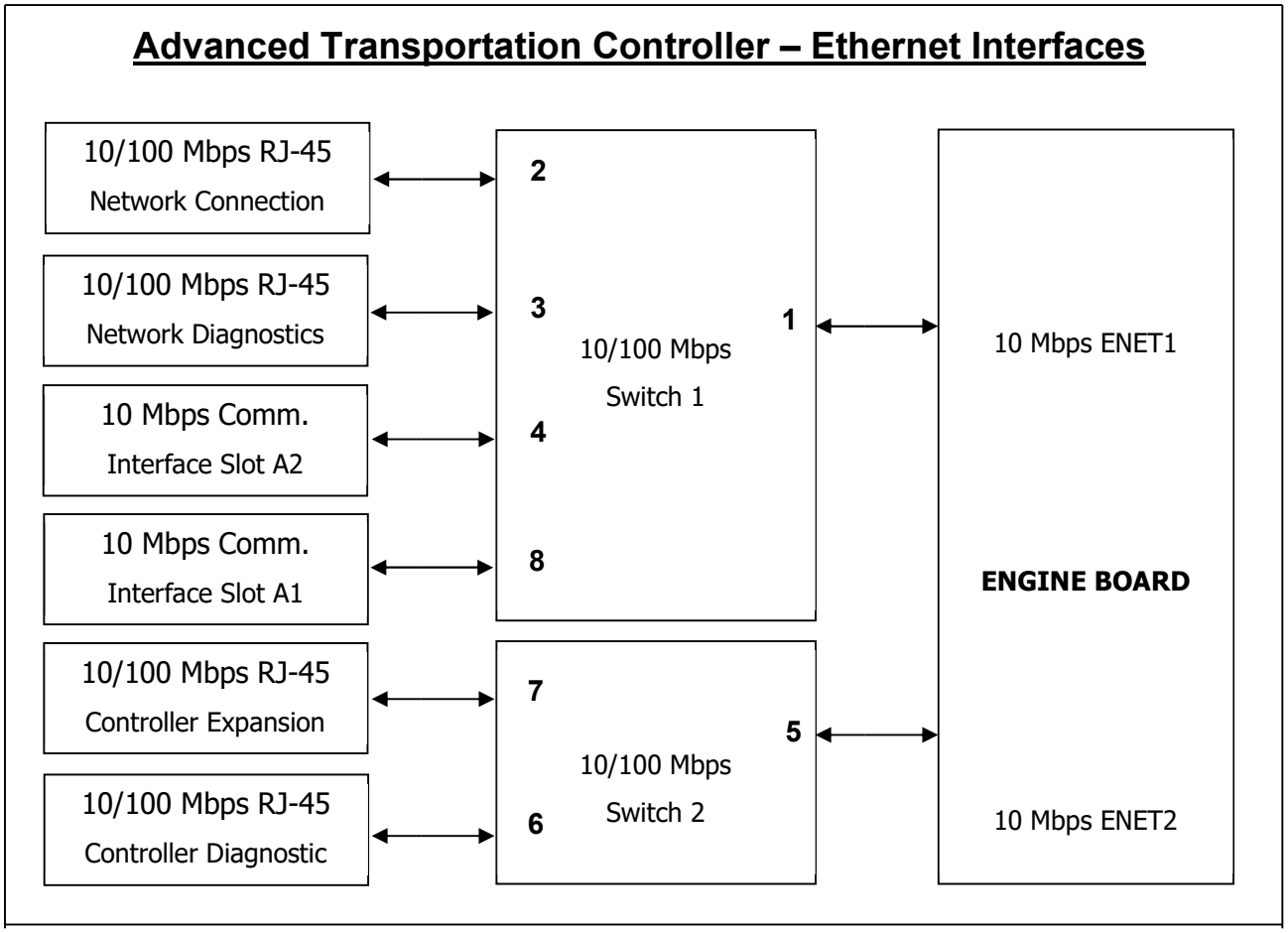


Figure 6-3. Details of Ethernet Switch Connections, Typical Use.

Typical Switch Logical Port Use:

Switch 1:

Port 1 connects to Engine Board ENET1, which handles the network traffic. Although ENET1 is not expected to handle 100 Mbps data streams, the ATC may find itself connected to a 100 Mbps network. Switch 1 handles the speed conversion, as well as provides two RJ-45 Ethernet connectors, Port 2 and Port 3, and internal expansion connections on Port 4 and Port 8.

Port 2 provides a permanent connection to the network backbone for network communications, such as NTCIP.

Port 3 is configured as an uplink for use with an Ethernet cable via the auto-MDIX capabilities of the port. This eliminates the need to disconnect the ATC from the network to connect a laptop computer for network diagnostics, or to connect other cabinet equipment to the network.

Port 4 is configured as a 10 BASE TX internal expansion port that is connected to the Ethernet signals on Communication Interface Slot A2. A communication interface module may connect directly to ENET1 via this port. This port is not required if no communications interface slots exist in the ATC.

Port 8 is configured as a 10 BASE TX internal expansion port that is connected to the Ethernet signals on Communication Interface Slot A1. A communication interface module may connect directly to ENET1 via this port. This port is not required if only one communications interface slot exists in the ATC.

Switch 2:

Port 5 connects to Engine Board ENET2, which communicates to local cabinet devices. Although ENET2 is not expected to handle 100 Mbps data streams, the ATC may find itself connected to a 100 mbps network. Switch 2 handles the speed conversion, as well as provides two RJ-45 Ethernet connectors, port 6 and port 7.

Port 7 provides a connection for external ATC expansion, such as connection to a network interface card residing in a computer rack (VME, for example). This provides a standard method to connect parallel devices such as analog to digital converters, disk storage and multiple computer boards, without specifying a particular computer bus.

Port 6 is configured as an uplink for use with an Ethernet cable via the auto-MDIX capabilities of the port. This eliminates the need to disconnect the ATC from the network to connect a laptop computer for controller diagnostics, or to load new controller software.

6.4 Isolation Requirements

The ATC shall maintain optical or magnetic isolation of signals from the ATC to field devices as described in the following paragraphs.

The need for isolating field connections is twofold:

- Isolation prevents electrical surge damage to the Engine Board due to lightning or nearby electrical equipment picked up by the field wires. Although the I/O circuitry may be damaged, isolation protects the Engine Board, allowing malfunction to be logged and reported.
- Isolation prevents “ground loops,” ensuring equipment is grounded at one place only. For example, desktop computers internally connect logic common to equipment ground. Attaching an un-isolated ATC serial port to a desktop PC will ground the ATC through the Engine Board or field I/O, creating ground loop current through the serial cable, resulting in data transmission errors.

Field Device Definition and Exceptions:

- Isolation is not required when the serial port is connected to another integrated ATC assembly. For example, an integrated front panel is considered to be part of the ATC, meaning the SP6 connection to the front panel device and any LCD heater installed need not be electrically isolated from the Engine Board. However, a removal front panel or SP6 external connector such as C60P is not considered to be part of the ATC and isolation from the Engine Board is required for all externally-accessible signals.

- SP4 (C50S/C50J) must always be isolated from the Engine Board; however, for non-integrated front panels, SP4 may share the same isolated ground reference as SP6 and any LCD heater.

Isolation Methods:

Electrical isolation shall be implemented via optical or magnetic methods. Capacitive isolation is not allowed, as capacitors act as high-pass filters, passing high-frequency common mode surges to the Engine Board. Optical isolators and magnetic transformers effectively block common mode surges at all frequencies.

6.4.1 Engine Board Isolation

The Engine Board shall be electrically isolated from all serial and parallel field connections with the exception of a front panel device if it is integrated within the ATC (for purposes of this standard, a removable front panel is not considered integrated). Engine board signals need not be electrically isolated from one another. The Engine Board, as well as +5 VDC, +12 VDC and -12 VDC are referenced to the minus of the controller power supply (DCGND1).

6.4.2 Parallel I/O Isolation

Every parallel input and output shall be electrically isolated from the Engine Board, and from each serial I/O signal. Parallel inputs and outputs need not be electrically isolated from one another. All parallel inputs and outputs, as well as +12 VDC ISO are referenced to DCGND2. External cabinet power supply (if used) shall be referenced to DCGND2.

6.4.3 Serial I/O Isolation

Serial I/O shall be isolated in the following manner:

- Serial inputs and outputs connected to field devices shall be electrically isolated from the Engine Board and from all parallel inputs and outputs
- Signals within a serial port connector need not be electrically isolated from one another
- Signals of different serial port connectors shall be electrically isolated from one another when on different communication interface modules
- Each serial port shall be referenced to the attached equipment
- Pictorially, dashed lines depict the isolation boundaries. Ports need not be electrically isolated from each other when located on the same communications interface module
- SP1, 2, 3, 4 are isolated on the communications interface modules

Ethernet ports are not shown, as they are magnetically isolated on the Engine Board

Guidance: DC Ground 2 will be used as the ground reference for both NEMA Port 1 and any NEMA parallel I/O, including the Type 1 Fault Monitor.

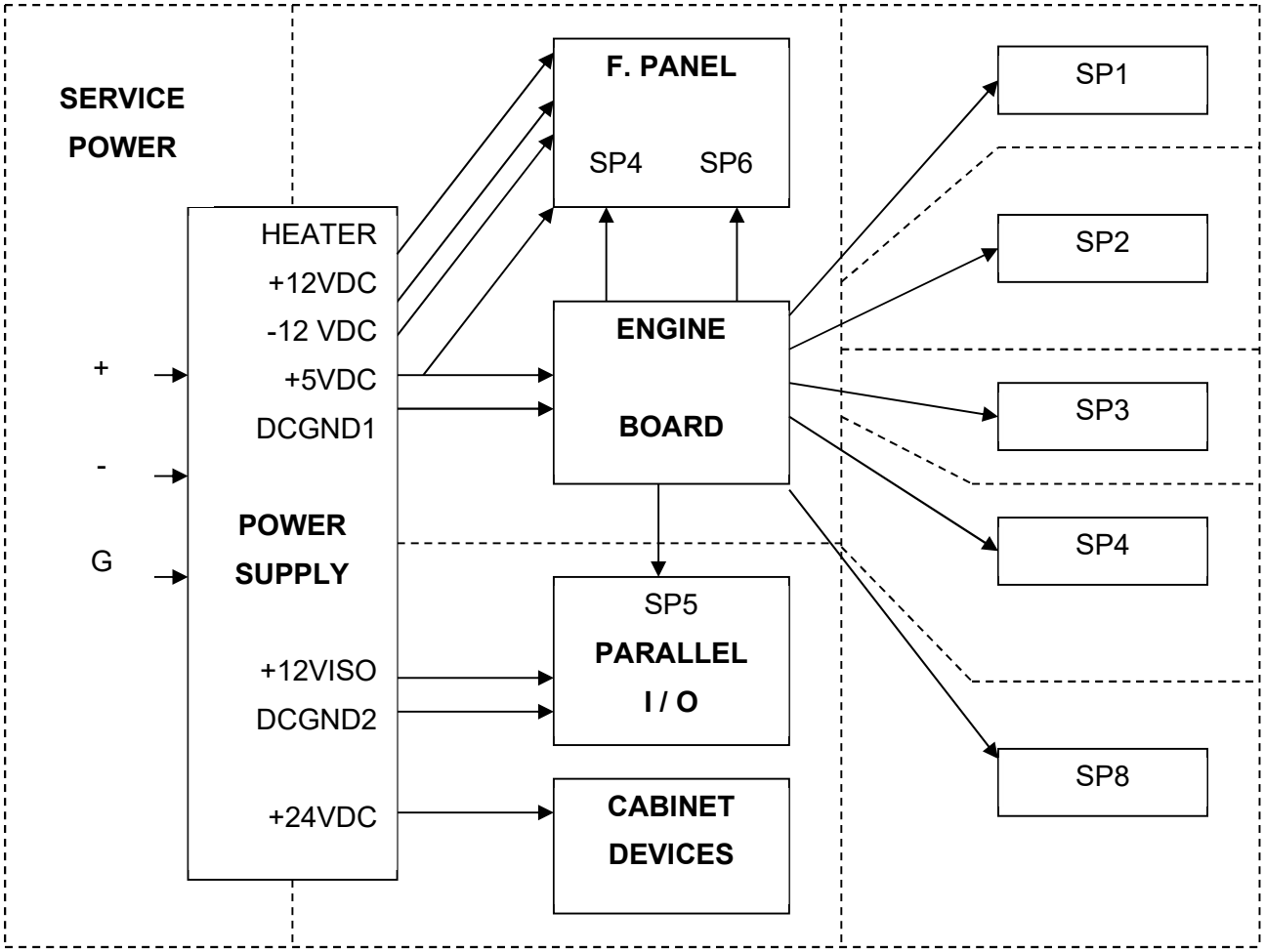


Figure 6-4. Typical Isolation Boundaries.

6.5 Electrostatic Discharge Protection Requirements

The ATC shall provide electrostatic discharge (ESD) protection to IEC 61000-4-2 (ESD) at ±15kV (air) and at ±8kV (contact) for the typical user communication ports C50 (SP4), C60 (SP6), USB port(s), and laptop Ethernet ports (Hub 1, Port 3 and Hub 2, Port 6 only).

7 ENVIRONMENTAL AND TEST PROCEDURES

NOTE: All references in this section to NEMA TS2 explicitly refer to the NEMA Standards Publication TS 2-2003 v02.06, *Traffic Controller Assemblies with NTCIP Requirements.*

7.1 General

This section establishes the environmental and operational conditions for a First Article ATC unit and defines the minimum test procedures which may be used to demonstrate conformance of an ATC unit with the provisions of the Standard.

Software shall be available from the manufacturer that contains a set of test programs to facilitate testing. This software shall be capable of running individual tests or combinational tests. The combinational tests shall include a single menu function that binds all of the tests into a single module. Tests may be run either from the front panel or by an external serial port. These tests shall include but are not limited to the items in the following outline:

- A testing program shall contain the following:
    1. Introduction to the Test
    2. Installation Instructions
    3. Starting the Software
    4. Running Individual Tests
    5. Test Suite Tree for combination tests
- Individual Processor tests shall include the following:
    1. Volatile / Dynamic RAM (DRAM) Test
    2. Short-Term Non-Volatile Memory (SRAM) Test
    3. Long-Term Non-Volatile Memory (FLASH) Read/Write Test
    4. Datakey Tests
    5. USB Tests
    6. Ethernet Tests
- Front Panel (when used) tests shall include the following:
    1. Display Tests
    2. Keyboard Tests
- I/O tests shall include the following:
    1. I/O Loop Back Tests
- Asynchronous/Synchronous Communication Port tests shall include the following:
    1. Loop Back Tests, Single Port and Port to Port
    2. Aggregate throughput tests as defined in Paragraph 7.1.1

Utility Function Tests:

- Time of Day Functions
    1. Display Time of Day (TOD) Clock

2. Set Time of Day (TOD) Clock
3. Enable Daylight Savings Time
4. Disable Daylight Savings Time
5. Line frequency tests
6. Clock accuracy tests

- Ethernet Functions
  1. Get Current IP Address
  2. Set Current IP Address
  3. Load IP Address from Datakey
  4. Save IP Address from Datakey
  5. Start Ethernet
- Clear Error Log
- Configure Continuous Tests
- Start Application

Testing shall be performed either within an environmental chamber or on a bench. The ATC unit is not required to be installed within a cabinet during these tests.

These test procedures do not verify equipment performance under every possible combination of environmental requirements covered by the Standard. However, nothing in this testing profile shall be construed as to relieve the requirement that the equipment provided must fully conform with the Standard under all environmental conditions stated herein.

7.1.1 Engine Board Communication Loading Test

The communication channel loads for load testing purposes, with only required test applications present, shall be the following:

- SP1 and SP2 shall each be operated continuously in a single port external loopback configuration at 115.2 kbps, asynchronous mode
- SP3, SP5, SP8: Any two of these ports shall be operated continuously at 614.4 kbps, synchronous SDLC mode. The third port shall be operated continuously at 115.2 kbps, asynchronous mode. Each port shall be configured as a single port external loopback.
- SP4 and SP6 shall be operated at 115.2 kbps asynchronous mode. One of these ports shall be configured to operate in a continuous external loopback mode while the other port shall be available for test interaction using either a display/keypad or serially connected terminal.
- ENET1 and ENET2 shall each be operated in continuous full-duplex mode at 5Mbps using UDP only with alternate 512 byte and 1024 byte packet sizes. Network traffic must be external to the Engine Board and without any other network traffic occurring (isolated subnet).

All serial ports under test shall continuously send the following text:

">The quick[0xF7][0xFE]brow[0xBE]n_fox jumps }over the~|azy dog?[0xFE]"

(in Hex)

"3E 54 68 65 20 71 75 69 63 6B F7 FE 62 72 6F 77 BE 6E 5F 66 6F 78 20 6A 75 6D 70 73 20 7D 6F 76 65 72 20 74 68 65 7E 7C 61 7A 79 20 64 6F 67 3F FE"

Flow control shall not be used for rate limiting. Test duration shall be 8 hours and must be error-free for successful completion.

Guidance: This test may require the use of a test fixture to expose the necessary serial ports and/or signals.

7.1.2 Timing Accuracy and Repeatability Test (TART)

A Timing Accuracy and Repeatability Test (TART) application program shall be installed and operate continuously and concurrently with the Engine Board Communications Loading Test program. The TART program shall be provided by the manufacturer and provide the following functionality:

1. Using Linux Timers, the TART program shall cause the CPU_ACTIVE signal to continuously toggle "on" (low voltage) and toggle "off" (high voltage) to rest during a one minute period as follows:
2. CPU_ACTIVE output shall toggle “on” at the 55.0 s mark and toggle “off” at the 0.0 s mark
3. A logging mechanism shall be available to timestamp each transition for future review.
4. The timestamps shall be reviewed upon test completion to ensure that the Timing Accuracy Tests and Repeatability tests of NEMA TS2 - 2008 (Section 2.2.2) are satisfied.

Changing the operating system time during the test should not impact the interval times of the test. LINESYNC shall be used as the timing reference, nominally at 60Hz. The ATC manufacturer shall provide access to the CPU_ACTIVE signal for test monitoring purposes.

7.2 Inspection

A visual and physical inspection of the ATC unit shall verify mechanical, dimensional, and assembly conformance to all parts of this standard.

7.3 Testing Certification

A complete quality control and final test report shall be supplied with each item (see Section 9.1.3).

7.4 Definitions of Design Acceptance Testing (DAT) and Production Testing

Design Acceptance Testing (DAT) is performed on the first article ATC unit and is a part of the pre-production process.

Production testing is performed on all units prior to their shipment to an agency.

7.5 Environmental and Operating Requirements

The requirements (voltage, temperature, etc.) of this section shall apply in any combination.

7.5.1 Voltage and Frequency

7.5.1.1 Operating Voltage

The nominal operating voltage shall be 120 VAC, unless otherwise noted.

7.5.1.2 Operating Frequency

The nominal AC line operating frequency range shall be 60 Hz (±3.0 Hz), unless otherwise noted.

7.5.2 Transients, Power Service (DAT)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.1.6, Transients, Power Service.

7.5.3 Nondestructive Transient Immunity (DAT)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.1.8, Nondestruct Transient Immunity, with the following variations:

- Test voltage amplitude shall be 2000 ± 100 V, both positive and negative polarity.

7.5.4 Temperature and Humidity

The ATC unit under test shall maintain all programmed functions when the temperature and humidity ambients are within the specified limits defined herein (Section 7.5.4.1 and Section 7.5.4.2).

7.5.4.1 Ambient Temperature

The operating ambient temperature range shall be from -37˚C to +74˚C. The storage temperature range shall be from -45˚C to +85˚C.

The rate of change in ambient temperature shall not exceed 18˚C per hour, during which the relative humidity shall not exceed 95 percent.

7.5.4.2 Humidity

The relative humidity shall not exceed 95 percent non-condensing over the temperature range of -37˚C to +74˚C.

Above +46˚C, constant absolute humidity shall be maintained. This will result in the relative humidities shown in Table 7-1 for dynamic testing.

Table 7-1. Ambient Temperature versus Relative Humidity At Barometric Pressures (29.92 In. Hg.) (Non-Condensing).

Ambient Temperature/ Dry Bulb (˚C)	Relative Humidity (in%)	Ambient Temperature/ Wet Bulb (˚C)
-37.0 to 1.1	10	-17.2 to 42.7
1.1 to 46.0	95	42.7
48.8	70	42.7
54.4	50	42.7
60.0	38	42.7
65.4	28	42.7
71.2	21	42.7
74.0	18	42.7

7.6 Test Facilities

All instrumentation required in the test procedures, such as voltmeters, ammeters, thermocouples, pulse timers, etc. shall be selected in accordance with good engineering practice. Calibration records for all test equipment shall be provided with test documentation. In all cases where time limit tests are required, the allowance for any instrumentation errors shall be included in the limit test.

1. Variable Voltage Source: A variable source capable of supplying 20 A from 0 VAC to 135 VAC.
2. Environmental Chamber: An environmental chamber capable of attaining temperatures of -37˚C to +74˚C and relative humidities given in Table 7-1.
3. Transient Generator(s): Transient generator(s) capable of supplying the transients outlined in Sections 7.5.2 through 7.5.3.

7.7 Test Procedures: Transients, Temperature, Voltage and Humidity

7.7.1 Test A: Placement in Environmental Chamber and Check-Out of Hook-Up (DAT and Production Testing)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.7.1, Test A: Placement in Environmental Chamber and Check-Out of Hook-Up, with the following variations:

For DAT purposes only, the transient generator shall be connected to the AC input circuit at a point at least 25 feet from the AC power source and not over 10 feet from the input to the test unit

Upon the satisfactory completion and verification of the test hook-up, proceed with Test B (DAT) or Test C (DAT and Production Testing). Figure 8-1 describes the test profile used for Tests C through G to demonstrate the ability of the test unit to function reliably under stated conditions of temperature, voltage, and humidity. Where the characteristics of Figure 8-1 differ from those specified in NEMA TS2, Figure 2-1, those in Figure 8-1 will apply unless specified otherwise.

7.7.2 Test B: Temperature Cycling and Applied Transient Tests (Power Service) (DAT)

Note: In this section the term “dwell” refers to an application in its resting state awaiting inputs, and the actual resting state depends on the application. The goal is to determine if the transients introduced affect the operation of the application by triggering false inputs and/or outputs causing inappropriate application program activity (e.g. improper timing, false calls, error alarms, etc).

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.7.2, Test B: Transient Tests (Power Service), with the following variations:

- Test voltage amplitude in Step 13 shall be 2000 ± 100 V, both positive and negative polarity

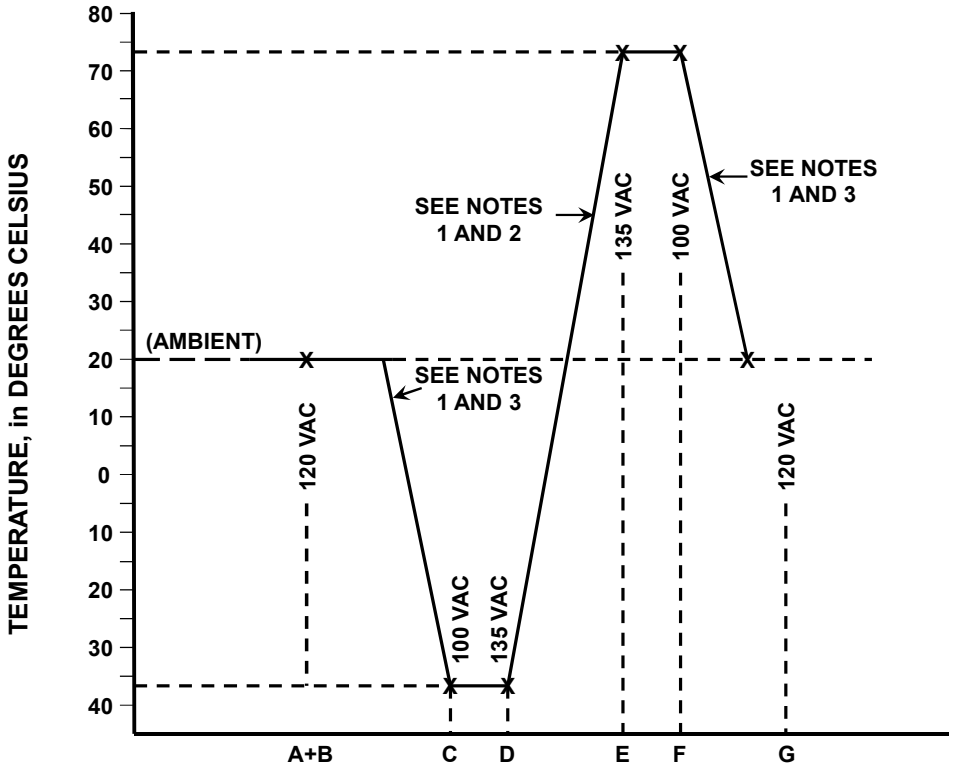


Figure 7-1. Test Profile.

NOTES:

1. The rate of change in temperature shall not exceed 18˚ C per hour.
2. Humidity controls shall be set in conformance with the humidities given in Table 7-1 during the temperature change between Test D and Test E.
3. If a change in both voltage and temperature are required for the next test, the voltage shall be selected prior to the temperature change.
4. When testing a NEMA unit, the LOW voltage shall be 89VAC in place of 100 VAC. All other units will use the 100VAC test level.

7.7.3 Test C: Low-Temperature Low-Voltage Tests (DAT and Production Testing)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.7.3, Test C: Low-Temperature Low-Voltage Tests, with the following variations:

- Utilize Figure 8-1 for the test profile (temperature: -37˚C, low voltage: 100VAC).
- Skip tests 2.c.2) 2.2.11 Timing Accuracy Tests and 2.c.3) 2.2.11.2 Repeatability.

7.7.4 Test D: Low-Temperature High-Voltage Tests (DAT and Production Testing)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.7.4, Test D: Low-Temperature High-Voltage Tests, with the following variations:

- Utilize Figure 8-1 for the test profile (temperature: -37˚C).
- Skip tests 2.c.2) 2.2.11 Timing Accuracy Tests and 2.c.3) 2.2.11.2 Repeatability.

7.7.5 Test E: High-Temperature High-Voltage Tests (DAT and Production Testing)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.7.5, Test E: High-Temperature High-Voltage Tests, with the following variations:

- Utilize Table 7-1 for the humidity controls
- In Step 3, utilize the temperature range of +1.1˚ C to +46˚ C and +42.7˚ C wet bulb
- Skip tests 2. 2.2.11 Timing Accuracy Tests and 2. 2.2.11.2 Repeatability

7.7.6 Test F: High-Temperature Low-Voltage Tests (DAT and Production Testing)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.7.6, Test F: High-Temperature Low-Voltage Tests, with the following variations:

- Utilize Figure 8-1 for the test profile (low voltage: 100VAC)
- In Step 2, utilize +42.7˚ C wet bulb
- Skip tests 2. 2.2.11 Timing Accuracy Tests and 2. 2.2.11.2 Repeatability

7.7.7 Test G: Test Termination (All tests)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.7.7, Test G: Test Termination, with the following variations:

- Utilize Figure 8-1 for the test profile.

7.7.8 Test H: Appraisal of Equipment Under Test

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.7.8, Test H: Appraisal of Equipment Under Test.

7.8 Vibration Test (DAT)

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.8, Vibration Test.

Shock and vibration tests shall be performed prior to environmental tests.

7.9 Shock (Impact) Test (DAT)

7.9.1 Purpose of Test

The ATC unit under test shall meet all requirements as defined in NEMA TS2, Section 2.2.9, Shock (Impact) Test, with the following modifications:

a) Program the table to subject the test unit to a 10G force having a duration of 11 ms.

b) The test shall use a waveform suitable to simulate a drop test, such as saw tooth. The waveform shall be a saw tooth with 11 ms rise time and 0 ms fall time.

c) Subject the test unit to the above test setup three times in both the positive and negative direction.

Shock and vibration tests shall be performed prior to environmental tests.

7.10 Power Interruption Test Procedures (DAT)

The following power interruption tests shall be conducted at low input voltage (100 VAC) and high input voltage (135 VAC) at -37˚ C, and +74˚ C.

Guidance:

Suggestion is to use a second ATC as part of a control group and accurately synchronize the test ATC and control ATC clocks prior to each specific test. One then subjects the test ATC to the desired test while the control ATC receives continuous, constant voltage at the test voltage level. After the tests have been performed, one then verifies that both ATCs still have their clocks accurately synchronized to one another (i.e., zero loss of time due to AC input signal failure and accuracy remains per Section 4.3.4 in cases where a restart occurs). Application test software may be required to facilitate verifying synchronization.

7.10.1 Short Power Interruption

While the ATC unit under test is cycling through normal operations, remove the input voltage for a period of 475 ms. Upon restoration of the input voltage, check to ensure that the CU under test continues normal operation as though no power interruption has occurred. Repeat this test three times.

Verify that the ATC unit clock has not drifted as a result of the power interruptions.

Additional power interruption testing is to be performed at 550 ms, 750 ms and one second outages to verify proper restart operation.

7.10.2 Voltage Variation

All circuits of the ATC unit under test shall be subjected to slowly varying line voltage during which the ATC unit under test shall be subjected to line voltage that is slowly lowered from a nominal 120 VAC line voltage to 0 VAC at a rate of not greater than 2 VAC per second. The line voltage shall then be slowly raised at a rate of not greater than 2 VAC per second to 100 VAC, at which point the ATC unit under test shall resume normal operation without operator intervention. This test shall be performed at both -37˚ C and +74˚ C, at a nominal 120 VAC line voltage. Repeat this test three times.

Verify that the CU clock has not drifted as a result of the power variations.

7.10.3 Rapid Power Interruption

The ATC unit under test shall be subjected to rapid power interruption testing of the form that the power shall be off for 350 ms and on for 650 ms for a period of two minutes. Power interruption shall be performed through electromechanical contacts of an appropriate size for the load. During this testing, the ATC unit shall function normally and shall continue normal sequencing (operation) at the conclusion of the test. This test shall be performed at both -37˚ C and +74˚ C, at a nominal 120 VAC line voltage. Repeat this test three times.

Verify that the ATC unit clock has not drifted as a result of the power interruptions.

Verification of memory may be performed at the conclusion of the overall test.

7.10.4 Extended AC Line Synchronization Verification

Verification of AC Line Synchronization for power outages of less than 2.7 minutes @ +25˚ C and less than 45 seconds at both -37˚ C and +74˚ C.

During transient and power interruption testing, the ATC unit shall operate at a minimum CPU loading of 15 percent and all serial and Ethernet ports shall operate with channel loading as

defined in Section 4.4.3. SPI and USB ports are not required to be operational for this test. This means that a simulation environment must be established to result in meaningful communications to an application program within the CU while conducting these tests.

1. The purpose of this test is to verify that time drift between multiple CUs does not exceed requirements while running on the same AC source and for short power outages of less than 2.7 minutes or 45 seconds as applicable.
2. This test requires two or more CUs connected to the same AC power source.
3. Each ATC unit shall be configured to drive an output through its field IO module.
4. The output shall continuously cycle, turning ON at the top of the minute and turning OFF 55 seconds later.
5. This test requires that all units under test are initially set to the same time using a GPS for time reference.
6. Time will be verified at different stages of testing. These stages, in no particular order are:
    a. On start up
    b. During / After power interruption testing
    c. During / After power supply transient testing
    d. During / After 8 hour continuous operation
    e. During / After varying line voltage test
7. The tests defined above are considered as one continuous test. Once testing has started, the ATC unit's times shall not be changed until completion of tests 6a through 6e.
8. To verify the time during each test, the 60 second toggling output shall be monitored. The monitored output will be checked to ensure that the CUs under test are synchronized. For the purpose of this test:
    a. Synchronization shall mean that each CU's monitored output shall go from the OFF to the ON state and the ON to the OFF state within +/- .5 AC cycles of each other.
    b. The waveforms monitored shall be checked to ensure that the on and off times occur within +/- .5 AC cycles (non-accumulating over multiple power interruptions of less than 2.7 minutes or 45 seconds) at the same edge for each ATC unit under test.

Power outages of more than 2.7 minutes @ +25˚ C and more than 45 seconds at both -37˚ C and +74˚ C:

1. This test verifies that the ATC units under test do not time drift by more than 0.0025% @ +25˚ C or more than 0.02% at both -37˚ C and +74˚ C. It shall also verify the operation of the time back up circuitry.
2. This test requires that all units under test are set to the same time as a GPS reference clock.
3. With time synchronized record the current time of each ATC unit and the GPS reference. The power to each ATC unit is then removed for 7 days. At the end of the 7 day period switch the units back on and record the time of each ATC unit.
4. Verify that the time drift between each ATC unit and the GPS reference is less than the maximum specified in step #1.

Guidance:

This test is provided for time critical applications such as traffic signal control, where short power interruptions must not introduce drift into the ATC line frequency clock used for time-of-day and offset/signal timing. This requires attention to both the software and hardware to ensure that the appropriate protections are taken to ignore such disruptions and maintain the accuracy of the internal clock. This testing is intended to verify that such transients do not introduce any drift into the clock used for offset determination. Such drifts can have an impact on desired operations. Therefore, it is important to verify that the design of the ATC meets the transient and power interruption requirements for clock accuracy.

8 PERFORMANCE AND MATERIAL REQUIREMENTS

8.1 General

8.1.1 Furnished Equipment

All equipment furnished in conformance to this standard shall be new and unused. Vacuum or gaseous tubes and electro-mechanical devices shall not be used unless specifically called for by this standard.

8.1.2 Edges

All sharp edges and corners shall be rounded.

8.1.3 Hardware

All washers, nuts, bolts, hinges, and hinge pins shall be stainless steel unless otherwise specified.

8.1.4 Electrical Isolation and Equipment Grounding

Within the circuit of any device, module, or printed circuit board (PCB), electrical isolation shall be provided between DC ground, Equipment Ground (EG) and AC. They shall be electrically isolated from each other by 500 MΩ, minimum, when tested at the input terminals with 500 VDC.

Equipment grounding practices specified in the NEMA TS2-2003 Standard shall be followed. In particular, all external metallic surfaces such as faceplates, chassis and connector housings shall be connected to the equipment ground signal input of the power supply.

Guidance: The USB port(s) provided by an ATC conformant controller are defined for use as temporary access points for the uploading and downloading of controller logs, diagnostics, firmware upgrades etc.

The user needs to be aware that many USB devices connect their logic ground pin and the metal shell of their connector together. This has the effect of connecting the controllers' logic ground and earth / chassis ground together when the USB device is plugged into the controller. This configuration, though it may be temporarily acceptable for a very short time, will seriously degrade the electrical, transient protection system of the controller and cabinet assembly if it remains permanently installed and should not be permitted.

If the user of an ATC desires to utilize the USB port(s) for permanent attachment of a device, such as a USB hard drive, it is the responsibility of the user to ensure that a path between the controllers' logic ground and earth ground is not provided when the USB device connector is inserted. Use of a USB device or cable that insulates its ground pin from its connector shell or some other USB isolation scheme should meet this requirement.

Guidance: Transient protection devices (TSPs) installed between DC ground, equipment ground, and AC may clamp at a voltage that is lower than the 500VDC test voltage. In this case, the TSPs may be removed to conduct isolation tests for the purpose of product qualification. Isolation tests of production units may be performed at the highest voltage that is compatible with the TSP clamp voltage.

8.1.5 Component Sources

All components shall be of such design, fabrication, nomenclature, or other identification as to be purchased from a wholesale distributor or from the component manufacturer, except as follows:

8.1.5.1 Circuit Designs

The electronic circuit design shall be such that all components of the same generic type, regardless of manufacturer, shall function equally in accordance with this standard. No component shall be applied contrary to its manufacturer's recommendations or data sheets.

8.1.5.2 Operational Envelopes

No component shall be operated above 80 percent of its maximum rated voltage, current or power ratings. Digital components shall not be operated above 3 percent over their nominal voltage, current, or power ratings. All components used shall be designed to operate within the full temperature range specified. The component data sheets shall be the only accepted form of validation of the temperature range. Testing and/or screening of commercial grade components is not permitted.

8.1.5.3 Component Design Life

The design life of each component, operating for 24 hours a day and operating in its circuit application, shall be 10 years or longer.

8.1.5.4 Component Packaging

Encapsulation of three or more discrete components into circuit modules is prohibited except for transient suppression circuits, resistor networks, diode arrays, solid-state switches, optical isolators, and transistor arrays. Components shall be arranged so they are easily accessible, replaceable, and identifiable for testing and maintenance. Where damage by shock or vibration exists, the component shall be supported mechanically by a clamp, fastener, retainer, or hold-down bracket.

8.1.6 Capacitors

The DC and AC voltage ratings as well as the dissipation factor of a capacitor shall exceed the worst-case design parameters of the circuitry by 1.5 times except for supercaps. Supercaps are capacitors rated less than 10 working volts DC with capacitance values greater than or equal to 1.0F. Supercaps shall be required to meet only their stated ratings. Capacitor encasements shall be resistant to cracking, peeling and discoloration. All capacitors shall be insulated and shall be marked with their capacitance values and working voltages. Electrolytic capacitors shall not be used for capacitance values of less than 1.0 µF and shall be marked with polarity.

8.1.7 Resistors

Fixed carbon film, deposited carbon, or composition-insulated resistors shall conform to the performance requirements of Military Specifications MIL-R-11F or MIL-R-22684. All resistors shall be insulated. Resistance values for all discrete resistors shall be indicated by the EIA color codes, or stamped value. The resistor value shall not vary by more than 5 percent for carbon film and deposited carbon types and 10 percent for composition–insulated type over the range of -37° C to 74° C. Special ventilation or heat sinking shall be provided for all resistors rated two W or higher. They shall be insulated from the PCB.

8.1.8 Semiconductors

All transistors, integrated circuits and diodes shall be a standard type listed by EIA and clearly identifiable. All metal oxide semiconductor components shall contain circuitry to protect their inputs and outputs against damage due to high static voltages or electrical fields. Device pin "1" locations shall be properly marked on the PCB adjacent to the pin.

8.1.9 Transformers and Inductors

With the exception of surface-mount components, all power transformers and inductors shall have the manufacturer's name or logo and part number clearly and legibly printed on the case or lamination. All transformers and inductors shall have their windings insulated, shall be protected to exclude moisture, and shall have their leads color coded with an approved EIA color code or identified in a manner to facilitate proper installation.

8.1.10 Single-Use Over-Current Protection Device

All single-use over-current protection devices (OCPD) shall be resident in a holder which is readily accessible. Single-use OCPDs shall not require any tools other than a screwdriver for their replacement. A label indicating the OCPD rating shall be located next to the OCPD holder.

Fuses shall not be dislodged during shipping and handling.

8.1.11 Switches

8.1.11.1 DIP Switches

Dual-inline-package, quick snap switches shall be rated for a minimum of 30,000 operations per position at 50 mA, 30 VDC. The switch contact resistance shall be 100 mΩ maximum at 2 mA, 30 VDC. The contacts shall be gold over brass (or silver). Contact for VAC or 28 VDC and shall be silver over brass (or equal).

8.1.11.2 Logic Switches

The switch contacts shall be rated for 0.4VA at 20V AC or DC. Contact material shall be copper alloy with gold over nickel plating (or equal). The switch shall be rated for a minimum of 40,000 operations.

8.1.11.3 Control Switches

The switch contacts shall be rated for 5A at 125 VAC or 28 VDC. Contact material shall be brass or copper alloy with silver plating (or equal). The switch shall be rated for a minimum of 40,000 operations.

8.1.11.4 Power Switches

The switch contacts shall be rated for a minimum of 5A resistive load at 120 VAC or 28 VDC and shall be silver over nickel (or equal). The switch shall be rated for a minimum of 10A inductive load at 125 VAC.

8.1.12 Wiring, Cabling, and Harnesses

8.1.12.1 Harnesses

Harnesses shall be neat, firm, and properly bundled with external protection. They shall be tie-wrapped and routed to minimize crosstalk and electrical interference. Each harness shall be of adequate length to allow any conductor to be connected properly to its associated connector or termination point. Conductors within an encased harness have no color requirements.

8.1.12.2 Bundling

Wiring containing AC shall be bundled separately or shielded separately from all DC logic voltage control circuits. Wiring shall be routed to prevent conductors from being in contact with metal edges. Wiring shall be arranged so that any removable assembly may be removed without disturbing conductors not associated with that assembly. Splicing or cutting/replacing of bundle wrapping is not allowed.

8.1.12.3 Conductor Construction

All conductors shall conform to MIL-W-16878E/1 or better and shall have a minimum of 19 strands of copper. The insulation shall be polyvinyl chloride with a minimum thickness of 10 mils or greater. Where insulation thickness is 15 mils or less, the conductor shall conform to MIL-W-16878/17. Conductor color identification shall be as follows:

- AC - gray or continuous white color
- EG - solid green or continuous green color with 1 or more yellow stripes
- DC logic ground - continuous white color with 1 red stripe
- AC+ - continuous black color or black with colored stripe

DC logic ungrounded or signal - any color not specified

8.1.13 Indicators and Character Displays

All indicators and character displays, when supplied, shall be readily visible at a radius of up to 4 feet within the cone of visibility when the indicator is subjected to 97,000 lux (9,000 foot-candles) of white light with the light source at 45 ±2˚ to the front panel.

Incandescent indicators shall not be used within the ATC unit.

8.1.13.1 Range of Visibility

All indicators and character displays shall have a minimum 90˚ cone of visibility with its axis perpendicular to the panel on which the indicator is mounted. All indicators shall be self-luminous. All indicators shall have a rated life of 100,000 hours minimum. Except for RJ-45 connectors with built-in LEDs, each LED indicator shall be white or clear when off.

8.1.13.2 LCDs

Liquid Crystal Displays (LCD), when used, shall operate at temperatures of -20 °C to +70 °C and shall not be damaged nor otherwise adversely affect the unit's operation at temperatures of -37 °C to +74 °C. Low temperature operation must have a sufficiently fast reaction time to be readable for the integer value displayed.

Some agencies may wish to specify faster LCD operation and lower temperature operation which may necessitate the use of heaters for the LCD. When such heaters are used, they shall only be energized at low temperature to support operator interaction and shall be controllable through the application software.

8.1.14 Connectors

All connectors shall be keyed to prevent improper insertion of the wrong connector. The mating connectors shall be designated as the connector number and male/female relationship, such as C1P (plug) and C1S (socket). The connector shall be called out base metal with minimum 0.00005 inch nickel plated with 0.000015 inch gold.

8.1.14.1 Plastic Circular and M Type Connectors

Pin and socket contacts, if used, for connectors shall be beryllium copper construction. Pin diameter shall be 0.062 inch. All pin and socket connectors shall use the AMP #601105-1 or #91002-1 contact insertion tool and the AMP #305183 contact extraction tool.

8.1.14.2 Flat Cable Connectors

All flat cable connectors, where used, shall be designed for use with 26 AWG cable; shall have dual cantilevered phosphor bronze contacts; and shall have a current rating of 1 A minimum and an insulation resistance of 5 MΩ minimum.

8.1.14.3 PCB Header Socket Connectors

Each PCB header socket block shall be nylon or diallyl phthalate. Each PCB header socket contact shall be removable but crimp-connected to its conductor. The manufacturer shall list the part number of the extraction tool recommended by its manufacturer. Each PCB header socket contact shall be brass or phosphor bronze.

8.1.14.4 Metallic Circular Connectors [NEMA]

Metallic Circular Connectors shall conform and interface with MS 116 Shell type.

8.1.15 PCB Design

No components, traces, brackets, or obstructions shall be within 0.125 inch of a PC board edge (guide edges). The manufacturer's name or logo, model number, serial number, and circuit issue or revision number shall appear and be readily visible on all PCBs. Devices to prevent the PCB from backing out of its assembly connectors shall be provided. All screw type fasteners shall utilize locking devices or locking compounds except for finger screws, which shall be captive. Solder quality should conform to IPC 610 specification for Industrial ratings. Serial numbers on PCBs shall be durable. All PCBs shall be mounted such that their component surface plane is vertically-oriented when the ATC is mounted in its normal operating position.

All Printed Circuit Boards (PCBs) shall be conformal coated per the coating manufacturer’s recommendations. Coating shall contain a UV light-traceable component to allow for inspection of coating coverage. Coating material shall meet the requirements of ANSI/IPC-CC-830B.

8.1.16 Assemblies

All assemblies shall be modular, easily replaceable and incorporate plug-in capability for their associated devices or PCBs. Assemblies shall be provided with two guides for each plug-in PCB or associated device (except relays). The guides shall extend to within 0.75 in from the face of either the socket or connector and front edge of the assembly. If Nylon guides are used, the guides shall be securely attached to the file or assembly chassis.

8.1.17 Potentiometers

Potentiometers with ratings from 1 to 2 watts shall meet Military Type RV4 requirements. Potentiometers with ratings less than 1 Watt shall be used only for trimmer type function. The potentiometer power rating shall be at least 100 percent greater than the maximum power requirements of the circuit.

8.1.18 Batteries

The ATC shall not utilize batteries.

8.1.19 Jumpers

All jumpers and contact pins shall be gold-plated.

8.1.20 Use of Fans

The ATC shall not use any fans or similar active mechanical devices to manage airflow.

8.1.21 PCB Connectors

8.1.21.1 PCB Edge Connectors

PCB Edge connectors shall have bifurcated gold-plated contacts. The PCB receptacle connector shall meet or exceed the following:

Operating Voltage: 600 VAC (RMS)

Current Rating: 5.0 Amperes

Insulation Material: Diallyl Phthalate or Thermoplastic

Insulation Resistance: 5,000 MΩ

Contact Material: Copper alloy plated with 0.00005 in of nickel and 0.000015 in of gold

Contact Resistance: 0.006 Ohm maximum

8.1.21.2 PCB Connectors (Two-Piece)

Two-piece PCB connectors shall meet or exceed the requirements of DIN 41612.

8.1.22 Tolerances

The following tolerances shall apply, except as specifically shown on the plans or in these specifications:

Sheet Metal	± 1.334 mm (0.0525 inch)
PCB	± 0.254 mm (0.010 inch)
Edge Guides	± 0.381 mm (0.015 inch)

9 QUALITY CONTROL

Material in this section is considered a supplement to that provided in Section 8. In the case of apparent inconsistencies, material in Section 8 of this standard shall prevail.

9.1 Components

All components shall be lot sampled to assure a consistent high conformance standard to the design specification of the equipment.

9.1.1 Subassembly, Unit Or Module

Complete electrical, environmental, and timing conformance testing shall be performed on each module, unit, printed circuit, or subassembly. Housing, chassis, and connection terminals shall be inspected for mechanical sturdiness, and harnessing to sockets shall be electrically tested for proper wiring sequence. The equipment shall be visually and physically inspected to assure proper placement, mounting, and compatibility of subassemblies.

9.1.2 Predelivery Repair

Any defects or deficiencies found by the inspection system involving mechanical structure or wiring shall be returned through the manufacturing process or special repair process for correction. PCB flow soldering is allowed a second time if copper runs and joints are not satisfactorily coated on the first run. Hand soldering is allowed for printed circuit repair.

9.1.3 Manufacturers' Quality Control Testing Certification

Guidance: If requested by the purchasing agency, quality control procedures shall be submitted prior to production. A compliant test report that is part of the quality control procedure shall be supplied with each delivered unit. Along with pass fail information this report shall include the quality control procedure, test report format and the name of the tester. It should be counter-signed by a corporate officer.

The quality control procedure shall include the following:

- Design acceptance testing of all supplied components
- Physical and functional testing of controller units
- Environmental testing report(s) and final acceptance
- Acceptance testing of all supplied components
- Physical and functional testing of all modules and items

Verification of a minimum burn-in of all equipment

9.1.4 Serial Numbers

Each complete ATC and all individual subassemblies shall have a unique serial number.

9.1.5 Delivery

Each item delivered for testing shall be complete, including manuals, and ready for testing. Documentation shall reflect as-built components.

APPENDIX A: Linux Operating system and Minimum KERNEL CONFIGURATION (NORMATIVE)

A.1 Linux Operating System and Development Requirements

The ATC shall use a Linux operating system (O/S) and shall include standard POSIX libraries for application support including real-time extensions of POSIX 1003.1b. To facilitate application level access to the ATC hardware, a Board Support Package (BSP) shall be provided for access to hardware-specific drivers.

After boot-up the ATC Linux O/S shall make available to applications, access to the low level drivers (block, character and network) provided by the kernel (subject to current open source requirements) or through kernel modules.

The BSP supplied by a manufacturer shall include the following components:

A Linux kernel that shall be configured to include, at minimum, the features specified in this Appendix.

1) Drivers that support all functionality defined by Section 4.3, On-Board Resources, and Section 4.4.3, Serial Interface Ports, of this document and be capable of operating in an interrupt-driven environment where appropriate. See Appendix B for BSP-specified driver interface details. Device drivers for the following Engine Board hardware are required.

Device	Function
/dev/sp1	SP1 Asynchronous Port
/dev/sp1s	SP1 Synchronous Port
/dev/sp2	SP2 Asynchronous Port
/dev/sp2s	SP2 Synchronous Port
/dev/sp3	SP3 Asynchronous Port
/dev/sp3s	SP3 Synchronous Port
/dev/sp4	SP4 Asynchronous Port
/dev/sp5s	SP5 Synchronous Port
/dev/sp6	SP6 Asynchronous Port
/dev/sp8	SP8 Asynchronous Port
/dev/sp8s	SP8 Synchronous Port
eth0	First Ethernet Interface
eth1	Second Ethernet Interface
/dev/sda	USB Storage Device
/dev/datakey	SPI Datakey Device
/dev/eeprom	SPI EEPROM Device
/dev/cpureset	CPU Reset Output
/dev/cpuactive	CPU Active LED
/dev/powerdown	Power Down Input
/dev/datakeypresent	Datakey Present Input
/dev/rtc0	RTC device
/	Long-Term Non-Volatile Memory (FLASH) File System
/sram	Short-Term Non-Volatile Memory (SRAM) File System

Utility applications, modules, libraries, and supporting data which include, but are not limited to, the following:

Table A-1. Additional BSP Elements

Package	Version *	Programs	Remarks
Busybox	**1.27.2**	**addgroup, adduser, adjtimex, ar, ash, basename, busybox, cat, chgrp, chmod, chown, chroot, clear, cmp, cp, crond, crontab, cut, date, dd, delgroup, deluser, df, dirname, dmesg, dos2unix, du, echo, egrep, env, expr, false, fgrep, find, freeramdisk, ftp, ftpd, getty, grep, gunzip, gzip, halt, head, hexdump, hostid, hostname, hwclock, id, ifconfig, ifdown, ifup, inetd, init, insmod, install, kill, killall, klogd, last, ln, logger, login, logname, logread, ls, lsmod, makedevs, md5sum, mesg, mkdir, mkfifo, mknod, mktemp, modprobe, more, mount, mv, netstat, nslookup, passwd, patch, pidof, ping, pivot_root, printf, ps, pwd, rdate, reboot, renice, reset, rm, rmdir, rmmod, route, rpm, run-parts, rx, sed, sh, sleep, sort, start-stop-daemon, stty, su, sulogin, sync, sysctl, syslogd, tail, tar, tee, telnet, telnetd, test, time, top, touch, tr, true, tty, umount, uname, uniq, unix2dos, unzip, uptime, usleep, vi, wc, which, who, whoami, xargs, yes, zcat**	**UNIX shell and commands collection** http://www.busybox.net **ftpd is optional in BusyBox if an alternate ftp server is provided telnetd and ftpd shall be disabled by default.**

Package	Version *	Programs	Remarks
Glibc	**2.23**	**ld.so, libc, libcrypt.so, libdl.so libm.so, libnsl.so, libpthread.so, libresolv.so, libutil.so, librt.so**	
Hrtimers			**per POSIX 1003.1b, with better than 100uS resolution**
Secure Shell	**SSH-2**	**sshd, ssh, scp**	
C++		**libstdc++.so**	
		libcap.so	**Linux Security Capabilities**
	manufacturer-specific	**libatc.so**	**ATC-specific library functions**

* Or higher

A root filesystem including the following directories, or symbolic links to directories, and compatible with version 2.3 (or later) of the Filesystem Hierarchy Standard:

Directory	Notes
/bin	
/dev	
/etc	
/etc/init.d	Linux Standard Base location for initialization scripts
/lib	
/media	
/media/sdcard1	Mount point for Engine Board-mounted SD card
/media/sdcard2	Mount point for Host Module-mounted SD card
/media/usb	Mount point for first USB storage device that is attached
/media/usb2	Mount point for second USB storage device that is attached
/media/usbN	Mount point for <Nth> USB storage device that is attached
/mnt	
/opt	
/sram	Mount point for SRAM filesystem
/sbin	
/tmp	
/usr	
/var	

Each Engine Board manufacturer shall provide all development tools (cross-compiler, linker, libraries, and header files) necessary to completely rebuild application programs for execution on their Engine Board. The development environment must operate on an x86-based platform and shall be available in both source and object format for both download from the manufacturer's website in an unrestricted area and by physical media if requested. These tools shall be available for a period of not less than five (5) years after the last date of delivery of the associated Engine Board.

The ATC manufacturer shall include any necessary utilities required to support the loading of the operating system, drivers, BSP, bootloader, and manufacturer-specific utilities using a USB memory device and/or communication connection (Ethernet or serial). It shall not be necessary to open or otherwise disassemble the unit, replace any device, or set any switch to affect the program update.

The C library shall use the /etc/localtime file to indicate the local system timezone. The /etc/localtime file shall be in the TZif2 format. The /etc/localtime file may be changed at any time; the C library shall always use the current file.

/dev/sp4 shall be used for the Linux console (and bootloader console if applicable). The default communications settings for the console shall be 115.2 kbps, no parity, 8 data bits, 1 stop bit.

/dev/sp4 shall also be available for use by application programs as a general purpose communication port. When thus configured, the console output shall be disabled at power up and the ATC shall send no output to /dev/sp4 during boot-up but may read it.

The startup mode of /dev/sp4, console or general, shall persist across reboots, but may be toggled using the <ESC> key on the /dev/sp6 front panel device. The ATC shall query for the <ESC> key at boot time to determine if the mode must be changed. The user must press and hold the <ESC> key until the mode is changed, as shown by the starting or stopping of console output. Note that the <ESC> key on the front panel device does not generate a solitary ASCII escape character, but a three-character escape sequence as shown in Table 6-6.

The CPURESET signal shall be asserted during the boot process with a minimum pulse width of 125 ms. It shall be deasserted no later than 250 ms from when the POWERUP signal is asserted. The bootloader may be required to perform this function. The CPURESET signal shall not be activated outside of application program control after this 250 ms period.

A.2 Minimum Linux Kernel Configuration

This O/S kernel shall be a Linux kernel of Version 3.12 or later.

Platform-specific or Linux version-specific configuration options shall be selected by the manufacturer based on satisfying the functional requirements implied below.

The following items or their equivalent shall be the minimum Linux kernel configuration features that are included in the kernel build. A manufacturer may include additional configuration options at their discretion.

Kernel Configuration

```
#
# General setup
#
CONFIG_SYSVIPC=y
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
CONFIG_EMBEDDED=y
CONFIG_SYSCTL=y
CONFIG_HOTPLUG=y
CONFIG_ELF_CORE=y
#
# Loadable module support
#
CONFIG_MODULES=y
CONFIG_MODULE_UNLOAD=y
CONFIG_KMOD=y
#
# Processor type and features
#
CONFIG_PREEMPT=y
#
# Executable file formats
#
CONFIG_BINFMT_ELF=y
CONFIG_BINFMT_AOUT=m
#
# Networking options
#
CONFIG_PACKET=y
CONFIG_PACKET_MMAP=y
CONFIG_UNIX=y
CONFIG_INET=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_PNP=y
CONFIG_IP_PNP_DHCP=y
CONFIG_IP_PNP_BOOTP=y
CONFIG_SYN_COOKIES=y
#
# IP: Virtual Server Configuration
#
CONFIG_NETFILTER=y
#
# Block devices
#
CONFIG_BLK_DEV_LOOP=y
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_INITRD=y
#
# SCSI device support
#
CONFIG_SCSI=y
```

```
CONFIG_SCSI_PROC_FS=y
#
# SCSI support type (disk, tape, CD-ROM)
#
CONFIG_BLK_DEV_SD=y
#
# Ethernet (10 or 100Mbit)
#
CONFIG_NET_ETHERNET=y
#
# Non-8250 serial port support
#
CONFIG_UNIX98_PTYS=y
CONFIG_LEGACY_PTYS=y
CONFIG_LEGACY_PTY_COUNT=256
#
# USB support
#
CONFIG_USB=y
#
# Miscellaneous USB options
#
CONFIG_USB_DEVICEFS=y
#
# NOTE: USB_STORAGE enables SCSI, and 'SCSI disk support'
# may also be needed; see USB_STORAGE Help for more information
#
CONFIG_USB_STORAGE=y
CONFIG_USB_STORAGE_FREECOM=y
CONFIG_USB_STORAGE_ISD200=y
CONFIG_USB_STORAGE_DPCM=y
#
# File systems
#
CONFIG_EXT2_FS=y
#
# DOS/FAT/NT Filesystems
#
CONFIG_FAT_FS=y
CONFIG_MSDOS_FS=y
CONFIG_VFAT_FS=y
CONFIG_FAT_DEFAULT_CODEPAGE=437
CONFIG_FAT_DEFAULT_IOCHARSET="iso8859-1"
#
# Pseudo filesystems
#
CONFIG_PROC_FS=y
CONFIG_SYSFS=y
CONFIG_TMPFS=y
CONFIG_RAMFS=y
CONFIG_SQUASHFS=y
CONFIG_SQUASHFS_LZO=y
#
# Network File Systems
#
CONFIG_NFS_FS=y
CONFIG_NFS_V3=y
CONFIG_NFSD=y
CONFIG_NFSD_V3=y
CONFIG_NFSD_TCP=y
CONFIG_ROOT_NFS=y
CONFIG_LOCKD=y
CONFIG_LOCKD_V4=y
CONFIG_EXPORTFS=y
```

```
CONFIG_NFS_COMMON=y
CONFIG_SUNRPC=y
#
# Native Language Support
#
CONFIG_NLS_DEFAULT="iso8859-1"
#
# Kernel hacking
#
# Security options
#
# Cryptographic options
#
CONFIG_CRYPTO=y
CONFIG_CRYPTO_HMAC=y
CONFIG_CRYPTO_MD5=y
#
# Library routines
#
CONFIG_CRC32=y
CONFIG_ZLIB_INFLATE=y
CONFIG_ZLIB_DEFLATE=y
#
# Control Groups
#
CONFIG_CGROUPS=y
CONFIG_MEMCG=y
CONFIG_CGROUP_SCHED=y
CONFIG_CGROUP_CPUACCT=y
```

A.3 Additional Linux Kernel Features

A.3.1 Network Configuration

The BSP shall support the standard persistent files:“/etc/network/interfaces,” “/etc/resolv.conf” and “/etc/hostname” for network interface, nameserver and hostname settings respectively.

A.3.2 NTP Configuration

The BSP shall provide a full implementation of the reference ntpd program (ntp.org) with support for the “Type 20 generic NMEA GPS reference clock” and the “/etc/ntp.conf” configuration file as a minimum. NTP v4.2.8 or higher is required.

A.3.3 Application Installation

Application software shall be installed under the directory:

“/opt/atcapps/vendor_name/application_name”

A startup script or executable program for each application shall be installed in that same directory and shall be named:

“/opt/atcapps/vendor_name/application_name/application_name”.

A.3.4 Security Capabilities

The BSP shall support non-root applications and protect from unnecessary privilege escalation by supporting the standard Linux Security Capabilities feature (via the libcap library and kernel support for its capabilities).

A.3.5 Initscripts

The BSP shall support the busybox convention for “initscripts”. Init script names shall take the form 'S' followed by a 2-digit number in the range 00-99 (giving a relative starting order), followed by the service name, e.g. "S40network". The init scripts shall reside in "/etc/init.d/".

A.3.6 BSP version

The Linux BSP version information shall be included in the file "/etc/os-release". As a minimum, the following identifiers shall be present:

NAME - vendor name

VERSION - vendor firmware release version string

VERSION_ID - vendor firmware release number

HOME_URL - vendor product URL

APPENDIX B: REQUIRED DEVICE DRIVER INTERFACES (NORMATIVE)

This appendix specifies the device driver interfaces required by this standard. Where practical, standard Linux drivers are specified and no further detail is given. Otherwise, each driver interface is described in full detail.

B.1 ATC CPU_RESET

The **atc_cpu_reset()** function has been deprecated.

B.2 ATC Engine Board PIO Driver Interface

Overview

The PIO Driver Interface allows the user to control the CPU Active LED, determine whether or not the Datakey is inserted, reset peripheral devices via the CPU Reset signal, and read the Powerdown signal status.

Supported Device File Operations:

- open();
- close();
- read();
- write();

fcntl();

open()

The following dev entries shall exist:

/dev/datakeypresent

/dev/cpuactive

/dev/powerdown

/dev/cpureset

close()

Closes the file descriptor.

read()

```
int read(int filp, void *buf, int count);
```

This allows for reading the state of the Powerdown signal and for determining whether or not the Datakey is inserted. It also allows reading of the current state of cpuactive and cpureset. The value passed in the count parameter must be 1 or no bytes will be read.

The value read for /dev/powerdown reflects the actual state of the Powerdown signal. For the Datakey, if the value read is 1, the Datakey is inserted; and is zero otherwise.

write()

```
int write(int filp, void *buf, int count);
```

Allows changing the state of the CPU Active LED and the CPU Reset signal.

Writing a single nonzero character to the /dev/cpuactive device shall turn on the CPU active LED. Writing a zero to the device will turn off the LED.

fcntl()

```
int fcntl(int fd, int cmd, ...);
```

With minimum supported commands:

F_GETFL, F_SETFL, F_GETOWN, F_SETOWN.

The fcntl() function shall be used to enable or disable asynchronous notification operations.

Examples

```
      /* blocking read of powerdown */
         int fd, result;
   char ch;
         fd = open("/dev/powerdown", O_RDONLY);
         while (1)
         {
            result = read (fd, &ch, 1); /* blocking read */
            if (result == 1)
            {  if (ch == 0) do_powerdown_handling();
               else if (ch == 1) recover_from_powerdown();
            }
   }
        /* non-blocking read of dkey_present*/
        int fd, result;
  char ch;
        fd = open ("/dev/datakeypresent", O_RDONLY | O_NONBLOCK);
        result = read (fd, &ch, 1);
  return (result);
/* setting cpuactive */
      int fd, result;
char ch;
        fd = open ("/dev/cpuactive", O_WRONLY);
        ch = 1; /* turn cpu_active ON */
        result = write(fd, &ch, 1);
        ch = 0; /* turn cpu_active OFF */
  result = write(fd, &ch, 1);
/* send SIGIO signal on change of state of powerdown */
      #include <fcntl.h>
      fd = open("/dev/powerdown", O_RDONLY);
      signal(SIGIO, &powerdown_handler);
      fcntl(fd, F_SETOWN, getpid());
      oflags = fcntl(fd, F_GETFL);
fcntl(fd, F_SETFL, oflags | FASYNC);
```

B.3 ATC Serial Peripheral Interface (SPI) Devices

B.3.1 SPI EEPROM Driver

The following standard Linux functions shall be supported by this driver:

- `open( )`
- `close( )`
- `read ( )`
- `write( )`
- `lseek()`

The special device file node for the host EEPROM shall be “/dev/eeprom.”

B.3.2 Host Board EEPROM Interface, Content and Organization

The optional Host EEPROM device shall have the following interface characteristics:

- Function in a manner similar to a 25020-type (2 kb) SPI EEPROM device
- Have a minimum size of 2 kb organized as 256 words of 8 bits each
- Provide 5V interface signals
- Operate properly with up to a 2.0 MHz SPI clock
- Utilize SPI Mode 0 (CPOL=0, CPHA=0)
- Be write-protected (using *WP pin) whenever POWERUP is LOW
- Be readable from application software during normal ATC operation
- Support the following instruction set:

Instruction	Description	Instruction Format
WREN	Write Enable	0000 X110
WRDI	Write Disable	0000 X100
RDSR	Read Status Register	0000 X101
WRSR	Write Status Register	0000 X001
READ	Read from Memory Array	0000 A8011
WRITE	Write to Memory Array	0000 A8010

Note: A8 represents MSB address bit A8

The following describes the content and organization of the Host EEPROM. If a Host EEPROM does not exist within the ATC, then application software should assume the default configuration which represents a Model 2070L. The data format in the Host EEPROM for the Latitude and Longitude fields shall conform with IEEE/ANSI 754-1985 STD. All other fields shall follow a Big Endian Format as implemented in Freescale 68000-family CPUs.

- Host EEPROM Version (uint8 0..255; Standard enumerated values are: v5.2b – 1; v06.25 and v06A – 2)
- Host EEPROM Length in bytes (uint16 0..65535; Use Defaults = 0)
    - (number of bytes from beginning of EEPROM through and including CRC1)

- # Modules (uint8 0..255). For each module:
    - Module Location (uint8 0..255; Standard enumerated values are: Other – 1, Host – 2, Display – 3, I/O – 4, Power Supply – 5, A1 – 6, A2 – 7)
    - Module Make (uint8 0..255; manufacturer – NEMA assigned NTCIP code except Generic = 0)
    - Module Model (OCTET STRING as defined by ASN.1 specification, see Section 1.4 References)
    - Module Version (seven bytes: YYYYMMDDrrmmpp … where YYYYMMDD is date & version is rr.mm.pp … stored in BCD)
    - Module Type (uint8 0..255; Standard enumerated values are: Other – 1, Hardware – 2, Software/Firmware – 3)
- Display properties:
    - # Char Lines (uint8 0..255 - use 0 if no display attached)
    - #Char Columns (uint8 0..255 - use 0 if no display attached)
    - #Graphic Rows-1 (y_max) (uint8 0..255 - use 0 if no display attached)
    - #Graphic Columns-1 (x_max) (uint16 0..1023 - use 0 if no display attached)
- # Ethernets (uint8 0..255). For each Ethernet:
    - Type (uint8 0..255; Standard enumerated values are: Other - 1, Hub/Phy/Other Direct Port - 2, Unmanaged Switch - 3, Managed Switch - 4, Router - 5)
    - IP Address (four bytes)
    - MAC Address (six bytes
    - Subnet Mask (four bytes)
    - Default Gateway (four bytes)
    - Engine Board Interface (integer 0..255; Standard enumerated values are: Other - 1, Phy – 2, RMII – 3, MII – 4)
- SPI3 Purpose (uint8 0..255; Standard enumerated values are: Unused – 0, reserved 1..255)
- SPI4 Purpose (uint8 0..255; Standard enumerated values are: Unused – 0, otherwise manufacturer specific code 1..255)
- Host Board Serial Ports Used – uint16 0..65535 bit encoded as follows:
    - Bit 0: SP1
    - Bit 1: SP1s
    - Bit 2: SP2
    - Bit 3: SP2s
    - Bit 4: SP3
    - Bit 5: SP3s
    - Bit 6: SP4
    - Bit 7: SP5s
    - Bit 8: SP6
    - Bit 9: SP8 (although always present in an ATC, it is optional in an Model 2070)
    - Bit 10: SP8s (although always present in an ATC, it is optional in a Model 2070)

- # Ports used for I/O (uint8 0..255). For each I/O Port:
    - Port ID (uint8 0..255; standard values are None – 0, SP1 – 1, SP2 – 2, SP3 – 3, SP4 – 4, SP5 – 5, SP6 – 6, Reserved – 7, SP8 – 8, ENET1 – 9, ENET2 – 10, SPI3 – 11, SPI4 – 12, USB – 13, Reserved 14..255)
    - Port Mode as applicable (uint8 0..255; standard values are Other – 0, Async – 1, Sync – 2, SDLC – 3, HDLC – 4, Reserved – 5..255)
    - Baud Rate as applicable (uint32) … Note: not specified for ENET1, ENET2, or USB
- Host Board Serial Ports Present – uint16 0..65535 bit encoded as follows:
    - Bit 0: SP1
    - Bit 1: SP1s
    - Bit 2: SP2
    - Bit 3: SP2s
    - Bit 4: SP3
    - Bit 5: SP3s
    - Bit 6: SP4
    - Bit 7: SP5s
    - Bit 8: SP6
    - Bit 9: SP8 (although always present in an ATC, it is optional in a Model 2070)
    - Bit 10: SP8s (although always present in an ATC, it is optional in a Model 2070)
- Serial Bus #1 Port – uint16 0..65535 bit encoded as follows (0 = SB #1 non-existent/unused):
    - Bit 0: Reserved
    - Bit 1: SP1s
    - Bit 2: Reserved
    - Bit 3: SP2s
    - Bit 4: Reserved
    - Bit 5: SP3s
    - Bit 6: Reserved
    - Bit 7: SP5s
    - Bit 8: Reserved
    - Bit 9: Reserved
    - Bit 10: SP8s (although always present in an ATC, it is optional in a Model 2070)
- Serial Bus #2 Port – uint16 0..65535 bit encoded as follows (0 = SB #2 non-existent/unused):
    - Bit 0: SP1
    - Bit 1: SP1s
    - Bit 2: SP2
    - Bit 3: SP2s
    - Bit 4: SP3
    - Bit 5: SP3s
    - Bit 6: SP4
    - Bit 7: SP5s
    - Bit 8: SP6
    - Bit 9: SP8 (although always present in an ATC, it is optional in a Model 2070)
    - Bit 10: SP8s (although always present in an ATC, it is optional in a Model 2070)

- TS 2 Port 1 Port – uint16 0..65535 bit encoded as follows (0 = NEMA Port 1 non-existent):
    - Bit 0: Reserved
    - Bit 1: SP1s
    - Bit 2: Reserved
    - Bit 3: SP2s
    - Bit 4: Reserved
    - Bit 5: SP3s
    - Bit 6: Reserved
    - Bit 7: SP5s
    - Bit 8: Reserved
    - Bit 9: Reserved
    - Bit 10: SP8s (although always present in an ATC, it is optional in a Model 2070)
- Expansion Bus type (uint8 0..255; Standard enumerated values are: None – 1, Other – 2, VME – 3)
- Encoded SPI Addressing (uint8 0..255; Standard enumerated values are: No – 1, Yes – 2)
- CRC #1 (16-bit)
- Latitude (4 bytes)
- Longitude (4 bytes)
- Controller ID (uint16 0..65535)
- Communication Drop # (uint16 0..65535)
- CAN Bus-specific data (one byte)

    b7: 0

    b6: 1 = CAN supported; 0 = CAN is NOT supported

    b5,b4: [0.0] = CAN GPIO not supported

    [0,1] = CAN GPI supported

    [1,0] = CAN GPO supported

    [1.1] = CAN GPIO supported

    b5: 1 = CAN Vbus Power on Connector

    b2,b1,b0 = /* defined use of CAN GPI / GPO / GPIO */

    [000] = Wake-on Bus Activity

    [001] = VbusSense

    [010] = CAN Vbus - If GPI then VbusSense, If GPO then Enable Vbus, If GPIO then Enable Vbus and Sense if active/overload

    [011] = Standby/Listen Only Mode (1=Standby/Listen Only, 0=Normal CAN operation)

    [100] = Enable Loopback (1=Enable, 0=Normal Operation)

    [101] = Reserved ATC

[110] = Reserved Manufacturer Specific (not an ATC defined configuration)

[111] = Reserved Manufacturer Specific (not an ATC defined configuration)

- Host Board Serial Port Signals Present – (0:not present; 1:present) uint8 0..255 bit encoded as follows:
    - Bit 0: SP1_TXC_INT
    - Bit 1: SP1_RXC_EXT
    - Bit 2: SP1_TXC_EXT
    - Bit 3: SP2_TXC_INT
    - Bit 4: SP2_RXC_EXT
    - Bit 5: SP2_TXC_EXT
    - Bit 6: SP3_TXC_EXT
    - Bit 7: SPI_IRQ
- Reserved for Agency (35 bytes … per TEES)
- CRC #2 (16-bit)

User Data (to end)

The default configuration is as follows:

- Host EEPROM Version = 2
- Host EEPROM Size in bytes = 0
- # Modules = 4 (Note: 2070L is 2070-1B, 2070-2A, 2070-3B, & 2070-4B only)
    - Module 1 Location = 2
    - Module 1 Make = 0
    - Module 1 Model = “2070-1B Host”
    - Module 1 Version = 20040608020200
    - Module 1 Type = 2
    - Module 2 Location = 4
    - Module 2 Make = 0
    - Module 2 Model = “2070-2A”
    - Module 2 Version = 20040608020200
    - Module 2 Type = 2
    - Module 3 Location = 3
    - Module 3 Make = 0
    - Module 3 Model = “2070-3B”
    - Module 3 Version = 20040608020200
    - Module 3 Type = 2
    - Module 4 Location = 5
    - Module 4 Make = 0
    - Module 4 Model = “2070-4B”
    - Module 4 Version = 20040608020200
    - Module 4 Type = 2
- Display properties:
    - # Char Lines = 8
    - #Char Columns = 40
    - #Graphic Rows-1 (y_max) = 63
    - #Graphic Columns-1 (x_max) = 239

- # Ethernets = 1
    - Ethernet 1 Type = 2
    - Ethernet 1 IP Address = 10.20.70.51
    - Ethernet 1 Switch/Router MAC Address = 000000 000000
    - Ethernet 1 Subnet Mask = 255.255.255.0
    - Ethernet 1 Default Gateway = 10.20.70.254
    - Ethernet 1 Engine Board Interface = 2
- SPI3 Purpose = 0
- SPI4 Purpose = 0
- Host Board Serial Ports Used = 2047:
- # Ports used for I/O = 1:
    - Port 1 ID = 5
    - Port 1 Mode = 3
    - Port 1 Baud Rate = 614400
- Serial Bus #1 Port = 128
- Serial Bus #2 Port = 0
- TS 2 Port 1 Port = 0
- Expansion Bus type = 1
- Encoded SPI Addressing = 1 (No)
- CRC #1 (16-bit) … N/A, simulate if required
- Latitude = 0.0 … or read from Datakey if present
- Longitude = 0.0 … or read from Datakey if present
- Controller ID = 65535 … or read from Datakey if present
- Communication Drop # = 65535 … or read from Datakey if present
- Can Bus = 0
- Host Board Serial Port Signals Present = 0x7F
- Reserved for Agency (35 bytes of 0xFF) … or read from Datakey if present
- CRC #2 (16-bit) … N/A, simulate if required … or read from Datakey if present

User Data (to end, each 0xFF) … N/A, or read from Datakey if present

CRC particulars are as follows:

- CRC polynomial to be TEES CCITT 16-4-2-1 with all ones preset.
- CRC1 runs from first EEPROM byte through CRC1.

CRC2 runs from Latitude through CRC2.

B.3.3 SPI Datakey Driver

The following standard Linux functions shall be supported by this driver:

- open( )
- close( )
- read ( )
- write( )
- lseek()
- ioctl()

The special device file node for the Datakey shall be “/dev/datakey.”

The ioctl function prototype is as follows:

int ioctl(int ***d***, int ***request***, ...);

The following ioctl requests shall be supported and are defined in the common atc.h header file.

ATC_DATAKEY_ERASE_ALL

Perform a Bulk Erase operation, which sets all bits to 1.

ATC_DATAKEY_ERASE_SECTOR

Perform a Sector Erase operation for the sector containing the address passed as a long integer parameter.

ATC_DATAKEY_READ_STATUS_BITS

Returns the contents of the device Status Register in the least significant byte of the value returned.

ATC_DATAKEY_WRITE_PROTECT_BITS

Sets the Block Protect Bits in the device Status Register to the value contained in the least significant bits of the value passed as a long integer parameter.

ATC_DATAKEY_GET_DEVICE_SIZE

Returns the total size of the device in bytes.

ATC_DATAKEY_GET_SECTOR_SIZE

Returns the size of one sector in bytes.

B.3.4 Datakey Header Format

The format of the first 28 bytes of the Datakey is defined in the following structure. Only Datakeys with **version** = 1 or 2 are currently defined. A non-defined value for **version** or **type** is an error. Bytes 29 to 63 are reserved for extension of the header format and shall contain the value 0xFF if unused. All additional memory following the 64th byte is undefined and available for application use.

```
typedef struct atc_datakey {
   uint16 fcs;          // 16 bit Frame Check Sequence (FCS) calculated as
                        // defined in clause 4.6.2 of ISO/IEC 3309. This FCS
                        // is calculated across bytes 3-64
   uint8 type;          // Key Type See table below
   uint8 version;       // Header Version: Only 1 and 2 are valid values
                        // currently.
   uint32 latitude;
   uint32 longitude;
   unit16 id;           // Controller ID
   unit16 drop;         // Communications drop number
   uint32 ipaddress;    // IP address
   uint32 subnet;       // Subnet mask
   uint32 gateway;      // Default gateway
   uint8 reserved[36]; // (padding for TEES match)
} ;
```

Structure member "**type**" shall contain the Key Type value as defined in the following table:

Key Type	Model No	Memory Size	Sector Size
3	SFK2M	2Mb	64 Kbytes
4	SFK4M	4Mb	64 Kbytes
5	SFK8M	8Mb	64 Kbytes
6	SFK32M	32Mb	64 Kbytes

The data format in the Datakey header for the Latitude and Longitude fields shall conform with IEEE/ANSI 754-1985 STD. All the other fields shall follow a Big Endian Format as implemented by Freescale 68000-family CPUs.

The Startup Override feature of the Model 2070 is not supported.

B.4 Standard Linux Drivers

B.4.1 Removable Storage Devices

Standard Linux drivers shall be provided to support removable USB and SD Card storage devices.

Removable storage devices using VFAT file systems shall be supported. Support for additional filesystems is optional.

The BSP shall support hotplug configuration, allowing removable storage devices to be attached and removed while the ATC is running. The ATC shall not require any removable storage device to be permanently attached.

When a removable storage device is attached to the ATC, the BSP shall automatically mount the device to the appropriate mount point listed in section A.1. When a removable storage device is removed from the ATC, the BSP shall automatically unmount the device.

B.4.1.1 USB

Standard Linux drivers required to implement a USB host interface with the mass storage device class shall be provided.

The first USB storage device to be attached shall be mounted to /media/usb. If multiple USB devices are attached simultaneously, then additional devices shall be mounted to /media/usb2, /media/usb3, etc. Devices shall always be mounted to the first-available mount point. The first partition of any attached USB device shall be auto-mounted at the device's designated mount point.

During boot-up, the ATC shall check for the presence of an executable binary or POSIX shell script named /media/usb/startup on an attached USB device. If found, the ATC shall automatically execute that file. This automatic startup from USB shall only be performed if the device is inserted at initial ATC boot-up, and NOT performed for devices newly inserted at any other time. For security purposes, the ATC shall include a user-configurable mechanism to permit enabling and disabling of this automatic startup feature. The maximum boot-up time requirement does not apply when the automatic startup feature is enabled.

B.4.1.2 SD Card

The first partition of each installed SD Card shall be auto-mounted at the mount point designated in Section A.1. SDSC and SDHC card capacities up to 32GB shall be supported. Support for SD cards of more than 32GB is optional.

B.4.2 Ethernet 1 and Ethernet 2

Standard Linux drivers required to implement two 10/100 Ethernet interfaces shall be provided.

B.4.3 Long-Term Non-Volatile Memory (FLASH) File System

Standard Linux drivers shall be provided to implement filesystem(s) on the provided non-volatile storage device(s). Features appropriate to the storage technology shall be included to maximize the lifespan of the storage device(s) and to prevent corruption due to power interruptions.

B.4.4 Static RAM

Standard Linux drivers required to implement an MTD-type persistent filesystem mounted as /sram on the +5 VDC Standby Power-backed static RAM shall be provided.

B.4.5 ATC SPx Asynchronous Serial Ports

Standard Linux drivers required to support the asynchronous serial ports shall be provided. These standard drivers shall provide the capability of reading (CTS, DCD) and controlling (RTS) the flow

control signals. These drivers shall also support the Linux tty flow control ioctl functions TIOCMGET/TIOCMSET with parameters TIOCM_RTS, TIOCM_CTS, TIOCM_CD and asynchronous notification of CTS/DCD changes.

B.5 ATC SPxs Synchronous Driver Interface

Overview

This section defines a generalized synchronous driver interface for the serial ports of the ATC Engine Board. The action and meaning of the functions and parameters selected by this standard shall follow those defined by the Open Group Base Specifications Issue 6 IEEE Std 1003.1, 2004 Edition. However, only those flags and commands defined here require implementation to ensure conformance.

B.5.1 Device Nodes

Each synchronous serial port shall have a special device file node located in the /dev directory. Device nodes shall be defined as follows:

Device	Port Node	/proc
SP1	/dev/sp1s	/proc/driver/sp1s
SP2	/dev/sp2s	/proc/driver/sp2s
SP3	/dev/sp3s	/proc/driver/sp3s
SP5	/dev/sp5s	/proc/driver/sp5s
SP8	/dev/sp8s	/proc/driver/sp8s

B.5.2 /proc File System

Each open port driver shall maintain a file entry in the /proc/driver/spxs directory. The driver shall generate ASCII data with the following format:

Owner PID	Bits/Sec	RX Count	TX Count	Receive Error Count

Example: “214 153600 5120 688 0” Defines process PID 214 running 153600 bps with 5120 bytes written, 688 bytes read, and zero errors.

B.5.3 SPxs Commands

open( )

This function allows an application to request exclusive ownership of a serial port. Upon opening, the port is made ready to be configured via an ioctl( ) function call and then accessed via the read( ), write( ) and close( ) functions.

Prototype

```
int open(const char* pathname, int flags);
```

Argument	Description
pathname	This is the name of the device to open.
flags	O_RDWR, O_NONBLOCK or O_NDELAY, O_SYNC, O_ASYNC

Return Value	Description
-1	An error occurred. Consult errno.
else	A valid file descriptor.

Errors	Description
ENODEV	Incorrect device specified or device cannot be found.
EBUSY	The device is already open or the device is busy
EINTR	The open system call was interrupted by a signal.

Example

```
      #include <errno.h>
      #include <stdio.h>
#include <fcntl.h>
      int fd;
      fd = open(/dev/sp5s, O_RDWR);              /* Get the file descriptor for
      sp5s */
      if (fd == -1))
            {
            printf( "open() failure on sp5s, errno = %d\n", errno ");
            }
            else
            …..
```

The returned file descriptor (**fd**) is the ownership “handle” passed by an application to the close, read, write, and ioctl functions.

close( )

This function allows an application to release ownership of the serial port pointed to by **fd**. This function should only be called after a successful open of the respective port. Upon closing the port, all settings and configurations are put in a reset state.

Prototype

```
      int close(int fd);
```

Argument	Description
fd	This is the file descriptor of the device to be closed.

Return Value	Description
-1	An error occurred. Consult errno.
0	The operation succeeded.

Errors	Description
EBADF	Incorrect file descriptor specified.
ENODEV	Incorrect device specified or device cannot be found.
EBUSY	The device is busy.
EINTR	The close system call was interrupted by a signal.

Example

```
      #include <errno.h>
      #include <stdio.h>
      #include <termios.h>
#include <fcntl.h>
int status;
      status = close(fd);
      if (status == -1)
      printf("close() failure, errno = %d\n", errno);
      return(status);
}
```

read( )

This function allows an application to read data from the open serial port pointed to by **fd**. This function should only be called after a successful open of the respective port. The function reads up to **count** bytes from the next available valid packet received. The function places the requested data in the memory location pointed to by ***buf.** When **count** bytes is less than the number of bytes in the next available packet, the remainder of the packet shall be discarded by the driver. The driver shall not copy the packet's CRC bytes to **buf**.

Prototype

int read(int fd, void *buf, size_t count);

Argument	Description
fd	This is the file descriptor of the device to access.
buf	The data read will be put here.
count	This is the desired number of bytes to read.

Return Value	Description
-1	An error occurred. Consult errno.
0 to count	The operation succeeded. If the return value is less than count, then the request timed out.
Errors	**Description**
EBADF	Incorrect file descriptor specified.
EFAULT	There was a problem copying data into the user specified buffer.
EAGAIN	Non-blocking has been specified and no data was immediately available for reading.
EINTR	The call was interrupted by a signal before any data was read.

Example

```
      #include <errno.h>
      #include <stdio.h>
      #include <termios.h>
#include <fcntl.h>
int status;
      status = read(fd, &buf, count);
      if (status == -1)
      printf("read() failure, errno = %d\n", errno);
      else
```

write( )

This function allows an application to write data to the open serial port pointed to by **fd**. This function should only be called after a successful open of the respective device. The function writes up to **count** bytes to be sent out as a single packet. The function takes the data from the memory location pointed to by ***buf.** The driver shall add the crc bytes to the data copied from **buf** as part of sending the out the packet. If **count** bytes cannot be sent as a single packet, the EINVAL error shall be returned and no packet shall be sent.

Prototype

int write(int fd, const void *buf, size_t count);

Argument	Description
fd	This is the file descriptor of the device to access.
buf	The data written comes from here.
count	This is the desired number of bytes to write.

Return Value	Description
-1	An error occurred. Consult errno.
0 to count	The operation succeeded. If the return value is less than count, then the request timed out

Errors	Description
EBADF	Incorrect file descriptor specified.
ENODEV	Incorrect device specified or device cannot be found.
EFAULT	There was a problem copying data from the user specified buffer.
EAGAIN	Non-blocking I/O has been selected using O_NONBLOCK and the write would block.
EINTR	The call was interrupted by a signal before any data was written
EINVAL	The maximum number of bytes to be written has been exceeded.

Example

```
      #include <errno.h>
      #include <stddef.h>
      #include <stdio.h>
      #include <unistd.h>
      #include <termios.h>
      #include <fcntl.h>
      int status;
      status = write(fd, &buf, count);
      if (status == -1)
printf("write() failure, errno = %d\n", errno);
```

poll( )

The poll( ) method shall be used to determine if a read or write to a device will block.

Prototype

```
int poll(struct pollfd fds[], nfds_t nfds, int timeout);
```

Argument	Description
Fds	This defines a table of file descriptor structures.
Nfds	Number of structures in the table.
timeout	Time out in milliseconds.

Return Value	Description
>0	Number of structures which have non-zero event fields
0	Call timed out.
-1	Error

Errors	Description
EBADF	An invalid file descriptor was given in one of the sets.
ENOMEM	There was no space to allocate file descriptor tables.
EINTR	A signal occurred before any requested event.
EINVAL	Invalid argument

Required Flags	
POLLOUT	Data may be written without blocking
POLLIN	Data is available for reading
POLLERR	There is some error with the port
POLLNVAL	Invalid request: fd not open

Example

```
      #include <stropts.h>
      #include <poll.h>
      ...
      struct pollfd fds[1];
      int timeout_msecs = 10;
      int ret;
int i;
      /* Open SCC device. */
      fds[0].fd = open("/dev/sp5s", ...);
fds[0].events = POLLOUT | POLLIN | POLLERR;
      ret = poll(fds, 1, timeout_msecs);

      if (ret > 0) {
      /* An event on one of the fds has occurred. */
      … if (fds[i].revents & POLLOUT) {
           /* Data may be written .*/
      ...
           }
           if (fds[i].revents & POLLIN) {
           /*Data is available for reading*/
      ...
           }
           if (fds[i].revents & POLLERR) {
           /* There is some error with the SCC*/
      ...
           }
}
```

fcntl( )

The fcntl( ) method shall be used to enable or disable driver blocking or asynchronous notification operations.

Prototype

```
int fcntl(int fd, int cmd);
int fcntl(int fd, int cmd, flags);
```

Argument	Description
Fd	File descriptor
Cmd	Command
Flags	Command options

Return Value	Description
F_GETOWN	Value of process owner.
F_GETFL	Value of file status flags.
F_GETSIG	Value of signal sent when read or write becomes possible or zero for traditional SIGIO behavior.
-1	Error

Errors	Description
EBADF	fd is not an open file descriptor
EINVAL	Invalid argument

Commands	
F_GETOWN	Get ID of process receiving signals from this file descriptor
F_SETOWN	Set calling process as owner
F_GETFL	Get the file status flags
F_SETFL	Set the file status flags
F_GETSIG	Get the signal sent when input or output becomes possible
F_SETSIG	Sets the signal sent when input or output becomes possible

Flags	
O_ASYNC	Set asynchronous notification mode
O_NONBLOCK	Non-blocking mode

Example

```
        #include “atc_spx.h”
  int fd, flags;
        fd = open(“/dev/sp5s”, O_RDWR, O_NONBLOCK);
        fcntl (fd, F_SETOWN, getpid());
        flags = fcntl (fd, FGETFL);
        fcntl (fd, F_SETFL, flags | FASYNC);
```

ioctl( )

This function allows an application to configure, control and monitor status of the serial port pointed to by **fd**. This function should only be called after a successful open of the respective device. The operation performed by the ioctl( ) function depends on the **command** argument. The **command** argument determines the interpretation of any additional arguments. The supported IOCTL services are defined below.

Prototype

```
      int ioctl(int fd, int command, parameters);
```

Argument	Description
fd	This is the file descriptor of the device to access
command	This specifies the desired operation to be performed
parameter	This argument is an integer or a pointer to a source structure containing port configuration data or an integer or a destination structure where status Information is placed by the ioctl function call

Return Value	Description
-1	An error occurred. Consult errno
0	The operation succeeded

Errors	Description
EBADF	Invalid file descriptor specified
EFAULT	There was a problem accessing data from the user specified buffer
ENOTTY	An invalid command parameter was specified.
EINVAL	An invalid command parameter was specified

Valid SPxs IOCTL Commands

ATC_SPXS_WRITE_CONFIG

This command passes the **atc_spxs_config** structure to the serial port pointed to by **fd.** This command is used to set the baud rate, protocol, and clocking options of a port.

ioctl( ) Argument	Description
fd	This is the file descriptor of the device to access
command	ATC_SPXS_WRITE_CONFIG
parameter	* struct atc_spxs_config

Example

```
      #include <errno.h>
      #include <stdio.h>
      #include <sys/ioctl.h>
      #include <termios.h>
#include <fcntl.h>
int status;
      status = ioctl(fd, ATC_SPXS_WRITE_CONFIG, & atc_sp1s_configure);
      if  (status == -1)
      printf("ioctl() failure, errno = %d\n", errno);
      …
…
```

ATC_SPXS_READ_CONFIG

This command copies the data from the port's **atc_spxs_config** structure to user space pointed to by **parameter**. This command is used to check the state of the baud rate, protocol and clocking options of a port.

ioctl( ) Argument	Description
fd	This is the file descriptor of the device to access.
command	ATC_SPXS_READ_CONFIG
parameter	Pointer to user space destination.

Example

```
      #include <errno.h>
      #include <stdio.h>
      #include <sys/ioctl.h>
      #include <termios.h>
#include <fcntl.h>

int status;

      status = ioctl(fd, ATC_SPXS_READ_CONFIG, & port_configuration);
      if (status == -1)

      printf("ioctl() failure, errno = %d\n", errno);
```

FIONREAD

This command returns the number of byes to be read from the next available valid packet received. If no packet is available the call will return success and a count of zero.

ioctl( ) Argument	Description
fd	This is the file descriptor of the device to access
command	FIONREAD
parameter	Pointer to user space destination.

Example

```
      #include <errno.h>
      #include <stdio.h>
      #include <sys/ioctl.h>
      #include <termios.h>
#include <fcntl.h>

int status;

      status = ioctl(fd, FIONREAD, &bytes_available);
      if (status == -1)

      printf("ioctl() failure, errno = %d\n", errno);
```

B.5.4 ATC SPxs Data Structures

The defined structure member values are shown in parentheses. Structure members that are not applicable to a port's selected protocol shall be ignored by the driver. Upon receipt of an invalid, as compared to not applicable, argument, the ioctl( ) function shall generate an EINVAL error.

The driver shall maintain one **atc_spxs_config** structure for each SPxs port.

```
typedef struct atc_spxs_config
   {
   uint8_t protocol;                  // set the port protocol
       // (ATC_SDLC)
   uint8_t baud;                      // set the port baud rate
                                      // (ATC_B1200,   ATC_B4800,  ATC_B9600,
                                      //  ATC_B19200,  ATC_B38400, ATC_B57600,
                                      //  ATC_B76800,  ATC_115200,
       //  ATC_B153600, ATC_B614400)
   uint8 transmit_clock_source;  // (ATC_CLK_INTERNAL, ATC_CLK_EXTERNAL)
   uint8 transmit_clock_mode;         // sets whether the sync transmit clock
                                      //  is on continuously or bursts with the
                                      //  data frame
                                      // (ATC_CONTINUOUS, ATC_BURST)
   } atc_spxs_config_t;
```

B.5.5 Constants Defined by this Standard for SPxs Synchronous Drivers

These constants shall reside in a file named **atc_spxs.h**. IOCTL commands are defined according to Linux conventions. See *include/asm/ioctl.h.*

```
#include <asm/ioctl.h>
      #ifndef _ATC_SPXS_H
      #define _ATC_SPXS_H

      // 'ioctl' commands
      #define ATC_SPXS_WRITE_CONFIG       0
#define ATC_SPXS_READ_CONFIG        1
      // Available synchronous channels
      #define ATC_LKM_SP1S          1
      #define ATC_LKM_SP2S          2
      #define ATC_LKM_SP3S          3
      #define ATC_LKM_SP5S          5
#define ATC_LKM_SP8S          8
      // Available communications protocols
      #define ATC_SDLC              0     // Default
      #define ATC_SYNC              1
#define ATC_HDLC              2
      // Available baud rates
      #define ATC_B1200       0
      #define ATC_B2400       1
```

```
      #define ATC_B4800       2
      #define ATC_B9600       3
      #define ATC_B19200      4
      #define ATC_B38400      5
      #define ATC_B57600      6
      #define ATC_B76800      7
      #define ATC_B115200     8
      #define ATC_B153600     9
#define ATC_B614400     10
const int ATC_B[] = {1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200,
153600, 614400};
      // Available clock sources
      #define ATC_CLK_INTERNAL            0     // Default
#define ATC_CLK_EXTERNAL            1
      // Available transmit clock modes
      #define ATC_GATED             0           // Default
      #define ATC_CONTINUOUS        1

      typedef struct atc_spxs_config {
          unsigned char protocol;
          unsigned char baud;
          unsigned char transmit_clock_source;
          unsigned char transmit_clock_mode;
} atc_spxs_config_t;
#endif /* _ATC_SPXS_H */
```

B.6 ATC SPxs Synchronous Driver Kernel Level Interface

B.6.1 Overview

This section defines a generalized synchronous driver Kernel Level interface for the serial ports of the ATC Engine Board. The action and meaning of the functions and parameters selected by this standard shall follow those defined by the Open Group Base Specifications Issue 6 IEEE STD 1003.1, 2004 Edition. However, only those flags and commands defined here require implementation to ensure conformance.

An additional Kernel Level interface is required to allow Kernel Level programming usage of the SPxs Synchronous Driver. The symbols mentioned herein must be exported to the Kernel so that Loadable Kernel Modules (LKM) may link to these functions.

B.6.2 Channel Numbers

Each synchronous serial port shall have a special channel number associated with it. Channel numbers shall be defined as follows:

Device	Channel Number
SP1	1
SP2	2
SP3	3
SP5	5
SP8	8

B.6.3 SPxs Functions

All function symbols listed herein must be exported to the Kernel environment to allow linking by Loadable Kernel Modules (LKM).

sdlc_kernel_open( )

This function allows kernel code to request exclusive ownership of a serial port, for read and write access. Upon opening, the port is made ready to be configured via an sdlc_kernel_ioctl( ) function call and then accessed via the sdlc_kernel_read( ), sdlc_kernel_write( ) and sdlc_kernel_close( ) functions.

Prototype

```
void * sdlc_kernel_open( int channel );
```

Argument	Description
channel	This is the channel number to open.

Return Value	Description
IS_ERR()	An error occurred, or the channel could not be opened. IS_ERR() is called with the returned void * to determine if an error occurred.
Pointer to driver block	A pointer to a device driver data block, specific to this driver and channel. This pointer will be passed in subsequent sdlc_kernel_ calls.

Errors	Description
ENODEV	Incorrect device specified or device cannot be found.
EBUSY	The device is already open or the device is busy.
EINTR	The open system call was interrupted by a signal.

Example

```
void *context;
context = sdlc_kernel_open(ATC_LKM_SP5S);              /* Get the file
context for sp5s */
if ( IS_ERR( context) )
      {
      printk( KERN_ALERT "sdlc_kernel_open() failure on sp5s, errno =
            %d\n", PTR_ERR( context ) );
      }
else
```

The returned context is the ownership “handle” passed by kernel code to the sdlc_kernel_close( ), sdlc_kernel_read( ), sdlc_kernel_write( ) and sdlc_kernel_ioctl( ) functions.

sdlc_kernel_close( )

This function allows kernel code to release ownership of the serial port pointed to by context. This function should only be called after a successful sdlc_kernel_open( ) of the respective port and after any previous write calls have had time to be transmitted. Upon closing the port, all settings and configurations are put in a reset state and the context is no longer valid. This call shall immediately perform the close and return without delay.

Prototype

```
int sdlc_kernel_close( void *context );
```

Argument	Description
context	This is the context of the port to be closed.

Return Value	Description
<0	An error occurred. The negative error code is returned.
0	The operation succeeded.

Errors	Description
EBADF	Incorrect file descriptor specified.
EINTR	The close system call was interrupted by a signal.

Example

```
int status;
status = sdlc_kernel_close( context);
if ( 0 != status )
      printk( KERN_ALERT "sdlc_kernel_close() failure, error = %d\n",
status);
return(status);
```

sdlc_kernel_read( )

This function allows kernel code to read data from the open serial port referenced by the void *context. This function should only be called after a successful sdlc_kernel_open() of the respective port. The function reads up to **count** bytes from the next available valid packet received. The function places the requested data in the memory location pointed to by ***buf.** When **count** bytes is less than the number of bytes in the next available packet, the remainder of the packet shall be discarded by the driver. The driver shall not copy the packet's CRC bytes to **buf**. This is a non-blocking operation. The function will return 0 (zero) if no data is available.

Prototype

ssize_t sdlc_kernel_read(void *context, void *buf, ssize_t count);

Argument	Description
Context	This is the context of the port to access.
Buf	The data sdlc_kernel_read will be put here.
Count	This is the desired number of bytes to read.

Return Value	Description
< 0	An error occurred. The negative error code is returned.
Count	The operation succeeded. The number of bytes actual read is returned.

Errors	Description
EBADF	Incorrect file descriptor specified.
EFAULT	There was a problem copying data into the user specified buffer.
EINTR	The call was interrupted by a signal before any data was read.

Example

```
ssize_t status;
      status = sdlc_kernel_read( context, &buf, count );
      if ( 0 > status )
            printk( KERN_ALERT "sdlc_kernel_read() failure, error = %d\n",
      status );
```

```
else
```

sdlc_kernel_write( )

This function allows kernel code to write data to the open serial port referenced by the void *context. This function should only be called after a successful open of the respective device. The function writes count bytes to the driver to be sent out as a single packet. The function takes the data from the memory location pointed to by ***buf.** The driver shall add the crc bytes to the data copied from **buf** as part of sending the out the packet. If count bytes cannot be sent as a single packet, the EINVAL error shall be returned and no packet shall be sent. This is a non-blocking operation; therefore if the write would block the call shall return immediately with an EAGAIN error.

Prototype

```
ssize_t write( void *context, const void *buf, ssize_t count);
```

Argument	Description
context	This is the context of the port to access.
buf	The data written comes from here.
count	This is the desired number of bytes to write.

Return Value	Description
< 0	An error occurred. The negative error code is returned.
count	The operation succeeded.

Errors	Description
EBADF	Incorrect file descriptor specified.
EFAULT	There was a problem copying data from the user specified buffer.
EINTR	The call was interrupted by a signal before any data was written.
EINVAL	The maximum number of bytes to be written has been exceeded.
EAGAIN	Returned if the write operation would block.

Example

```
ssize_t status;
status = sdlc_kernel_write( context, &buf, count );
if ( 0 > status )
      printk( KERN_ALERT "sdlc_kernel_write() failure, error = %d\n",
status );
```

sdlc_kernel_ioctl( )

This function allows kernel code to configure, control and monitor status of the serial port referenced. by void *context. This function should only be called after a successful sdlc_kernel_open( ) of the respective port. The operation performed by the sdlc_kernel_ioctl( )

function depends on the command argument. The command argument determines the interpretation of any additional arguments. The supported IOCTL services are defined below.

Prototype

```
int sdlc_kernel_ioctl( void *context, int command, parameters );
```

Argument	Description
context	This is the context of the port to access.
command	This specifies the desired operation to be performed.
parameters	This argument is an integer or a pointer to a source structure containing port configuration data or an integer or a destination structure where status Information is placed by the sdlc_kernel_ioctl( ) function call.

Return Value	Description
< 0	An error occurred. The negative error code is returned.
0	The operation succeeded.

Errors	Description
EBADF	Invalid file descriptor specified.
EFAULT	There was a problem accessing data from the user specified buffer.
ENOTTY	An invalid command parameter was specified
EINVAL	An invalid command parameter was specified.

Valid SPxs IOCTL Commands

ATC_SPXS_WRITE_CONFIG

This command passes the atc_spxs_config structure to the serial port pointed to by context. This command is used to set the baud rate, protocol and clocking options of a port. The atc_spxs_config structure is defined in Section B.5.4

ioctl( ) Argument	Description
context	This is the context of the device to access.
command	ATC_SPXS_WRITE_CONFIG
parameter	struct atc_spxs_config * Pointer to a kernel space config structure to write.

Example

```
int status;

      status = sdlc_kernel_ioctl( context, ATC_SPXS_WRITE_CONFIG,
      &atc_sp1s_configure);
      if  ( 0 > status )

      printk(  KERN_ALERT "sdlc_kernel_ioctl() failure, error = %d\n", status);
```

ATC_SPXS_READ_CONFIG

This command copies the data from the port's atc_spxs_config structure to the buffer pointed to by parameter. This command is used to check the state of the baud rate, protocol and clocking options of a port. The atc_spxs_config structure is defined in the Advanced Transportation Controller (ATC) standard Section B.5.4.

ioctl( ) Argument	Description
context	This is the context of the device to access.
command	ATC_SPXS_READ_CONFIG
parameter	struct atc_spxs_config * Pointer to a kernel space destination.

Example

```
int status;
      status = sdlc_kernel_ioctl( context, ATC_SPXS_READ_CONFIG,
      &port_configuration );
      if ( 0 > status )
printk( KERN_ALERT "sdlc_kernel_ioctl() failure, error = %d\n", status );
```

FIONREAD

This command returns the number of byes to be read from the next available valid packet received. If no packet is available the call will return success and a count of zero.

ioctl( ) Argument	Description
context	This is the context of the device to access.
command	FIONREAD
parameter	Pointer to user space destination

Example

```
      #include <errno.h>
      #include <stdio.h>
      #include <sys/ioctl.h>
      #include <termios.h>
#include <fcntl.h>
int status;
      status = sdlc_kernel_ioctl (fd, FIONREAD, &bytes_available);
      if (status == -1)
      printf("ioctl() failure, errno = %d\n", errno);
```

B.7 Time of Day Driver

The following standard Linux functions shall be supported by this driver:

- `open( )`
- `close( )`
- `read( )`
- `write( )`
- `ioctl( )`

The special device file node for the time of day device shall be “/dev/tod.”

open()

Allows an application to request shared access to the time of day device. The device supports the O_RDONLY, O_WRONLY, and O_RDWR open flags.

close()

Closes the time of day device. Closing the device cancels any requested signals.

read()

Read operations are ignored. All reads shall return 0.

write()

Write operations are ignored. All writes shall return 0.

ioctl()

Allows an application to configure and control the time of day device. This function shall only be called after a successful open of the respective device. The operation performed by the ioctl( ) function depends the command argument. The command argument determines the interpretation of any additional arguments. The supported IOCTL services are defined below.

Prototype

int ioctl(int fd, int command, parameters);

Argument	Description
fd	This is the file descriptor of the device to access.
command	This specifies the desired operation to be performed.
parameter	This argument is an integer or a pointer to a source structure containing port configuration data or an integer or a destination structure where status Information is placed by the ioctl function call.

Return Value	Description
-1	An error occurred. Consult errno.
else	The operation succeeded.

Errors	Description
EBADF	Invalid file descriptor specified.
EFAULT	There was a problem accessing data from the user specified buffer.
EINVAL	An invalid command parameter was specified.

ATC_TOD_SET_TIMESRC

This commands sets the time source for the TOD driver.

External time references, such as NTP and GPS, may be used to set controller time but are not directly supported by the TOD driver. When an external time reference is in use, set the TOD driver time source to ATC_TIMESRC_CRYSTAL to allow standard Linux timekeeping functions to operate normally.

Guidance: The ATC Application Programming Interface (API) may provide direct support for the use of external time references.

Available time sources:

ATC_TIMESRC_LINESYNC

(LINESYNC pulses from AC power line)

ATC_TIMESRC_RTCSQWR

(square wave output from Real Time Clock (RTC) IC

ATC_TIMESRC_CRYSTAL

(primary microprocessor crystal)

(standard Linux timekeeping function)

ATC_TIMESRC_EXTERNAL1

(deprecated, non-functional)

ATC_TIMESRC_EXTERNAL2

(deprecated, non-functional)

Ioctl() Argument	**Description**
fd	This is the file descriptor of the device to access.
command	ATC_TOD_SET_TIMESRC
parameter	Time source integer.

Errors:

EINVAL Hardware does not support setting the time source requested.

ATC_TOD_GET_TIMESRC

This command returns the current time source. The time sources are defined above.

Ioctl() Argument	Description
fd	This is the file descriptor of the device to access.
command	ATC_TOD_GET_TIMESRC
parameter	None.

ATC_TOD_GET_INPUT_FREQ

This command returns the current frequency that is driving the time of day clock. 0 is returned if the frequency is unknown or unusable as a tick signal.

Ioctl() Argument	Description
fd	This is the file descriptor of the device to access.
command	ATC_TOD_GET_INPUT_FREQ
parameter	None.

ATC_TOD_REQUEST_TICK_SIG

This command requests a signal to be sent at each tick of the time of day clock as long as the file device remains opened. The param value passed to ioctl is the signal number that should be sent to the calling process at each time of day clock tick.

Ioctl() Argument	Description
fd	This is the file descriptor of the device to access.
command	ATC_TOD_REQUEST_TICK_SIG
parameter	Signal number integer.

Errors:

EINVAL The signal number is not a valid signal or the input frequency is unknown or unusable as a tick signal.

ATC_TOD_CANCEL_TICK_SIG

This releases the signal from being sent when the time of day clock ticks. If the file device is closed, the signal is automatically released.

Ioctl() Argument	Description
fd	This is the file descriptor of the device to access.
command	ATC_TOD_CANCEL_TICK_SIG
parameter	None.

ATC_TOD_REQUEST_ONCHANGE_SIG

This command requests a signal to be sent each time the system time is changed. The param value passed to ioctl is the signal number that should be sent to the calling process.

Ioctl() Argument	Description
fd	This is the file descriptor of the device to access.
command	ATC_TOD_REQUEST_ONCHANGE_SIG
parameter	Signal number integer.

Errors:

EINVAL The signal number is not a valid signal.

ATC_TOD_CANCEL_ONCHANGE_SIG

This releases the signal from being sent when the system time is changed. If the file device is closed, the signal is automatically released.

Ioctl() Argument	Description
fd	This is the file descriptor of the device to access.
command	ATC_TOD_CANCEL_ONCHANGE_SIG
parameter	None.

B.8 General ATC Include File Definitions

These constants shall reside in a file named **atc.h**.

```
#ifndef __ATC_H
#define __ATC_H
// Device File Names
#define ATC_HOST_EEPROM_DEV "/dev/eeprom"
#define ATC_DATAKEY_DEV "/dev/datakey"
#define ATC_GPIO_POWERDOWN_DEV "/dev/powerdown"
#define ATC_GPIO_DATAKEY_DEV "/dev/datakeypresent"
#define ATC_GPIO_CPUACTIVE_DEV "/dev/cpuactive"
#define ATC_GPIO_CPURESET_DEV "/dev/cpureset"
#define ATC_TIMING_TOD_DEV "/dev/tod"
#define ATC_SP1 "/dev/sp1"
#define ATC_SP2 "/dev/sp2"
#define ATC_SP3 "/dev/sp3"
#define ATC_SP4 "/dev/sp4"
#define ATC_SP5 "/dev/sp5"
#define ATC_SP6 "/dev/sp6"
#define ATC_SP8 "/dev/sp8"
#define ATC_SP1S "/dev/sp1s"
```

```
#define ATC_SP2S "/dev/sp2s"
#define ATC_SP3S "/dev/sp3s"
#define ATC_SP5S "/dev/sp5s"
#define ATC_SP8S "/dev/sp8s"
// Datakey Driver Definitions
#define ATC_DATAKEY_ERASE_ALL             _IO('D', 1)
#define ATC_DATAKEY_ERASE_SECTOR          _IOW('D', 2, unsigned long)
#define ATC_DATAKEY_READ_STATUS_BITS      _IOR('D', 3, unsigned long)
#define ATC_DATAKEY_WRITE_PROTECT_BITS    _IOW('D', 4, unsigned long)
#define ATC_DATAKEY_GET_DEVICE_SIZE       _IOR('D', 5, unsigned long)
#define ATC_DATAKEY_GET_SECTOR_SIZE _IOR('D', 6, unsigned long)
// Time of Day Driver Definitions
#define ATC_TOD_SET_TIMESRC               _IOW('T', 1, unsigned long)
#define ATC_TOD_GET_TIMESRC               _IO('T', 2)
#define ATC_TOD_GET_INPUT_FREQ            _IO('T', 3)
#define ATC_TOD_REQUEST_TICK_SIG          _IOW('T', 4, unsigned long)
#define ATC_TOD_CANCEL_TICK_SIG           _IO('T', 5)
#define ATC_TOD_REQUEST_ONCHANGE_SIG      _IOW('T', 6, unsigned long)
#define ATC_TOD_CANCEL_ONCHANGE_SIG _IO('T', 7)
enum timesrc_enum
{
   ATC_TIMESRC_LINESYNC,
   ATC_TIMESRC_RTCSQWR,
   ATC_TIMESRC_CRYSTAL,
   ATC_TIMESRC_EXTERNAL1,
   ATC_TIMESRC_EXTERNAL2
};
#endif
```

APPENDIX C: HISTORICAL BACKGROUND OF TRAFFIC CONTROLLERS

Many of the design choices in this standard are based on historical trends. This history is included to provide a framework for the decisions represented in this standard. It is also recognized that many legacy systems are presently deployed and that any new technology, such as that specified here, must be capable of interfacing accurately and readily within existing networks of deployed equipment. Therefore, it is appropriate to document the known characteristics of elements of the deployed network.

In the early 1970s, two concurrent traffic controller standards efforts were initiated in North America. These were the Model 170 standard and the National Electrical Manufacturers Association (NEMA) standard. A brief history of these two standards efforts and the later Model 2070 Controller Standard are presented in the following subsections: .

NEMA

The NEMA standard(s) stemmed from a group of manufacturers who joined NEMA and assembled a core of experienced traffic and electronic engineers to define the first NEMA traffic signal controller. The controller development consisted of an interchangeable electronic device with standard connectors. The NEMA standard further defined traffic terminology and minimum traffic signal control software functionality. Various user agencies that included state, city, and county government officials were included in this initial definition of the standard.

The initial standard included the standardization of connectors and connections for three MS-style connectors. The inputs and outputs were defined and standardized with respect to electrical levels as well as function.

The development process ultimately yielded a document labeled the “TS-1” *Traffic Controller Assemblies* - Standard in 1983. The NEMA standard also defined peripheral devices used in the controller industry and eventually defined the cabinet. The NEMA process requires that every 6 years the standard is updated and re-ratified. The standard did not cover communications between devices, nor did the standard provide for interchangeability of software functions. During subsequent years, the demand for communications to provide data transfers between local controllers and central control or on-street master systems increased rapidly. The original TS-1 standard had not defined communication and subsequently a non-standard fourth connector evolved that did not allow interchangeability. The TS-1 1989 revision defined/standardized actuated intersection control; provided standards for all cabinet components and added test procedures; and improved interchangeability between manufacturers equipment.

Over the years, further definitions were recommended to define a safer cabinet to controller interface. This new recommendation included a full SDLC communication protocol to allow the traffic controller and the conflict monitor to communicate between each device and check the intended output with what was actually being displayed by the cabinet.

This effort generated the most recent "TS-2" standard in 1992, later updated in 1998 and 2003, and most recently updated in 2016. The standard outlines an expandable and interchangeable traffic controller, cabinets, and peripherals. The TS2 standard replaced individual Parallel I/O lines with time slots in a high-speed serial data stream, reducing the amount of cabinet wiring and allowing easier addition of new features. The standard, however, did not accommodate interchangeable software among the various manufacturers. Features found in one software package were not available in other packages. The front panel displays and the information displayed were also unique and non-standardized.

Model 170 Specification

The Model 170 specification was developed by Caltrans and NYSDOT to address needs for an "open systems" controller for transportation applications. Unlike the NEMA standard, the Model 170 defined controller hardware but not software functionality. The Model 170 approach allows software from any source to be loaded and executed on the controller. The Model 170 obtains its hardware / software independence by requiring, by part number specification, the use of specific integrated circuit chips (for CPU and Serial Communications functions). In addition, a memory map was defined so that software developers would know precisely where to address input and output functions regardless of who manufactured the hardware unit.

While the Model 170's architecture has been enormously successful and achieves the desired independence of the hardware and software, the Model 170 relied heavily on the specific Motorola CPU and serial communications chips (or suitable substitutes). Unfortunately, these chips have been designated for phased-out obsolescence. The issue is further compounded by the relatively poor computational performance of the Model 170, compared to today's controller systems. The applications software written for the Model 170 CU is written in assembly language which makes it difficult to move to a different CPU. Also, the Model 170, without a dedicated CPU for communications, cannot handle the performance demands of today's modern packet-based, high-speed communications networks. Few options currently exist for those agencies heavily invested in Model 170 software/hardware to preserve their investments in Model 170 applications software.

Model 2070 Standard

The Model 2070 is an "open systems" controller system and is recognized explicitly within this standard. It was originally developed by Caltrans and City of Los Angeles to address some of the shortfalls associated with the Model 170 as discussed above. Its designers tried to mitigate some of the potential parts obsolescence issues associated with the Model 170. Instead of relying on the efficiency of assembly language programming, the Model 2070 CU includes the necessary resources to execute programs written in high level programming languages such as ANSI C or C++. Such high-level language programs are more easily written and debugged and are capable of being ported to other hardware platforms as necessary. The Model 2070 makes use of an O/S (OS-9 or Linux depending on the configuration) to separate the hardware from the application software). By specifying an O/S, the explicit mapping of user memory and field I/O, as was done with the Model 170, is no longer necessary. The O/S and associated standardized support functions take care of many of the basic execution management and scheduling tasks required by application software programs. The O/S further extends the hardware/software independence through I/O and memory resource sharing capabilities. These capabilities allow multiple independent applications to be run simultaneously on a single controller unit in a multi-tasking mode. This was not the case with a Model 170. The Caltrans TEES now has a configuration of the Model 2070 controller that meets this standard.

The Model 2070 Controller Standard also provides for greater subcomponent interchangeability and modularity than the Model 170. Model 2070 component modules are defined through specification such that they are interchangeable among different manufacturers. With the Model 170, only the modem/communication and memory modules are interchangeable among controllers produced by different manufacturers.

In the early years of the ATC program, Model 2070 unit of any configuration was considered an ATC. This is no longer the case. Unless the Model 2070 is of a configuration that conforms to this standard, it is not an ATC. Purchasers need to be aware of this and explicitly require conformance to the ATC 5201 Standard along with their 2070 specifications. Unfortunately, some vendors may use the ATC identifier on controller products that are not ATCs.