Advanced Transportation Controller (ATC)
Standard

January 25, 2006

This is a draft document that is distributed for review and ballot purposes only. You may reproduce and distribute this document within your organization, but only for the purposes of and only to the extent necessary to facilitate review and ballot to AASHTO, ITE, or NEMA. Please ensure that all copies reproduced or distributed bear this legend. This document contains recommended information that is subject to approval.

Published by

American Association of State Highway and Transportation Officials (AASHTO)
444 North Capitol St. NW, Suite 249
Washington, DC  20001

Institute of Transportation Engineers (ITE)
1099 14th St. NW, Suite 300 West
Washington, DC  20005-3438

National Electrical Manufacturers Association (NEMA)
1300 N. 17th St., Suite 1847
Rosslyn, VA  22209-3801

© Copyright 2006 AASHTO/ITE/NEMA. All rights reserved.
NOTICE

These materials are delivered "AS IS" without any warranties as to their use or performance.

AASHTO/ITE/NEMA AND THEIR SUPPLIERS DO NOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THESE MATERIALS. AASHTO/ITE/NEMA AND THEIR SUPPLIERS MAKE NO WARRANTIES, EXPRESSED OR IMPLIED, AS TO NONINFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AASHTO, ITE, OR NEMA OR THEIR SUPPLIERS BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY CLAIM OR FOR ANY CONSEQUENTIAL, INCIDENTAL, OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, ARISING FROM YOUR REPRODUCTION OR USE OF THESE MATERIALS. EVEN IF AN AASHTO, ITE, OR NEMA REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential, or special damages, or exclusion of implied warranties, so the above limitations may not apply to you.

Use of these materials do not constitute an endorsement or affiliation by or between AASHTO, ITE, or NEMA and you, your company, or your products and services.

If you are not willing to accept the foregoing restrictions, you should immediately return these materials.

ATC is a trademark of NEMA/AASHTO/ITE.
# Advanced Transportation Controller (ATC) Standard

## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FOREWORD</td>
</tr>
<tr>
<td>2</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>2.1</td>
<td>SCOPE</td>
</tr>
<tr>
<td>2.2</td>
<td>KEY ELEMENTS OF THE ATC CONTROLLER STANDARD</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Form/Fit/Function</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Engine Board</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Communications and User Interfaces</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Parallel and Serial I/O</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Operating System &amp; Board Support Package Requirements</td>
</tr>
<tr>
<td>2.3</td>
<td>REFERENCES</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Normative References</td>
</tr>
<tr>
<td>3</td>
<td>CONCEPT OF OPERATIONS</td>
</tr>
<tr>
<td>3.1</td>
<td>PROBLEM STATEMENT</td>
</tr>
<tr>
<td>3.2</td>
<td>HISTORICAL BACKGROUND</td>
</tr>
<tr>
<td>3.2.1</td>
<td>NEMA</td>
</tr>
<tr>
<td>3.2.2</td>
<td>The Model 170 Specification</td>
</tr>
<tr>
<td>3.2.3</td>
<td>The ATC 2070 Standard</td>
</tr>
<tr>
<td>3.3</td>
<td>FUNCTIONAL NEEDS</td>
</tr>
<tr>
<td>3.4</td>
<td>OPERATIONAL ENVIRONMENT</td>
</tr>
<tr>
<td>3.5</td>
<td>REPRESENTATIVE USAGE</td>
</tr>
</tbody>
</table>
3.5.1 Manage/Configure Controller
Applications ....................................................... 3-9

3.5.2 Manage External Devices .................................. 3-11

3.5.3 Facilitate Ease of Maintenance & Future
Hardware and Software Updates ............................... 3-11

3.6 SECURITY ............................................................. 3-12

3.7 MODES OF OPERATION ........................................ 3-13

4 FUNCTIONAL REQUIREMENTS ............................... 4-1

4.1 MANAGE/CONFIGURE CONTROLLER
APPLICATIONS .......................................................... 4-1

4.1.1 Install and Update Applications
Software .................................................................. 4-1

4.1.2 Installing and Upgrading the Operating
System Software ...................................................... 4-2

4.1.3 Maintain Clock/Calendar Function and
Synchronize with External Sources ......................... 4-2

4.1.4 Configure and Verifying Parameter(s) ............... 4-3

4.1.5 Uploading/Downloading Data Block(s) ............... 4-3

4.1.6 Monitoring and Verifying Present
Application Status .................................................. 4-4

4.1.7 Allowing Operator Control of
Application(s) ........................................................ 4-4

4.1.8 Facilitate the Long Term Retention of
Data ......................................................................... 4-5

4.2 MANAGE EXTERNAL DEVICES ............................ 4-5

4.2.1 Manage/Control a Variety of External
Field Devices .......................................................... 4-5

4.2.2 Monitor the Status of External Field
Devices ..................................................................... 4-6
## 4.3 Facilitate Ease of Maintenance & Future Hardware or Software Updates

4.3.1 Board Support Package (BSP)

4.3.2 Provide a Platform that Allows for Hardware Upgrades

4.3.3 Facilitate Software Application Portability

4.3.4 Facilitate Diagnostic Capabilities

## 5 Engine Board Details

5.1 General Information

5.1.1 Engine Board

5.1.2 Host Module

5.2 Mechanical and Physical

5.2.1 Board Dimensions and Mechanical Requirements

5.2.2 Connector Pinout and Signal Names

5.2.3 Environmental Requirements

5.3 On-Board Resources

5.3.1 Central Processing Unit

5.3.2 Startup Considerations

5.3.3 Memory

5.3.4 Real-Time Clock (RTC)

5.3.5 ATC Controller Board Support Package (BSP)

5.4 Electrical Interface

5.4.1 Power

5.4.2 Synchronization
6 COMMUNICATION INTERFACE DETAILS ......6-1

6.1 GENERAL DESCRIPTION .............................................6-1
  6.1.1 Interchangeability Control .................................. 6-2
  6.1.2 Communication Interface Slot Identification .............. 6-3
  6.1.3 Serial Port Identification ................................ 6-3

6.2 MECHANICAL DESCRIPTION ...........................................6-3
  6.2.1 Mechanical Outline Dimensions ............................. 6-3
  6.2.2 ATC Communications Connector Mechanical Pin Assignments 6-5
  6.2.3 Mechanical Field Connections .............................. 6-7

6.3 OPERATIONAL DESCRIPTION ...........................................6-11
  6.3.1 Interface to ATC .............................................. 6-11
  6.3.2 Modulation and Demodulation ............................... 6-12

6.4 COMMUNICATIONS INTERFACE VERSIONS .............6-21

7 PHYSICAL AND USER INTERFACE DETAILS ........................................7-1

7.1 USER INTERFACE GENERAL DESCRIPTION ............7-1
  7.1.1 Minimum User Interface ...................................... 7-1
  7.1.2 Optional User Interfaces ..................................... 7-2
  7.1.3 User Interface Pin Connections ............................ 7-2
  7.1.4 User Interface Operation ................................... 7-3
  7.1.5 User Interface Power Requirements ..................... 7-15
# TABLE OF CONTENTS

## 7.2 POWER SUPPLY GENERAL DESCRIPTION
- 7.2.1 "ON/OFF" Power Switch
- 7.2.2 LED DC Power Indicators
- 7.2.3 Service Voltage Fuse
- 7.2.4 +5 VDC Standby Power
- 7.2.5 Monitor Circuitry
- 7.2.6 External Power Supply Requirements

## 7.3 MECHANICAL AND PHYSICAL GENERAL DESCRIPTION
- 7.3.1 Chassis

## 8 PARALLEL AND SERIAL I/O DETAILS
- 8.1 GENERAL INFORMATION
  - 8.1.1 Parallel Input / Output Overview
  - 8.1.2 Serial I/O Overview
- 8.2 PARALLEL INPUT / OUTPUT (PI/O)
  - 8.2.1 Parallel Connection to Model 332 Cabinets
  - 8.2.2 Parallel Connection to NEMA TS-1 or TS-2 Type 2 Cabinets
  - 8.2.3 Connection to NEMA TS-2 Type 1 Cabinets
- 8.3 SERIAL INPUT / OUTPUT
- 8.4 ISOLATION REQUIREMENTS
  - 8.4.1 Engine Board Isolation
  - 8.4.2 Parallel I/O Isolation
  - 8.4.3 Serial I/O Isolation
## 8.5 Electrostatic Discharge Protection Requirements

## 9 Environmental and Test Procedures

### 9.1 General

#### 9.1.1 Communication Loading Test

### 9.2 Inspection

### 9.3 Testing Certification

### 9.4 Definitions of Design Acceptance Testing (DAT) and Production Testing

### 9.5 Environmental and Operating Requirements

#### 9.5.1 Voltage and Frequency

#### 9.5.2 Transients, Power Service (DAT)

#### 9.5.3 Nondestructive Transient Immunity (DAT)

#### 9.5.4 Temperature and Humidity

### 9.6 Test Facilities

### 9.7 Test Procedures: Transients, Temperature, Voltage, and Humidity

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

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

#### 9.7.3 Test C: Low-Temperature Low-Voltage Tests (DAT and Production Testing)
9.7.4 Test D: Low-Temperature High-Voltage Tests (DAT and Production Testing) .......... 9-10
9.7.5 Test E: High-Temperature High-Voltage Tests (DAT and Production Testing) .......... 9-10
9.7.6 Test F: High-Temperature Low-Voltage Tests (DAT and Production Testing) .......... 9-11
9.7.7 Test G: Test Termination (All tests) ............... 9-12
9.7.8 Test H: Appraisal of Equipment under Test ......................................................... 9-12

9.8 VIBRATION TEST (DAT) ................................................. 9-13
9.8.1 Purpose of Test .................................................. 9-13
9.8.2 Test Equipment Requirements .............................................. 9-13
9.8.3 Resonance [Mechanical Resonant Frequency] Search (DAT) ............................... 9-13
9.8.4 Endurance Test (DAT) ............................................. 9-14
9.8.5 Disposition of Test Unit ........................................ 9-14

9.9 SHOCK (IMPACT) TEST (DAT) ........................................ 9-14
9.9.1 Purpose of Test .................................................. 9-14
9.9.2 Test Equipment Requirements .............................................. 9-15
9.9.3 Test Procedure (DAT) ............................................. 9-16
9.9.4 Disposition of Test Unit ........................................ 9-17

9.10 POWER INTERRUPTION TEST PROCEDURES (DAT) 9-17
9.10.1 Short Power Interruption ............................................. 9-18
9.10.2 Voltage Variation .................................................. 9-18
9.10.3 Rapid Power Interruption ............................................. 9-18

10 PERFORMANCE AND MATERIAL REQUIREMENTS................................. 10-1
## TABLE OF CONTENTS

1 **10.1** GENERAL .................................................................10-1
   1
2  10.1.1 Furnished Equipment ..............................................10-1
3
4  10.1.2 Edges .........................................................................10-1
5
6  10.1.3 Hardware .......................................................................10-1
7
8  10.1.4 Electrical Isolation and Equipment Grounding ..............10-1
9
10  10.1.5 Component Sources ..................................................10-1
11
12  10.1.6 Capacitors ...............................................................10-2
13
14  10.1.7 Resistors ......................................................................10-2
15
16  10.1.8 Semiconductors .........................................................10-3
17
18  10.1.9 Transformers and Inductors .......................................10-3
19
20  10.1.10 Fuses ..........................................................................10-3
21
22  10.1.11 Switches ......................................................................10-3
23
24  10.1.12 Wiring, Cabling, and Harnesses ...............................10-4
25
26  10.1.13 Indicators and Character Displays ...........................10-5
27
28  10.1.14 Connectors ..................................................................10-5
29
30  10.1.15 PCB Design ...............................................................10-6
31
32  10.1.16 Tolerances ..................................................................10-6
33
34 **11** QUALITY CONTROL ...................................................11-1
35
36  11.1 COMPONENTS .............................................................11-1
37
38   11.1.1 Subassembly, Unit Or Module ....................................11-1
39
40   11.1.2 Predelivery Repair ......................................................11-1
41
42   11.1.3 Manufacturers’ Quality Control Testing
43      Certification .........................................................................11-1
44
45 **12** GLOSSARY ....................................................................12-1
46
47   12.1 PHYSICAL UNITS .......................................................12-1
48
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2 MODIFIERS</td>
<td>12-1</td>
</tr>
<tr>
<td>12.3 ACRONYMS AND DEFINITIONS</td>
<td>12-2</td>
</tr>
<tr>
<td>ANNEX A Minimum Linux Kernel Configuration</td>
<td>A-1</td>
</tr>
<tr>
<td>ANNEX B Required Device Driver Interfaces</td>
<td>B-1</td>
</tr>
</tbody>
</table>
# LIST OF TABLES AND FIGURES

<table>
<thead>
<tr>
<th>Figure/Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Component Parts of the ATC Controller and their Connections</td>
</tr>
<tr>
<td>3-1</td>
<td>BSP in the ATC Architecture</td>
</tr>
<tr>
<td>3-2</td>
<td>View of a Typical ATC System Environment</td>
</tr>
<tr>
<td>3-3</td>
<td>Main Maintenance/ Support Diagram</td>
</tr>
<tr>
<td>3-4</td>
<td>Anticipated ATC Applications</td>
</tr>
<tr>
<td>3-5</td>
<td>Manage External Devices’ Sub-feature Areas</td>
</tr>
<tr>
<td>3-6</td>
<td>Facilitate Ease of Maintenance &amp; Future Hardware and Software Updates’ Sub-feature Areas</td>
</tr>
<tr>
<td>3-7</td>
<td>Engine Board Top View</td>
</tr>
<tr>
<td>3-8</td>
<td>Engine Board / Host Module Stackup (not to scale)</td>
</tr>
<tr>
<td>3-9</td>
<td>Connector Pinout and Signal Names</td>
</tr>
<tr>
<td>3-10</td>
<td>Connector Pinout and Signal Names (Continued)</td>
</tr>
<tr>
<td>3-11</td>
<td>Standard Specification &amp; Option Choices</td>
</tr>
<tr>
<td>3-12</td>
<td>Power Supply Signals during Restoration of AC Service Power</td>
</tr>
<tr>
<td>3-13</td>
<td>Power Supply Signals during Loss of AC Service Power</td>
</tr>
<tr>
<td>3-14</td>
<td>Power Supply Signals during Short Loss of AC Service Power</td>
</tr>
</tbody>
</table>
# Advanced Transportation Controller (ATC) Standard

## TABLE OF CONTENTS

1. **Figure 7-4:** Relationship of AC Service Voltage and LINESYNC ................................................................. 7-20
2. **Figure 7-6:** Maximum Basic Dimensions, Rack Mount .......... 7-23
3. **Figure 8-1:** C1S and C11S Pin Configuration: Refer to ATC 2070 Standard .......................................................... 8-19
4. **Figure 8-2:** Connector Diagram .................................................. 8-21
5. **Figure 8-3:** Details of Ethernet Hub Connections, Typical Use 8-24
6. **Figure 8-4:** Isolation Boundaries ............................................................. 8-26
7. **Table 9-1:** Ambient Temperature versus Relative Humidity At Barometric Pressures (29.92 In. Hg.) (Non-Condensing) .................................................. 9-5
8. **Figure 9-1:** Test Profile ................................................................. 9-9
9. **Figure 9-2:** Shock Test Fixture ......................................................... 9-16
1 FOREWORD

The purpose of this document is to define the minimum, required functionality of the hardware, operating system and support software for an Advanced Transportation Controller (ATC) compliant device.

The effort to develop standards for the ATC began with the Federal Highway Administration gathering together a group of users interested in furthering the development of open architecture hardware and software to meet the future needs of Intelligent Transportation Systems. The ATC users group gained the support of the Institute of Transportation Engineers to continue their work in developing standards for the ATC. The American Association of State Highway and Transportation Officials (AASHTO) and the National Electrical Manufacturer’s Association joined with ITE to create a joint effort.

In March 1999, a formal agreement was reached among NEMA, ITE and AASHTO to jointly develop, approve and maintain the ATC standards. Under the guidance of a Joint AASHTO/ITE/NEMA Committee on the ATC, a Working Group was created in order to develop a standard for the Advanced Transportation Controller. The first official meeting of this working group was in November 2002.

In preparation of this Standards Publication, input of users and other interested parties was sought and evaluated. Inquiries, comments and proposed or recommended revisions should be submitted to:

Standards Development Engineer
Institute of Transportation Engineers
1099 14th Street, NW, Suite 300 West
Washington, DC 20005-3438

voice:202-289-0222
fax:202-289-7722
2 INTRODUCTION

2.1 Scope

With the growth of Intelligent Transportation Systems, transportation management increasingly relies on electronically controlled devices deployed in the field and the controllers that coordinate and relay data from those devices. This standard describes a family of advanced, ruggedized, field communications and process controllers that are configurable for a variety of traffic management applications. Typically, they provide communication, control, and data gathering from and to

- Central control computers when appropriate
- Other controllers when appropriately configured
- Control units for devices deployed in the field, typically in the vicinity of and linked to the controller.

Essentially, an ATC is a special function computer that must be able to operate remotely in a largely unattended mode in the harsh environment of field deployments throughout the United States.

The goal of this standard is to provide an open architecture design for the next generation of transportation controller applications. These controllers are modular in design and intended to be compatible with or inclusive of existing (present day) traffic controller capabilities. First, the design specified in this standard is based on the concentration of computing power in a single component (the Engine Board) that is interchangeable with Engine Boards designed by other manufacturers. Second, the standard provides for required and optional features, all of which are based on open standard, common protocol communication standards. Third, the standard is responsive to the functional requirements identified in Section 4 below.

Guidance: The NEMA/AASHTO/ITE API shall run on controllers designed to this ATC standard. (Authorized Engineering Information).

Figure 2-1 provides details of the component parts of the ATC and their connections.
Figure 2-1: Component Parts of the ATC Controller and their Connections.
2.2 Key Elements of the ATC Controller Standard

2.2.1 Form/Fit/Function

The ATC provides for easy hardware upgrades to adapt to newer processors, operating systems, and increased memory size and speed. It does this by requiring that the Engine Board (CPU module) 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 no-connects on both Engine Board and Host modules.

While the ATC packaging is ultimately left open to allow manufacturers to be responsive to special needs, this standard describes packaging and interfaces that allow the ATC Controller to be deployed in industry standard cabinet configurations.

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

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

Note that many of the design choices in this standard reflect the basic requirement that the ATC provide backward interface compatibility with existing NEMA TS1, TS2, Caltrans Model 170, NYDOT Model 179, and ATC 2070 controllers and NEMA Model 332 and ITS cabinets.

2.2.2 Engine Board

All computational functions are concentrated on an Engine Board within the ATC that meets designated minimum requirements on:

- CPU and RAM memory
- FLASH memory storage
- Serial ports
- Ethernet interface
- Standardized (form, fit and function) pin out interface
- Clock/calendar maintenance
- Board Support Package
2.2.3 Communications and User Interfaces

This standard requires at least one and at most two Communications Interface slots be provided by an ATC. These slots are further described in Section 6 of this standard and must adhere to the form, fit and electrical interface specifications of the communications interface slots provided by the ATC 2070. This standard does not require that either of these slots be populated.

Section 7 of this standard defines the required front panel interfaces of the ATC and defines the allowable optional interfaces. In this standard, alternative user interfaces may be included provided that the minimal interface is also provided.

2.2.4 Parallel and Serial I/O

The ATC provides industry-standard communication interfaces for asynchronous and synchronous serial communications. This standard also requires a minimum of one synchronous serial port to interface to ITS Cabinet or TS2 Type 1 Cabinet. Optional interface modules defined in this standard include:

- Serial to parallel interface module for connection to NEMA TS1 or TS2 type 2 cabinet
- Serial to parallel interface module for connection to Model 332 cabinet

2.2.5 Operating System & Board Support Package 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:

1) A Linux compatible kernel that shall be configured to include, at minimum, the features specified in Annex A.
2) Drivers that support all functionality defined by sections 5.3, On-Board Resources, and 5.4.3, Serial Interface Ports of this document, and be capable of operating in an interrupt driven environment where appropriate. Drivers for the following engine board hardware are required:

<table>
<thead>
<tr>
<th>Function</th>
<th>Character</th>
<th>Block</th>
<th>Network</th>
<th>Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sp1 Asynchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp1s Synchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp2 Asynchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp2s Synchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp3 Asynchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp3s Synchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp4 Asynchronous Port (Note 1)</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp5 Asynchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp5s Synchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp6 Asynchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp8 Asynchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>/dev/sp8s Synchronous Port</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ethernet 1</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet 2</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USB</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SPI</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CPU Reset</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU Active LED</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Down</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RTC</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Datakey Present</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flash File System</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SRAM File System</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Notes:
1. A Linux Console shall be provided on /dev/tty4 (/dev/console) at boot-up. All terminal outputs during boot time shall be made to that interface. All inputs are made via a terminal program to that interface. This interface is the default stdin / stdout of Linux. The communication parameters are initially 38400 baud, n, 8, 1. The same communication parameters shall be used by the boot-loader in order to ensure a continuous output to the serial terminal / console. After booting and all applications are loaded, /dev/tty4 shall be available to applications as /dev/sp4 unless the single-user mode Linux feature has been invoked via Ctrl-C during the boot process before control is passed to any application programs.

See Annex B for BSP-specified driver interface details.

3) Utility applications, modules, libraries, and supporting data which include but are not limited to the following:

<table>
<thead>
<tr>
<th>Package</th>
<th>Version *</th>
<th>Programs</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Busybox</td>
<td>1.00p8</td>
<td>addgroup, adduser</td>
<td>UNIX shell and commands</td>
</tr>
</tbody>
</table>
### INTRODUCTION

| 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, 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, msg, 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, sed, sh, sleep, sort, start-stop-daemon, stty, su, sulogin, sync, syslog, 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 |
|---|---|---|
| collection | http://www.busybox.net/ |
| UclibC | 0.9.27 | ld.so, libc, libcrypt.so, libdl.so, libm.so, libnc.so, libnss_dns.so, libnss_files.so, libPTHread.so, libresolv.so, libutil.so |
| Uclibc Equivalent | http://www.uclibc.org/downloads |

* Or higher

### 2.3 References

#### 2.3.1 Normative References

- This standard assumes and is consistent with known versions of ITS cabinet
- ATC Standard for the type ATC 2070 Controller
NEMA TS2-2003 Traffic Controller Assemblies with NTCIP Requirements
http://www.nema.org/stds/ts2.cfm

USB Specifications


http://www.usb.org/developers/docs/usbspec.zip

**USB Mass Storage Overview 1.2**
http://www.usb.org/developers/devclass_docs/usb_msc_overview_1.2.pdf

**USB Mass Storage Bulk Only 1.0**

**USB Mass Storage Control/Bulk/Interrupt (CBI) Specification 1.1**
http://www.usb.org/developers/devclass_docs/usb_msc_cbi_1.1.pdf

**USB Mass Storage UFI Command Specification 1.0**

*Guidance: If the links listed above are no longer valid, it is recommended that the user reference the www.USB.ORG web site and the http://www.usb.org/developers/devclass_docs for a listing of the most recent documents which are relevant to the application of USB to the ATC.*

Ethernet 802.3 Specifications

**IEEE 802.3-2002 Specification**
3 CONCEPT OF OPERATIONS

This standard describes a general, field-located computing device that must be capable of executing applications software from various developers. Generally accepted systems engineering practices begin from user needs. This section identifies the presently known user requirements for an ATC and begins to identify the associated functions. Because the users’ needs and applications are expected to expand in unknown ways in the future, the standard explicitly recognizes that the details of particular future applications use are not completely known at this writing. It is important nonetheless that the support and usage needs of the most commonly known and anticipated applications be defined.

As indicated above, it is the intent of this standard to describe a general-purpose computing device. As such, the ATC can be seen as analogous to a Personal Computer (PC). A difference between this standard and the PC is that a device meeting this standard must be able to withstand the harsh environment of a field-located device with no special cabinet or environmental conditioning beyond that specified separately in the ITS cabinet standard. Another difference is that the ATC must be able to operate remotely in a largely unattended mode. Similar to the PC, the ATC Controller must adhere to a set of programming conventions and interfaces standards such that the applications software that runs in the device can be developed using the Engine Board vendor-supplied BSP as guidance.

The ATC Controller must also have a high degree of reliability, and be easily maintained.

3.1 Problem Statement

One of the largest component costs of today’s Intelligent Transportation Systems is associated with the development, testing, deployment and maintenance of applications software. As the current trend continues towards distributing more of the intelligence of ITS out closer to the field, there is an increasing demand for more and more capable field deployable devices. This hardware must run more sophisticated applications software and operate in modern networking environments. The ATC Controller is intended to address these needs.

The ATC Controller is intended as a next generation, “Open Systems” controller [in which hardware interfaces are generically defined, standardized, and adopted by multiple manufacturers] which follows the “Open Systems” lineage of the ATC 2070 and California Model 170 and New York Model 179 controllers. “Open Systems” in this context refers to the concept of separation of hardware from software by standardizing the interface between the two. This allows software to be developed independent of the hardware. “Open Systems” help protect an agency’s investment by guarding against premature obsolescence due to a manufacturer’s discontinuance of a particular line of equipment or the manufacturer’s ceasing of business operations altogether. Additionally, “Open Systems” typically increase equipment procurement competition;
3.2 Historical Background

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 1970’s two concurrent traffic controller standards efforts were initiated in North America. These were the Model 170 standard and the NEMA standard. A brief history of these two standards efforts and the later ATC 2070 standard are presented in the subsections below.

3.2.1 NEMA

The NEMA standard(s) stemmed from a group of manufacturers who joined the NEMA (National Electrical Manufacturers Association) 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 six 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 lastly updated in 2003. 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 another's package. Also the front panel displays and the information displayed were all different and non standard. The ATC standard addresses both the interchangeability of software, the standardization of displays and the reliance upon a single operating system.

3.2.2 The Model 170 Specification

The Model 170 specification was developed by Caltrans and New York State DOT 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.

3.2.3 The ATC 2070 Standard

The ATC 2070 is a current generation “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 ATC 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 ATC 2070 also specifies the use of an O/S (OS-9 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 ATC 2070 standard also provides for greater subcomponent interchangeability and modularity than the Model 170. ATC 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.

However, the ATC 2070 requires that a specific CPU chip and a specific commercial O/S be used. Unfortunately, the embedded hardware and O/S market place is not as large as is the PC marketplace. As a result, longevity concerns are surfacing for the ATC 2070 related to its particular O/S and CPU selections. Many users are concerned that additional retrofit and software porting costs would be required should either this O/S and/or the CPU become unavailable.
3.3 Functional Needs

ATC Engine Board manufacturers will provide a BSP that is compatible with their ATC hardware. Developers can port their software to various ATC controllers by compiling and linking their application with the appropriate drivers for the target controller as shown in Figure 3.1.

Advanced communication capabilities are becoming increasingly important for ITS field controllers. ITS data communications networks are deploying NTCIP and Internet Protocol (IP) based data communications networks. Peer-to-peer networking capabilities are also increasingly required for advanced control algorithm implementations. For such networks, Ethernet is the connection interface of choice at field controllers.

![Figure 3-1: BSP in the ATC Architecture](image-url)
3.4 Operational Environment

Typically, an operator interfaces to an ATC through one of 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, PDA, 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 3-2 depicts the physical architecture of the key components related to a typical ATC based system run from a central location.

Figure 3-2: 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.

As a minimum, the ATC must provide the necessary interfaces to support the ITS Cabinet standard. Additionally, the ATC should provide optional interface support for common legacy cabinets including Model 332, NEMA TS1, and NEMA TS2 types.

3.5 Representative Usage

As previously indicated, the functionality of a deployed ATC will depend on the applications software loaded into it. Typical ITS applications to be hosted on the ATC are listed in Table 3-1.

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

Table 3-1: Anticipated ATC Applications

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 3-3 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 4 then uses these definitions to organize and define the various functional requirements of an ATC.
Figure 3-3: Main Maintenance/Support Diagram

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

- Manage/Configure Applications
- Manage External Devices
- Facilitate ease of maintenance & future hardware and software updates

The Maintenance and Support function includes features for maintenance and update/enhancement of the controller unit’s hardware and/or software.
3.5.1 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.

![Manage/Configure Controller Applications](image)

3.5.1.1 Install/Update Applications Software Quickly and Efficiently

This feature allows the local operator or a remote computer to install or update the application software resident on the ATC.
3.5.1.2 Install/Upgrade O/S Quickly and Efficiently

This feature allows the local operator to install or update the O/S resident on the ATC. Local upgrade capability is required while remote upgrade capability is considered an optional feature.

3.5.1.3 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.

3.5.1.4 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.

3.5.1.5 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.

3.5.1.6 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.

3.5.1.7 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.

3.5.1.8 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.
3.5.2 **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 3-5: Manage External Devices’ Sub-feature Areas**

- **Manage/Control a Variety of External Field Devices**
- **Monitor the Status of a Variety of External Devices**
- **Local Operator / Remote Computer**
- **External Device**

3.5.2.1 **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.

3.5.2.2 **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.

3.5.3 **Facilitate Ease of Maintenance & Future Hardware and Software Updates**

The various sub-features for “facilitating ease of maintenance & future hardware and software” are shown in the following figure. The subsequent sections detail these sub-features.
3.5.3.1 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.

3.5.3.2 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.

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

3.6 Security
The standard does not explicitly address security issues. However, network communication interfaces have been defined with provisions for data security in mind. 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. These are outside the scope of this particular standard.

3.7 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 is 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.
4 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 & Future Hardware or 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.

4.1 Manage/Configure Controller Applications

4.1.1 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:

- Front panel serial port for interfacing with laptop computer, PDA or similar locally connected device with software for performing this function
- Front panel Ethernet port for interfacing with laptop computer, PDA 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 or PDA device 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:
• Separate Ethernet port for possible use to communicate with a remote device having the necessary software for performing this function.

• Separate Serial port for possible use to communicate with a remote device having the necessary software for performing this function.

4.1.2 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 4.1.1.

4.1.3 Maintain Clock/Calendar Function and Synchronize with External Sources

The ATC shall provide hardware to support a clock/calendar function:

• The Engine Board shall include a clock/calendar device to support the maintenance and backup of current time and date by the ATC unit in the absence of AC service power.
  o Clock/calendar device shall maintain time/date for a minimum of 30 days without AC service power applied to the controller.
  o Clock/calendar device drift shall be less than ± 1 minute per 30 days at 25°C.

• Applications software executing in the ATC shall be able to set time and date on the clock/calendar device to the nearest 0.1 seconds via the BSP.

• When AC service power is present, current time/date information shall be maintained by the O/S and easily accessed by the application software utilizing the BSP.

• Power transients and short term power outages shall not introduce clock drift.

• The Engine Board/BSP shall utilize the LINESYNC signal and the Engine Board Real Time Clock (RTC) to maintain an accurate Operating System Time (OST) by following these rules:
  o Under normal AC service power conditions (as defined in section 7.2.6.1) or during power failure conditions of less than 500 ms as indicated by the POWERDOWN signal, OST timing shall be derived from the 8.33 ms LINESYNC signal.
  o Once each hour, the BSP RTC driver shall automatically copy the current OST time to the RTC with an accuracy of 0.1 seconds.
Upon power failure, after reapplication of power and system initialization, the BSP RTC driver shall copy the RTC time values into the OST registers with an accuracy of 0.01 seconds.

- Accuracy requirements of the LINESYNC signal is stated in section 7.2.5.2.

**Guidance:** The accuracy requirements allow for 0.1 second accuracy in the ability to set the RTC, 0.01 second accuracy to synchronize the OST to a transition edge of the RTC when re-establishing the OST, and an additional 17 ms due to the asynchronous nature of the LINESYNC signal. Thus, the net error is 0.127 seconds plus the accuracy of the RTC. (Authorized Engineering Information)

### 4.1.4 Configure and Verifying Parameter(s)

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

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

- Front panel display and keyboard(s) to support operator configuring/verifying of application parameter(s) and/or
- Serial communication port for locally connected laptop, PDA or similar device with software to support operator configuring/verifying application parameter(s) from this device

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

- Serial communications port or
- Ethernet port

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

### 4.1.5 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 locally connected laptop, PDA 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:

• Communications port (no provisions for operator data entry), and
• Presence of application support and/or BSP support for NTCIP transfers through
communications port

4.1.6 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 locally connected laptop, PDA 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, and
• Presence of BSP and/or applications support for NTCIP transfers through
communications port(s).

4.1.7 Allowing Operator Control of Application(s)

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

If performed locally, this requirement shall be satisfied by:

• Communication port(s) for interface to locally connected laptop, PDA or similar
device with necessary software to support operator control of applications
(start/stop/run times, etc.).

• BSP description of controller resident operator interface software to control other
applications tasks (start/stop/run time/etc.).

Remote performance of this function is not supported.
4.1.8 Facilitate the Long Term Retention of Data

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

- SRAM memory for applications to store data
- BSP-described support of FLASH memory file management system

4.2 Manage External Devices

The ATC shall include hardware to provide control, management, and monitoring of a variety of field devices through conventional parallel I/O that conforms to the existing NEMA (TS2-2003) and Type 170 and 179 (as reflected by CALTRANS TEES, and the New York State TRANSPORTATION MANAGEMENT EQUIPMENT SPECIFICATIONS) controller interface specifications, and shall provide serial interfaces as listed below. The four (4) otherwise undedicated serial ports required in the following two subsections may be shared by the requirements of these two subsections.

4.2.1 Manage/Control a Variety of External Field Devices

The ATC shall include hardware to provide management/control of a variety of external field devices. This standard describes required interfaces to provide standardized communication with external devices via industry-standard asynchronous and synchronous serial communication connections.

In support of this requirement, this standard calls for a minimum of four (4) otherwise undedicated general-purpose serial communications ports for possible interface to external field devices:

- Each port shall support asynchronous or synchronous communications
- Each port shall support a range of baud rates as defined in section 5.4.3 Serial Interface Ports of this standard
- Ports shall be configurable to the various mechanical field connections defined in section 6.2.3 and the respective modulation and demodulation methods defined in section 6.3.2 of this standard

The standard also provides details of packaging and interfaces that allow this controller to be deployed in industry standard cabinet configurations including: NEMA TS2 Types 1 and 2, ITS and Model 332 cabinets. The ATC must provide backward interface compatibility with existing NEMA, Models 170, 179, and ATC 2070 controllers.
One dedicated synchronous serial port shall be provided to directly interface (select one as appropriate)

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

### 4.2.2 Monitor the Status of External Field Devices

The ATC shall provide hardware to monitor the status of a variety of external field devices. This standard describes the required interfaces to provide standardized communication with external devices via industry-standard asynchronous and synchronous serial communication connections.

In support of this requirement, this standard calls for a minimum of four (4) otherwise undedicated general-purpose serial communications ports for possible interface to external field devices. (Note that these ports are shared with the four ports required for the management and control of external devices listed in section 4.2.1).

- Each port shall support asynchronous or synchronous communications
- Each port shall support a range of baud rates as defined in section 5.4.3 Serial Interface Ports of this standard
- Ports shall be configurable to the various mechanical field connections defined in section 6.2.3 and the respective modulation and demodulation methods defined in section 6.3.2 of this standard

The standard also provides details of packaging and interfaces that allow this controller to be deployed in industry standard cabinet configurations including: NEMA TS2 Types 1 and 2, ITS and Model 332 cabinets. The ATC must provide backward interface compatibility with existing NEMA, Models 170, 179, and ATC 2070 controllers.

One dedicated synchronous serial port to directly interface (select one as appropriate)

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

### 4.3 Facilitate Ease of Maintenance & Future Hardware or Software Updates

#### 4.3.1 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 will support, at a minimum, the following classes of functions:

- Serial communications
- Field cabinet I/O
- FLASH memory file management
- Portable memory devices, as needed
- Applications task control
- Time & date management functions
- User interface support

4.3.2 Provide a Platform that Allows for Hardware Upgrades

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

4.3.2.1 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 mounted in rack cabinet including, but not limited to, cabinets adhering to the new ITS Cabinet standard and the Model 332 cabinet specifications.
- If used in standard NEMA TS1 or TS2 cabinet, the controller unit shall be shelf-mounted.

4.3.2.2 Standardize Engine Board Contents

A key design goal of this ATC standard is that it provides for easy hardware upgrades to adapt to newer processors, and increased memory size and speed. It does this by requiring that all computational functions be concentrated on an Engine Board within the ATC. To maintain interchangeability, the Engine Board (CPU module) 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 no-connects on both Engine Board and Host modules. Section 5 of this standard designates minimum Engine Board requirements on:

- CPU and RAM memory
- FLASH memory storage
- Operating System Software
- Serial ports
- Ethernet interfaces
- Standardized (form, fit and function) pin out interface
- Real-time clock

### 4.3.2.3 Standardize Communication Interfaces

The ATC 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.

### 4.3.3 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 open source operating system as defined in Section 2.2.5 and Annex A. 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.

### 4.3.4 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.
5  ENGINE BOARD DETAILS

5.1  General Information

5.1.1  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 ATC 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.

These specifications for the Engine Board require a minimum level of real-time processing capability. Section 9.1 specifies the set of tests that shall be used to determine Engine Board compliance 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.

Guidance: There has been much discussion and debate regarding the approach taken in this document regarding the Engine Board, in particular the obvious ties to the ATC 2070. The consensus of the Project Team from the beginning has been that this work should represent an evolution of that design, rather than a revolutionary new design, and should build upon and enhance the strengths of that design while addressing the shortcomings which prevent the ATC 2070 from adequately meeting current and future requirements as outlined in this document.

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.

5.1.2 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 this standard, manufacturers are free to construct virtually any type and form of Host Module to meet any specific market need.

5.2 Mechanical and Physical

5.2.1 Board Dimensions and Mechanical Requirements

The maximum horizontal dimensions of the Engine Board shall be 5.000" L x 3.937" W. The nominal thickness of the PCB material shall be 0.062". 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 5-1. 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" +0.010"/-0" in diameter. A 0.250" 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" 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 9, 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, shall not exceed 0.742" on the top including the Engine Board PCB material thickness, nor exceed 0.100" on the bottom excluding the Engine Board PCB material thickness.
Figure 5-1: Engine Board Top View
Figure 5-2: Engine Board / Host Module Stackup (not to scale)
5.2.2 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. These connectors shall be dual-row, DIN 41612 pin headers with the following specifications:

- distance post-to-post, same row: 0.100” nominal
- distance post-to-post, between rows: 0.100” nominal
- representative connector: Hirose PCN10-50P-2.54DSA (or equivalent)
- Host Module mating connector: Hirose PCN10C-50S-2.54DSA (or equivalent)

Table 5-1 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 5-1: Connector Pinout and Signal Names

<table>
<thead>
<tr>
<th>Connector P1</th>
<th>Connector P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 VPRIMARY</td>
<td>1 VSTANDBY_5</td>
</tr>
<tr>
<td>2 VPRIMARY</td>
<td>2 RESERVED</td>
</tr>
<tr>
<td>3 VPRIMARY</td>
<td>3 RESERVED</td>
</tr>
<tr>
<td>4 VPRIMARY</td>
<td>4 RESERVED</td>
</tr>
<tr>
<td>5 GROUND</td>
<td>5 SP2_TXD</td>
</tr>
<tr>
<td>6 GROUND</td>
<td>6 SP2_RXD</td>
</tr>
<tr>
<td>7 GROUND</td>
<td>7 SP2_RTS</td>
</tr>
<tr>
<td>8 GROUND</td>
<td>8 SP2_CTS</td>
</tr>
<tr>
<td>9 SP1_TXD</td>
<td>9 SP2_CD</td>
</tr>
<tr>
<td>10 SP1_RXD</td>
<td>10 SP2_TXC_INT</td>
</tr>
<tr>
<td>11 SP1_RTS</td>
<td>11 SP2_TXC_EXT</td>
</tr>
<tr>
<td>12 SP1_CTS</td>
<td>12 SP2_RXC_EXT</td>
</tr>
<tr>
<td>13 SP1_CD</td>
<td>13 SP5_TXD</td>
</tr>
<tr>
<td>14 SP1_TXC_INT</td>
<td>14 SP5_RXD</td>
</tr>
<tr>
<td>15 SP1_TXC_EXT</td>
<td>15 SP5_TXC_INT</td>
</tr>
<tr>
<td>16 SP1_RXC_EXT</td>
<td>16 SP5_RXC_EXT</td>
</tr>
<tr>
<td>17 SP3_TXD</td>
<td>17 USB_POWER_SWITCH</td>
</tr>
<tr>
<td>18 SP3_RXD</td>
<td>18 USB_OVERCURRENT</td>
</tr>
<tr>
<td>19 SP3_RTS</td>
<td>19 USB_DPLUS</td>
</tr>
<tr>
<td>20 SP3_CTS</td>
<td>20 USB_DMINUS</td>
</tr>
<tr>
<td>21 SP3_CD</td>
<td>21 SP8_TXD</td>
</tr>
<tr>
<td>22 SP3_TXC_INT</td>
<td>22 SP8_RXD</td>
</tr>
<tr>
<td>23 SP3_TXC_EXT</td>
<td>23 SP8_RTS</td>
</tr>
<tr>
<td>24 SP3_RXC_EXT</td>
<td>24 SP8_CTS</td>
</tr>
</tbody>
</table>
5.2.3 Environmental Requirements

Engine boards shall meet all environmental requirements specified in Section 9, 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.

5.3 On-Board Resources
5.3.1 Central Processing Unit

The Engine Board shall incorporate a CPU and support circuitry that shall have a minimum computational capability of 60 MIPS calculated using the Dhrystone v2.1 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.

5.3.2 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 4.5 seconds from the rise of both the POWERUP and POWERDOWN signals to the HIGH state (see Figure 7-1). 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 5.3.5.1 outlines the typical startup sequence.

The Engine Board shall provide circuitry to prevent writing to the 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 or POWERDOWN signals.

5.3.3 Memory

FLASH Memory

The Engine Board shall provide FLASH for the storage of O/S software and user application programs. A minimum of 6MB of FLASH shall be provided exclusively for application program storage. FLASH devices shall use a segmented architecture allowing erasing, writing and reading of individual segments. Access to this memory shall be accomplished with wait states totaling no more than 100 ns and a data bus width of no less than 16 bits.

Application software shall be capable of reading from and writing to the FLASH without the FLASH being corrupted by the power fail conditions specified in Section 5.4.1.

Dynamic RAM (DRAM)

The Engine Board shall contain a minimum of 16MB of DRAM or equivalent volatile memory for application and O/S program execution. The preferred memory organization
shall be in the native word length of the CPU for maximum performance and is preferred to operate with zero wait states. The minimum DRAM bus width shall be 16 bits. If the native bus width and zero wait states are not used in the engine board design, the Engine Board Manufacturer must publish DHRYSTONE test results to prove that the MIPS requirement is met.

**Static RAM (SRAM)**

The Engine Board shall contain a minimum of 1MB of SRAM memory for non-volatile parameter storage. Access to this memory shall be accomplished with wait states totaling no more than 100 ns and a data bus width of no less than eight bits. In the absence of VPRIMARY the SRAM shall be supported and maintained by VSTANDBY_5.

### 5.3.4 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 and shall maintain the accuracy requirements of section 4.1.3. Upon reapplication of power, the RTC shall be used to set the current time/date of the Operating System Time (OST) per section 4.1.3.

*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.*

### 5.3.5 ATC Controller Board Support Package (BSP)

The BSP supplied by the vendor shall provide operating-system-level support for the Engine Board. See “Operating System & Board Support Package Requirements” section of this standard.

#### 5.3.5.1 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 EPROM to 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 EPROM 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:

• Install any additional time-critical BSP driver modules.

(The steps listed above must be completed within the time period specified in Section 5.3.2.)

• 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.

5.4 Electrical Interface

5.4.1 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.

The Engine Board does not provide standby power in support of the SRAM or RTC. In
the absence of VPRIMARY, these components shall be supported and maintained by
VSTANDBY_5 provided from the Host Module (+5 VDC Standby Power from the Power
Supply). That is, standby power shall be provided only from the Host and not from any
source located on the Engine Board itself. 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 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 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.

**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 7.2.5.1 and diagrams of their states under
various operational conditions are shown in Figures 7-1, 7-2, and 7-3.

The Engine Board shall provide circuitry to prevent writing to the 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 7.2.5.1 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 7-2), 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 7-1). 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 7.2.5.1 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 7-2 and 7-3). 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 7-3), the application software shall continue operating normally without a restart. This condition is considered a short power outage.

5.4.2 Synchronization

LINESYNC

The LINESYNC signal is specified in section 7.2.5.2 and is an input to the Engine Board and provides a 50% duty cycle square-wave at 60Hz. 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.

5.4.3 Serial Interface Ports

Serial Communications Interface 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. The following communication channel loads for test purposes, with no other activity present shall be the following:

SP1,2,8 = Continuous full-duplex, asynchronous communications at 19.2Kbps
SP4,6 = Continuous full-duplex, asynchronous communications at 38.4Kbps
SP3S = Continuous full-duplex, synchronous communications at 153.6Kbps
SP5S = Continuous full-duplex synchronous communications at 614.4Kbps
All Ethernet Ports: At 10% loading, with 3% average hits to processor per minute

No other applications or I/O activities are required to be operational during this test.

Serial Port 1 (SP1)

Principal Usage: general-purpose
Operating Modes: ASYNC / SYNC / HDLC / SDLC
Asynchronous Rates (bps): 1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k
Optional: 57.6k / 115.2k
Synchronous Rates (bps): 19.2k / 38.4k / 57.6k / 76.8k / 153.6k
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)
Serial Port 2 (SP2)

Principal Usage: general-purpose
Operating Modes: ASYNC / SYNC / HDLC / SDLC
Asynchronous Rates (bps): 1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k
Optional: 57.6k / 115.2k
Synchronous Rates (bps): 19.2k / 38.4k / 57.6k / 76.8k / 153.6k
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)
- SP2_TXC_EXT: Transmit Clock External (I)
- SP2_RXC_EXT: Receive Clock External (I)

Serial Port 3 (SP3)

Principal Usage: in-cabinet devices
Operating Modes: ASYNC / SYNC / HDLC / SDLC
Asynchronous Rates (bps): 1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k
Optional: 57.6k / 115.2k
Synchronous Rates (bps): 153.6k / 614.4k
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)
- SP3_RXC_EXT: Receive Clock External (I)

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
Optional: 57.6k / 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: SYNC / HDLC / SDLC
Synchronous Rates (bps): 153.6k / 614.4k
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)

Serial Port 6 (SP6)

Principal Usage: front panel user-interface
Operating Modes: ASYNC
Asynchronous Rates (bps): 1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k
Optional: 57.6k / 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 / SYNC / HDLC / SDLC
Asynchronous Rates (bps): 1200 / 2400 / 4800 / 9600 / 19.2k / 38.4k
Optional: 57.6k / 115.2k
Synchronous Rates (bps): 19.2k / 38.4k / 57.6k / 76.8k / 153.6k
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)
Serial Peripheral Interface Port

The Engine Board shall provide a synchronous Serial Peripheral Interface Port. All SPI interface pins shall be at HCT logic-levels. Input pins are indicated by (I), output pins by (O).

The implementation of SPI_SEL_1 is required to support DataKey operations.

The implementation of SPI_SEL_2 is required to support a Host Module serial EEPROM device containing controller configuration information. The content and organization of the information will be described in the BSP. This EEPROM device shall have the following characteristics:

- shall function in a manner similar to a 25020-type (1K-bit) SPI EEPROM device
- shall have a minimum size of 2 Kbit organized as 256 words of 8 bits each
- shall provide 5V interface signals
- shall operate properly with up to a 2.0 MHz SPI clock
- shall utilize SPI Mode 0 (CPOL=0, CPHA=0)
- shall be write-protected (using *WP pin) whenever POWERUP is LOW
- shall be readable from application software during normal ATC operation
- shall support the following instruction set:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
<th>Instruction Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>WREN</td>
<td>Write Enable</td>
<td>0000 X110</td>
</tr>
<tr>
<td>WRDI</td>
<td>Write Disable</td>
<td>0000 X100</td>
</tr>
<tr>
<td>RDSR</td>
<td>Read Status Register</td>
<td>0000 X101</td>
</tr>
<tr>
<td>WRSR</td>
<td>Write Status Register</td>
<td>0000 X001</td>
</tr>
<tr>
<td>READ</td>
<td>Read from Memory Array</td>
<td>0000 A0011</td>
</tr>
<tr>
<td>WRITE</td>
<td>Write to Memory Array</td>
<td>0000 A0010</td>
</tr>
</tbody>
</table>

Note: A₈ represents MSB address bit A₈

SPI_SEL_3 is currently unimplemented and is reserved for future SPI-related expansion.

SPI_SEL_4 is manufacturer specific.

Serial Peripheral Interface (SPI)

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: Select 1 (O)
- SPI_SEL_2: Select 2 (O)
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, as a minimum, shall conform to the appropriate sections of the USB v1.1 specification for both hardware and software operation in order to support bulk transfer operations.
- To facilitate the transfer of files between dissimilar equipment, all USB memory devices shall be formatted using the FAT16 file system. This provides for a maximum per-device storage capacity of 2GB (assuming 32kB clusters).

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.

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.
Guidance: Independent hubs 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)

### 5.4.4 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 Controller 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 5.2.1.

### 5.4.5 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.

**EQUIPMENT GROUND**

The Engine Board mounting hole nearest P1-50 is 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.
6 COMMUNICATION INTERFACE DETAILS

6.1 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, interchangeable among vendors. To meet this standard, a Communications Interface shall comply with:

- Mechanical dimensions and ATC connector of this standard
- Front panel connectors of this standard
- Modulation methods of this standard
6.1.1 Interchangeability Control

1. The ATC shall provide at least one and a maximum of two communication interface slots.

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

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

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

5. Communications circuitry embedded inside the ATC does NOT exempt the ATC from providing at least one communications interface slot.
6.1.2 Communication Interface Slot Identification

Each Communications Interface Slot shall be externally labeled as either A1 or A2 as appropriate. These labels shall be clearly visible when the Communication Module(s) are installed.

6.1.3 Serial Port Identification

Each Communications Interface Module may use one or more ATC Serial Ports. For clarity, each Communications Interface Module front panel connector shall be identified as either Port A, Port B, Port C, etc. 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-574 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).

6.2 Mechanical Description

6.2.1 Mechanical Outline Dimensions

The ATC Communications Interface uses the ATC 2070 communications slot mechanical form factor and pin configurations. The mechanical dimensions are as follows:
Figure 6-2: Mechanical Dimensions

NOTES:
1. SOUTHC 0 47 02 101 10 OR EQUAL
2. ELCD 00 8272 96 000 013 OR EQUAL
### ATC Communications Connector Mechanical Pin Assignments

Mandatory Communications Slot:

<table>
<thead>
<tr>
<th>PIN</th>
<th>ROW A</th>
<th>ROW B</th>
<th>ROW C</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>SP1TXD+</td>
<td>SP6TXD+</td>
<td>SP5TXD-</td>
</tr>
<tr>
<td>8</td>
<td>SP1TXD-</td>
<td>SP6TXD-</td>
<td>SP5TXD-</td>
</tr>
<tr>
<td>9</td>
<td>SP1RXD+</td>
<td>SP6RXD+</td>
<td>SP5TXC+</td>
</tr>
<tr>
<td>10</td>
<td>SP1RXD-</td>
<td>SP6RXD-</td>
<td>SP5TXC-</td>
</tr>
<tr>
<td>11</td>
<td>SP1RTS+</td>
<td>SP1TXC0+</td>
<td>SP5RXD+</td>
</tr>
<tr>
<td>12</td>
<td>SP1RTS-</td>
<td>SP1TXC0-</td>
<td>SP5RXD-</td>
</tr>
<tr>
<td>13</td>
<td>SP1CTS+</td>
<td>SP1TXCI+</td>
<td>SP5RXC+</td>
</tr>
<tr>
<td>14</td>
<td>SP1CTS-</td>
<td>SP1TXCI-</td>
<td>SP5RXC-</td>
</tr>
<tr>
<td>15</td>
<td>SP1DCD+</td>
<td>SP1RXC+</td>
<td>SP3TXD+</td>
</tr>
<tr>
<td>16</td>
<td>SP1DCD-</td>
<td>SP1RXC-</td>
<td>SP3TXD-</td>
</tr>
<tr>
<td>17</td>
<td>SP2TXD+</td>
<td>SP4TXD+</td>
<td>SP3RXD+</td>
</tr>
<tr>
<td>18</td>
<td>SP2TXD-</td>
<td>SP4TXD-</td>
<td>SP3RXD-</td>
</tr>
<tr>
<td>19</td>
<td>SP2RXD+</td>
<td>SP4RXD+</td>
<td>SP3RTS+</td>
</tr>
<tr>
<td>20</td>
<td>SP2RXD-</td>
<td>SP4RXD-</td>
<td>SP3RTS-</td>
</tr>
<tr>
<td>21</td>
<td>SP2RTS+</td>
<td>SP2TXCO+</td>
<td>SP3CTS+</td>
</tr>
<tr>
<td>22</td>
<td>SP2RTS-</td>
<td>SP2TXCO-</td>
<td>SP3CTS-</td>
</tr>
<tr>
<td>23</td>
<td>SP2CTS+</td>
<td>SP2TXCI+</td>
<td>SP3DCD+</td>
</tr>
<tr>
<td>24</td>
<td>SP2CTS-</td>
<td>SP2TXCI-</td>
<td>SP3DCD-</td>
</tr>
<tr>
<td>25</td>
<td>SP2DCD+</td>
<td>SP2RXC+</td>
<td>SP3TXCO+</td>
</tr>
<tr>
<td>26</td>
<td>SP2DCD-</td>
<td>SP2RXC-</td>
<td>SP3TXCO-</td>
</tr>
<tr>
<td>27</td>
<td>DCGND1</td>
<td>NA</td>
<td>SP3TXC+</td>
</tr>
<tr>
<td>28</td>
<td>NETWK1 Tx+</td>
<td>NA</td>
<td>SP3TXC-</td>
</tr>
<tr>
<td>29</td>
<td>NETWK2 Tx-</td>
<td>INSTALLED</td>
<td>SP3RXC+</td>
</tr>
<tr>
<td>30</td>
<td>NA</td>
<td>LINESYNC</td>
<td>SP3RXC-</td>
</tr>
<tr>
<td>31</td>
<td>NETWK3 Rx+</td>
<td>POWERUP</td>
<td>CPU_RESET</td>
</tr>
<tr>
<td>32</td>
<td>NETWK4 Rx-</td>
<td>POWERDOWN</td>
<td>CPU_ACTIVE</td>
</tr>
<tr>
<td>33</td>
<td>DCGND1</td>
<td>DCGND1</td>
<td>DCGND1</td>
</tr>
<tr>
<td>34</td>
<td>+12 VDC</td>
<td>-12 VDC</td>
<td>+5 VDCSTANDBY</td>
</tr>
<tr>
<td>35</td>
<td>+5 VDC</td>
<td>+5 VDC</td>
<td>+5 VDC</td>
</tr>
<tr>
<td>36</td>
<td>DCGND1</td>
<td>DCGND1</td>
<td>DCGND1</td>
</tr>
<tr>
<td>37</td>
<td>+12 ISO VDC</td>
<td>+12 ISO VDC</td>
<td>+12 ISO VDC</td>
</tr>
<tr>
<td>38</td>
<td>DCGND2</td>
<td>DCGND2</td>
<td>DCGND2</td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Optional Second Communications Slot:

<table>
<thead>
<tr>
<th>PIN</th>
<th>ROW A</th>
<th>ROW B</th>
<th>ROW C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SP3TXD+</td>
<td>SP6TXD+</td>
<td>SP5TXD+</td>
</tr>
<tr>
<td>2</td>
<td>SP3TXD-</td>
<td>SP6TXD-</td>
<td>SP5TXD-</td>
</tr>
<tr>
<td>3</td>
<td>SP3RXD+</td>
<td>SP6RXD+</td>
<td>SP5TXC+</td>
</tr>
<tr>
<td>4</td>
<td>SP3RXD-</td>
<td>SP6RXD-</td>
<td>SP5TXC-</td>
</tr>
<tr>
<td>5</td>
<td>SP3RTS+</td>
<td>SP3TXC0+</td>
<td>SP5RXD+</td>
</tr>
<tr>
<td>6</td>
<td>SP3RTS-</td>
<td>SP3TXC0-</td>
<td>SP5RXD-</td>
</tr>
<tr>
<td>7</td>
<td>SP3CTS+</td>
<td>SP3TXC1+</td>
<td>SP5RXC+</td>
</tr>
<tr>
<td>8</td>
<td>SP3CTS-</td>
<td>SP3TXC1-</td>
<td>SP5RXC-</td>
</tr>
<tr>
<td>9</td>
<td>SP3DCD+</td>
<td>SP3RXC+</td>
<td>SP3TXD+</td>
</tr>
<tr>
<td>10</td>
<td>SP3DCD-</td>
<td>SP3RXC-</td>
<td>SP3TXD-</td>
</tr>
<tr>
<td>11</td>
<td>SP4TXD-</td>
<td>SP4TXD-</td>
<td>SP3RXD+</td>
</tr>
<tr>
<td>12</td>
<td>SP4TXD+</td>
<td>SP4TXD+</td>
<td>SP3RXD-</td>
</tr>
<tr>
<td>13</td>
<td>SP4RXD-</td>
<td>SP4RXD-</td>
<td>SP3RTS+</td>
</tr>
<tr>
<td>14</td>
<td>SP4RXD+</td>
<td>SP4RXD+</td>
<td>SP3RTS-</td>
</tr>
<tr>
<td>15</td>
<td>NA</td>
<td>NA</td>
<td>SP3CTS+</td>
</tr>
<tr>
<td>16</td>
<td>NA</td>
<td>NA</td>
<td>SP3CTS-</td>
</tr>
<tr>
<td>17</td>
<td>NA</td>
<td>NA</td>
<td>SP3DCD+</td>
</tr>
<tr>
<td>18</td>
<td>NA</td>
<td>NA</td>
<td>SP3DCD-</td>
</tr>
<tr>
<td>19</td>
<td>NA</td>
<td>NA</td>
<td>SP3TXC0+</td>
</tr>
<tr>
<td>20</td>
<td>NA</td>
<td>NA</td>
<td>SP3TXC0-</td>
</tr>
<tr>
<td>21</td>
<td>DCGND1</td>
<td>C50 ENABLE</td>
<td>SP3TXC1+</td>
</tr>
<tr>
<td>22</td>
<td>NETWK1 Tx+</td>
<td>NA</td>
<td>SP3TXC1-</td>
</tr>
<tr>
<td>23</td>
<td>NETWK2 Tx-</td>
<td>INSTALLED</td>
<td>SP3RXC+</td>
</tr>
<tr>
<td>24</td>
<td>NA</td>
<td>LINESYNC</td>
<td>SP3RXC-</td>
</tr>
<tr>
<td>25</td>
<td>NETWK3 Rx+</td>
<td>POWERUP</td>
<td>CPU_RESET</td>
</tr>
<tr>
<td>26</td>
<td>NETWK4 Rx-</td>
<td>POWERDOWN</td>
<td>CPU_ACTIVE</td>
</tr>
<tr>
<td>27</td>
<td>DCGND1</td>
<td>DCGND1</td>
<td>DCGND1</td>
</tr>
<tr>
<td>28</td>
<td>+12 VDC</td>
<td>-12 VDC</td>
<td>+5 VDCSTANDBY</td>
</tr>
<tr>
<td>29</td>
<td>+5 VDC</td>
<td>+5 VDC</td>
<td>+5 VDC</td>
</tr>
<tr>
<td>30</td>
<td>DCGND1</td>
<td>DCGND1</td>
<td>DCGND1</td>
</tr>
<tr>
<td>31</td>
<td>+12 ISO VDC</td>
<td>+12 ISO VDC</td>
<td>+12 ISO VDC</td>
</tr>
<tr>
<td>32</td>
<td>DCGND2</td>
<td>DCGND2</td>
<td>DCGND2</td>
</tr>
</tbody>
</table>

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.
6.2.3 Mechanical Field Connections

Guidance:

The following Foreword reprinted from the EIA-574 Standard explains the change from EIA-232 to EIA-574:

“The EIA-574 standard was developed in recognition of the fact that a defacto interface standard had appeared in industry which, although it used the Circuit Definitions and Electrical Characteristics of EIA-232-D was implemented on a 9-pin connector instead of the 25-pin connector specified in that Standard. As no standard existed for this interface many manufacturers incorrectly labeled this defacto interface “RS-232” causing confusion to the user community. EIA-574 provides a solution to the problem of incorrect referencing. It also provides the flexibility of a new interface which specifies the use of EIA-562 Electrical Characteristics which, although they are interworkable with EIA-232-D Electrical Characteristics, are capable of higher data signaling rates and being driven from a ± 5 volt supply.”

6.2.3.1 EIA-232 and 574 Field Connections

EIA-574 Port (9-pin)

EIA-574 field connection to the Communications Interface shall be via a 9-pin “D” connector (sockets) mounted on the Communications Interface front panel. The EIA-574 drivers must be capable of supporting either 2x the maximum asynchronous baud rate for the port or 1Mbps, which ever is less. The pin assignments are as follows:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DCD</td>
<td>Carrier Detect</td>
<td>In</td>
</tr>
<tr>
<td>2</td>
<td>RXD</td>
<td>Received Data</td>
<td>In</td>
</tr>
<tr>
<td>3</td>
<td>TXD</td>
<td>Transmitted Data</td>
<td>Out</td>
</tr>
<tr>
<td>4</td>
<td>NA</td>
<td>Not Used</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DC GND</td>
<td>DC Reference</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>NA</td>
<td>Not Used</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>RTS</td>
<td>Request to Send</td>
<td>Out</td>
</tr>
<tr>
<td>8</td>
<td>CTS</td>
<td>Clear to Send</td>
<td>In</td>
</tr>
<tr>
<td>9</td>
<td>NA</td>
<td>Not Used</td>
<td></td>
</tr>
</tbody>
</table>

EIA-232 Port (25-pin)

EIA-232 field connection to the Communications Interface shall be via a 25-pin “D” connector (sockets) compliant to NEMA TS2 Section 3.3.2, including latch mechanism, but excluding support for DTE Ready (pin 20). The EIA-232 drivers must be capable of supporting either 2x the maximum asynchronous baud rate for the port or 1Mbps, which ever is less.
6.2.3.2 EIA-485 Field Connections

EIA-485 field connections to the Communications Interface shall be via a choice of two connector arrangements. The EIA-485 drivers must be capable of supporting either 2x the maximum applicable baud rate for the port or 1Mbps, which ever is less.

EIA-485 Port for Asynchronous Operation

EIA-485 field connection for asynchronous operation shall be via a 9-pin “D” connector (sockets) mounted on the Communications Interface front panel. The pin assignments are as follows:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TXD+</td>
<td>Transmitted Data</td>
<td>Out</td>
</tr>
<tr>
<td>2</td>
<td>TXD-</td>
<td>Transmitted Data</td>
<td>Out</td>
</tr>
<tr>
<td>3</td>
<td>RXD+</td>
<td>Received Data</td>
<td>In</td>
</tr>
<tr>
<td>4</td>
<td>RXD-</td>
<td>Received Data</td>
<td>In</td>
</tr>
<tr>
<td>5</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>RTS+</td>
<td>Request to Send</td>
<td>Out</td>
</tr>
<tr>
<td>7</td>
<td>RTS-</td>
<td>Request to Send</td>
<td>Out</td>
</tr>
<tr>
<td>8</td>
<td>CTS+</td>
<td>Clear to Send</td>
<td>In</td>
</tr>
<tr>
<td>9</td>
<td>CTS-</td>
<td>Clear to Send</td>
<td>In</td>
</tr>
</tbody>
</table>

EIA-485 Port for Synchronous Operation

EIA-485 field connection for synchronous operation shall be via a 15-pin “D” connector (sockets) mounted on the Communications Interface front panel. The pin assignments are as follows:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TXDATA+</td>
<td>Transmitted Data</td>
<td>Out</td>
</tr>
<tr>
<td>2</td>
<td>DC GND</td>
<td>DC Reference</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TXCLOCK+</td>
<td>Transmitter Clock</td>
<td>In/Out</td>
</tr>
<tr>
<td>4</td>
<td>DC GND</td>
<td>DC Reference</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RXDATA+</td>
<td>Receiver Data</td>
<td>In</td>
</tr>
<tr>
<td>6</td>
<td>DC GND</td>
<td>DC Reference</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>RXCLOCK+</td>
<td>Receiver Clock</td>
<td>In</td>
</tr>
<tr>
<td>8</td>
<td>NA</td>
<td>Not Used</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TXDATA-</td>
<td>Transmitted Data</td>
<td>Out</td>
</tr>
<tr>
<td>10</td>
<td>DC GND</td>
<td>DC Reference</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>TXCLOCK-</td>
<td>Transmitter Clock</td>
<td>In/Out</td>
</tr>
<tr>
<td>12</td>
<td>DC GND</td>
<td>DC Reference</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>RXDATA-</td>
<td>Receiver Clock</td>
<td>In</td>
</tr>
<tr>
<td>14</td>
<td>DC GND</td>
<td>DC Reference</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>RXCLOCK-</td>
<td>Receiver Clock</td>
<td>In</td>
</tr>
</tbody>
</table>
6.2.3.3 Internal Private Line Modulator/Demodulator (Modem) Connection

Private phone line twisted pair field connections to the Communications Interface shall be via a choice of two connector arrangements:

**Internal and External Modem Connections**

M14 AMP connector (sockets) mounted on the Communications Interface front panel. This connector includes signals for transmit twisted pair and receive twisted pair phone lines for use with internal modem, plus EIA-574 signals for use with external modem. The pin assignment is as follows:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>AUDIO IN</td>
<td>Phone Line Receive Pair</td>
<td>In (4 Wire)</td>
</tr>
<tr>
<td>B</td>
<td>AUDIO IN</td>
<td>Phone Line Receive Pair</td>
<td>In (4 Wire)</td>
</tr>
<tr>
<td>C</td>
<td>AUDIO OUT</td>
<td>Phone Line Transmit Pair</td>
<td>Out (4 Wire, I/O 2 Wire)</td>
</tr>
<tr>
<td>D</td>
<td>-5 VDC</td>
<td>+5 VDC</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>AUDIO OUT</td>
<td>Phone Line Transmit Pair</td>
<td>Out (4 Wire, I/O 2 Wire)</td>
</tr>
<tr>
<td>F</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>DCD</td>
<td>Carrier Detect</td>
<td>In</td>
</tr>
<tr>
<td>J</td>
<td>RTS</td>
<td>Request to Send</td>
<td>Out</td>
</tr>
<tr>
<td>K</td>
<td>TXD</td>
<td>Transmitted Data</td>
<td>Out</td>
</tr>
<tr>
<td>L</td>
<td>RXD</td>
<td>Received Data</td>
<td>In</td>
</tr>
<tr>
<td>M</td>
<td>CTS</td>
<td>Clear to Send</td>
<td>In</td>
</tr>
<tr>
<td>N</td>
<td>DC GND</td>
<td>DC Reference</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Internal Private Line Only Modem Only Connections**

Nine pin “D” connector (pins) mounted to the Communications Interface front panel. This connector includes signals for transmit and receive twisted pair phone lines for use with internal modem. The NEMA TS-2 pin out assignments are as follows:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AUDIO OUT</td>
<td>Phone Line Transmit Pair</td>
<td>Out (4 Wire), I/O (2 Wire)</td>
</tr>
<tr>
<td>2</td>
<td>AUDIO OUT</td>
<td>Phone Line Transmit Pair</td>
<td>Out (4 Wire), I/O (2 Wire)</td>
</tr>
<tr>
<td>3</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>AUDIO IN</td>
<td>Phone Line Receive Pair</td>
<td>In (4 Wire Only)</td>
</tr>
<tr>
<td>5</td>
<td>AUDIO IN</td>
<td>Phone Line Receive Pair</td>
<td>In (4 Wire Only)</td>
</tr>
<tr>
<td>6</td>
<td>EG</td>
<td>Equipment Ground</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>EG</td>
<td>Equipment Ground</td>
<td></td>
</tr>
</tbody>
</table>
6.2.3.4 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:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Ring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Tip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>NC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>NC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.2.3.5 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.

6.2.3.6 Multi Mode Fiber Connections

Multi-Mode fiber connections to the Communications Interface shall be via 820 nM ST connectors for both transmitters and receivers.

6.2.3.7 Wide Area Radio Connections

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

6.2.3.8 Infrared Connections

Wireless infrared field connections to an external device are via a red transparent window.

6.2.3.9 Ethernet Connections

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

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Description</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TXD+</td>
<td>Transmitter Pair +</td>
<td>Out</td>
</tr>
<tr>
<td>2</td>
<td>TXD-</td>
<td>Transmitter Pair -</td>
<td>Out</td>
</tr>
<tr>
<td>3</td>
<td>RXD+</td>
<td>Receiver Pair +</td>
<td>In</td>
</tr>
<tr>
<td>4</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 6.3 Operational Description

#### 6.3.1 Interface to ATC

##### 6.3.1.1 EIA-485 Signals

Except NETWK1-NETWK4, CPU_RESET, 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 insure 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.

##### 6.3.1.2 Ethernet Signals

NETWK1-NETWK4 are 10 Base-T Ethernet 1 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.

##### 6.3.1.3 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.
6.3.1.4 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 10.1.4 “Electrical Isolation”. Equipment ground (EG) shall maintain the isolation specifications of Section 10.1.4 “Electrical Isolation”.

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

Field connections of the EIA-574 and EIA-485 versions of the Communications Interface shall be optically 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.

6.3.1.5 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.

6.3.2 Modulation and Demodulation

6.3.2.1 EIA-574

(Guidance: This paragraph is intended to represent the present 2070-7A).

Description:

The EIA-574 versions of the Communications Interface shall convert the ATC EIA-485 signals to EIA-574 bipolar simplex, meaning each signal is unidirectional, point-to-point, without ability to disable the transmitter, unless C50_Enable is active.

Indicators:

EIA-574 versions of the Communications Interface shall include the following indicators:
6.3.2.2 EIA-485

(Guidance: This paragraph is intended to represent the present 2070-7B).

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:

<table>
<thead>
<tr>
<th>Front Panel</th>
<th>Legend</th>
<th>Indicator Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX</td>
<td>ON=DATA+ at Field Wire is 0V, DATA- at Field Wire is Positive V</td>
<td></td>
</tr>
<tr>
<td>RX</td>
<td>ON=DATA+ at Field Wire is 0V, DATA- at Field Wire is Positive V</td>
<td></td>
</tr>
</tbody>
</table>

Specifications:

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

6.3.2.3 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 Clear to Send (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:**

<table>
<thead>
<tr>
<th>Front Panel</th>
<th>Indicator Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legend</td>
<td></td>
</tr>
<tr>
<td>TX</td>
<td>ON= SPACE Tone at Field Wire</td>
</tr>
<tr>
<td>RX</td>
<td>ON= SPACE Tone at Field Wire</td>
</tr>
<tr>
<td>CD</td>
<td>ON= Received Tone Within Specified Sensitivity and Filter Band</td>
</tr>
</tbody>
</table>

**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
## Advanced Transportation Controller (ATC) Standard
### COMMUNICATION INTERFACE DETAILS

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTS to CTS Delay</td>
<td>11 ms ± 3 ms</td>
</tr>
<tr>
<td>Carrier Detect</td>
<td>8 ms ± 2 ms MARK frequency</td>
</tr>
<tr>
<td>Receiver Squelch</td>
<td>6.5 ms ± 1 ms, 0 ms (OUT)</td>
</tr>
<tr>
<td>Soft Carrier OFF</td>
<td>10 ms ± 2 ms</td>
</tr>
<tr>
<td>Recovery Time</td>
<td>22 ms maximum from Transmit to Receive</td>
</tr>
<tr>
<td>Error Rate</td>
<td>Less than 1 bit in 100,000 bits</td>
</tr>
<tr>
<td>Signal to Noise</td>
<td>16 dBm measured with flat-weight over a 300 to Controller Hz band</td>
</tr>
<tr>
<td>Transmit Noise</td>
<td>-50 dBm maximum into 600 Ω resistive load within frequency spectrum of 300 to Controller Hz band at maximum output</td>
</tr>
<tr>
<td>Anti-Stream Time</td>
<td>6 to 8 seconds</td>
</tr>
</tbody>
</table>

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 19.2KHz and 38.4KHz ), 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

6.3.2.4 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:

<table>
<thead>
<tr>
<th>Front Panel Legend</th>
<th>Indicator Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX</td>
<td>ON= Transmitted Data Activity</td>
</tr>
<tr>
<td>RX</td>
<td>ON= Received Data Activity</td>
</tr>
<tr>
<td>CD</td>
<td>ON= Received Tone Activity</td>
</tr>
<tr>
<td>RS</td>
<td>ON= RTS Asserted</td>
</tr>
</tbody>
</table>
Specifications:

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

6.3.2.5 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:

<table>
<thead>
<tr>
<th>Front Panel</th>
<th>Indicator Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX</td>
<td>ON= Transmitter MARK state</td>
</tr>
<tr>
<td>RX</td>
<td>ON= Receiver MARK state</td>
</tr>
</tbody>
</table>

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

6.3.2.6 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:

<table>
<thead>
<tr>
<th>Front Panel</th>
<th>Legend</th>
<th>Indicator Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX</td>
<td>ON</td>
<td>Transmitter MARK state</td>
</tr>
<tr>
<td>RX</td>
<td>ON</td>
<td>Receiver MARK state</td>
</tr>
</tbody>
</table>

**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

**6.3.2.7 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 insures 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:

<table>
<thead>
<tr>
<th>Front Panel</th>
<th>Legend</th>
<th>Indicator Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX</td>
<td>ON</td>
<td>Transmitted Data Activity</td>
</tr>
<tr>
<td>RX</td>
<td>ON</td>
<td>Received Data Activity</td>
</tr>
</tbody>
</table>

**Specifications:**

- Shall be specified by radio manufacturer.

**6.3.2.8 Infrared**

*(Guidance: This paragraph is intended to represent an interface to a standard PDA. Agencies with single door cabinets shall insure, 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                                    |

### 6.3.2.9 Ethernet

**Description:**

The Ethernet version of the Communications Interface shall adapt the ATC NETWK1-4 signals. The Ethernet port may be directly tied to NETWK1-4 or buffered as a hub.

**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:

<table>
<thead>
<tr>
<th>Front Panel Legend</th>
<th>Indicator Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>ON= Transmitted Data or Received Data is logic “1”</td>
</tr>
<tr>
<td>100</td>
<td>ON= 100 MBPS Data Rate</td>
</tr>
</tbody>
</table>

**Specifications:**

The electrical specifications and signal definition shall conform to the requirements of IEEE 802.3.
6.4 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 Paragraph 6.2.1
- A connection to the ATC serial ports and power, as described in Paragraph 6.2.2
- One or more communications ports described in Paragraph 6.2.3
- Modulation / demodulation circuitry for each port, described in Paragraph 6.3.2

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

Guidance:

The following is a list of the existing Communications Interface Versions:

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2070-6A</td>
<td>Dual 300 to 1200 bps Modem</td>
</tr>
<tr>
<td>2070-6B</td>
<td>Dual 0 to 9600 bps Modem</td>
</tr>
<tr>
<td>2070-7A</td>
<td>Dual EIA-574 Serial Interface</td>
</tr>
<tr>
<td>2070-7B</td>
<td>Dual EIA-485 Serial Interface</td>
</tr>
</tbody>
</table>

As new versions are defined, this list will expand.

Each version may be implemented using any of the following three design methods:

- Dedicated circuit design, each version ordered as separate vendor part numbers
- Common base board, with selectable modulation via plug-in circuit assemblies
- Common board, with selectable modulation via digital signal processor (DSP) software

Please refer to the Joint NEMA/AASHTO/ITE ATC 2070 Standard for detailed specifications of the 2070-6A, 2070-6B, 2070-7A and 2070-7B.
7 PHYSICAL AND USER INTERFACE DETAILS

7.1 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 personal computers and PDAs. For example, the User Interface of a NEMA controller is normally a keyboard and display, with NEMA Port 2 allocated to a personal computer or PDA. The ATC 2070 provides its User Interface via either a keyboard and display mounted in its Front Panel Assembly, or a serial port connector for a personal computer or PDA. Going forward, it is the intent of this specification to:

- Preserve compatibility with existing ATC 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:

- Preserve User Interface interchangeability among vendors
- Dictate User Interface requirements, other than minimum and optional
- Limit the choices of User Interfaces

7.1.1 Minimum User Interface

The User Interface performs two separate and necessary functions

- User Interface to the Application (Keyboard and Display, for example)
- User Interface to the O/S (Updating application software, O/S and drivers)

This standard specifies a minimum interface to the Application, plus a minimum interface to the O/S. This minimum interface provides a common method to enter data and update software for all hardware and software suppliers. In this standard, alternative user interfaces may be included provided that the minimal interface is provided also.

7.1.1.1 Minimum User Interface to the Application

The minimum user interface to the Application shall consist of the following:

- EIA-574 SP6 connector, 9 pin “D” (Guidance: C60P of ATC 2070)
7.1.1.2 Minimum User Interface to the O/S

The minimum User Interface to the O/S shall consist of the following:

- EIA-574 SP4 connector for O/S, 9 pin “D” (Guidance: C50S of ATC 2070)
- CPU ACTIVE LED Indicator
- Ethernet Port (Guidance: Internal ATC 10/100 hub Port 2)
- USB Port, for removable memory device, only.

7.1.2 Optional User Interfaces

In addition to the minimum User Interface, the ATC may include one or more optional User Interfaces.

7.1.2.1 Optional User Interfaces to the Application

Option 1: Keyboard, LCD, Bell, and AUX Switch (Guidance: traditional ATC 2070)
Option 2: Infrared Port for PDA or Laptop (Guidance: PDA IRDA COM2)
Option 3: Ethernet interface to graphics device (Guidance: Flat panel LCD)

7.1.2.2 Optional User Interfaces to the O/S

Option 1: Infrared Port for PDA or Laptop (Guidance: PDA IRDA COM2)

7.1.3 User Interface Pin Connections

<table>
<thead>
<tr>
<th>SP4 Connector Pinout</th>
<th>SP6 Connector Pinout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin</td>
<td>Function</td>
</tr>
<tr>
<td>1</td>
<td>C50_ENABLE</td>
</tr>
<tr>
<td>2</td>
<td>SP4_RXD</td>
</tr>
<tr>
<td>3</td>
<td>SP4_TXD</td>
</tr>
<tr>
<td>4</td>
<td>NA</td>
</tr>
<tr>
<td>5</td>
<td>DCGND3</td>
</tr>
<tr>
<td>6</td>
<td>NA</td>
</tr>
<tr>
<td>7</td>
<td>NA</td>
</tr>
</tbody>
</table>
Guidance:

1. 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.

2. DCGND3 is the ground reference for all signals on front panel SP4 and SP6 connectors and the LCD Heater Power Supply

7.1.4 User Interface Operation

7.1.4.1 Keyboard, LCD and Bell Operation

Keyboard
The Keyboard, at a minimum, shall be capable of the complete single keystroke functionality (without key translations) of the standard ATC 2070 front panel. Each key shall be engraved or embossed with its function character. Minimum key size shall be 0.3” x 0.3”. Minimum key spacing shall be 0.5” on centers. The actual keypad arrangement is not specified here.

CPU_ACTIVE LED Indicator
The cathode of the CPU_ACTIVE LED Indicator shall be electrically connected to the CPU_ACTIVE LED signal and shall have the pull-up resistor on the front panel.

Display
The Display shall consist of a Liquid Crystal Display (LCD), backlight and a contrast control. Other dot matrix display technologies are allowed, but shall meet all requirements of this document. The contrast control can either be a potentiometer or a software-controlled contrast adjustment. If using a potentiometer contrast control, the contrast shall increase with clockwise rotation. If using a software controlled contrast, the contrast control shall be accomplished by pressing the (*) key to enable the adjustment, followed by the (+) key to darken and the (-) key to lighten the contrast. By pressing the (*) key again will disable the contrast adjustment. The contrast adjustment shall provide the entire contrast range of the LCD.

The Display shall have an LED or electro-luminescent backlight. The backlight shall be turned on and off by the Controller Circuitry. The backlight and associated circuitry shall consume no power when in off state. The Display shall have a minimum of 8 lines with 40 characters each with typical 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.

**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**

The User Interface shall be reset once power is applied or have a momentary control reset switch on the PCB that is logic ORed with the CPU_RESET Line, producing a USER Interface RESET. 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 (hex value 20).
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 backlight shall be extinguished.
6. The display shall be cleared (all ASCII space).
7. 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”, (hex values “1B 5B 50 55”).

**Key Press**

When a key press is detected, the appropriate key code shall be transmitted to SP6 RXD. If two or more keys are depressed simultaneously, no code shall be sent. If a key is depressed while another key is depressed, no additional 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 5 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 ATC 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% ON / 40% 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 shall be refreshed at least 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:

1. **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' (hex value 68) and 'l' (hex value 6C) are used to indicate binary operations. 'h' represents ON (high), 'l' represents OFF (low).

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

3. 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 38.4 kbps, 8 data bits, 1 stop bit, and no parity.

Bell
The User Interface shall include an electronic bell to signal receipt of ^G (hex value 07). Receipt of all other characters and ESC codes shall continue during the time the bell sounds. Bell shall be rated at 70 decibels minimum sound pressure at 4 KHz.

Configuration Command Codes

<table>
<thead>
<tr>
<th>ASCII REPRESENTATION</th>
<th>HEX VALUE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>HT</td>
<td>09</td>
<td>Move cursor to next tab stop</td>
</tr>
<tr>
<td>CR</td>
<td>0D</td>
<td>Position cursor at first position on current line</td>
</tr>
<tr>
<td>LF</td>
<td>0A</td>
<td>(Line Fee) Move cursor down one line</td>
</tr>
<tr>
<td>BS</td>
<td>08</td>
<td>(Backspace) Move cursor one position to the left and write space</td>
</tr>
<tr>
<td>ESC 0 Py j Px f</td>
<td>1B 5B Py 3B Px 66</td>
<td>Position cursor at (Px, Py)</td>
</tr>
<tr>
<td>ESC 0 Pn C</td>
<td>1B 5B Pn 43</td>
<td>Position cursor Pn positions to right</td>
</tr>
<tr>
<td>ESC 0 Pn D</td>
<td>1B 5B Pn 44</td>
<td>Position cursor Pn positions to left</td>
</tr>
<tr>
<td>ESC 0 Pn A</td>
<td>1B 5B Pn 41</td>
<td>Position cursor Pn positions up</td>
</tr>
<tr>
<td>ESC 0 Pn B</td>
<td>1B 5B Pn 42</td>
<td>Position cursor Pn positions down</td>
</tr>
<tr>
<td>ESC 0 H</td>
<td>1B 5B 48</td>
<td>Home cursor (move to 1,1)</td>
</tr>
<tr>
<td>ESC 0 2 j</td>
<td>1B 5B 32 4A</td>
<td>Clear screen with spaces without moving cursor</td>
</tr>
<tr>
<td>ESC c</td>
<td>1B 63</td>
<td>Soft reset</td>
</tr>
<tr>
<td>ESC 0 P1 [ Pn j Pn...f</td>
<td>1B 50 P1 5B Pn 3B...Pn 66</td>
<td>Compose special character number Pn (1-8) at current cursor position</td>
</tr>
<tr>
<td>ESC 0 &lt; Pn V</td>
<td>1B 5B 3C Pn 56</td>
<td>Display special character number Pn (1-8) at current cursor position</td>
</tr>
<tr>
<td>ESC 0 25 h</td>
<td>1B 5B 32 35 68</td>
<td>Turn Character blink on</td>
</tr>
<tr>
<td>ESC 0 25 j</td>
<td>1B 5B 32 35 6C</td>
<td>Turn Character blink off</td>
</tr>
<tr>
<td>ESC 0 &lt; 5 h</td>
<td>1B 5B 3C 35 68</td>
<td>Illuminate Backlight</td>
</tr>
<tr>
<td>ESC 0 &lt; 5 l</td>
<td>1B 5B 3C 35 6C</td>
<td>Extinguish Backlight</td>
</tr>
</tbody>
</table>
### Configuration Command Codes

<table>
<thead>
<tr>
<th>ASCII Representation</th>
<th>HEX Value</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC 0 33 h</td>
<td>1B 5B 33 33 68</td>
<td>Cursor blink on</td>
</tr>
<tr>
<td>ESC 0 33 l</td>
<td>1B 5B 33 33 6C</td>
<td>Cursor blink off</td>
</tr>
<tr>
<td>ESC 0 27 h</td>
<td>1B 5B 32 37 68</td>
<td>Reverse video on</td>
</tr>
<tr>
<td>ESC 0 27 l</td>
<td>1B 5B 32 37 6C</td>
<td>Reverse video off</td>
</tr>
<tr>
<td>ESC 0 24 h</td>
<td>1B 5B 32 34 68</td>
<td>Underline on</td>
</tr>
<tr>
<td>ESC 0 24 l</td>
<td>1B 5B 32 34 6C</td>
<td>Underline off</td>
</tr>
<tr>
<td>ESC 0 0 m</td>
<td>1B 5B 30 6D</td>
<td>All attributes off</td>
</tr>
<tr>
<td>ESC H</td>
<td>1B 48</td>
<td>Set tab stop at current cursor position</td>
</tr>
<tr>
<td>ESC 0 Pn g</td>
<td>1B 5B Pn 67</td>
<td>Clear tab stop Pn = 0,1,2 at cursor = 3 all tab stops</td>
</tr>
<tr>
<td>ESC 0 ? 7 h</td>
<td>1B 5B 30 37 68</td>
<td>Auto-wrap on</td>
</tr>
<tr>
<td>ESC 0 ? 7 l</td>
<td>1B 5B 30 37 6C</td>
<td>Auto-wrap off</td>
</tr>
<tr>
<td>ESC 0 ? 8 h</td>
<td>1B 5B 30 38 68</td>
<td>Auto-repeat on</td>
</tr>
<tr>
<td>ESC 0 ? 8 l</td>
<td>1B 5B 30 38 6C</td>
<td>Auto-repeat off</td>
</tr>
<tr>
<td>ESC 0 ? 25 h</td>
<td>1B 5B 30 32 35 68</td>
<td>Cursor on</td>
</tr>
<tr>
<td>ESC 0 ? 25 l</td>
<td>1B 5B 30 32 35 6C</td>
<td>Cursor off</td>
</tr>
<tr>
<td>ESC 0 ? 47 h</td>
<td>1B 5B 3C 34 37 68</td>
<td>Auto-scroll on</td>
</tr>
<tr>
<td>ESC 0 ? 47 l</td>
<td>1B 5B 3C 34 37 6C</td>
<td>Auto-scroll off</td>
</tr>
<tr>
<td>ESC 0 &lt; Pn S</td>
<td>1B 5B 3C Pn 53</td>
<td>Set Backlight timeout value to Pn (0-63)</td>
</tr>
<tr>
<td>ESC 0 Pu</td>
<td>1B 5B 50 55</td>
<td>String sent to ENGINE BOARD when EIPA power up</td>
</tr>
</tbody>
</table>

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

### Inquiry Command – Response Codes

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine Board to Front Panel Module</td>
<td>Front Panel Module to Engine Board</td>
<td></td>
</tr>
<tr>
<td>ACIII Representation</td>
<td>HEX Value</td>
<td>ASCII Representation</td>
</tr>
<tr>
<td>ESC 0 6 n</td>
<td>1B 5B 36 60</td>
<td>ESC O Py, Px</td>
</tr>
<tr>
<td>ESC 0 B n</td>
<td>1B 5B 42 60</td>
<td>ESC 0 P1; P2;….P6 R</td>
</tr>
<tr>
<td>ESC 0 A n</td>
<td>1B 5B 41 6E</td>
<td>ESC 0 P1 R</td>
</tr>
</tbody>
</table>

*Table 7-1: Configuration Command Codes*
### Key Codes

<table>
<thead>
<tr>
<th>Key</th>
<th>ASCII DATA (TEXT)</th>
<th>ASCII DATA (HEX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>34</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>36</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>37</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>38</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>39</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>41</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>42</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>43</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>44</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>45</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>46</td>
</tr>
<tr>
<td>(UP ARROW)</td>
<td>ESC [ A</td>
<td>1B 5B 41</td>
</tr>
<tr>
<td>(DOWN ARROW)</td>
<td>ESC [ B</td>
<td>1B 5B 42</td>
</tr>
<tr>
<td>(RIGHT ARROW)</td>
<td>ESC [ C</td>
<td>1B 5B 43</td>
</tr>
<tr>
<td>(LEFT ARROW)</td>
<td>ESC [ D</td>
<td>1B 5B 44</td>
</tr>
<tr>
<td>ESC</td>
<td>ESC O S</td>
<td>1B 4F 53</td>
</tr>
<tr>
<td>NEXT</td>
<td>ESC O P</td>
<td>1B 4F 50</td>
</tr>
<tr>
<td>YES</td>
<td>ESC O Q</td>
<td>1B 4F 51</td>
</tr>
<tr>
<td>NO</td>
<td>ESC O R</td>
<td>1B 4F 52</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>2A</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>2B</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>2D</td>
</tr>
<tr>
<td>ENTER</td>
<td>CR</td>
<td>OD</td>
</tr>
</tbody>
</table>

### AUX Switch Codes

<table>
<thead>
<tr>
<th>SWITCH POSITION</th>
<th>ASCII DATA (TEXT)</th>
<th>ASCII DATA (HEX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>ESC O T</td>
<td>1B 4F 54</td>
</tr>
<tr>
<td>OFF</td>
<td>ESC O U</td>
<td>1B 4F 55</td>
</tr>
</tbody>
</table>

### 7.1.4.2 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, such as a PDA may be used. Note that the HEATER POWER signal is optional with actual Voltage and Current at the manufacturer’s discretion and with no expectation of interchangeability. DCGND3 is the
related ground signal (pin 5). Intent is that Heater is controlled by Front Panel activity only.

7.1.4.3 Infrared Port
This option specifies short distance wireless communications via a modulated light beam. Please refer to the IRDA standards for operation.

7.1.4.4 Ethernet Port
A 10/100 Ethernet port is used for hardwired communications to external devices. Please refer to the IEEE 802.3 standard for operation.

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

7.1.4.6 Data Key
Datakey Keyceptacle™ (KC4210, KC4210PCB or equal) with a 2Mb Datakey™ (SFK2Mb with 64 KB sector size).

7.1.4.7 Graphics Interface
If a Display device such as an LCD is provided, then it shall support a mandatory graphics interface. The graphics interface shall operate as specified herein. The graphic interface commands are supplemental to the normal “configuration command codes” and functionality specified in section 7.1.4.1.

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 x 1 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 be 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.
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:

<table>
<thead>
<tr>
<th>Function</th>
<th>ASCII Representation (“fn”)</th>
<th>HEX Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>C</td>
<td>43</td>
</tr>
<tr>
<td>Fill</td>
<td>F</td>
<td>46</td>
</tr>
<tr>
<td>AND</td>
<td>&amp;</td>
<td>26</td>
</tr>
<tr>
<td>OR</td>
<td>+</td>
<td>2B</td>
</tr>
<tr>
<td>XOR</td>
<td>X</td>
<td>58</td>
</tr>
<tr>
<td>Write</td>
<td>W</td>
<td>57</td>
</tr>
</tbody>
</table>

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 (00 hex) can be used to clear the block of graphic bits, while a value of 255 (FF hex) 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 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:

```
  1  2  3  4  5  6  7  8  9 10 11 12 13
 14
 15
```

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

As a second example, the graphic block:

```
  1  2  3  4  5  6  7  8  9 10 11 12
 13
 14
 15
```

would be represented by the data bytes (hex values) “CC CC CC C0” and it could also be created using a Fill command with the single data value “CC”.

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 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>
<thead>
<tr>
<th>ASCII REPRESENTATION</th>
<th>HEX VALUE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC G U 0</td>
<td>1B 47 61 30</td>
<td>Disable Underlay/User graphic layer</td>
</tr>
<tr>
<td>ESC G U 1</td>
<td>1B 47 61 31</td>
<td>Enable Underlay/User graphic layer in 1x1 pixel mode</td>
</tr>
<tr>
<td>ESC G U 2</td>
<td>1B 47 61 32</td>
<td>Enable Underlay/User graphic layer in 2x2 pixel mode</td>
</tr>
<tr>
<td>ESC G O 0</td>
<td>1B 47 5B 30</td>
<td>Disable Overlay graphic layer</td>
</tr>
<tr>
<td>ESC G O 1</td>
<td>1B 47 5B 31</td>
<td>Enable Overlay graphic layer in 1x1 pixel mode</td>
</tr>
<tr>
<td>ESC G O 2</td>
<td>1B 47 5B 32</td>
<td>Enable Overlay graphic layer in 2x2 pixel mode</td>
</tr>
<tr>
<td>ESC G C U</td>
<td>1B 47 43 30</td>
<td>Disable Graphic Cursor</td>
</tr>
<tr>
<td>ESC G C U</td>
<td>1B 47 43 31</td>
<td>Enable Graphic Cursor on Underlay/User graphic layer</td>
</tr>
<tr>
<td>ESC G C O</td>
<td>1B 47 43 32</td>
<td>Enable Graphic Cursor on Overlay graphic layer</td>
</tr>
<tr>
<td>ESC G P x y</td>
<td>1B 47 5C XX xx yy</td>
<td>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.</td>
</tr>
<tr>
<td>ESC G p sx sy</td>
<td>1B 47 6F XX xx YY yy</td>
<td>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.</td>
</tr>
<tr>
<td>ESC G O fn x1 y1 x2 y2 [data]</td>
<td>1B 47 5B fn XX xx YY yy [data]</td>
<td>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.</td>
</tr>
<tr>
<td>ESC G U fn x1 y1 x2 y2 [data]</td>
<td>1B 47 61 fn XX xx YY yy [data]</td>
<td>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.</td>
</tr>
<tr>
<td>ESC G o fn Δx Δy [data]</td>
<td>1B 47 6F fn XX xx yy [data]</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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 =
Advanced Transportation Controller (ATC) Standard
PHYSICAL AND USER INTERFACE DETAILS

GRAPHIC CONFIGURATION COMMAND CODES

<table>
<thead>
<tr>
<th>ASCII REPRESENTATION</th>
<th>HEX VALUE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(unsigned) 16-bit with XX = MSB and xx = LSB, and ∆y = (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.</td>
</tr>
<tr>
<td>ESC G G +</td>
<td>1B 47 47 2B</td>
<td>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).</td>
</tr>
<tr>
<td>ESC G G &amp;</td>
<td>1B 47 47 26</td>
<td>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).</td>
</tr>
<tr>
<td>ESC G G X</td>
<td>1B 47 47 58</td>
<td>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).</td>
</tr>
<tr>
<td>ESC G D +</td>
<td>1B 47 44 2B</td>
<td>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).</td>
</tr>
<tr>
<td>ESC G D &amp;</td>
<td>1B 47 44 26</td>
<td>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).</td>
</tr>
<tr>
<td>ESC G D X</td>
<td>1B 47 44 58</td>
<td>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).</td>
</tr>
</tbody>
</table>

7.1.5 User Interface Power Requirements

The User Interface shall be powered by 4.8 to 5.2 VDC. Any additional voltages required by the User Interface, such as backlight and communications, shall be derived from this single power source. If the User Interface can be mounted remotely, the typical and maximum current requirements of each User Interface shall be published for each device.
7.2 Power Supply General Description

The Power Supply shall be an independent module, 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 7.2.6.

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.

7.2.1 "ON/OFF" Power Switch

An "On/Off" POWER Switch shall be provided to disconnect AC from the Power Supply. The “Power On” shall be in the up position.

7.2.2 LED DC Power Indicators

Four LED DC Power Indicators shall be provided to indicate that all required DC voltages meet the following conditions:

a. +5 VDC is within 4.8V to 5.25V and the ± 12 VDC voltages are within ±8% of nominal.

b. 332 Parallel I/O versions, the +12 VDC ISO shall be within ±8% of nominal.

c. NEMA versions, the +24 VDC shall be within NEMA TS-2 tolerance.

7.2.3 Service Voltage Fuse

A replaceable 3AG slow blow fuse shall be provided. Fuse label shall indicate rating.

7.2.4 +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, hold up 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 Hold Up 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.

7.2.5 Monitor Circuitry

Monitor Circuitry shall be provided to monitor incoming Service Power for Power Failure and Restoration and LiNESYNC generation.

7.2.5.1 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 of when 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 7.2.6.

The POWERUP Output signal shall transition to LOW 525 ± 20 ms after Loss of AC Service Power. 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 7.2.6).

Power Loss and Restoration Calibration for NEMA Cabinets:

The POWERDOWN Output signal shall go LOW (ground true) within 50 ± 10 ms of when the AC service voltage falls below 85 ±2 VAC ("Loss of AC Service Power"). The signal shall transition to HIGH within 50 ms of when both the AC service voltage exceeds 90 ± 2 VAC ("Restoration of AC Service Power") and the +5VDC supply is within the range specified in section 7.2.6.

The POWERUP Output signal shall transition to LOW 525 ± 20 ms after Loss of AC Service Power. 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 7.2.6).
Figure 7-1: Power Supply Signals during Restoration of AC Service Power

![Power Supply Signals during Restoration of AC Service Power](image1)

Notes:
1. Restoration of AC Service Power occurs at $t = 0$ ms
2. Power supply fully recovered
3. $t_2 = (t_1 + 225 \text{ms} \pm 25 \text{ms}) \leq 750$ ms

Figure 7-2: Power Supply Signals during Loss of AC Service Power

![Power Supply Signals during Loss of AC Service Power](image2)

Notes:
1. Loss of AC Service Power occurs at $t = 0$ ms
7.2.5.2 LINESYNC

The LINESYNC signal shall be generated by a crystal oscillator, which shall synchronize to the 60-Hz AC service power line at 120 and 300°. A continuous square-wave signal shall be +5 VDC amplitude, 60 Hz, 8.333 ms half-cycle pulse duration, and 50 ± 1% duty cycle during loss of AC service power. The output shall have drive sink capability of 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 7.2.6 (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% at 25 C. The relationship of AC Service Voltage (top trace) and LINESYNC (bottom trace) is shown in Figure 7-4.
7.2.6 External Power Supply Requirements

The following external voltages shall be within these parameters.

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

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Tolerances</th>
<th>I Minimum</th>
<th>I Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>+5 VDC</td>
<td>+4.875 to +5.125 VDC</td>
<td>0.050 A</td>
<td>0.500 A</td>
</tr>
<tr>
<td>+12 VDC</td>
<td>+11.4 to +12.6 VDC</td>
<td>0.050 A</td>
<td>0.100 A</td>
</tr>
<tr>
<td>-12 VDC</td>
<td>-11.4 to -12.6 VDC</td>
<td>0.050 A</td>
<td>0.100 A</td>
</tr>
</tbody>
</table>

For NEMA TS1 and NEMA TS2 Type 2 versions:

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Tolerances</th>
<th>I Minimum</th>
<th>I Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>+24 VDC</td>
<td>+22.0 to +26.0VDC</td>
<td>0.050 A</td>
<td>0.500 A</td>
</tr>
</tbody>
</table>

For 332 Parallel I/O version:

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Tolerances</th>
<th>I Minimum</th>
<th>I Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>+12 VDC ISO</td>
<td>+11.4 to +12.6 VDC</td>
<td>0.050 A</td>
<td>0.750 A</td>
</tr>
</tbody>
</table>

For the Required USB Port:

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Tolerances</th>
<th>I Minimum</th>
<th>I Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>+5 VDC</td>
<td>+4.875 to +5.125 VDC</td>
<td>0.000 A</td>
<td>0.500 A</td>
</tr>
</tbody>
</table>
For the optional Display (LCD) Heater:

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Tolerances</th>
<th>I Minimum</th>
<th>I Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to +15 VDC</td>
<td>n/a</td>
<td>0.000 A</td>
<td>1.000 A</td>
</tr>
</tbody>
</table>

7.2.6.1 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: 100 VAC to 135 VAC ± 2 VAC
b. NEMA: 89 VAC to 135 VAC ± 2 VAC

7.2.6.2 Ripple and Noise
Less than 0.5% rms, 2% peak to peak, whichever is greater.

7.2.6.3 Over Voltage
The Power Supply shall clamp at 130% Vout for all outputs.

7.2.6.4 Inrush Current
Cold Start Inrush shall be less than 25A at 115VAC.

7.2.6.5 Holdup Time
The power supply shall supply +5 VDC current budget for 550 ms after power loss at 100 VAC. 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 100 VAC. Since the Engine Board is powered completely by +5 VDC, no other power supply output voltages shall need to be maintained during power loss to prevent reboot.

7.2.6.6 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.

7.3 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:

- 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:

- Interchange electronic modules and mechanical assemblies among vendors (except Communications Interface and Engine Board)
- Dictate mechanical details
- Preserve existing controller sizes and form factors

7.3.1 Chassis

7.3.1.1 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.
7.3.1.2 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.

7.3.1.3 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.

7.3.1.4 Dimensions (All dimensions are given in inches)
Details of Maximum Basic Dimensions (not restricted to shape shown)

Figure 7-6: Maximum Basic Dimensions, Rack Mount

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

Minimum and Optional components located on front of assembly are as follows:

SP4 - 9-Pin D Socket Type
SP6 - 9-Pin D Plug Type
Infrared Communication Port
Front Panel Display
Keyboard
ON/OFF Power Switch
Power Supply Service Fuse Holder (with 3AG fuse)
LED for each DC power source and “ACTIVE” indications
USB series A 4 Pin Receptacle Communication Port
RJ-45 Connector ETHERNET Hub Port 3 (Network Diagnostics)
RJ-45 Connector ETHERNET Hub Port 6 (Controller Diagnostics)
Two Ethernet LEDs, labeled “100” and “ACTIVE” for Hub Port 3
Two Ethernet LEDs, labeled “100” and “ACTIVE” for Hub Port 6

Minimum and Optional components located on rear of assembly (rack mount) or on the front of assembly (shelf mount) are as follows:

Parallel I/O C1S - M104 Type
C11S – 37-Pin Circular Plastic Type
Serial I/O Port 1 – 15-Pin D Socket Type
C13S – 25-Pin D Socket Type
Datakey Keyceptacle™KC4210, KC4210PCB or equal
ATC Communications Module Slots, One or More
RJ-45 Connector ETHERNET Hub Port 2 (Network Connection)
Two Ethernet LEDs, labeled “100” and “ACTIVE” for Hub Port 2
NEMA MSA, MSB, MSC, and MSD connectors

Transmitter and Receiver activity is displayed on the “ACTIVITY” LED. The “100” LED is illuminated when the hub port is linked at 100 Mbps and extinguished at all other times.
8 PARALLEL AND SERIAL I/O DETAILS

8.1 General Information

The ATC Input / Output (I/O) provides both serial and parallel connections to field devices connected to the ATC, as well as the input of service power.

8.1.1 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
- Model 332
- NEMA/AASHTO/ITE ITS

Guidance: Access to devices residing on a high-speed computer bus may, for example, be interfaced via the ATC Ethernet hub port 6. The user is responsible for providing an external rack, power supply, high-speed bus and Ethernet interface module residing in this external rack, as well as all Ethernet software drivers.

8.1.2 Serial I/O Overview

The serial connections described here provides 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
- Joint NEMA/AASHTO/ITE/ATC 2070
8.2 Parallel Input / Output (PI/O)

8.2.1 Parallel Connection to Model 332 Cabinets

The parallel connection to a Model 332 Cabinet shall consist of the FCU, Parallel Input/Output Ports, Connectors C1S, and C11S, C12S and other Circuit Functions including muzzle jumper.

8.2.1.1 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.

8.2.1.2 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.
8.2.1.3 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% (±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.

**Communications Loss**
Following Power Up initialization, the FCU loss of communications timer shall indicate loss of communications with the ATC until the user program sends the Request Module Status message to reset the “E” Bit and a subsequent set output command is processed.
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.

8.2.1.4 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:

<table>
<thead>
<tr>
<th>Input Transition Entry</th>
<th>Description</th>
<th>msb</th>
<th>Isb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transition Entry Identifier</td>
<td>S</td>
<td>Input Number</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Timestamp NLSB</td>
<td>x x x x x x</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp LSB</td>
<td>x x x x x x x x</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Millisecond Counter Rollover Entry</th>
<th>Description</th>
<th>msb</th>
<th>Isb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rollover Entry Identifier</td>
<td>1 1 1 1 1 1 1 1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp MSB</td>
<td>x x x x x x x x</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp NMSB</td>
<td>x x x x x x x x</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.2.1.5 I/O Functions

Inputs

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. The condition of the outputs shall only be “ON” if the PI/O continues to receive active communications from the Engine Board. If there is no valid communications with the Engine Board for 2.0 s, all outputs shall revert to the OFF condition, and the PI/O status byte shall be updated to reflect the loss of communication from the Engine Board.

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

<table>
<thead>
<tr>
<th>Case</th>
<th>Output Data Bit</th>
<th>Output Control Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>Output in the OFF state</td>
</tr>
</tbody>
</table>

Output Bit Translation
Advanced Transportation Controller (ATC) Standard
PARALLEL AND SERIAL I/O DETAILS

<table>
<thead>
<tr>
<th>B</th>
<th>1</th>
<th>1</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0</td>
<td>Output is in the ON state.</td>
</tr>
</tbody>
</table>

**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.

**8.2.1.6 Other Processor Functions**

**Interrupts**

All interrupts shall be capable of asynchronous operation with respect to all processing and all other interrupts. 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. LINESYNC Interrupt - This 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.
1  **Input Processing**
2  This task shall process the raw input data scanned in by the 1 ms interrupt routine,  
3  perform all filtering, and maintain the transition queue entries.

4  
5  **8.2.1.7 Data Communications Protocols**

6  **Protocols**
7  All communication with the Engine Board shall be via command-response protocol. The 
8  Engine Board shall always initiate the communication and should the command frame 
9  be incomplete or in error, no PI/O response shall be transmitted. The amount of bytes of 
10  a command or response is dependent upon the I/O Module identification. The physical 
11  interface is not controlled by this specification, and interchangeability among vendors 
12  from PI/O to Engine Board is not intended.

13  
14  **Guidance:**
15  
16  For example, communications to PI/O module may be implemented via EIA-485 at 
17  614 K bps, 5V TTL, or via a 1 GHz fiber channel provided all PI/O specifications 
18  herein are met, including:
19  - Command and Response Message Content
20  - Command and Response Timing
21  - Error Checking
22  - Electrical Isolation

23  
24  Therefore, a “frame” is merely a field in the data stream, not related to the 
25  physical interface between the CPU and the PI/O, in contrast to the ATC 2070 
26  which requires serial communications via SP5 to its Field I/O.

27  
28  **8.2.1.7.1.1 Frame Types**
29  The frame type shall be determined by the value of the first byte of the message. The 
30  command frames type values 112 – 127 and associated response frame type values 
31  240 – 255 are allocated to the Manufacturer diagnostics. All other frame types not called 
32  out are reserved. The command-response Frame Type values and message times shall 
33  be as follows:

34  **Guidance:**
35  
36  The following Commands and Responses are intended to match the 
37  NEMA/AASHTO/ITE ATC 2070 commands and responses. The Minimum and 
38  Maximum Message times specified are representative of a protocol using SDLC at 
39  614,000 bps.
8.2.1.7.1.2 ITS Cabinet Frames

Messages 61 / 189, 62 / 190 and 65 / 193 are for ITS Cabinet Monitor Unit. See ITS Cabinet Monitor System Serial Bus #1 for Command and Response Frames. 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.

### Request Module Status

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

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 49)</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Reset Status Bits</td>
<td>P</td>
<td>E</td>
<td>K</td>
</tr>
</tbody>
</table>

#### Frame Types

<table>
<thead>
<tr>
<th>Module Command</th>
<th>I/O Module Response</th>
<th>Description</th>
<th>Minimum Message Time</th>
<th>Maximum Message Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-43</td>
<td>128-171</td>
<td>Reserved for NEMA TS2</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>44-48</td>
<td>172-176</td>
<td>Reserved</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>49</td>
<td>177</td>
<td>Request Module Status</td>
<td>250 µs</td>
<td>275 µs</td>
</tr>
<tr>
<td>50</td>
<td>178</td>
<td>MILLISECOND CTR. Mgmt.</td>
<td>222.5 µs</td>
<td>237.5 µs</td>
</tr>
<tr>
<td>51</td>
<td>179</td>
<td>Configure Inputs</td>
<td>344.5 µs</td>
<td>6.8750 ms</td>
</tr>
<tr>
<td>52</td>
<td>180</td>
<td>Poll Raw Input Data</td>
<td>317.5 µs</td>
<td>320 µs</td>
</tr>
<tr>
<td>53</td>
<td>181</td>
<td>Poll Filtered Input Data</td>
<td>317.5 µs</td>
<td>320 µs</td>
</tr>
<tr>
<td>54</td>
<td>182</td>
<td>Poll Input Transition Buffer</td>
<td>300 µs</td>
<td>10.25 ms</td>
</tr>
<tr>
<td>55</td>
<td>183</td>
<td>Command Outputs</td>
<td>405 µs</td>
<td>410 µs</td>
</tr>
<tr>
<td>56</td>
<td>184</td>
<td>Config. Input Tracking Functions</td>
<td>340 µs</td>
<td>10.25 ms</td>
</tr>
<tr>
<td>57</td>
<td>185</td>
<td>Config. Complex Output Functions</td>
<td>340 µs</td>
<td>6.8750 ms</td>
</tr>
<tr>
<td>58</td>
<td>186</td>
<td>Configure Watchdog</td>
<td>222.5 µs</td>
<td>222.5 µs</td>
</tr>
<tr>
<td>59</td>
<td>187</td>
<td>Controller Identification</td>
<td>222.5 µs</td>
<td>22.5 µs</td>
</tr>
<tr>
<td>60</td>
<td>188</td>
<td>I/O Module Identification</td>
<td>222.5 µs</td>
<td>222.5 µs</td>
</tr>
<tr>
<td>61-62-65</td>
<td>189-190-193</td>
<td>Reserved (note below)</td>
<td>317.5 µs</td>
<td>320 µs</td>
</tr>
<tr>
<td>63</td>
<td>191</td>
<td>Poll variable length raw input</td>
<td>405 µs</td>
<td>410 µs</td>
</tr>
<tr>
<td>64</td>
<td>192</td>
<td>Variable length command outputs</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>65-111</td>
<td>193-239</td>
<td>Reserved</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>112-127</td>
<td>240-255</td>
<td>Manufacturer Diagnostics</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>
### Request Module Status Response

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 177)</td>
<td>1</td>
<td>0110001100001</td>
<td>1</td>
</tr>
<tr>
<td>System Status</td>
<td>P</td>
<td>EKRTMLW</td>
<td>2</td>
</tr>
<tr>
<td>SCC Receive Error Count</td>
<td></td>
<td>Receive Error Count</td>
<td>3</td>
</tr>
<tr>
<td>SCC Transmit Error Count</td>
<td></td>
<td>Transmit Error Count</td>
<td>4</td>
</tr>
<tr>
<td>Timestamp MSB</td>
<td></td>
<td>Timestamp MSB</td>
<td>5</td>
</tr>
<tr>
<td>Timestamp NMSB</td>
<td></td>
<td>Timestamp NMSB</td>
<td>6</td>
</tr>
<tr>
<td>Timestamp NLSB</td>
<td></td>
<td>Timestamp NLSB</td>
<td>7</td>
</tr>
<tr>
<td>Timestamp LSB</td>
<td></td>
<td>Timestamp LSB</td>
<td>8</td>
</tr>
</tbody>
</table>

8.2.1.7.1.3 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

8.2.1.7.1.4 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.

#### 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:

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 50)</td>
<td>0</td>
<td>0110001010</td>
<td>1</td>
</tr>
<tr>
<td>New Timestamp MSB</td>
<td>X</td>
<td>x x x x x x x</td>
<td>2</td>
</tr>
<tr>
<td>New Timestamp NMSB</td>
<td>X</td>
<td>x x x x x x x</td>
<td>3</td>
</tr>
<tr>
<td>New Timestamp NLSB</td>
<td>X</td>
<td>x x x x x x x</td>
<td>4</td>
</tr>
<tr>
<td>New Timestamp LSB</td>
<td>X</td>
<td>x x x x x x x</td>
<td>5</td>
</tr>
</tbody>
</table>

ATC Standard
Version 5.2a2
Millisecond Counter Management Response

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>Isb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 178)</td>
<td>1</td>
<td>1</td>
<td>0 1 1 0 0 1 0</td>
</tr>
<tr>
<td>Status</td>
<td>0</td>
<td>0</td>
<td>0 0 0 0 0 0 S</td>
</tr>
</tbody>
</table>

2

Configure Inputs

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

Configure Inputs Command

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>Isb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 51)</td>
<td>0</td>
<td>0</td>
<td>1 1 0 0 1 1</td>
</tr>
<tr>
<td>Number of Items (n)</td>
<td>n</td>
<td>n</td>
<td>n n n n n n</td>
</tr>
<tr>
<td>Item # - Byte 1</td>
<td>E</td>
<td>Input Number</td>
<td>3(I-1)+3</td>
</tr>
<tr>
<td>Item # - Byte 2</td>
<td>Leading edge filter (e)</td>
<td>3(I-1)+4</td>
<td></td>
</tr>
<tr>
<td>Item # - Byte 3</td>
<td>Trailing edge filter (r)</td>
<td>3(I-1)+5</td>
<td></td>
</tr>
</tbody>
</table>

Configure Inputs Response

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>Isb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 179)</td>
<td>1</td>
<td>0</td>
<td>1 1 0 0 1 1</td>
</tr>
<tr>
<td>Status</td>
<td>0</td>
<td>0</td>
<td>0 0 0 0 0 S</td>
</tr>
</tbody>
</table>

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

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
Advanced Transportation Controller (ATC) Standard
PARALLEL AND SERIAL I/O DETAILS

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 52)</td>
<td>0 0 1 1 0 1 0 0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Poll Raw Input Data Response

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 180)</td>
<td>1 0 1 1 0 1 0 0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Inputs I0 (lsb) to I7 (msb)</td>
<td>x x x x x x x</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Inputs I8 to I119</td>
<td>x x x x x x x</td>
<td>3 to 16</td>
<td></td>
</tr>
<tr>
<td>Timestamp MSB</td>
<td>x x x x x x x</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Timestamp NMSB</td>
<td>x x x x x x x</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Timestamp NLSB</td>
<td>x x x x x x x</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Timestamp LSB</td>
<td>x x x x x x x</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 53)</td>
<td>0 0 1 1 0 1 0 1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Poll Filter Input Data Response

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 181)</td>
<td>1 0 1 1 0 1 0 1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Inputs I0 (lsb) to I7 (msb)</td>
<td>x x x x x x x</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Inputs I8 to I119</td>
<td>x x x x x x x</td>
<td>3 to 16</td>
<td></td>
</tr>
<tr>
<td>Timestamp MSB</td>
<td>x x x x x x x</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Timestamp NMSB</td>
<td>x x x x x x x</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Timestamp NLSB</td>
<td>x x x x x x x</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Timestamp LSB</td>
<td>x x x x x x x</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

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
8.2.1.7.1.5 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

8.2.1.7.1.6 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.
### 8.2.1.8 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:

<table>
<thead>
<tr>
<th>Description</th>
<th>mslsbi</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 55)</td>
<td>0 0 1 1 0 1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>O0 (lsb) to O7 (msb) Data</td>
<td>X x x X x x x X</td>
<td>2</td>
</tr>
<tr>
<td>O8 to O103 Data</td>
<td>X x x X x x x x</td>
<td>3 to 14</td>
</tr>
<tr>
<td>O0 (lsb) to O7 (msb) Control</td>
<td>X x x X x x x x</td>
<td>15</td>
</tr>
<tr>
<td>O8 to O103 Control</td>
<td>X x x X x x x x</td>
<td>16 to 27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 183)</td>
<td>1 0 1 1 0 1 1 1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>0 0 0 0 0 0 L E</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### 8.2.1.9 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 8 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:

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 56)</td>
<td>0 0 1 1 0 0 0 0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Number of Items (N)</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Item I - Byte 1</td>
<td>E</td>
<td>Output Number</td>
<td>2(I-1)+3</td>
</tr>
<tr>
<td>Item I - Byte 2</td>
<td>I</td>
<td>Input Number</td>
<td>2(I-1)+4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>mslsbi</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 184)</td>
<td>1 0 1 1 0 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>Status</td>
<td>0 0 0 0 0 0 0 V</td>
<td>2</td>
</tr>
<tr>
<td>Timestamp MSB</td>
<td>x x x X x x x x</td>
<td>3</td>
</tr>
<tr>
<td>Timestamp NMSB</td>
<td>x x x X x x x x</td>
<td>4</td>
</tr>
<tr>
<td>Timestamp NLSB</td>
<td>x x x X x x x x</td>
<td>5</td>
</tr>
</tbody>
</table>
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.

**Timestamp**

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

**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.

**Number of Item Field**

The “Number of Item” field is valid from 0 to 16 (most that is sent at one time is 8 enables and 8 disables). If processing a command resulting in more than 8 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 8 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.

**Configure Complex Output Functions**

The Configure Complex Output Functions frame shall be used to specify a complex output for one to eight of any of the outputs. If a Configure Complex Output Function command results in more than eight outputs being configured, the ‘V’ bit in the response message shall be set to a ‘1’, and the command shall not be implemented. Two output forms shall be provided, single pulse and continuous oscillation. These output forms shall be configurable to begin immediately or on a specified input trigger and, in the case of continuous oscillation, to continue until otherwise configured or to oscillate only while
gated active by a specified input. If the command gate bit is active, the command trigger bit shall be ignored and the specified input shall be used as a gate signal. The command and response frames are as follows:

Configure Complex Output Functions Command

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 57)</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Number of Items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item i - Byte 1</td>
<td>0</td>
<td>Output Number</td>
<td>7(i-1)+3</td>
</tr>
<tr>
<td>Item i - Byte 2</td>
<td>Primary Duration (MSB)</td>
<td>7(i-1)+4</td>
<td></td>
</tr>
<tr>
<td>Item i - Byte 3</td>
<td>Primary Duration (LSB)</td>
<td>7(i-1)+5</td>
<td></td>
</tr>
<tr>
<td>Item i - Byte 4</td>
<td>Secondary Duration (MSB)</td>
<td>7(i-1)+6</td>
<td></td>
</tr>
<tr>
<td>Item i - Byte 5</td>
<td>Secondary Duration (LSB)</td>
<td>7(i-1)+7</td>
<td></td>
</tr>
<tr>
<td>Item i - Byte 6</td>
<td>0</td>
<td>Input Number</td>
<td>7(i-1)+8</td>
</tr>
<tr>
<td>Item i - Byte 7</td>
<td>P</td>
<td>W</td>
<td>G</td>
</tr>
</tbody>
</table>

Configure Complex Output Functions Response

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 185)</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp (MSB)</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Timestamp (NMSB)</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Timestamp (NLSB)</td>
<td>x</td>
<td>X</td>
<td>x</td>
</tr>
<tr>
<td>Timestamp (LSB)</td>
<td>x</td>
<td>X</td>
<td>x</td>
</tr>
</tbody>
</table>

Configure Complex Outputs Bit Fields

The bit fields of the command frame are defined as follows:

- E '1' enable complex output function for this output
- E '0' disable complex output function for this output
- J '1' During the primary duration, the output shall be written as a logic '1'. During the secondary duration, the output shall be written as a logic '0'.
- J '0' During the primary duration, the output shall be written as a logic '0'. During the secondary duration, the output shall be written as a logic '1'

Output Number - 7-bit output number identifying outputs

Primary Duration - For single pulse operation, this shall determine the number of 'ticks' preceding the pulse. For continuous oscillation, this shall determine the length of the inactive (first) portion of the cycle.

Secondary Duration - For single pulse operation, this shall determine the number of 'ticks' the pulse is active. Subsequent to the secondary duration, the output shall return to the state set according to the most recently received Set Outputs command. For
continuous oscillation, this shall determine the length of the active (second) portion of the cycle. 0 = hold output state until otherwise configured.

F '1' - The trigger or gate shall be acquired subsequent to filtering the specified input. The raw input signal shall be used if filtering is not enabled for the specified input. '0' - The trigger or gate shall be derived from the raw input.

R '1' - For triggered output, the output shall be triggered by an ON-to-OFF transition of the specified input and shall be triggered immediately upon command receipt if the input is OFF. For gated output, the output shall be active while the input is OFF. '0' - For triggered output, the output shall be triggered by an OFF-to-ON transition of the specified input and shall be triggered immediately upon command receipt if the input is ON. For gated output, the output shall be active while the input is ON.

Input Number - 7-bit input number identifying inputs 0 Up.

P '1' - The output is configured for single-pulse operation. Once complete, the complex output function shall be disabled. '0' - The output is configured for continuous oscillation.

W '1' - It is triggered by the specified input. Triggered complex output shall commence within 2 ms of the associated trigger. '0' - Operation shall begin within 2 ms of the command receipt.

G '1' - Operation shall be gated active by the specified input. '0' - Gating is inactive.

L '1' - The LINESYNC based clock shall be used for the time ticks. '0' - The MC shall be used for the time ticks.

V '1' - Indicates maximum number of configurable outputs is exceeded. '0' - No error

Number of items - The number of entries in the frame. If 0, all outputs currently configured as complex outputs shall be disabled.

Sampling Rate
Controlling input signals shall be sampled at least once per ms.

Data Range
The “Number of Items” field is valid from 0 to 16. Zero means disable all Complex Output functions. Sixteen is the maximum because the most that is sent at one time is 8
enables and 8 disables. If processing a command results in more than 8 Complex Output functions being enabled, none of the command shall be implemented and the response message “V” bit shall be set to 1. If an invalid output or input number (the “G” or “W” bits being set to 1) is specified for a function, that function definition is not done by the FCU software. It shall also not be counted towards the maximum of 8 Complex Output functions allowed. The rest of the message shall be processed. When a Complex Output function is disabled, the output is set according to the most recently received Set Outputs command. When a complex output function for an output is superseded, that is, redefined as whether another Complex Output function, or as an Input Tracking function, nothing special is done with the output. The most recent value remains until the new function changes it. The “G” bit (gating) set to 1 takes precedence over the “W” bit (triggering). If gating is ON, triggering is turned OFF, regardless of the value of the “W” bit in the command message. If a Complex Output is configured with the “G” bit set to 1 (gating) and the “P” bit set to 0 (continuous oscillation), the output is set to OFF (0) whenever the specified input changes state so that the oscillation should cease (output inactive). For a single pulse operation (“G” bit set to 1), after the secondary duration completes the Complex Output function shall be disabled, and the output shall be set according to the most recently received Set Outputs command.

8.2.1.11 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:

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 58)</td>
<td>0 0 1 1 1 0 1 0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Timeout Value</td>
<td>x X x x</td>
<td>x X</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 186)</td>
<td>1 0 1 1 1 0 1 0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>0 0 0 0 0 0 0 Y</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**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.
8.2.1.12 Controller Identification

Frame Type 59 is reserved. Frame Type 59 and Response 187 are not supported by ATC standard controllers.

Guidance:
This is a legacy ATC 2070 message command / response for Field I/O modules with Datakey resident. Upon receipt of the command, a response frame containing the 128 bytes of the Datakey is returned. Whenever NRESET transitions to High (inactive) or immediately prior to any interrogation of the Datakey, the FCU shall test the presence of the Key. If absent, the FCU Status Bit “K” shall be set and no interrogation shall take place. If an error occurs during the interrogation, Bit “K” shall be set. If “K” bit set, only the first two bytes shall be returned. The Command Response frames are as follows:

Controller Identification Command

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 59)</td>
<td>0 0 1 1 1</td>
<td>0 1 1</td>
<td>1</td>
</tr>
</tbody>
</table>

Controller Identification Response

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 187)</td>
<td>1 0 1 1 0</td>
<td>1 1</td>
<td>1</td>
</tr>
<tr>
<td>Status</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 K</td>
<td>2</td>
</tr>
<tr>
<td>Datakey Data</td>
<td>x x x x x</td>
<td>x x x x x</td>
<td>3 to 130</td>
</tr>
</tbody>
</table>

8.2.1.13 Module Identification

The PI/O Identification command frame shall be used to request the PI/O Identification value Response of “1” for the Model 332 PI/O, “2” for the NEMA TS-2 Type 2 PI/O, “3” for the NEMA TS-2 Type 1 PI/O. The Identification value response for ITS Cabinet SIUs and CMU shall be frame address. The command and response frames are shown as follows:

I/O Module Identification Command

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type Number = 60)</td>
<td>0 0 1 1 1</td>
<td>1 0 0</td>
<td>1</td>
</tr>
</tbody>
</table>

I/O Module Identification Response

<table>
<thead>
<tr>
<th>Description</th>
<th>msb</th>
<th>lsb</th>
<th>Byte Number</th>
</tr>
</thead>
</table>
8.2.1.14 Mechanical Details

C11S

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 8-1: C1S and C11S Pin Configuration: Refer to ATC 2070 Standard

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

8.2.2.1 Description

This PI/O shall consist of an FCU Controller, 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.

8.2.2.2 Front Panel

The Front Panel shall be furnished with the following:

- Incoming VAC fuse protection
- Four NEMA Connectors, A, B, C & D
8.2.2.3 Functional Requirements Exceptions

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

- 118 bits of input and 102 bits of output shall 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.

8.2.2.4 Fault Monitor and Voltage Monitor

NEMA TS-2 Controller FAULT and VOLTAGE Monitor functions (outputs to the cabinet monitor) shall be provided.

Monitor Logic

Two 3-input OR gates shall be provided. The gate 1 output shall be connected to Connector A, Pin A (Fault Monitor) and gate 2 output shall be connected to Connector A, Pin C (Voltage Monitor). Any FALSE state input shall cause a gate output FALSE (+24V) state.

Watchdog

O78 shall normally change its state every 100 ms. A Watchdog Timer (WDT) circuit shall monitor the output. No state change for 2 ± 0.1 s shall cause the circuit output to generate a FALSE (+24 VDC) output (input to gates 1 and 2). Should the FCU begin changing state, the WDT output shall return to TRUE (0 VDC) state.

The 5 VDC Monitor

The +5 VDC shall be monitored. When 5 VDC supply falls out of regulation (± 0.25 V), this monitor circuit shall generate a FALSE output (input to gates 1 and 2) Normal operation shall return the output state to TRUE state.

Fault Monitor Logic

The FCU microprocessor output shall be assigned to FAULT Monitor (input to gate 1) and another output shall be assigned to VOLTAGE MONITOR (input to gate 2).

Monitor Control from Application Software

CPU Port 5 SET OUTPUT COMMAND Message

OUTPUTs O78 and O79 shall be assigned to FAULT (O78) and VOLTAGE (O79). The bit logic “1” shall be FCU output FALSE.
Monitor Output Power Up Conditions

CPU / FCU operation at POWERUP shall be as follows:

FCU Comm Loss Flag set. FAULT & VOLTAGE MONITOR outputs set FALSE (i.e. inactive). CPU REQUEST MODULE STATUS COMMAND Message with “E” bit set is sent to FCU to clear Comm Loss Flag and responds to CPU with “E” bit reset.

Before the Comm Loss timer expires, the SET OUTPUTS COMMAND data must be sent. In that data, the O78 and O79 logically set to “0” will cause the FCU microprocessor port pins assigned for FM and VM outputs to go to their TRUE (i.e. active) state. At this point, the signal outputs defined in the message will be permitted at 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 VM & FM shall retain the FALSE (inactive) output state.

Communications Loss

A CPU/FCU Communications Loss during operation shall cause all outputs to go FALSE (inactive) state and shall set the Comm Loss Flag. FM and VM outputs shall be in the FALSE (inactive) state.

8.2.2.5 Mechanical Details

NOTES:

1. A = NEMA “A” Connector, Type MS-3112-22-55P
2. B = NEMA “B” Connector, Type MS-3112-22-55S
3. C = NEMA “C” Connector, Type MS-3112-24-61S
4. D = NEMA “D” Connector, Type MS-3112-24-61P
5. Spacing Between A, B, C, D Connectors = 3.0 “ min, Center to Center.
8.2.3 Connection to NEMA TS-2 Type 1 Cabinets

Description
The TS-2 Type 1 PI/O provides a TS2-1 compatible interface, AC Power to the ATC, Fault Monitor Logic Output and Output Frame Byte 9 Bit 6 to the NEMA TS2 Cabinet Monitor Unit (CMU).

Front Panel
The Front Panel shall be furnished with the following:
- Incoming VAC fuse protection
- One NEMA Connector, A
- One NEMA Port 1 Connector

Functional Requirements Exceptions
This PI/O shall meet all requirements of 7.2.2, except:
- No C1 and C11 Connectors
- No 64 inputs / 64 outputs requirements
- There is no “Muzzle Jumper” and O39 is defined by TS2 instead of being used as the toggling watchdog output required for the 332 cabinet.

Parallel Connector
The parallel connector is a 10 Pin NEMA Connector 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
An FCU output 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.

Connector A Pin Assignment
Connectors A pin assignment: Refer to NEMA TS-2 Specification

8.3 Serial Input / Output

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

USB Removable Memory Device
ENET2 Ethernet for Local Cabinet (Expansion Rack, Upload to Laptop)

Refer to Engine Board section for a description of each serial port operation.

The ATC shall provide two internal 100BASE-TX hubs per the IEEE 802.3 specification for 100 Mbps Ethernet signaling with CSMA/CD over two pairs of Category 5 UTP or STP wire. 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(s)
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)
Typical Hub Port Use:

Hub 1:

Hub1, 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. Hub1 handles the speed conversion, as well as provides two RJ-45 Ethernet connectors, Port 2 and Port 3, and internal expansion connection on Port 4.

Port 2 provides a permanent connection to the network backbone for network communications, such as NTCIP. Port 2 does not act as a router or communications switch. Because the Engine Board cannot respond to every Ethernet packet on the trunk, the network topology should be designed to forward only packets destined for the ATC.

Port 3 is configured as an uplink for use with a “straight through” 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 NETWK1-4 signals on the Communication Interface Slot(s). A Communication Interface Module may connect directly to ENET1 via this port.

Hub 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. Hub2 handles the speed conversion, as well as provides two RJ-45 Ethernet connectors, Port 6 and Port 7.

Port 7 provides a permanent connection for internal 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 a “straight through” 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.

8.4 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”; insuring 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 need not be electrically isolated. However, a removal front panel or SP6 external connector such as C60P is not considered to be part of the ATC and isolation is required.
- SP4 (C50S), SP6 (C60P), and the LCD Heater supply must be isolated from the Engine Board; however, they do not have to isolated from each other.

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.

8.4.1 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).

### 8.4.2 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.

### 8.4.3 Serial I/O Isolation

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. 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:

---

**Figure 8-4: Isolation Boundaries**
8.5 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).
9 ENVIRONMENTAL AND TEST PROCEDURES

NOTE: This Electrical, Environmental, and Testing Requirements section was developed based on information excerpted from NEMA TS2-2003 Traffic Controller Assemblies with NTCIP Requirements, and is not intended to replicate those requirements.

9.1 General

This section establishes the limits of the environmental and operational conditions in which the First Article ATC Controller Unit (CU) will perform. This section defines the minimum test procedures which may be used to demonstrate conformance of a device type with the provisions of the standard.

Software shall be provided 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:
  1. DRAM Test
  2. SRAM Test
  3. FLASH Read/Write Test
  4. Datakey Tests
  5. USB Tests.

- Front Panel (when used) tests shall include:
  1. Display Tests
  2. Keyboard Tests.

- I/O tests shall include:
  1. I/O Loop Back Tests.

- Asynchronous/Synchronous Communication Port tests shall include:
  1. Loop Back Tests. Single Port and Port to Port.
  2. Aggregate throughput tests as defined in Paragraph 9.1.1
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 controller 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 this standard. However, nothing in this testing profile shall be construed as to relieve the requirement that the equipment provided must fully comply with these standards/specifications under all environmental conditions stated herein.

### 9.1.1 Communication Loading Test

The communication channel loads for load testing purposes, with no other activity present, shall be the following:

- SP1, SP2, & SP8 = Continuous full-duplex, asynchronous communications at 19.2Kbps
- SP4 & SP6 = Continuous full-duplex, asynchronous communications at 38.4Kbps
- SP3S = Continuous full-duplex, synchronous communications at 153.6Kbps
- SP5S = Continuous full-duplex synchronous communications at 614.4Kbps
- All Ethernet Ports: At 10% loading, with 3% average hits to processor per minute

No other applications or I/O activities are required to be operational during this test.
9.2 Inspection
A visual and physical inspection of the CU shall verify mechanical, dimensional and assembly conformance to all parts of this standard.

9.3 Testing Certification
A complete quality control and final test report shall be supplied with each item (see Section 11.1.3).

9.4 Definitions of Design Acceptance Testing (DAT) and Production Testing.
Design Acceptance Testing (DAT) is performed on the First Article ATC Controller Unit (CU), and is a part of the pre-production process.
Production Testing is performed on all units prior to their shipment to an agency.

9.5 Environmental and Operating Requirements
The requirements (voltage, temperature, etc.) of this section shall apply in any combination.

9.5.1 Voltage and Frequency

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

9.5.1.2 Operating Frequency
The operating frequency range shall be 60 Hz (±3.0 Hz), unless otherwise noted.

9.5.2 Transients, Power Service (DAT)
The CU under test shall maintain all defined functions when the independent test pulse levels specified in Section 9.5.2.1 on an individual unit, and Section 9.5.2.2 occur on the alternating-current power service in a controller unit.

9.5.2.1 High-Repetition Noise Transients (DAT)
The test pulses shall not exceed the following conditions:
1. Amplitude: 300 V, both positive and negative polarity.

2. Peak Power: 2500 W.

3. Repetition: 1 pulse approximately every other cycle moving uniformly over the full wave in order to sweep across 360° of the line cycle once every 3 seconds.


5. Pulse Width: 10 µs.

9.5.2.2 Low-Repetition High-Energy Transients (DAT)

The test pulses shall not exceed the following conditions:

1. Amplitude: 600 V, both positive and negative polarity.

2. Energy Source: Capacitor, oil filled, 10 ± 1 µF, internal surge impedance less than 1 Ω.

3. Repetition: 1 discharge every 10 seconds.


9.5.3 Nondestructive Transient Immunity (DAT)

The CU under test shall be capable of withstanding a high energy transient having the following characteristics repeatedly applied to the alternating current input terminals (no other power connected to terminals) without failure of the test specimen:

1. Amplitude: 2000 ± 100 V, both positive and negative polarity.

2. Energy Source: Capacitor, oil filled, 15 ± 1.5 µF, internal surge impedance less than 1 Ω.

3. Repetition: Applied to the CU under test once every 2 seconds for a maximum of three applications for each polarity.

4. After the foregoing, the CU under test shall perform all defined functions upon the application of nominal alternating current power.

5. Repetition: 1 pulse per second, for a minimum of 5 pulses per selected terminal.

6. Pulse rise time: 1 µs.
7. Pulse width: 10 µs.

9.5.4 Temperature and Humidity
The CU under test shall maintain all programmed functions when the temperature and humidity ambients are within the specified limits defined herein (9.5.4.1 and 9.5.4.2).

9.5.4.1 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%.

9.5.4.2 Humidity
The relative humidity shall not exceed 95% 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 9-1 for dynamic testing.

<table>
<thead>
<tr>
<th>Ambient Temperature/Dry Bulb (°C)</th>
<th>Relative Humidity (in%)</th>
<th>Ambient Temperature/Wet Bulb (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-37.0 to 1.1</td>
<td>10</td>
<td>-17.2 to 42.7</td>
</tr>
<tr>
<td>1.1 to 46.0</td>
<td>95</td>
<td>42.7</td>
</tr>
<tr>
<td>48.8</td>
<td>70</td>
<td>42.7</td>
</tr>
<tr>
<td>54.4</td>
<td>50</td>
<td>42.7</td>
</tr>
<tr>
<td>60.0</td>
<td>38</td>
<td>42.7</td>
</tr>
<tr>
<td>65.4</td>
<td>28</td>
<td>42.7</td>
</tr>
<tr>
<td>71.2</td>
<td>21</td>
<td>42.7</td>
</tr>
<tr>
<td>74.0</td>
<td>18</td>
<td>42.7</td>
</tr>
</tbody>
</table>

9.6 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 9-1.

3. Transient Generator(s): Transient generator(s) capable of supplying the transients outlined in Sections 9.5.2 through 9.5.3.

9.7 Test Procedures: Transients, Temperature, Voltage, and Humidity

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

1. Place the CU under test in the environmental chamber. Connect the CU under test AC input circuit to a variable voltage power transformer, voltmeter, and transient generator. 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 CU under test.

2. Connect test switches to the appropriate terminals to simulate the various features incorporated into the CU under test. Place these switches in the proper position for desired operation.

3. Verify the test hook-up. Adjust the variable-voltage power transformer to 120 VAC and apply power to the CU under test. Verify that the CU under test goes through its prescribed startup sequence and cycles properly in accordance with the operation determined by the positioning of test switches in item 2.

Upon the satisfactory completion and verification of the test hook-up, proceed with Test B (DAT) or Test C (DAT and Production Testing). Figure 9-1 describes the test profile used for Tests C through G to demonstrate the ability of the CU under test to function reliably under stated conditions of temperature, voltage, and humidity.

9.7.2 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).

1. Program the CU under test to dwell. Verify the input voltage is 120 VAC.

2. Set the transient generator to provide high-repetition noise transients as follows:
   a. Amplitude: 300 ± 15 V, both positive and negative polarity.
   b. Peak Power: 2500 W.
   c. Repetition Rate: One pulse every other cycle moving uniformly over the full wave in order to sweep once every 3 s across 360° of line cycle.
   d. Pulse Rise Time: 1 µs.
   e. Pulse Width: 10 µs.

3. Apply the transient generator output to the AC voltage input for at least 5 minutes. Repeat this test for at least two conditions of dwell for the CU under test. The CU under test must continue to dwell without malfunction.

4. Program the CU under test to cycle through normal operations. Turn on the transient generator (output in accordance with item 2) for 10 minutes, during which time the CU under test shall continue to cycle without malfunction.

5. Set a transient generator to provide high-repetition noise transients as follows:
   a. Amplitude: 300 ± 15 V, both positive and negative polarity.
   b. Source Impedance: Not less than 1000 Ω nominal impedance.
   c. Repetition: One pulse per second for a minimum of five pulses per selected terminal.
   d. Pulse Rise Time: 1 µs.
   e. Pulse Width: 10 µs.

Program the CU under test to dwell. Verify the input voltage is 120 VAC.

6. Apply the transient generator (output in accordance with item 5) between logic ground and the connecting cable termination of selected Field I/O input/output terminals of the CU under test.

A representative sampling of selected input/output terminations shall be tested. The CU under test shall continue to dwell without malfunction.

7. Program the CU under test to cycle. Turn on the transient generator (output in accordance with item 5) and apply its output to the selected Field I/O input/output terminations. The CU under test shall continue to cycle without malfunction.

8. Set a transient generator to provide low-repetition high-energy transients as follows:
   a. Amplitude: 600 ± 30 V, both positive and negative polarity.
   b. Energy Discharge Source: Capacitor, oil-filled, 10 µF.
c. Repetition Rate: One discharge each 10 s.

d. Pulse Position: Random across 360 degrees of line cycle.

9. Program the CU under test to dwell. Verify the input voltage is 120 VAC.

10. Discharge the oil-filled 10 µF capacitor ten times for each polarity across the AC voltage input. Repeat this test for at least two conditions of dwell. The CU under test shall continue to dwell without malfunction.

11. Program the CU under test to cycle through normal operations. Discharge the capacitor ten times for each polarity while the CU under test is cycling, during which time the CU under test shall continue to cycle without malfunction.

12. During the preceding transient tests (item 3 through 11), the CU under test must continue its programmed functions.

The CU under test shall not skip normal program intervals/steps or portions thereof when in normal operation; place false inputs or produce false outputs while in dwell; disrupt normal sequences in any manner; or change parameters. Details of requirements established by appropriate DAT program.

13. Nondestructive Transient Immunity:

a. Turn off the AC power input to the CU under test from the variable-voltage power source.

b. Apply the following high-energy transient to the AC voltage input terminals of the CU under test (no other power connected to terminals):

   (1) Amplitude: 2000 V, both positive and negative polarity.

   (2) Peak Power Discharge: Capacitor, oil-filled, 15 µF.

   (3) Maximum Repetition Rate: Applied to the Controller Assembly once every 2 s for a maximum of three applications for each polarity.

c. Upon completion of the foregoing, apply 120 VAC to the CU under test and verify that the CU under test goes through its prescribed startup sequence and cycles properly in accordance with the programmed functions. The first operation of the over-current protective device during this test shall not be considered a failure of the CU under test.
NOTES:
1. The rate of change in temperature shall not exceed 18°F per hour.
2. Humidity controls shall be set in conformance with the humidities given in Table 9-1 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.

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

1. Definition of Test Conditions
   a. Environmental Chamber Door: Closed.
   b. Temperature: -37°C.
   c. Low Voltage: 100 VAC.
   d. Humidity Control: Off.
2. Test Procedure: While at room temperature, adjust the input voltage to 100 VAC and verify that the CU under test is still operable.
   a. With the CU under test cycling through normal operations, lower the test chamber to -37˚C at a rate not exceeding 18˚C per hour. Allow the CU under test to cycle for a minimum of 5 hours at -37˚C with the humidity controls in the off position. Then operate the test switches as necessary to determine that all functions are operable.
   b. Power shall then be removed from the CU under test for a minimum period of 5 hours. Upon restoration of power, the CU under test shall go through its prescribed startup sequence and then resume cycling.
   c. With the CU under test at -37˚C and the input voltage at 100 VAC, the following items shall be evaluated against the respective standards:
      1) Given in Section 9.10 Power Interruption Tests
      2) Transient voltage tests (DAT)

On satisfactory completion of this test, proceed with Test D.

9.7.4 Test D: Low-Temperature High-Voltage Tests (DAT and Production Testing)

1. Definition of Test Conditions
   a. Environmental Chamber Door: closed.
   b. Low Temperature: -37˚C.
   c. High Voltage: 135 VAC.
   d. Humidity Controls: Off.

2. Test Procedure: While at -37˚C and with humidity controls off, adjust the input voltage to 135 VAC and allow the CU under test to cycle for 1 hour. Then operate the test switches as necessary to determine that all functions are operable.

3. With the CU under test at -37˚C and the input voltage at 135 VAC (humidity controls off), the following items shall be evaluated against the respective standards:
   i. Given in Section 9.10 Power Interruption Tests
   ii. Transient voltage tests (DAT)

On satisfactory completion of this test, proceed to Test E.

9.7.5 Test E: High-Temperature High-Voltage Tests (DAT and Production Testing)

1. Definition of Test Conditions
   a. Environmental Chamber Door: Closed.
2. Test Procedure—With the CU under test cycling, raise the test chamber to +74°C at a rate not to exceed 18°C per hour. Verify the input voltage is 135 VAC.

3. Set the humidity controls to not exceed 95% relative humidity over the temperature range of +1.1°C to +46°C. When the temperature reaches +46°C, readjust the humidity control to maintain constant absolute humidity; +42.7°C wet bulb which results in the relative humidities shown in Table 9-1. Verify that the CU under test continues to cycle satisfactorily during the period of temperature increase and at established levels of relative humidity.

   a. Allow the CU under test to cycle for a minimum of 15 hours at +74°C and 18% relative humidity. Then operate the test switches as necessary to determine that all functions are operable.

   b. With the CU under test at +74°C and 18% relative humidity and the input voltage at 135 VAC, the following items shall be evaluated against the respective standards:

      i. Given in Section 9.10 Power Interruption Tests

      ii. Transient voltage tests (DAT)

On satisfactory completion of this test, proceed to Test F.

9.7.6 Test F: High-Temperature Low-Voltage Tests (DAT and Production Testing)

1. Definition of Test Conditions

   a. Environmental Chamber Door: Closed.

   b. High Temperature: +74°C.

   c. Low Voltage: 100 VAC.

   d. Humidity Controls: 18% relative humidity and +42.7°C wet bulb.

2. Test Procedure: Adjust the input voltage to 100 VAC and proceed to operate the test switches to determine that all functions are operable. With the CU under test at +74°C and 18% relative humidity, +42.7°C wet bulb, and the input voltage at 100 VAC, the following items shall be evaluated against the respective standards:

   a. Given in Section 9.10 Power Interruption Tests

   b. Transient voltage tests (DAT)

On satisfactory completion of this test, proceed to Test G.
9.7.7  **Test G: Test Termination (All tests)**

1. Program the CU under test to cycle according to DAT.

2. Adjust the input voltage to 120 VAC.

3. Set the controls on the environmental chamber to return to room temperature, +20° C ± 5° C, with the humidity controls in the off position. The rate of temperature change shall not exceed 18° C per hour.

4. Verify the CU under test continues to cycle through normal operations properly.

5. Allow the CU under test to stabilize at room temperature for 1 hour. Proceed with test program to determine that all functions are operable.

9.7.8  **Test H: Appraisal of Equipment under Test**

1. A failure shall be defined as any occurrence which results in other than normal operation of the equipment. (See item 2 for details.) If a failure occurs, the CU under test shall be repaired or components replaced, and the test during which failure occurred shall be restarted from its beginning.

2. The CU under test is considered to have failed if any of the following occur:
   a. If the CU under test skips normal program intervals/steps or portions thereof when in normal operation, places false inputs, presents false outputs, exhibits disruption of normal sequence of operations, or produces changes in parameters beyond specified tolerances, or
   b. If the CU under test fails to satisfy the requirements of Section 9.7 Tests A to G, inclusive.

3. An analysis of the failure shall be performed and corrective action taken before the CU under test is retested in accordance with this standard. The analysis must outline what action was taken to preclude additional failures during the tests.

4. When the number of failures exceeds two, it shall be considered that the CU under test fails to meet these standards. The CU under test may be completely retested after analysis of the failure and necessary repairs have been made in accordance with item 3.

5. Upon completion of the tests, the CU under test shall be visually inspected. If material changes are observed which will adversely affect the life of the CU under test, the cause and conditions shall be corrected before making further tests.
6. Upon satisfactory completion of all of the tests described in Sections 9.7.1 through 0, the CU under test shall be tested in accordance with Section 0.

9.8 Vibration Test (DAT)

Guidance:
Units such as the ATC 2070 and certain Controller Assemblies that are controlled by specific mechanical design specifications are not subject to the tests in Sections 9.8 through 9.9 (Authorized Engineering Information).

Shock and vibration tests shall be performed prior to environmental tests.

9.8.1 Purpose of Test
This test is intended to duplicate vibrations encountered by the CU under test (individual major components) when installed at its field location.

The CU under test shall be fastened securely to the vibration test table prior to the start of the test.

9.8.2 Test Equipment Requirements

1. Vibration table with adequate table surface area to permit placement of the CU under test.

2. Vibration test shall consist of:
   a. Vibration in each of three mutually perpendicular planes.
   b. Adjustment of frequency of vibration over the range from 5 Hz to 30 Hz.
   c. Adjustment of test table excursion (double amplitude displacement) to maintain a ‘g’ value, measured at the test table, of 0.5g; as determined by the following formula:

   \[ g = 0.0511d^2 \]

   Where:
   \( d \) = excursion in inches
   \( f \) = frequency in Hz

9.8.3 Resonance [Mechanical Resonant Frequency] Search (DAT)

1. With the CU under test securely fastened to the test table, set the test table for a double amplitude displacement of 0.015 inch.
2. Cycle the test table over a search range from 5 Hz to 30 Hz and back within a period of 12.5 minutes.

3. Conduct the resonant frequency search in each of the three mutually perpendicular planes.

4. Note and record the resonant frequency determined from each plane.
   a. In the event of more than one resonant frequency in a given plane, record the most severe resonance.
   b. If resonant frequencies appear equally severe, record each resonant frequency.
   c. If no resonant frequency occurs for a given plane within the prescribed range, 30 Hz shall be recorded.

9.8.4 **Endurance Test (DAT)**

1. Vibrate the CU under test in each plane at its resonant frequency for a period of 1 hour at amplitude resulting in 0.5 G acceleration.

2. When more than one resonant frequency has been recorded in accordance with Section 9.8.4, item number 4, the test period of 1 hour shall be divided equally between the resonant frequencies.

3. The total time of the endurance test shall be limited to 3 hours, 1 hour in each of three mutually perpendicular planes.

9.8.5 **Disposition of Test Unit**

1. The CU under test shall be examined to determine that no physical damage has resulted from the vibration tests.

2. The CU under test shall be checked to determine that it is functionally operable in all modes of its prescribed operation.

3. The CU under test may be removed from the test table. Upon satisfactory completion of the vibration test, proceed with the shock (impact) test described in Section 9.8.1.

9.9 **Shock (Impact) Test (DAT)**

9.9.1 **Purpose of Test**
Shock and vibration tests shall be performed prior to environmental tests.

The purpose of this test is to determine that the CU under test is capable of withstanding the shock (impact) to which it may reasonably be subjected during handling and transportation in the process of installation, repair, and replacement. It is to be noted that the CU under test is not, at this time, in its shipping carton.

The CU under test shall be firmly fastened to the specimen table. In each of its three planes the CU under test shall be dropped from a calibrated height to result in a shock force of 10 G.

**9.9.2 Test Equipment Requirements**

1. Shock (impact) test fixture equivalent to that suggested by the simplified sketch shown in Figure 9-2.

2. The test table shall have a surface area sufficient to accommodate the CU under test.

3. The test table shall be calibrated and the items tested as indicated. This shock test defines the test shock to be 10 ± 1 G.
   a. Calibration of the test equipment for these shock tests shall be measured by three accelerometers having fixed shock settings of 9 G, 10 G, and 11 G. They shall be Inertia Switch Incorporated ST-355, or the equivalent. These devices shall be rigidly attached to the test table.
   b. Calibration of the fixture for each item to be tested shall be as follows:
      1) Place a dummy load weighing within 10% of the CU under test on the table.
      2) Reset the three accelerometers and drop the test table from a measured height.
      3) Observe that the accelerometers indicate the following:
         a) The 9 G accelerometer shall be activated.
         b) The 10 G unit may or may not be actuated.
         c) The 11 G unit shall not be actuated.
   c. Repeat calibration test (a) and (b) adjusting the height of the drop until, on ten successive drops, the following occurs:
      1) The 9 G unit is actuated ten times.
      2) The 10 G unit is actuated between four to eight times.
      3) The 11 G unit is not actuated on any of the ten drops.
9.9.3 Test Procedure (DAT)

1. The calibration height of the drop for the particular item under test as determined in Section 9.9.2 shall be used in this procedure.

2. Secure the CU under test to the test table surface so that the CU under test rests on one of its three mutually perpendicular planes.
3. Raise the test table to the calibrated height.

4. Release the test table from the calibrated height, allowing a free fall into the box of energy absorbing material below.

5. Repeat the drop test for each of the remaining two mutually perpendicular planes, using the same calibrated height for each drop test of the same CU under test.

6. The observations of the accelerometer for the three tests of the test item shall be:
   a. The 9 G unit is actuated for all three tests. (Repeat the calibration if the unit is not actuated.)
   b. The 10 G unit may or may not be actuated in these tests.
   c. The 11 G unit is not actuated on any drop. (If the unit is actuated, repeat the calibration only if the CU under test has suffered damage.)

7. Production Testing drop test procedure: while the unit is running, tilt and lift the controller from the front four inches high and drop.

9.9.4 Disposition of Test Unit

   1. Check the CU under test for any physical damage resulting from the drop tests.
   2. Check the CU under test to determine that it is functionally operable in all modes of its prescribed operation.
   3. Satisfactory completion of all environmental tests, including the shock (impact) is required.

9.10 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 4.1.3 in cases where a restart occurs). Application test software may be required to facilitate verifying synchronization.
9.10.1 Short Power Interruption

While the CU 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 insure that the CU under test continues normal operation as though no power interruption has occurred. Repeat this test three times.

Verify that the clock in the CU under test has not drifted as a result of the power interruptions.

Additional power interruption testing is to be performed at 550 ms, 750 ms, and 1 second outages to verify proper restart operation.

9.10.2 Voltage Variation

All circuits of the CU under test shall be subjected to slowly varying line voltage during which the CU 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 CU 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 clock in the CU under test has not drifted as a result of the power variations.

9.10.3 Rapid Power Interruption

The CU 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 2 minutes. Power interruption shall be performed through electromechanical contacts of an appropriate size for the load. During this testing, the CU under test 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 clock in the CU under test has not drifted as a result of the power interruptions.
10 PERFORMANCE AND MATERIAL REQUIREMENTS

10.1 General

10.1.1 Furnished Equipment

All equipment furnished in compliance 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.

10.1.2 Edges

All sharp edges and corners shall be rounded.

10.1.3 Hardware

All washers, nuts, bolts, hinges and hinge pins shall be stainless steel unless otherwise specified.

10.1.4 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 mega-\(\Omega\), 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.

10.1.5 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:
10.1.5.1 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.

10.1.5.2 Operational Envelopes
No component shall be operated above 80% of its maximum rated voltage, current or power ratings. Digital components shall not be operated above 3% over their nominal voltage, current or power ratings.

10.1.5.3 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.

10.1.5.4 Component Packaging
Encapsulation of 3 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.

10.1.6 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. Capacitors which 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.

10.1.7 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\% for carbon film and deposited carbon types and 10\% 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 2 W or higher. They shall be insulated from the PCB.

10.1.8 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.

10.1.9 Transformers and Inductors

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 their leads color coded with an approved EIA color code or identified in a manner to facilitate proper installation.

10.1.10 Fuses

All fuses shall be 3AG Slow Blow type and resident in a holder. Fuse size rating shall be labeled on the holder. Fuses shall be easily accessible and removable without use of tools.

Fuses shall not be dislodged during shipping and handling.

10.1.11 Switches

10.1.11.1 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).
10.1.11.2 Logic Switches
The switch contacts shall be rated for a minimum of 1 A resistive load at 120 VAC and shall be silver over brass (or equal). The switch shall be rated for a minimum of 40,000 operations.

10.1.11.3 Control Switches
The switch contacts shall be rated for a minimum of 5 A resistive load at 120 VAC or 28 VDC and shall be gold over brass (or equal). The switch shall be rated for a minimum of 40,000 operations.

10.1.11.4 Power Switches
The switch contacts shall be rated for a minimum of 5 A resistive load at 120 VAC or 28 VDC and shall be gold over brass (or equal). The switch shall be rated for a minimum of 10 A at 125 VAC.

10.1.12 Wiring, Cabling, and Harnesses

10.1.12.1 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.

10.1.12.2 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.

10.1.12.3 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.
10.1.13 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.

10.1.13.1 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. Each LED indicator shall be white or clear when off.

10.1.13.2 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 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.

10.1.14 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.

10.1.14.1 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.
10.1.14.2 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.

10.1.14.3 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.

10.1.14.4 Metallic Circular Connectors [NEMA]

Metallic Circular Connectors shall comply and interface with MS 116 Shell type.

10.1.15 PCB Design

No components, traces, brackets or obstructions shall be within 0.125 inch of a PCB 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 permanent.

10.1.16 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)
11 QUALITY CONTROL

Material in this section are considered a supplement to that provided in Section 9. In the case of apparent inconsistencies, material in Section 9 of this standard shall prevail.

11.1 Components

All components shall be lot sampled to assure a consistent high conformance standard to the design specification of the equipment.

11.1.1 Subassembly, Unit Or Module

Complete electrical, environmental and timing compliance 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.

11.1.2 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.

11.1.3 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.
1. Environmental testing report(s) and final acceptance.
2. Acceptance testing of all supplied components.
3. Physical and functional testing of all modules and items.
4. Verification of a minimum burn-in of all equipment.
12 GLOSSARY

12.1 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

12.2 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
### 12.3 Acronyms and Definitions

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AASHTO</td>
<td>American Association of State Highway and Transportation Officials</td>
</tr>
<tr>
<td>AC</td>
<td>Alternating Current</td>
</tr>
<tr>
<td>AC-</td>
<td>120 VAC, 60 Hz neutral (grounded return to the power source)</td>
</tr>
<tr>
<td>AC+</td>
<td>120 VAC, 60 Hz line source (ungrounded)</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standard Institute</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>Assembly</td>
<td>A complete machine, structure, or unit of a machine that was manufactured by fitting together parts and/or modules</td>
</tr>
<tr>
<td>ASTM</td>
<td>American Society for Testing and Materials</td>
</tr>
<tr>
<td>ATC</td>
<td>Advanced Transportation Controller</td>
</tr>
<tr>
<td>AWG</td>
<td>American Wire Gage</td>
</tr>
<tr>
<td>BSP</td>
<td>Board Support Package</td>
</tr>
<tr>
<td>Cabinet</td>
<td>An outdoor enclosure generally housing the controller unit and associated equipment</td>
</tr>
<tr>
<td>Caltrans</td>
<td>California Department of Transportation</td>
</tr>
<tr>
<td>CD</td>
<td>Carrier Detect</td>
</tr>
<tr>
<td>Component</td>
<td>Any electrical or electronic device</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear to send (data)</td>
</tr>
<tr>
<td>CU</td>
<td>Controller Unit, that portion of the controller assembly devoted to the operational control of the logic decisions programmed into the assembly</td>
</tr>
<tr>
<td>DAT</td>
<td>Design Acceptance Testing</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>DCD</td>
<td>Data Carrier Detect (receive line signal detector)</td>
</tr>
<tr>
<td>DRAM</td>
<td>Dynamic Random Access Memory</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically Erasable Programmable Read-Only Memory</td>
</tr>
<tr>
<td>EG</td>
<td>Equipment Ground</td>
</tr>
<tr>
<td>EIA</td>
<td>Electronic Industries Association</td>
</tr>
<tr>
<td>EL</td>
<td>Electro-luminescent</td>
</tr>
<tr>
<td>EMI</td>
<td>Electromagnetic Interference</td>
</tr>
<tr>
<td></td>
<td>Term</td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>ENET</td>
</tr>
<tr>
<td>2</td>
<td>EPROM</td>
</tr>
<tr>
<td>3</td>
<td>Equal</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ETL</td>
</tr>
<tr>
<td>6</td>
<td>FCU</td>
</tr>
<tr>
<td>7</td>
<td>Firmware</td>
</tr>
<tr>
<td>8</td>
<td>FLASH</td>
</tr>
<tr>
<td>9</td>
<td>FPA</td>
</tr>
<tr>
<td>10</td>
<td>FSK</td>
</tr>
<tr>
<td>11</td>
<td>HDLC</td>
</tr>
<tr>
<td>12</td>
<td>I/O</td>
</tr>
<tr>
<td>13</td>
<td>IEEE</td>
</tr>
<tr>
<td>14</td>
<td>IP</td>
</tr>
<tr>
<td>15</td>
<td>ISO</td>
</tr>
<tr>
<td>16</td>
<td>ITE</td>
</tr>
<tr>
<td>17</td>
<td>ITS</td>
</tr>
<tr>
<td>18</td>
<td>Jumper</td>
</tr>
<tr>
<td>19</td>
<td>Keyed</td>
</tr>
<tr>
<td>20</td>
<td>LCD</td>
</tr>
<tr>
<td>21</td>
<td>LED</td>
</tr>
<tr>
<td>22</td>
<td>LOGIC</td>
</tr>
<tr>
<td>23</td>
<td>logic-level</td>
</tr>
<tr>
<td>24</td>
<td>Isb</td>
</tr>
<tr>
<td>25</td>
<td>LSB</td>
</tr>
<tr>
<td>26</td>
<td>MIPS</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Module</td>
<td>A functional unit that plugs into an assembly</td>
</tr>
<tr>
<td>msb</td>
<td>Most Significant Bit</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Byte</td>
</tr>
<tr>
<td>NA</td>
<td>Presently Not Assigned. Cannot be used by the contractor for other purposes</td>
</tr>
<tr>
<td>NEMA</td>
<td>National Electrical Manufacturer's Association</td>
</tr>
<tr>
<td>NETA</td>
<td>National Electrical Testing Association, Inc.</td>
</tr>
<tr>
<td>NLSB</td>
<td>Next Least Significant Byte</td>
</tr>
<tr>
<td>NMSB</td>
<td>Next Most Significant Byte</td>
</tr>
<tr>
<td>NTCIP</td>
<td>National Transportation Communication for ITS Protocols</td>
</tr>
<tr>
<td>OST</td>
<td>Operating System Time</td>
</tr>
<tr>
<td>O/S</td>
<td>Operating System</td>
</tr>
<tr>
<td>Open System</td>
<td>standardized hardware interfaces in a module</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Data Assistant (electronic)</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RMS</td>
<td>Root mean square</td>
</tr>
<tr>
<td>ROM</td>
<td>Read only memory</td>
</tr>
<tr>
<td>RTC</td>
<td>Real Time Clock</td>
</tr>
<tr>
<td>RTS</td>
<td>Request to send (data)</td>
</tr>
<tr>
<td>RX</td>
<td>Abbreviation for “Receive” when used to describe communication signals. Typically a prefix for other character(s).</td>
</tr>
<tr>
<td>RXC</td>
<td>Receive Clock</td>
</tr>
<tr>
<td>RXD</td>
<td>Receive Data</td>
</tr>
<tr>
<td>SDLC</td>
<td>Synchronous Data Link Control</td>
</tr>
<tr>
<td>SP</td>
<td>Serial Port</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static Random Access Memory</td>
</tr>
<tr>
<td>TEES</td>
<td>Transportation Electrical Equipment Specifications</td>
</tr>
<tr>
<td>TMC</td>
<td>Transportation Management Center</td>
</tr>
<tr>
<td>TOD</td>
<td>Time Of Day Clock</td>
</tr>
<tr>
<td>TTL</td>
<td>Transistor-Transistor Logic</td>
</tr>
<tr>
<td></td>
<td>Abbreviation</td>
</tr>
<tr>
<td>---</td>
<td>--------------</td>
</tr>
<tr>
<td>1</td>
<td>TX</td>
</tr>
<tr>
<td>2</td>
<td>TXC</td>
</tr>
<tr>
<td>3</td>
<td>TXD</td>
</tr>
<tr>
<td>4</td>
<td>UL</td>
</tr>
<tr>
<td>5</td>
<td>USB</td>
</tr>
<tr>
<td>6</td>
<td>VAC</td>
</tr>
<tr>
<td>7</td>
<td>VDC</td>
</tr>
<tr>
<td>8</td>
<td>WDT</td>
</tr>
</tbody>
</table>
Annex A

Minimum Linux Kernel Configuration
(Normative)

This Annex specifies the minimum Linux kernel configuration features that shall be included in all kernel builds and the minimum required set of utilities. This standard shall only specify non-platform-specific configuration options (i.e. a specific manufacturer’s implementation may require additional supplemental configuration options to be specified to support their particular hardware). Currently, only a Linux compatible kernel in the range of 2.6.x, (where x >= 9) is permitted under this standard.

Kernel Configuration

# General setup
# CONFIG_SYSVIPC=y
CONFIG_SYSCTL=y
CONFIG_KCORE_ELF=y
CONFIG_HOTPLUG=y
CONFIG_KOBJECT_UEVENT=y
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y

# Loadable module support
CONFIG_MODULES=y
CONFIG_MODULE_UNLOAD=y
CONFIG_OBSOLETE_MODPARM=y
CONFIG_KMOD=y

# Platform options
CONFIG_EMBEDDED=y
CONFIG_PREEMPT=y
CONFIG_BINFMT_ELF=y
CONFIG_KERNEL_ELF=y

# Block devices
CONFIG_BLK_DEV_LOOP=y
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_INITRD=y

# SCSI device support
1 #
2 CONFIG_SCSI=y
3 CONFIG_SCSI_PROC_FS=y
4
5 #
6 # SCSI support type (disk, tape, CD-ROM)
7 #
8 CONFIG_BLK_DEV_SD=y
9
10 #
11 # Networking support
12 #
13 CONFIG_INET=y
14
15 #
16 # Networking options
17 #
18 CONFIG_PACKET=y
19 CONFIG_PACKET_MMAP=y
20 CONFIG_NETFILTER=y
21 CONFIG_FILTER=y
22 CONFIG_UNIX=y
23 CONFIG_IP_PNP=y
24 CONFIG_IP_PNP_DHCP=y
25 CONFIG_IP_PNO_BOOTP=y
26 CONFIG_IP_MULTICAST=y
27 CONFIG_SYN_COOKIES=y
28
29 #
30 # Ethernet (10 or 100Mbit)
31 #
32 CONFIG_NETETHERNET=y
33
34 #
35 # Character devices
36 #
37 CONFIG_GEN_RTC=y
38 CONFIG_GEN_RTC_X=y
39
40 #
41 # Non-8250 serial port support
42 #
43 CONFIG_UNIX98_PTYS=y
44 CONFIG_LEGACY_PTYS=y
45 CONFIG_LEGACYPTY_COUNT=256
46
47 #
48 # USB support
49 #
50 CONFIG_USB=y
51
52 #
# Miscellaneous USB options

# 1. CONFIG_USB_DEVICEFS=y
2. CONFIG_USB_STORAGE=y
3. CONFIG_USB_STORAGE_FREECOM=y
4. CONFIG_USB_STORAGE_ISD200=y
5. CONFIG_USB_STORAGE_DPCM=y
6. CONFIG_USB_STORAGE_SDDR09=y
7. CONFIG_USB_STORAGE_SDDR55=y

# File systems

# 12. CONFIG_EXT2_FS=y
13. CONFIG_FAT_FS=y
14. CONFIG_MSDOS_FS=y
15. CONFIG_VFAT_FS=y
16. CONFIG_FAT_DEFAULT_CODEPAGE=437
17. CONFIG_FAT_DEFAULT_IOCHARSET="iso8859-1"
18. CONFIG_DNOTIFY=y

# Pseudo File Systems

# 23. CONFIG_PROC_FS=y
24. CONFIG_SYSFS=y
25. CONFIG_TMPFS=y
26. CONFIG_RAMFS=y

# Network File Systems

# 33. CONFIG_NFS_FS=y
34. CONFIG_NFS_V3=y
35. CONFIG_ROOT_NFS=y
36. CONFIG_LOCKD=y
37. CONFIG_LOCKD_V4=y
38. CONFIG_SUNRPC=y

# Native Language Support

# 41. CONFIG_NLS_DEFAULT="iso8859-1"

# Library routines

# 46. CONFIG_CRC32=m
47. CONFIG_ZLIB_INFLATE=y
CONFIG_ZLIB_DEFLATE=y

# Cryptographic options
#
CONFIG_CRYPTO=y
CONFIG_CRYPTO_HMAC=y
Annex B

Required Device Driver Interfaces
(Normative)

This Annex 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

Overview

This section defines a generalized driver interface for generating a pulse on the CPU_RESET port pin defined on the ATC Controller Engine Board. Although an underlying general purpose interface to the ATC Engine Board PIO pins exists, only the simplified interface of this section is exposed to the User Applications for controlling the CPU_RESET pin.

The BSP shall provide an external function, void atc_cpu_reset (void), that generates a 125 ± 20 ms active-low pulse on the CPU_RESET signal. This function first sets the CPU_RESET signal low, and resets a timer to count 125 ms. When the timer expires, the CPU_RESET signal is raised HIGH again. This function is non-blocking and non-exclusive in nature and may be called repetitively to generate a longer active-low pulse if necessary. If the CPU_RESET pin is already low when called, then the pulse width timer is simply reset and begins timing 125 ms.

Examples

extern void atc_cpu_reset (void)

/* single 125 ms pulse on CPU_RESET */
atc_cpu_reset();
...

/* generate a 200 ms pulse on CPU_RESET */
atc_cpu_reset ();
pause ( 75 ms); /* blocking delay of 75 ms */
atc_cpu_reset ();
...

B.2 ATC Engine Board PIO Driver Interface

Overview

This section defines a generalized driver interface for port pins defined on the ATC Controller Engine Board.

To access a port pin, the following procedures are required:
1. Open the port to obtain its file descriptor.
2. Read / Write.
3. Close the port.

The port pins shall have special device files located in the /dev directory.

<table>
<thead>
<tr>
<th>Device</th>
<th>Node</th>
<th>Input/Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWERDOWN pin</td>
<td>/dev/powerdown</td>
<td>Input</td>
</tr>
<tr>
<td>DKEY_PRESENT pin</td>
<td>/dev/datakeypresent</td>
<td>Input</td>
</tr>
<tr>
<td>CPU_ACTIVE pin</td>
<td>/dev/cpuactive</td>
<td>Output</td>
</tr>
<tr>
<td>CPU_RESET pin</td>
<td>/dev/cpureset</td>
<td>Output</td>
</tr>
</tbody>
</table>

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_NONBLOCKING);
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);
B.3  ATC Serial Peripheral Interface (SPI) Bus Driver Interface

Overview

This section defines a generalized driver interface for the SPI bus defined on the ATC Controller Engine Board. This bus is used to access the Datakey, Host Board serial EEPROM, one manufacturer defined device and one reserved device.

To access a SPI bus device, the following procedures are required:

1. Open the bus port to obtain its file descriptor.
2. Read data from or write data to the bus port.
3. Close the port.

B.3.1  SPI Bus Functions (SPIx)

The following functions are used this driver:

- open()
- close()
- read()
- write()
- ioctl()

Each SPI port shall have a special device file node located in the /dev directory. Device nodes shall be defined as follows:

<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
<th>Port Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI_SEL_1</td>
<td>Datakey</td>
<td>/dev/datakey</td>
</tr>
<tr>
<td>SPI_SEL_2</td>
<td>Host EEPROM</td>
<td>/dev/eeprom</td>
</tr>
<tr>
<td>SPI_SEL_3</td>
<td>Reserved</td>
<td>/dev/reserved</td>
</tr>
<tr>
<td>SPI_SEL_4</td>
<td>Manufacturer</td>
<td>/dev/manufacturer</td>
</tr>
</tbody>
</table>

specific device

open()

This function allows an application to request exclusive ownership of the SPI bus. Upon opening, the record pointer for the device is set to the beginning of the device so that the next read or write starts at the first byte. The port is then accessed via read(), write(), ioctl() and close() functions.

close()

This function allows an application to release ownership of the SPI bus 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.

read()
This function allows an application to read data from the open SPI bus pointed to by `fd`. This function should only be called after a successful open of the respective port. The function reads up to `count` sequential bytes from the SPI device starting from the device’s current record pointer and advances the record pointer as each byte is read. If the number of bytes requested is not available within the configured time limit, the read operation times out if time out operation is configured. The function places the requested data in the memory location pointed to by `*buf`.

`write()`

This function allows an application to write data to the open SPI bus pointed to by `fd`. This function should only be called after a successful open of the respective device. The function writes up to `count` sequential bytes to the SPI device starting at the device’s current record pointer and advances the record pointer as each byte is written. If the number of bytes requested cannot be sent within the configured time limit, the write operation times out. The function takes the data from the memory location pointed to by `*buf`.

`ioctl()`

This function allows an application to configure and control the SPI bus 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 the `command` argument. The `command` argument determines the interpretation of any additional arguments. The supported IOCTL services are defined below.

**Prototype**

```c
int ioctl(int fd, int command, parameters);
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fd</code></td>
<td>This is the file descriptor of the device to access.</td>
</tr>
<tr>
<td><code>command</code></td>
<td>This specifies the desired operation to be performed.</td>
</tr>
<tr>
<td><code>parameter</code></td>
<td>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.</td>
</tr>
</tbody>
</table>

**Valid SPI IOCTL Commands**

**ATC_SPI_READ**

This command copies `len` bytes of data from the SPI device starting at byte `#offset` from the beginning of the device. At the conclusion of the read operation, the SPI device record pointer will be set to byte number `offset + len`.
Advanced Transportation Controller (ATC) Standard

ANNEX B

command ATC_SPI_READ
parameter1 Pointer to SPI device first origination byte: offset
parameter2 Number of bytes to read: len

**ATC_SPI_WRITE**

This command copies len bytes of data from the user space to the SPI device starting at byte #offset within the device. At the conclusion of the write operation, the SPI device record pointer will be set to byte number offset + len.

<table>
<thead>
<tr>
<th>ioctl( ) Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fd</td>
<td>This is the file descriptor of the device to access.</td>
</tr>
<tr>
<td>command</td>
<td>ATC_SPI_WRITE</td>
</tr>
<tr>
<td>parameter1</td>
<td>Pointer to SPI device first destination byte: offset</td>
</tr>
<tr>
<td>parameter2</td>
<td>Number of bytes to read: len</td>
</tr>
</tbody>
</table>

**B.3.2 Datakey**

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. All additional memory following the 28th 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;
    uint16 id; // Controller ID
    uint16 drop; // Communications drop number
    uint32 ipaddress; // IP address
    uint32 subnet; // Subnet mask
    uint32 gateway; // Default gateway
} ;

Structure member “type” shall contain the Key Type value as defined in the following table:

<table>
<thead>
<tr>
<th>Key Type</th>
<th>Model No</th>
<th>Sector Size</th>
<th>Memory Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>SFK2Mb</td>
<td>2Mb</td>
<td>64 Kbytes</td>
</tr>
</tbody>
</table>

The data format in the Datakey header for the Latitude and Longitude fields shall comply with IEEE/ANSI 754-1985 STD. All the other fields shall follow a Big Endian Format as implemented by Freescale 68000-family CPUs.
B.3.3 Host Board EEPROM Content and Organization

The following describes the content and organization of the Host EEPROM. If a Host EEPROM does not exist within the ATC, then the BSP should assume the default configuration which represents an ATC 2070L. The data format in the Host EEPROM for the Latitude and Longitude fields shall comply with IEEE/ANSI 754-1985 STD. All the 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: v5.2 – 1)
- Host EEPROM Size in bytes (uint16 0..65535; Use Defaults = 0)
- # 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)
  - 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)
  - Switch/Router MAC Address (six bytes); use 000000 000000 if Hub/Phy/Other Direct Port
  - 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 ATC2070)
  - Bit 10: SP8s (although always present in an ATC, it is optional in an ATC 2070)
- # Ports used for I/O (uint8 0..255). For each I/O Port:
o 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)

4 o Port Mode as applicable (uint8 0..255; standard values are Other – 0, Async – 1, Sync – 2, SDLC – 3, HDLC – 4, Reserved – 5..255)

6 o 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 an ATC 2070)
  - Bit 10: SP8s (although always present in an ATC, it is optional in an ATC 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 an ATC 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 an ATC 2070)
  - Bit 10: SP8s (although always present in an ATC, it is optional in an ATC 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
Advanced Transportation Controller (ATC) Standard
ANNEX B

1. Bit 5: SP3s
2. Bit 6: Reserved
3. Bit 7: SP5s
4. Bit 8: Reserved
5. Bit 9: Reserved
6. Bit 10: SP8s (although always present in an ATC, it is optional in an ATC 2070)

- Expansion Bus type (uint8 0..255; Standard enumerated values are: None – 1, Other – 2, VME – 3)
- CRC #1 (16-bit)
- Latitude (4 bytes)
- Longitude (4 bytes)
- Controller ID (uint16 0..65535)
- Communication Drop # (uint16 0..65535)
- Reserved for Agency (35 bytes ... per TEES)
- CRC #2 (8-bit)
- User Data (to end)

The default configuration is as follows:

- Host EEPROM Version = 1
- 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
ANNEX B

Advanced Transportation Controller (ATC) Standard

B.3.4 Reserved Device

This port is reserved for future expansion of this standard. No driver requirements are currently defined for this port.

B.3.5 Manufacturer Specific Device

Manufacturer specific devices that use this port shall use the SPI Driver functions identified within section B.3.1; however, no content, structure, or functionality of the connected device is specified within this standard.

B.3.6 Constants Defined by this Standard for SPI Drivers

These constants shall reside in a file named `atc_spi.h`. IOCTL commands are defined according to Linux conventions. See `include/asm/ioctl.h`.

```c
#include <asm/ioctl.h>
#define ATC_SPI_WRITE 0
#define ATC_SPI_READ 1
#define ATC_SPI_MAX 4
```

B.4 Standard Linux Drivers

B.4.1 USB
Standard Linux drivers required to implement a USB host interface shall be provided. The USB
Mass Storage device shall use the FAT file system as configured in Annex A. The BSP shall support hot-plug configuration.

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 Flash File System
Standard Linux drivers required to implement a file system on the Flash EPROM(s) shall be provided.

B.4.4 Static RAM
Standard Linux drivers required to implement a ramfs file system 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 manipulating the flow control signals.

B.5 ATC SPxs Synchronous Driver Interface

Overview
This section defines a generalized synchronous driver interface for the serial ports of the ATC Controller 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:

<table>
<thead>
<tr>
<th>Device</th>
<th>Port Node</th>
<th>/proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP1</td>
<td>/dev/sp1s</td>
<td>/proc/driver/sp1s</td>
</tr>
<tr>
<td>SP2</td>
<td>/dev/sp2s</td>
<td>/proc/driver/sp2s</td>
</tr>
<tr>
<td>SP3</td>
<td>/dev/sp3s</td>
<td>/proc/driver/sp3s</td>
</tr>
<tr>
<td>SP5</td>
<td>/dev/sp5s</td>
<td>/proc/driver/sp5s</td>
</tr>
<tr>
<td>SP8</td>
<td>/dev/sp8s</td>
<td>/proc/driver/sp8s</td>
</tr>
</tbody>
</table>

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:
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

```c
int open(const char* pathname, int flags);
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pathname</td>
<td>This is the name of the device to open.</td>
</tr>
<tr>
<td>flags</td>
<td>O_RDWR, O_NONBLOCK or O_NDELAY, O_SYNC, O_ASYNC</td>
</tr>
</tbody>
</table>

Return Value | Description
---------------|-----------------
-1            | An error occurred. Consult errno.
else          | A valid file descriptor.

Errors

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENODEV</td>
<td>Incorrect device specified or device cannot be found.</td>
</tr>
<tr>
<td>EBUSY</td>
<td>The device is already open or the device is busy</td>
</tr>
<tr>
<td>EINTR</td>
<td>The open system call was interrupted by a signal.</td>
</tr>
</tbody>
</table>

Example

```c
#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 "\n");
}
else
    .....
```
The returned file descriptor (\texttt{fd}) is the ownership “handle” passed by an application to the close, read, write and ioctl functions.

\textbf{close( )}

This function allows an application to release ownership of the serial port pointed to by \texttt{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.

\textbf{Prototype}

\begin{verbatim}
int close(int fd);
\end{verbatim}

\begin{table}[h]
\begin{tabular}{|l|l|}
\hline
\textbf{Argument} & \textbf{Description} \\
\hline
\texttt{fd} & This is the file descriptor of the device to be closed. \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\begin{tabular}{|l|l|}
\hline
\textbf{Return Value} & \textbf{Description} \\
\hline
-1 & An error occurred. Consult \texttt{errno}. \\
0 & The operation succeeded. \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\begin{tabular}{|l|l|}
\hline
\textbf{Errors} & \textbf{Description} \\
\hline
EBADF & Incorrect file descriptor specified. \\
ENODEV & Incorrect device specified or device cannot be found. \\
EBUSY & The device is busy \\
EINVAL & The close system call was interrupted by a signal. \\
\hline
\end{tabular}
\end{table}

\textbf{Example}

\begin{verbatim}
#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);
}
\end{verbatim}
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 receive FIFO. If the number of bytes requested is not available within the configured time limit, the read operation times out if time out operation is configured. The function places the requested data in the memory location pointed to by *buf.

Prototype

int read(int fd, void *buf, size_t count);

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fd</td>
<td>This is the file descriptor of the device to access.</td>
</tr>
<tr>
<td>buf</td>
<td>The data read will be put here.</td>
</tr>
<tr>
<td>count</td>
<td>This is the desired number of bytes to read.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>An error occurred. Consult errno.</td>
</tr>
<tr>
<td>0 to count</td>
<td>The operation succeeded. If the return value is less than count, then the request timed out.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBADF</td>
<td>Incorrect file descriptor specified.</td>
</tr>
<tr>
<td>EFAULT</td>
<td>There was a problem copying data into the user specified buffer.</td>
</tr>
<tr>
<td>EAGAIN</td>
<td>Non blocking has been specified and no data was immediately available for reading.</td>
</tr>
<tr>
<td>EINTR</td>
<td>The call was interrupted by a signal before any data was read.</td>
</tr>
</tbody>
</table>

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 the transmit FIFO. If the number of bytes requested cannot be sent within the configured time limit, the write operation times out. The function takes the data from the memory location pointed to by `*buf`.

**Prototype**

```c
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**

```c
#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

```c
int poll(struct pollfd fds[], nfds_t nfds, int timeout);
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fds</td>
<td>This defines a table of file descriptor structures</td>
</tr>
<tr>
<td>Nfds</td>
<td>Number of structures in the table</td>
</tr>
<tr>
<td>timeout</td>
<td>Time out in milliseconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0</td>
<td>Number of structures which have non-zero event fields.</td>
</tr>
<tr>
<td>0</td>
<td>Call timed out</td>
</tr>
<tr>
<td>-1</td>
<td>Error</td>
</tr>
</tbody>
</table>

Errors

- 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

```c
#include <stropts.h>
#include <poll.h>
...
struct pollfd fds[1];
int timeout_msecs = 10;
int ret;
int i;
```
/* Open SCC device. */
fd[0].fd = open("/dev/sp5s", ...);
fd[0].events = POLLOUT | POLLIN | POLLERR;

ret = poll(fd, 1, timeout_msecs);

if (ret > 0) {
    /* An event on one of the fds has occurred. */
    ... if (fd[i].revents & POLLOUT) {
        /* Data may be written */
        ...
    }
    if (fd[i].revents & POLLIN) {
        /* Data is available for reading */
        ...
    }
    if (fd[i].revents & POLLERR) {
        /* There is some error with the SCC */
        ...
    }
}
The `fcntl()` method shall be used to enable or disable driver blocking or asynchronous notification operations.

### Prototype

```c
int fcntl(int fd, int cmd);
int fcntl(int fd, int cmd, flags);
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fd</td>
<td>File descriptor</td>
</tr>
<tr>
<td>cmd</td>
<td>Command</td>
</tr>
<tr>
<td>flags</td>
<td>Command options</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_GETOWN</td>
<td>Value of process owner</td>
</tr>
<tr>
<td>F_GETFL</td>
<td>Value of file status flags</td>
</tr>
<tr>
<td>F_GETSIG</td>
<td>Value of signal sent when read or write becomes possible, or zero for traditional SIGIO behavior.</td>
</tr>
<tr>
<td>-1</td>
<td>Error</td>
</tr>
</tbody>
</table>

### Errors

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBADF</td>
<td>fd is not an open file descriptor</td>
</tr>
<tr>
<td>EINVAL</td>
<td>Invalid argument</td>
</tr>
</tbody>
</table>

### Commands

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F_GETOWN</td>
<td>Get ID of process receiving signals from this file descriptor</td>
</tr>
<tr>
<td>F_SETOWN</td>
<td>Set calling process as owner</td>
</tr>
<tr>
<td>F_GETFL</td>
<td>Get the file status flags</td>
</tr>
<tr>
<td>F_SETFL</td>
<td>Set the file status flags</td>
</tr>
<tr>
<td>F_GETSIG</td>
<td>Get the signal sent when input or output becomes possible.</td>
</tr>
<tr>
<td>F_SETSIG</td>
<td>Sets the signal sent when input or output becomes possible.</td>
</tr>
</tbody>
</table>

### Flags

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>O_ASYNC</td>
<td>Set asynchronous notification mode</td>
</tr>
<tr>
<td>O_NONBLOCK</td>
<td>Non-blocking mode</td>
</tr>
</tbody>
</table>

### Example

```c
#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

```c
int ioctl(int fd, int command, parameters);
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fd</code></td>
<td>This is the file descriptor of the device to access.</td>
</tr>
<tr>
<td><code>command</code></td>
<td>This specifies the desired operation to be performed.</td>
</tr>
<tr>
<td><code>parameter</code></td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>An error occurred. Consult errno.</td>
</tr>
<tr>
<td>0</td>
<td>The operation succeeded.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBADF</td>
<td>Invalid file descriptor specified.</td>
</tr>
<tr>
<td>EFAULT</td>
<td>There was a problem accessing data from the user specified buffer.</td>
</tr>
<tr>
<td>ENOTTY</td>
<td>An invalid command parameter was specified</td>
</tr>
<tr>
<td>EINVAL</td>
<td>An invalid command parameter was specified</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>ioctl( ) Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fd</code></td>
<td>This is the file descriptor of the device to access.</td>
</tr>
<tr>
<td><code>command</code></td>
<td>ATC_SPXS_WRITE_CONFIG</td>
</tr>
<tr>
<td><code>parameter</code></td>
<td>* struct atc_spxs_config</td>
</tr>
</tbody>
</table>
Example

```c
#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.

<table>
<thead>
<tr>
<th>ioctl() Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fd</td>
<td>This is the file descriptor of the device to access.</td>
</tr>
<tr>
<td>command</td>
<td>ATC_SPXS_READ_CONFIG</td>
</tr>
<tr>
<td>parameter</td>
<td>Pointer to user space destination</td>
</tr>
</tbody>
</table>

Example

```c
#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);
...
```
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 protocol; // Set the port protocol (ATC_SDLC, ATC_SYNC, ATC_HDLC)
    uint8 baud; // Set the port baud rate (ATC_B1200, ATC_B2400, ATC_B4800, ATC_B9600, ATC_B19200, ATC_B38400, ATC_B57600, ATC_B76800, ATC_B115200, 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)
} ;

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>
#endif _ATC_SPXS_H
#define _ATC_SPXS_H
#define ATC_SPXS_MAX 5

// 'ioctl' commands
#define ATC_SPXS_WRITE_CONFIG 0
#define ATC_SPXS_READ_CONFIG 1

// Available communications protocols
#define ATC_SDLC 0 // Default
#define ATC_SYNC 1
#define ATC_HDLC 2

// Available baud rates
#define ATC_B1200 1200
#define ATC_B2400 2400
#define ATC_B4800 4800
#define ATC_B9600 9600
#define ATC_B19200 19200
#define ATC_B38400 38400
#define ATC_B57600 57600
#define ATC_B76800 76800
#define ATC_B115200 115200
#define ATC_B153600 153600 // NEMA default
#define ATC_B614400 614400 // ITS Cabinet default

// 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

#endif /* _ATC_SPXS_H */