

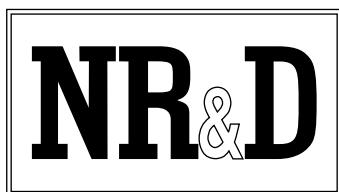
# AIM4

## Installation and Programming Manual

---

This manual describes the AIM4 standard, and PLUS, Rack mount ASCII Module, its uses and set up. It also describes the use of the AIM4SW emulation and programming software.

Effective: 26 February 1996



Niobrara Research and Development Corporation  
P.O. Box 3418 Joplin, MO 64803

Telephone: (800) 235-6723 or (417) 624-8918  
Facsimile: (417) 624-8920  
BBS: (417) 624-2028  
Internet: <http://niobrara.com>

---

SY/MAX and SY/NET are registered trademarks of Square D Company.  
MS/DOS is a registered trademark of Microsoft Corporation.  
Allen-Bradley, A-B, and Data Highway are trademarks of Allen-Bradley Company.  
Subject to change without notice.

© Niobrara Research & Development Corporation 1994, 1995, 1996. All Rights Reserved.

---

# Contents

<b>1 Introduction</b>	9
AIM4 <i>PLUS</i> Additional Functions	10
<b>2 Installation</b>	13
<b>3 Quick Start</b>	17
TERM Setup	18
PC Setup	18
INPUT Operations	19
Rack Addressing	20
OUTPUT Operations	20
Periodic Poll	20
ASCII Demultiplexing	21
<b>4 Configuring the Port Parameters</b>	23
<b>5 Parameter Guide</b>	27
Port Parameters	27
Baud Rate	27
Data Bits	27
Parity	28
Stop bits	28
Capitalize	28
Accept Characters	28
Terminate Characters	28
Periodic Polling	28
Terminate Count	28
Terminate Timeout	29
Trigger/Inhibit register	29
Path Parameters	29
Pattern	29
Mask	30
Start Register	31

Register Count .....	31
Editing .....	31
Continue .....	32
Global Parameters .....	32
Print/Prog baud .....	32
Rack Base .....	32
<b>6 Output Operations .....</b>	<b>33</b>
<b>7 Periodic Polling .....</b>	<b>35</b>
Dash 2 PLUS Output Programming .....	36
Output Trigger/Inhibit Register Allocation .....	36
Register Output Substitution Sequences .....	37
Register Substitution Example .....	38
Redirected Port Output .....	38
Additional Output Comments .....	39
<b>8 AIM4-2M Operation .....</b>	<b>41</b>
Full Duplex .....	41
Half Duplex .....	42
Multidrop .....	42
Point-to-Point .....	42
<b>9 SY/MAX Operations .....</b>	<b>43</b>
AIM4-2 Additional Setup Registers .....	48
Accept and Terminate sets .....	48
Patterns, Masks, and Queries .....	49
Poll Interval .....	49
Start Registers and the Rack address start .....	49
Data Registers .....	49
Max Count Message Termination .....	49
Maximum Silence Message Termination .....	50
Options Register Values .....	50
<b>10 Configuration Software AIM4SW .....</b>	<b>51</b>
AIM4SW .....	51
Data Entry Keys .....	52
Hex Editor .....	52
Online and Offline Editing Screen .....	54
Offline Functions .....	55
"Read from disk to memory" .....	55
"Write from memory to disk" .....	55
"Edit configuration in memory" .....	55
"Fetch memory from AIM4" .....	56
"Send memory to the AIM4" .....	56
"Print configuration in memory" .....	56
Emulator .....	56

Utilities .....	57
View registers .....	57
CTS Monitor .....	58
Terminal Emulator .....	59
SETUP .....	60
SY/MAX SETUP .....	60
Personal Computer COM: port .....	60
SY/LINK Connection .....	61
Terminal Emulator SETUP .....	62
Command Line Parameters .....	63
 <b>11 AIM4SW Errors .....</b>	 <b>65</b>
 <b>12 Troubleshooting .....</b>	 <b>69</b>
 <b>13 Examples .....</b>	 <b>73</b>
Example 1 .....	73
Port 1 Configuration .....	73
Port 2 Configuration .....	75
Example 2 .....	77
Example 3 .....	79
Setup .....	80
Example 4 .....	82
 <b>14 Throughput and Latency .....</b>	 <b>85</b>
 <b>15 Connector Pinouts .....</b>	 <b>87</b>
RS-232 data ports (DE9P with screw lock posts) .....	87
RS-232 data port with SD034B adapter attached (DB25P) .....	87
RS-422 data port (DE9S with slide lock posts) .....	88
RS-422 Print/Prog port (DE9S with slide lock posts) .....	88
 <b>16 Recommended Cabling .....</b>	 <b>89</b>
AIM4 to SY/MAX RS-422 port .....	89
RS-232 DCE (modem) to AIM4 RS-232 data .....	89
RS-232 DTE (terminal) to AIM4 RS-232 data port .....	90
AIM4 data port to separate RS-232 DTE input and output devices .....	90
Male RS-232 DTE (personal computer) to AIM4 data port .....	91
Personal Computer to AIM4 Print/Prog port .....	91
 <b>Appendix A Serial Communication Overview .....</b>	 <b>93</b>
Hardware Overview .....	93
RS-232 .....	93

RS-422 .....	98
RS-485 (four wire) .....	99
RS-485 (two wire) .....	99
20mA Current Loop .....	100
Hardware Handshaking .....	101
Software Overview .....	102
Binary Representation of Data .....	102
Start Bit .....	102
Data Bits .....	102
Parity Bit .....	102
Stop Bit .....	102
Message Determination .....	103
Hexadecimal numbers .....	103
ASCII characters .....	104
Software Handshaking .....	104
X-ON .....	104
X-OFF .....	104

## **Appendix B NR&D/Online BBS and Internet ..... 107**

### **Figures**

Figure 1-1 AIM4 Memory Map .....	11
Figure 1-2 AIM4 Front Panel .....	12
Figure 2-1 AIM4SW Startup Screen .....	14
Figure 2-2 Online Screen .....	15
Figure 3-1 Quick Start Setup .....	17
Figure 3-2 QSTART Setup .....	19
Figure 3-3 Utility View Register Screen .....	19
Figure 4-1 Trigger Bit Capture Ladder Rungs .....	24
Figure 4-2 Trigger Bit Capture Ladder Rungs (better solution) .....	25
Figure 4-3 ASCII Input Flow Chart .....	26
Figure 6-1 ASCII Demultiplexing .....	33
Figure 7-1 Periodic Polling Screen .....	36
Figure 10-1 Hex Editor screen .....	53
Figure 10-2 AIM4SW Startup Screen .....	53
Figure 10-3 AIM4SW oFfline Menu .....	54
Figure 10-4 oFfline Read from Disk to Memory .....	55
Figure 10-5 AIM4 Emulator .....	57
Figure 10-6 View Registers .....	58
Figure 10-7 CTS Monitor Screen .....	59
Figure 10-8 Terminal Emulator .....	59
Figure 10-9 SY/MAX Setup Screen .....	60
Figure 10-10 SY/LINK Setup Screen .....	61
Figure 10-11 Terminal Emulator Setup Screen .....	62
Figure 12-1 Accept Character Table .....	71
Figure 12-2 Terminator Character Table .....	71
Figure 13-1 Example 1 - Port 1 - Setup .....	74
Figure 13-2 Example 1 - Port 1 - Accept Table .....	74
Figure 13-3 Example 1 - Port 1 - Terminate Table .....	75

Figure 13-4	Example 1 - Port 2 - Configuration .....	76
Figure 13-5	Example 1 - Port 2 - Accept Table .....	76
Figure 13-6	Example 1 - Port 2 - Terminate Table .....	77
Figure 13-7	Example 2 - Configuration .....	78
Figure 13-8	Example 2 - Accept Table .....	78
Figure 13-9	Example 2 - Terminate Table .....	79
Figure 13-10	Example 3 - Configuration .....	80
Figure 13-11	Example 3 - Accept Table .....	81
Figure 13-12	Example 3 - Terminate Table .....	81
Figure 13-13	Example 3 - Periodic Poll Configuration .....	82
Figure 13-14	Example 4 Periodic Poll .....	83
Figure A-1	DTE to Modem connection .....	93
Figure A-2	Null Modem connection .....	97
Figure A-3	RS-422 Setup .....	98
Figure A-4	RS-485 Four Wire Setup .....	99
Figure A-5	RS-485 Two wire Multidrop Setup .....	100
Figure A-6	20mA Current Loop (Full Duplex) .....	101

## Tables

Table 7-1	Output Substitution Sequences .....	37
Table 7-2	Options Register 7435 .....	39
Table 8-1	Options Register 7435 .....	41
Table 9-1	Max Count Registers .....	50
Table 9-2	Max Silence Registers .....	50
Table 9-3	Options Register 7435 .....	50
Table A-1	25 pin RS-232 port .....	95
Table A-2	Type A 9 pin RS-232 port .....	96
Table A-3	Type B 9 pin RS-232 port .....	96
Table A-4	DB25 Null Modem .....	97
Table A-5	DE9 Null Modem .....	97
Table A-6	SY/MAX DE9S RS-422 Pinout .....	98
Table A-7	Decimal, Hex, Binary .....	103
Table A-8	ASCII Table .....	105





# Introduction

The Niobrara AIM4 is a SY/MAX<sup>®</sup> compatible register rack module that serves as a coprocessor for ASCII data inputs to a programmable controller system through the register bus and as an output multiplexer to drive multiple ASCII peripherals from a single SY/MAX communication port. The AIM4 accepts ASCII coded serial data inputs on up to four data ports and translates the ASCII character stream to PLC register values more conveniently processed by the limited ASCII capabilities of ladder programmed controllers. Preprocessing of data by the AIM4 can include ignoring certain characters, recognizing delimiters, pattern matching, and masking or modifying ASCII strings before conversion to register values. AIM4 processing behavior is determined by a set of operational parameters configured through an MS/DOS<sup>®</sup> compatible computer program. Processed data is transferred to the attached PLC through rack addressed dual-port memory. From the processor point of view, the AIM4 is like any input module. A SY/MAX RS-422 port is needed only when it is desired to drive output devices.

The AIM4 is available in the following configurations:

- AIM4-1S Rack Input only RS-232 version.
- AIM4-1D Rack Input only RS-422 version.
- AIM4-2S *PLUS* Bi-directional Rack RS-232 version.
- AIM4-2D *PLUS* Bi-directional Rack RS-422 version.
- AIM4-2M *PLUS* Bi-directional Rack RS-485 (2 wire compatible) version.

The RS-422 model can also be used in RS-485 applications either directly for four-wire RS-485 or with the NR&D SM285 adapter for two-wire RS-485. The port type can be identified by the sex of the connector. RS-232 data ports use male connectors (DE9P) with jackscrew sockets. RS-422 and RS-485 data ports are equipped with female connectors (DE9S) with slide lock posts. The fifth Print/Prog port which is provided for communication with a personal computer during parameter configuration or for receiving data to be output through the data ports is an RS-422 port on both RS-232 and RS-422 data port models. This port can be connected to a personal computer with Niobrara's SC406 smart cable or to a SY/MAX PLC with a Niobrara DC1 or Square D CC100 cable. The Print/Prog port will also accept SY/MAX read, write, and print messages allowing a PLC or other SY/MAX compatible device to read and write the module's rack addressed data registers or to modify AIM4 configuration parameters dynamically.

ASCII input devices which can be interfaced through the AIM4 include keyboards, barcode scanners, scales, vision systems, badge readers, intelligent test instruments, and any device with an ASCII coded serial data output. If the output function is not needed, the AIM4 does not require a SY/MAX serial

connection. SY/MAX compatible devices may write data directly to the rack by sending write messages to the Print/Prog port. In this way, one SY/MAX PLC can write data to another through the AIM4 without converting it to ASCII or using a PLC communication port.

Output devices which can be driven include displays, printers, host computers, speech synthesizers, and the control inputs of intelligent peripherals such as barcode scanners and vision systems. Multiple AIM4's can be daisy chained to drive more than four output devices. Two AIM4's can drive seven outputs. Five AIM4's can drive sixteen outputs. If processed data will be read through the Print/Prog port by SY/MAX read messages, the AIM4 can be plugged into a power only register slot such as slot 17 or 18 of an RRK-300 register rack.

Because input and output data paths are independent, unrelated input and output devices may be connected to a single port as long as baud rate and parity can be set the same. This means that the bi-directional data ports can be used not only with bi-directional devices such as terminals, computers, and operator interfaces but also to interface two unrelated devices such as driving a printer while accepting input from a barcode scanner. A Y-cable is available which splits a single AIM4 RS-232 data port into two 9 pin male jacks for attaching separate devices. Order Niobrara part number SP9Y.

Input and output devices are connected to the data ports of the AIM4 and the RS-422 Print/Prog port is connected to an MS/DOS personal computer for configuration of the AIM4 or to a SY/MAX PLC, NIM, or D-LOG for print output. The ASCII input processor of the AIM4 continuously monitors all attached input devices, waiting to recognize a valid message as defined by configuration parameters including acceptable characters and pattern matching formulae. When received, the message may be converted to a register-compatible form (binary integer, floating point, binary coded decimal, or packed or null-padded ASCII characters) or may simply generate a constant register value to signal the PLC of a recognized pattern. Either way, the formatted register value is then presented to the PLC through the register bus. A signaling register bit toggles each time the AIM4 processes a new input event. The PLC can use this signal to trigger events in the ladder program in response to the new data.

ASCII output through the AIM4 is by one of two methods; either via PRINT rungs in the ladder program or via periodic output polling from the AIM4. Using print rungs, the first field of the PRINT must be an ASCII digit 1 through 4 which identifies the AIM4 port to emit the printed message. The port selection digit is stripped off by the AIM4 and does not appear in the output message. To print through daisy chained AIM4's, two or more ASCII digits can be used. Refer to Output Operations on page 33 for daisy chain configuration details.

ASCII output using the periodic polling feature of the AIM4 requires no processor intervention. The AIM4 will output between 1 and 16 messages at a user defined interval, which is useful in prompting ASCII devices to transmit data to the AIM4 for processing and subsequent use by the PLC. Output messages and polling intervals are configured independently for each port. Refer to the Periodic Polling section on page 35 for configuration details.

## **AIM4 PLUS Additional Functions**

The AIM4-2S, AIM4-2D, and AIM4-2M models of the AIM4, also denoted by the *PLUS* label on the front panel, have the capability to generate output (i.e. print) strings under control of rack-addressed registers and therefore without requiring a SY/MAX connection to the processor. Output messages are controlled by programming special character sequences in the polling query strings. Output messages can be triggered by the polling timer or, if that timer is set to zero, by toggling bits in a rack-addressable trigger register. If the polling timer is non-zero, this register becomes an inhibit register and can be used to turn polling on or off and to individually inhibit specific queries.

AIM4 *PLUS* firmware later than 15Jun95 also includes the ability to send triggered queries out of other ports. For instance, if a printer is connected to port 1 and the periodic poll on port 2 is not being used, port 2's polling queries may be configured to be sent out port 1 under PLC control. If all four ports are configured, up to 64 queries may be sent from one port.

The *PLUS* models also feature message termination by programmable character count and /or intercharacter timeout. These new termination capabilities are useful when the peripheral device being interfaced does not supply a terminator character.

A memory map of the internal registers is shown in Figure 1-1. A front panel layout is shown in Figure 1-2.

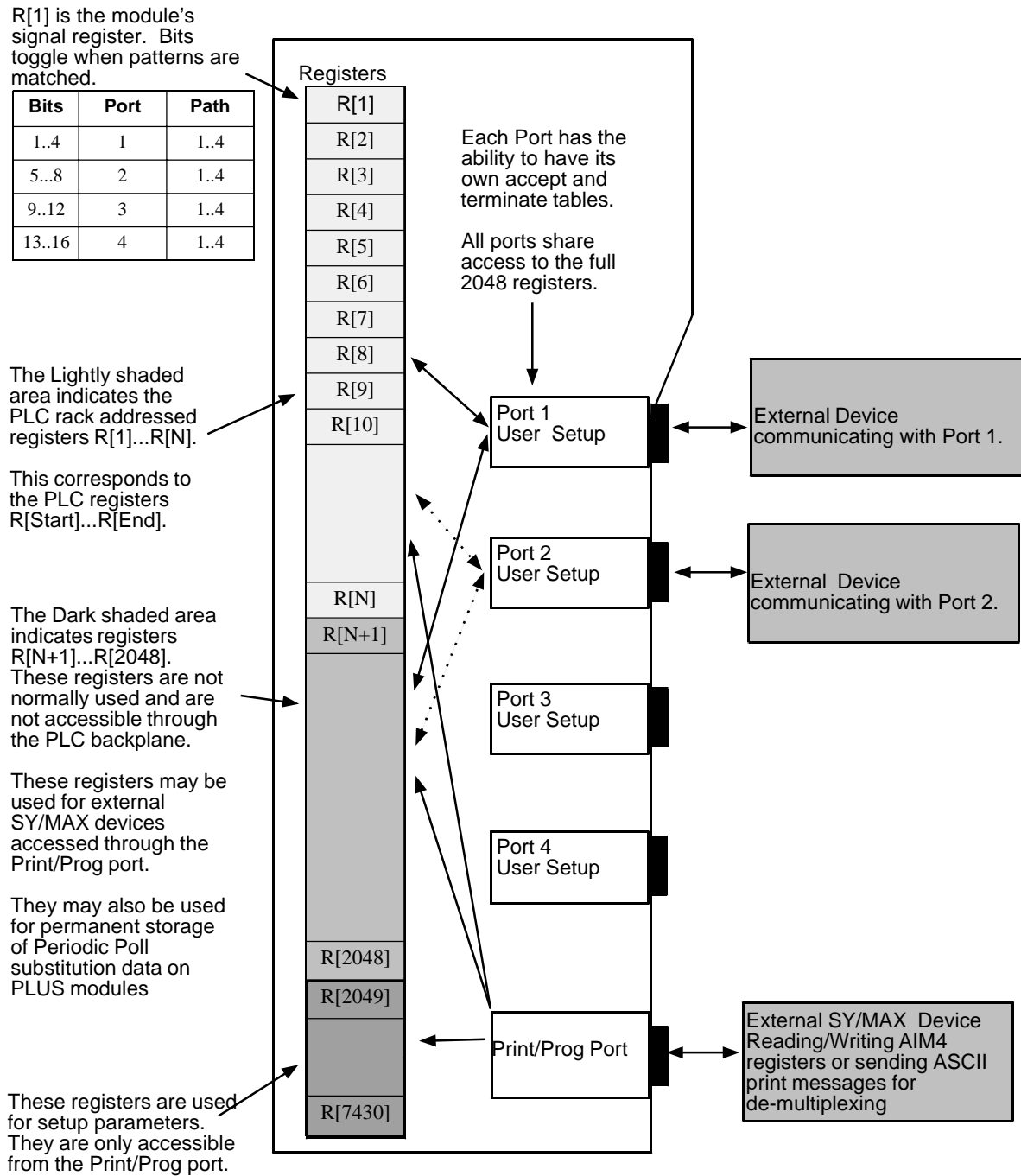


Figure 1-1 AIM4 Memory Map

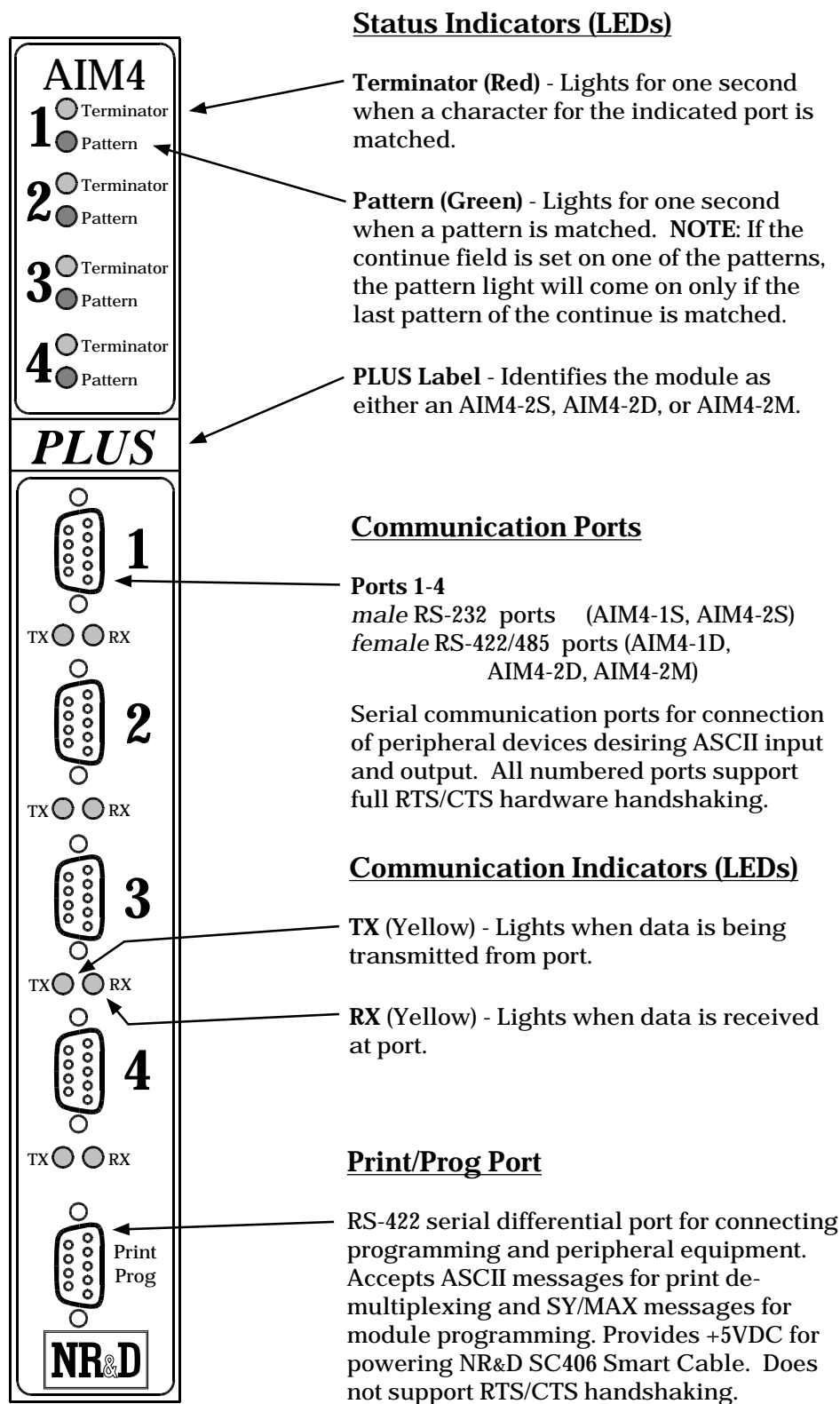


Figure 1-2 AIM4 Front Panel

## Installation

- 1 Mount the AIM4 module in an available slot in the register rack.
- 2 For configuration, the Print/Prog port may be connected to an MS/DOS personal computer. This connection may be made through a COM: port on the personal computer or through the Square D SY/LINK® card. If the connection is direct from the personal computer COM: port, Niobrara's SC406 RS-232 to RS-422 converter cable can be used to make this connection. The AIM4 provides power for the smart cable. If the Square D SY/LINK card is installed in the personal computer, it may also be directly connected from its RS-422 port to the Print/Prog port with a Niobrara DC1 or Square D CC100 cable. Indirect connection to the personal computer may be made through the SY/NET Local Area Network, through SY/MAX compatible modems, or through the Niobrara SPE4 SY/MAX port expander. Consequently the Print/Prog port should be connected to the SY/MAX compatible device using a DC1 or CC100 cable and the personal computer should be attached to the "network" using a COM: port or SY/LINK card. For output operations or SY/MAX communications, the Print/Prog port will later be connected to a SY/MAX port with a Niobrara DC1 or Square D CC100 cable. This chapter assumes a direct connection from the personal computer's COM: port to the AIM4.
- 3 Apply power to the rack. The red and green lights in the top of the AIM4 panel should flash enthusiastically for about a second.
- 4 Using Square D programming software, allocate rack address register space to the module. You may allocate as few as one or as many as 2048 registers depending on the needs of the application. Note the first register address assigned to the AIM4 register slot. This is the Rack Base and will be used in configuring the AIM4.
- 5 The software for assisting in setting up the AIM4 is sent on a 720K 3.5 in. MS-DOS diskette. This diskette includes all of the public domain software from Niobrara in compressed form. The software for the AIM4 is called AIM4SW.ZIP. This file must be expanded using the PKUNZIP.EXE program. This is described in Step 6. The INSTALL program on the diskette allows a simple automatic installation of selected software from the diskette. **To run the INSTALL program simply insert the disk into an available floppy drive. Change to that drive, and enter INSTALL. The INSTALL program will prompt the user for which software to install and where to place that software.**
- 6 To manually install the AIM4SW software, create a directory on your hard disk for the AIM4 configuration software. A directory name of "AIM4" is suggested. Copy the AIM4SW.ZIP and PKUNZIP.EXE files from the included floppy disk to this directory. The following sequence of MS-DOS commands will accomplish these steps on most computers:

```

C:
CD \
MD AIM4
COPY A:AIM4SW.ZIP C:\AIM4\*.
COPY A:PKUNZIP.EXE C:\AIM4\*.
CD \AIM4

```

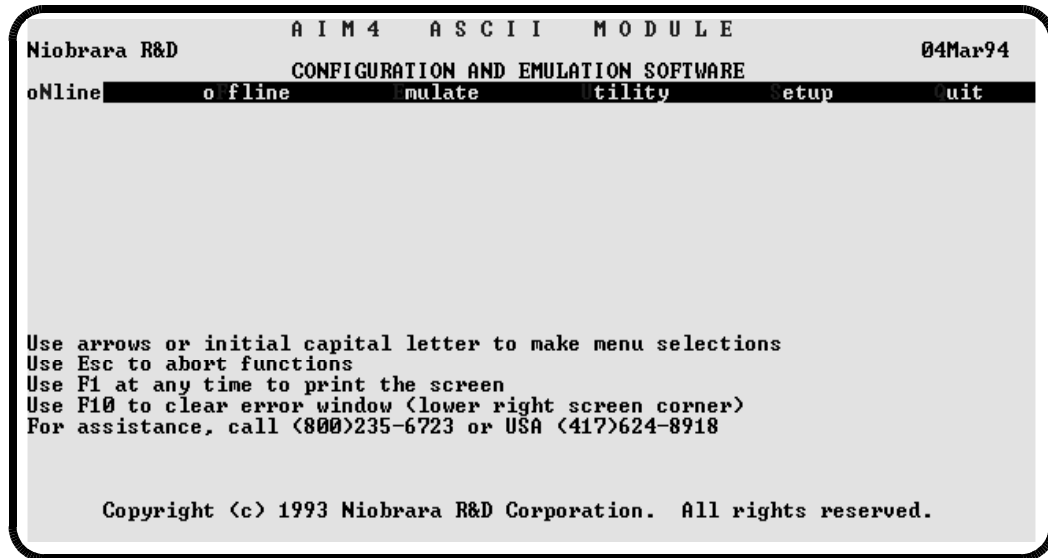
Now expand the AIM4 software using the following command:

```
PKUNZIP AIM4SW.ZIP
```

- 7 Run the AIM4 software with the command:

```
AIM4SW
```

The screen should look like Figure 2-1.



**Figure 2-1 AIM4SW Startup Screen**

- 1 Type the **S** key to select the **SETUP** option from the menu.
- 2 Type the **S** key for **Sy/Max**. A window will appear in the center of the screen. Press the **space bar** until the appropriate mode of communication to the AIM4 or devices connected between the personal computer and then AIM4. For a direct connection from the personal computer COM: port to the AIM4, choose: **Sy/Max COM:**. Press **ENTER**.
- 3 Press the **space bar** until the appropriate COM: port of the personal computer is displayed. Press **ENTER**.
- 4 If the baud rate is not set to 9600, press the **space bar** until 9600 appears. You will probably never need to change this setting. A new AIM4 from the factory is set to 9600 baud, so you must start with this setting even if your application will use a different baud rate later. Press **ENTER**.
- 5 The cursor will skip over the Data Bit field and stop at the Stop Bit field. This value should be set for 1, if it is not press the **space bar** until a 1 appears. Press **ENTER**.
- 6 If the parity is not set to **EVEN**, press the **space bar** until **EVEN** appears. Press **ENTER**.
- 7 If a direct connection between the personal computer and the AIM4 is made, no route is needed. Press the **DEL** key and **NONE** will be displayed. Press **ENTER**.
- 8 A prompt will appear in the lower left hand corner of the screen asking you if you want to save the setup information entered in steps 6 through 11. Type **Y**. Future invocations of the AIM4SW program will use these values and will not require setting the COM port and baud rate as in steps 2 and 4-7 above.

- 9 Type the **N** key to enter the oNline section of the software. After a few seconds, you should see the screen shown in Figure 2-2 Online Screen on page 15. If you get an error message in the lower right corner of the screen, press **F10** to acknowledge the error. Then recheck your connections and try oNline again. If the baud rate of the AIM4 has previously been changed, you may have to alternate between the Setup menu and the oNline mode until you discover what the baud rate setting of the AIM4 is.
- 10 Attach one or more ASCII input devices to the AIM4 data ports. Refer to Connector Pinouts page 87 and Recommended Cabling page 89 for port and cable pinout diagrams. An RS-232 or RS-422 breakout box is useful in verifying the correct connection of input devices.
- 11 Use the AIM4SW program to configure port parameters for the input devices attached. See Figure 2-2 for parameter configuration information.
- 12 Using a programmer or SFW software on a PC, monitor the registers in the programmable controller. Trigger input devices one at a time to test transmission of data.

```

Niobrara R&D      AIM4  ASCII  MODULE
Type 1 to select a port. Type 2 through 4, ., or 5 through 7 change parameters.
1 Port 1
2 Baud rate 9600
3 Data bits 8
4 Parity NONE
5 Stop bits 2
6 Capitalize NO
7 Accept characters
8 Terminate characters
9 Periodic Poll
0 Terminate Count 0
1 Terminate Timeout 0
-----
2 Pattern *-.....
3 Mask -
4 Start Register 2
5 Register Count 1
6 Editing INTEGER
7 Continue NO
-----
8 Pattern .....
9 Mask -
0 Start Register 3
1 Register Count 0
2 Editing ASCII
3 Continue NO
-----
4 Pattern .....
5 Mask -
6 Start Register 4
7 Register Count 0
8 Editing ASCII
9 Continue NO
-----
0 Pattern .....
1 Mask -
2 Start Register 5
3 Register Count 0
4 Editing ASCII
-----
5 Print/Prog baud rate 9600
6 Rack start address 1
7 Trigger/Inhibit register 0
Press Esc to return to menu.

```

Figure 2-2 Online Screen



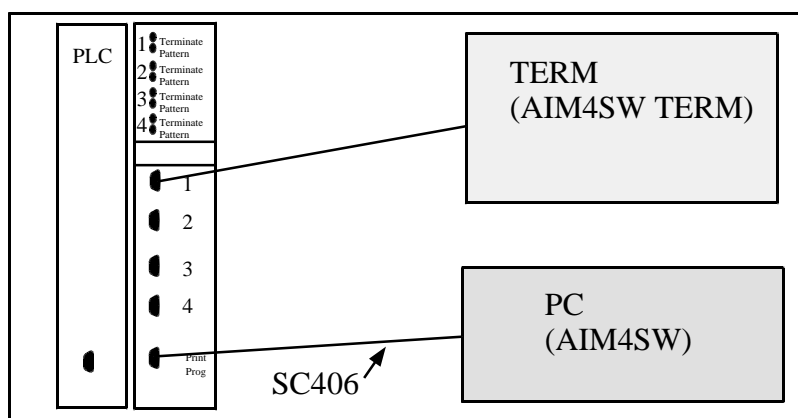


## Quick Start

This instructions in this chapter allow the user to quickly operate the AIM4 from the box. The following equipment is needed:

- AIM4
- SY/MAX PLC
- Register Rack and Power Supply
- PLC programmer (May be the same PC used for AIM4 programming.)
- Personal computer running AIM4SW(PC)
- ASCII Terminal or Personal computer running AIM4SW Terminal Emulator (TERM)
- Cable from the PC to the AIM4 Print/Prog port (SC406 or SC902)
- Cable from the TERM to the AIM4 Data port.

Install the AIM4 as directed in Chapter 2. Power up the rack. The Terminator and Pattern lights should blink quickly and extinguish. Connect the PC to the Print/Prog port with the SC406. Connect the TERM to Port 1 of the AIM4. The setup should be as shown in Figure 3-1.



**Figure 3-1 Quick Start Setup**

## TERM Setup

Setup the TERM for operation at 9600 baud, 8 data bits, No parity, and 1 stop bit. If using a PC with AIM4SW, use the following command from the AIM4 directory:

>AIM4SW

To setup COM1 for the terminal emulator, select S for Setup, from the main menu. Now select T for Terminal emulator. Now select COM1:, 9600 baud, 8 Data bits, 1 stop bit, and NONE parity. Press ENTER and Y when asked to save to disk.

To use the terminal emulator within AIM4SW, select U for Utility, and then T for Terminal Emulator.

Once the connection to the TERM is made, check to see that the cabling is proper by pressing the SPACE bar. The RX light on Port 1 should blink each time the space bar is pressed. If the SPACE bar is held, the light should blink continuously.

Now press the ENTER key. The RX light should blink once. The Terminator and Pattern lights should come on and then go off.

Press the ENTER key again and notice that the Pattern and Terminator lights do not light this time. This shows that there must be other data besides the terminator for pattern matching with this setup. So press the 1 key and then the ENTER. Both the terminator and pattern lights should light.

Notice that the TERM program does not echo characters typed to the screen. It only displays characters received from the COM port. Characters which are not displayable, such as carriage return or line feed, are displayed as the hexadecimal representation of the raw ASCII value of the character within <>. For instance if a carriage return is received, the character <0D> will be displayed.

## PC Setup

On the programming PC, start the AIM4SW software by entering the following command from the AIM4 directory.

>AIM4SW

- 1 Move the cursor to the Setup section to check for proper PC configuration. Select S for Sy/Max connection. The software will default to a direct connection on COM1:, 9600 baud, 8 data bits, EVEN parity, and 1 Stop bit.
- 2 Select OFFLINE from the main menu.
- 3 Select "Read from disk to memory" to load a configuration file into the PC memory.
- 4 Move the cursor to the file named QSTART and press enter. The QSTART file has now been loaded into the PC's memory.
- 5 Press F for OFFLINE again.
- 6 Press S for Send memory to AIM4. This will pull up a OFFLINE TO AIM4 menu.
- 7 Press A for ALL. The AIM4SW will send the configuration to the AIM4, replacing the configuration that was in the module.
- 8 Select ONLINE from the main menu. The AIM4SW will poll the AIM4 to obtain the setup parameters. The parameters for Port 1 will be displayed. The screen should appear as in Figure 3-2.

Notice that the Rack start address is 1, the Path 1 parameters are:

- Pattern is the \* wildcard. (Accepts all characters in the Accept table.)
- Mask is all underscores. (Passes through all characters matched in pattern.)
- Start Register is 2. (Register to start placing data.)
- Register Count is 1. (Number of registers affected.)
- Editing is INTEGER. (Data will be placed in register as integers if possible.)
- Continue is NO. (If pattern A is matched, it will not check the rest of the patterns.)

```

Niobrara R&D  AIM4  ASCII  MODULE
Type 1 to select a port. Type 2 through 4, or 7 through 9 change parameters.
1 Port 1 5 Stop bits 2 7 Periodic Poll
2 Baud rate 9600 6 Capitalize NO 8 Terminate Count 0
3 Data bits 8 7 Accept characters 9 Terminate Timeout 0
4 Parity NONE 8 Terminate characters
----- DATA PATH 1 -----
1 Pattern *-----
2 Mask -----
3 Start Register 2 4 Register Count 1 6 Editing INTEGER 8 Continue NO
----- DATA PATH 2 -----
1 Pattern *-----
2 Mask -----
3 Start Register 3 4 Register Count 0 6 Editing ASCII 8 Continue NO
----- DATA PATH 3 -----
1 Pattern *-----
2 Mask -----
3 Start Register 4 4 Register Count 0 6 Editing ASCII 8 Continue NO
----- DATA PATH 4 -----
1 Pattern *-----
2 Mask -----
3 Start Register 5 4 Register Count 0 6 Editing ASCII
-----
7 Print/Prog baud rate 9600 8 Rack start address 1
Press Esc to return to menu. 9 Trigger/Inhibit register 0

```

Figure 3-2 QSTART Setup

To see the registers in the AIM4 the Utility Register Viewer is very useful.

- 1 Press ESC to return to the main menu.
- 2 Press U for Utility menu.
- 3 Press V for View registers.

The AIM4SW is now showing the values in registers R[1] through R[20] in the AIM4. (See Figure 3-3.)

Niobrara R&D AIM4 ASCII MODULE									
CONFIGURATION AND EMULATION SOFTWARE									
REGSTR	HEX	UNSIGN	SIGNED	BINARY				STAT	
1	0001	1	1	0000	0000	0000	0001	E000	
2	007B	123	123	0000	0000	0111	1011	E000	
3	0000	0	0	0000	0000	0000	0000	A000	
4	0000	0	0	0000	0000	0000	0000	A000	
5	0000	0	0	0000	0000	0000	0000	A000	
6	0000	0	0	0000	0000	0000	0000	E000	
7	0000	0	0	0000	0000	0000	0000	A000	
8	0000	0	0	0000	0000	0000	0000	A000	
9	0000	0	0	0000	0000	0000	0000	A000	
10	0000	0	0	0000	0000	0000	0000	E000	
11	0000	0	0	0000	0000	0000	0000	A000	
12	0000	0	0	0000	0000	0000	0000	A000	
13	0000	0	0	0000	0000	0000	0000	A000	
14	0000	0	0	0000	0000	0000	0000	E000	
15	0000	0	0	0000	0000	0000	0000	A000	
16	0000	0	0	0000	0000	0000	0000	A000	
17	0000	0	0	0000	0000	0000	0000	A000	
18	0000	0	0	0000	0000	0000	0000	A000	
19	0000	0	0	0000	0000	0000	0000	A000	
20	0000	0	0	0000	0000	0000	0000	A000	

04Mar94  
AIM4 Sy/Max  
Register Uiewer  
Press F2 for Help

Figure 3-3 Utility View Register Screen

## INPUT Operations

To demonstrate the input capability of the AIM4 the TERM will be used to supply ASCII strings for the AIM4 to process.

On the TERM press 123Enter.

The following should have occurred:

- The RX light should have blinked on each character received.
- The Terminator light should have come on then off.
- The Pattern Light should have come on then off.
- The decimal value 123 should have appeared in register R[2] in the AIM4.
- Bit 1 in register R[1] should have toggled from 0 to 1 or from 1 to 0.

On the TERM now press 456Enter.

The value 456 should now be in R[2]. Bit 1 in R[1] should have toggled.

On the TERM now press 456Enter again.

Bit 1 in R[1] will toggle again but the data in R[2] will be the same. This feature allows the PLC to know when new data arrives, regardless if it is the same data.

## Rack Addressing

Now that we can get data into the module, let's see how to get it into the PLC. Connect the PLC programmer to the PLC and rack address the slot that the AIM4 occupies for registers 40 through 50.

Now with the PC connected to the AIM4 Prog port. Use AIM4SW to go to the ONLINE screen and perform the following steps:

- 1 Press Y and enter 41 for the Rack Start Address.
- 2 Press Y to confirm the Adjust Start Register in all paths. AIM4SW will automatically adjust the start register in all the paths. When it is finished notice that the start registers that were 2, 3, 4, and 5 are now 42, 43, 44, and 45. These are the PLC registers where the data will be placed. The registers in the AIM4 that are viewed through the Print/Prog port will still be 2, 3, 4, and 5.
- 3 To illustrate the above, PRESS ESC to return to the main menu in AIM4SW.
- 4 Press U for Utility, and V for View registers to display registers R[1] through R[20] in the AIM4.
- 5 Now disconnect the SC406 from the AIM4 Print/Prog port and move it to one of the COM ports of the PLC. The Register Viewer is now displaying R[1] through R[20] of the PLC.
- 6 Press the left arrow to move the cursor to the REGSTR column.
- 7 Press 40Enter to move the displayed range of register from R[40] to R[59].
- 8 On the TERM press 789Enter. Notice that bit 1 of R[41] toggled and the decimal value 789 appeared in R[42].
- 9 On the TERM press 123Enter. Again bit 1 of R[41] toggled and the value 123 appeared in R[42].
- 10 Now move the SC406 back to the AIM4 and notice that R[40] through R[59] are zero. Press the HOME key to return to R[1] through R[20]. Notice that R[2] has the value 123.

You have just witnessed the relationship between the PLC registers and the AIM4 registers. Make sure that the Rack Start Address matches the value in the PLC in your application. Otherwise you may be placing data in other locations.

## OUTPUT Operations

### Periodic Poll

To demonstrate the Periodic Polling output feature of the AIM4, perform the following tasks:

- 1 Connect the PC to the AIM4 Print/Prog port and go ONLINE with AIM4SW.

- 2 Press 9 for Periodic Poll for Port 1.
- 3 Press 1.5 followed by the down arrow to move to the Query strings.
- 4 For Query 1 type: Hello
- 5 Press ENTER to accept the periodic poll setup and return to the ONLINE screen.

Notice that the TERM is now displaying HelloHelloHello... with a new Hello being added every 1.5 seconds. This type of query is very useful for prompting attached devices without the PLC's involvement.

- 6 Press 9 to return to Periodic Poll.
- 7 Press down arrow to Query 1.
- 8 Press the END key to go to the end of the string in Query 1.
- 9 Press F9 to enter the Hex Edit Mode. This mode allows direct hexadecimal editing of the query.
- 10 Press 0D to insert the carriage return character after the 6F character.
- 11 Press 0A to insert the line feed character after the 0D character.
- 12 Press ENTER to return to the Query 1 line.
- 13 Press ENTER to accept the Periodic Poll and return to the ONLINE screen.

Notice that the TERM is now displaying Hello<0D><0A> on each line.

- 14 Press 9 to return to the Periodic Poll.
- 15 Press 0 and ENTER to set the Poll Rate to 0 seconds and consequently disabling the Periodic Poll.

Notice that the TERM has stopped being updated by the AIM4.

## ASCII Demultiplexing

To demonstrate the demultiplexing capabilities of the AIM4 perform the following tasks:

- 1 Press ESC to exit from the ONLINE screen.
- 2 Press S for Setup.
- 3 Press T for Terminal emulator.
- 4 Select COM1:, 9600 baud, 8 data bits, 1 stop bit, NONE parity.
- 5 Press Enter and Y for yes to save to disk.
- 6 Press U for Utility.
- 7 Press T for Terminal Emulator. You should now have a terminal emulator connected to the Print/Prog port of the AIM4 and a terminal emulator connected to Port 1 of the AIM4.
- 8 To cause the AIM4 to establish a link between the Print/Prog port and a data port the first character sent into the Print/Prog port must be the number of the data port.

Press the number 1 key to start the link to port 1.

Notice that nothing appeared on the TERM.

- 9 Now type any sequence of characters. Notice that each character typed appears on the TERM, and the TX light blinks with the transmission of each character.
- 10 To break the link from the Print/Prog port to a data port one of the three following characters must be sent: carriage return <0D>, line feed <0A>, or form feed <0C>. Press the ENTER key on the PC to send a carriage return.

Notice that a <0D> was received at the TERM.

- 11 To re-establish the link or to create a new link with another port, the first character received will need to be the number 1 through 4. So move the TERM cable from port 1 to port 2 and try to send data to port 2.

This concludes the Quick Start Chapter. It has hopefully given the user a brief introduction to the major features of the AIM4 and to get a "feel" for the module. These exercises have just touched the surface of the capabilities of the AIM4. It is recommended that the user now read the rest of the manual to for further explanation of these features.

## Configuring the Port Parameters

The configuration of the port parameters is the key to accessing the AIM4's powerful data input capabilities to simplify ASCII data collection and enhance PLC system performance. There are many ways to combine the effects of the various parameters to achieve desired results. The following discussion explains the flow of data through the input logic of a single data port of the AIM4. A flow chart is shown in Figure 4-3 on page 26.

Characters are received one at a time according to serial port configuration parameters. A list of characters, the Accept Character table, determines whether each character received is placed in a buffer or ignored. A second list, the Terminate Character table, determines which character or characters indicate the end of the message. When a terminate character is recognized, the message is passed on to the pattern recognition, mask editing, and register format conversion. Recognition of a terminate character with a non-empty message buffer is indicated by the red Terminator light flashing. This is the sequence of events for each character:

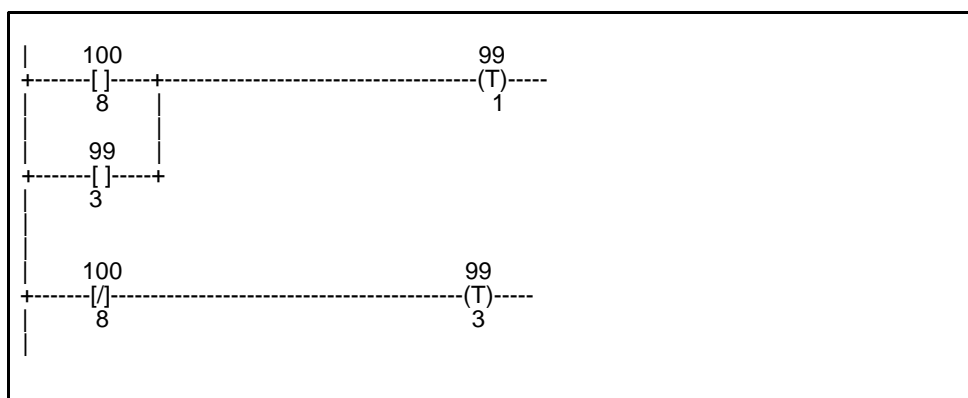
- 1 The character is received on a data port according to the settings of communications parameters Baud Rate, Data Bits, and Parity.
- 2 If the character is found in the Accept Character table, it is placed in the input message buffer.
- 3 If the character is found in the Terminate Character table, the input message is considered complete and processing continues with step 4, if the character is not in the Terminate Character table, repeat steps 1 through 3 until a terminate character is found. Note that a character which is listed in both the Accept Character table and the Terminate Character table will be included in the message (as the last character) and also trigger subsequent processing of the message. Characters which match neither the Accept Character table nor the Terminate Character table are ignored.

Note: The "Terminate Character" may not necessarily be a character on the -2 Model. It may be a maximum message length count or a certain time has expired between characters. Refer to Page for more information on message termination.

- 4 The red Terminator light is illuminated.
- 5 The most significant bit of each character is set to zero to make all characters valid ASCII codes. The most significant bit is tested by the Accept and Terminate Character tables but is ignored by Pattern matching, Masking and register value conversion.
- 6 If the Capitalize parameter is set to YES, all lower case letters are converted to upper case.
- 7 The input message is compared to the Pattern of the first Data Path and if a match is found, the input message is processed through that data path. If the input message does not match that pat-

tern, it is compared to the patterns of Data Path 2, 3, and 4. The first matching pattern will trigger its data path and the others will not be tested unless the Continue option for that path is set to YES. If the input message does not match any of the patterns, it is discarded and processing of a new message resumes at step 1 above.

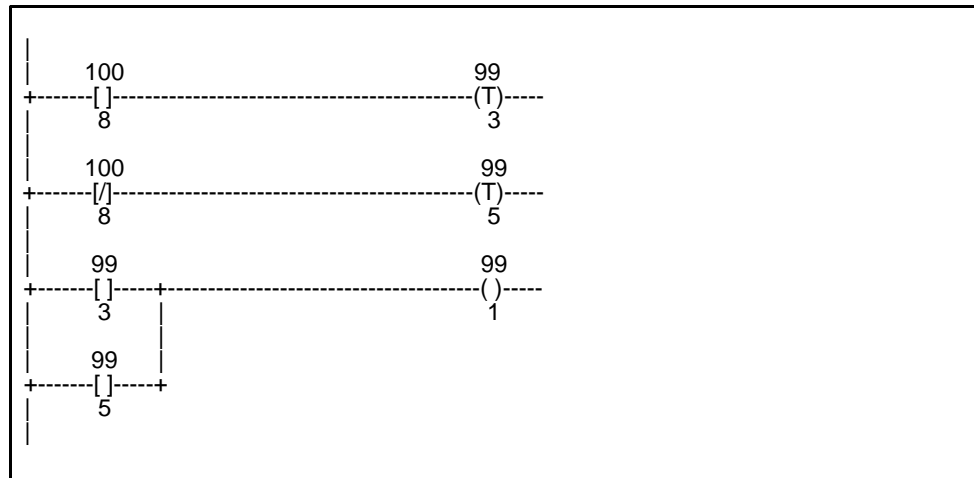
- 8 The green Pattern light is illuminated, indicating that one or more of the four Data Paths has been triggered.
- 9 The message is processed by the Mask of the activated data path. Each position of the mask with an underscore character \_ retains the character in the corresponding position of the input message. Any other character in the Mask replaces the corresponding position of the message with that character. The length of the masked message will become that of the shorter of the input message or the mask.
- 10 The selected editing operation is applied to the masked message to prepare data values suitable for placement in registers. ASCII and PACKED editing restore the most significant bit of the original input data.
- 11 Edited data in register format is placed in the AIM4 rack image table at the location indicated by the Start Register parameter for the triggered path. The number of registers affected is controlled by the Count parameter. For INTEGER and FLOAT editing, the number of registers required is fixed by the nature of the output format (one for INTEGER and two for FLOAT). Additional registers specified by Count will be set to zero.
- 12 If Continue is YES for the path, a copy of the input message is tested against the pattern of subsequent paths and, if a match is found, steps 7 through 12 are repeated for the new path.
- 13 The Edited data is transferred to the SY/MAX processor at the end of the next scan. The first register in the module (i.e. Rack Base) is used by the AIM4 to signal the triggering of any path. Bits 1 through 4 are for Port 1, Paths 1 through 4. Bits 5 through 8 are for Port 2, Paths 1 through 4. Bits 9 through 12 and 13 through 16 are for ports 3 and 4 respectively. Each time a Data Path is triggered, the corresponding bit of the signaling register changes state. This change informs the processor of the arrival of new data (or simply recognition of a pattern if the Register Count is zero). A signaling bit is toggled by the same conditions which flash the Pattern light. A pair of transition coils can be used to detect the toggling of this bit and generate a one scan time long pulse to trigger other logic when an AIM4 data path is triggered. For example, the following ladder program segment, in Figure 4-1, will generate a pulse on transition coil 99-1 when the AIM4 with a Rack Base setting of 100 triggers on Data Path 4 of Port 2:



**Figure 4-1 Trigger Bit Capture Ladder Rungs**

The latency in this example is one scan time longer for input events which clear bit 100-8 than for those which set it. This can be avoided, if necessary, at the expense of an additional rung and internal relay, see Figure 4-2.

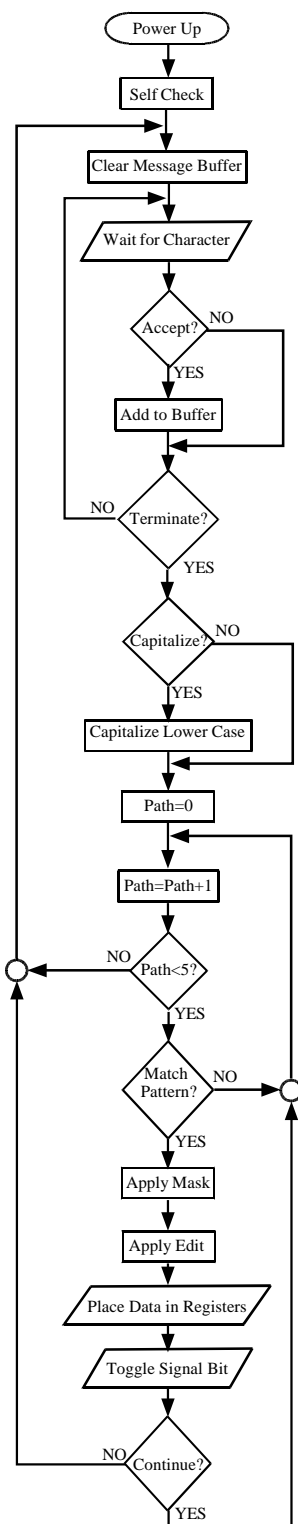




**Figure 4-2 Trigger Bit Capture Ladder Rungs (better solution)**

Monitoring the signaling bits is superior to just watching for the register data value to change since the newly arrived data could have the same value as the current data. It is not possible for the processor to zero an AIM4 register because the AIM4 is an input only device. (Dash 2 modules contain PLC output registers which can be zeroed. If the direction of a specific register is in question, look at the status register for that data register. Input registers have the hex value E000 in their status register; Output registers have the value A000 hex.) If it is desired to zero data after processing it, use the ladder logic above to trigger a matrix LET to copy the data to a separate group of internal relay registers that can be zeroed by the ladder program.

- 14 If internal message buffers are filling faster than the AIM4 can process the messages, the AIM4 will negate its RTS signal until it has a chance to catch up with the backlog and then it will again assert RTS. On RS-232 data ports RTS is pin 7 and is positive when the AIM4 is ready to accept data. On RS-422 data ports, RTS is the differential pair of pins 6 and 8. Pin 8 will be more positive than pin 6 when the AIM4 is prepared to accept more input. Due to the high speed of the AIM4 processor, this restraint function will only rarely be active. These pins can be left uncommitted in almost all applications. If you drive all four data ports continuously at greater than 9600 baud, you may need to connect the RTS output to your data source. Refer to Throughput and Latency on page 85 for more information.



**Figure 4-3 ASCII Input Flow Chart**

## Parameter Guide

Three types of parameters control operation of the unit. Global parameters, Port parameters and Path parameters. Global parameters affect operation of the entire unit. Port parameters affect all data passing through a port. Path parameters affect only characters routed through one of the four paths provided for each port.

Parameters are configured using the supplied AIM4SW program on an MS/DOS computer. All of the parameters for a single port (together with the global parameters) are displayed on a single screen. Typing a **1** key scrolls the display through the four ports. Each parameter on the screen is displayed beside a digit, 2 through 8, or a letter, A through W. Typing this selection character either changes the value of a multiple-choice parameter, such as baud rate or parity, or opens an entered parameter, like Pattern or Mask for editing. When entering edited parameters, the **Backspace** or **Del** keys can be used to erase mistakes, left and right arrows can be used to move the cursor within the field, and the **RETURN** or **ENTER** key is used to accept the displayed value of the parameter and return to the single-key selection of parameters. AIM4 configuration parameters are maintained in internal memory by AC line power or batteries in the rack power supply or by an internal capacitor when the module is removed from the rack.

### Port Parameters

The following are the **port parameters**. There is one set of port parameters for each of the four data ports.

#### Baud Rate

Communications data rate in baud or bits per second. This setting must match the baud rate of the data source. Use of 9600 baud if possible is recommended. The slower the baud rate, the more reliable is the message transmission and the less likely are problems with dropped characters. In most cases, high data rates do not improve system throughput significantly. A barcode scanner which only sees a label every three seconds has time to transfer over 350 characters of label information between readings at 1200 baud. In general, and especially when using many ports simultaneously, use the lowest baud rate which can transfer a message with acceptable latency.

#### Data Bits

Set to 7 or 8 to match the data bits transmitted by the connected source device. If this parameter is seven bits, the eighth bit of all received characters will be treated as a 0. The eight bit character is tested against the Accept Character and Terminate Character tables but truncated to seven bits for Pat-

tern matching, masking, and editing. In 8 bit mode and if ASCII or PACKED editing is selected, the full eight bit character will be transferred to the PLC.

## Parity

Normally set to match the connected data source. A special case is when you want to test parity in the Accept Character table. The data source must be seven bits with parity. Set the Parity parameter to NONE and the Data Bits parameter to 8. The eighth bit (described above) will be the parity bit from the data source and the Accept Character table can be loaded with only those character patterns with the desired parity.

## Stop bits

Controls the stop bits of the transmitter only. The receiver only looks for one stop bit but is not affected by more. In almost all situations, a setting of 2 will work best.

## Capitalize

If this parameter is set to YES, lower case letters (a through z) are converted to their upper case equivalents (A through Z). This conversion takes place after matching the character to the Accept and Terminate character tables and before Pattern matching. To accept both upper and lower case characters, both must be enabled in the Accept Character table regardless of the setting of Capitalize.

## Accept Characters

This is a list of the character codes which the data ports will recognize and place in the input buffer for processing. If a character is enabled in this table and in the Terminate Character table, it will be placed in the buffer (Accepted) and then the buffer will be processed (Terminated). The table is controlled by typing a 7 to get the Editing Accept Character table display. The display shows all possible eight bit characters as hexadecimal codes. Enabled characters are shown in reverse video. A cursor can be moved around the table with the arrow keys. The ASCII character or control code of the cursor position is shown on the right side of the table. Use the + key to turn on a character, the - key to turn it off, or the space bar to change its state.

## Terminate Characters

This is a list of the characters that will terminate input and start processing of the input buffer. Typical terminators are carriage return (hex 0D) or line feed (0A). The terminate character is not placed in the buffer unless it is also in the Accept Character table. See Accept Characters for a description of the character table editing procedure. Terminate characters need not be the obvious end of a message. Other delimiters are often found in barcode labels and scale data streams. Creative selection of terminate characters can be used to break a single string from an attached device into multiple messages so that various fields may be loaded into individual registers.

## Periodic Polling

The periodic polling parameters consist of a polling interval and up to sixteen poll queries. At the specified interval, a poll query is output through the port. If more than one query is provided, they will be used in rotation.

## Terminate Count

(PLUS units only) In addition to the terminate character set provided on all AIM4 models, the AIM4-S2 and AIM4-D2 provide the ability to terminate a message and begin processing based on a specified message length. Set this parameter to zero (default) to disable maximum count termination. This method is not highly recommended due to the fact that synchronization is easily lost but may be useful in conjunction with maximum silence termination and implemented to resynchronize in the event of a dropped or extra character.

## Terminate Timeout

(PLUS units only) A message may be terminated by a certain time expiring between characters. The time is units of hundredths of a second (i.e. the value 100 specifies one second). Set this value to zero (default) to disable maximum silence termination.

## Trigger/Inhibit register

(PLUS units only) The values stored here are the register numbers of the trigger/inhibit registers as defined by the PLC: as specified in the rack base module parameter. For instance, a module has been rack addressed the PLC registers 200 to 350, and the Base Address in the module is set to 200. The Trigger/Inhibit registers to be used in the ladder program are 347, 348, 349, and 350 for ports 1, 2, 3, and 4 respectively. The values stored in the module registers 7431, 7432, 7433, and 7434 would be 347, 348, 349, and 350.

It is a good idea to place these registers at the end of the group of rack addressed registers. The PLC updates the rack sequentially from the lowest assigned register to the highest. The AIM4 will trigger on the data in the Trigger/Inhibit register almost immediately upon the PLC's writing of that register. If the Trigger/Inhibit register is located near the end of the assigned group of registers, all the output registers that may be used in the associated Periodic Poll queries will be written by the PLC before they are triggered. Likewise, if the Trigger/Inhibit registers are near the beginning of the assigned registers, the AIM4 may be triggered and start outputting strings before the PLC is able to write new data to the rest of the module on that scan.

Note: It is not necessary for each port to have an independent trigger/inhibit register. Two or more ports may share the same register allowing one condition to govern the output of several ports. In fact the trigger/inhibit register may be assigned to the signaling register of the module (the first register assigned to the module). This way a certain input condition on a particular path on a port could trigger a corresponding query on that or another port without PLC intervention.

## Path Parameters

The following are the **path parameters**. There are four data paths for each data port or sixteen sets of these parameters overall:

### Pattern

This parameter is used to select which input messages will be processed by a data path. When an input message is received by a port (one or more Accept characters followed by a Terminate character), the message is tested against the Pattern of each Data Path of the port. The first Pattern which matches selects that data path and the message is processed by masking and editing and placed in registers for transfer to the PLC processor. Pattern matching proceeds by testing the characters in the input message against the characters in the pattern i.e. a pattern like ABC will match only the input message ABC (not even abc unless Capitalize is YES).

Patterns can also include the special symbols \* = # and ?. The = symbol matches any single character, the ? symbol matches any letter A through Z, the # symbol matches any digit 0 through 9, and the \* symbol matches any group of 0 or more characters.

Two special constructions are used to match numeric or alphanumeric subsets.

- A list of characters in square brackets [ ] matches any single character from the list. The list may include a hyphen to indicate a range of characters such as [A-Z]. A leading or trailing hyphen is literal. **Note:** The square brackets [ ] may be used to match one of the wildcard symbols as a literal character; [\*] will match the \* character, [=] will match the = sign.
- A range of numeric values may be matched by a list of values enclosed in parentheses ( ) such as (1-100). In this case, a leading or trailing hyphen indicates an open ended range. (-100) matches numbers 0 through 100 and (2-) matches numbers greater than 1. Commas may be used to define a discontinuous range such as (-10,12-) which matches any group of digits that don't evaluate as 11.

Below are some examples of patterns:

*	Matches any input message
A*	Matches any input message starting with A
====	Matches any four character input message
????	Matches any four alpha letter input message
####	Matches any four numeric digit input message
*K	Matches any input message ending in K
G#####	Matches a G followed by 6 digits.
?###*	Matches any letter followed by at least three digits.
ZONE	Matches only the message ZONE
*ZONE*	Matches any message containing ZONE
*?*	Matches any message with a letter in it
[A-K]	Matches A,B,C,D,E,F,G,H,I,J, or K
[A-K]*	Matches any string starting with a through K
O[NF]*	Matches any string starting with ON or OF
[+-]#####	Matches any five digit number with an explicit sign
[RB]O[BY]	Matches ROB, ROY, BOB, or BOY
[=]	Matches the literal character =
(1-100)	Matches any sequence of digits that equals a number between 1 and 100 inclusive.
[A-Za-z]	Same as ?, matches any letter
[0-9]	Matches any single digit, same as #
(0-9)	Matches any string that equals 1 through 9 including 9, 09, 009, 0009, etcetera.
(-)	Matches any sequence of digits.

## Mask

When Pattern is found to match the input message, the Mask of the selected data path is applied to select which characters of the input are to be edited and placed in the registers. The mask replaces the input message except for any underscore \_ characters in the mask which retain the corresponding character from the input message. The Mask consisting of sixty four underscore characters therefore, will not change the input message at all.

Some example masks are shown below:

___	Only process the first three characters of the input message.
0___	Replace the first character of the input message with a zero. This might be used with a Pattern like A#### to identify an input with a certain first character but which will be converted (by Editing) to a number.
___A	Append an A to the end of a three character input message. This might be used so that the host (PLC) can determine which port or path was the source of a value when more than one Data Path is configured with the same Start Register.
___000	Might be used with a Pattern like ##### to ignore the three least significant digits of a number and replace them with zeroes. This might be used to truncate numbers coming from a scale, for instance.
1	This mask always produces the message 1. Use this type of mask with INTEGER editing to place a constant bit pattern in a register when the input message matches the pattern. For instance, this could place a code (like 1) in a register when a certain type of barcode label (matching the corresponding Pattern) was scanned.

The special character 7F(hex) acts as a skip character. The corresponding character in the pattern will be treated as though it were not present.

## Start Register

Where to place edited data in the register memory of the PLC processor. This is not the register in the AIM4, the Rack Base address is used to modify the Start Register so that the actual AIM4 register, when transferred to the processor, will be the PLC register numbered as Start Register. Usually a different range of registers is used for each path and port but it is possible to configure multiple data paths to affect the same registers, for example when Pattern matching is used with a constant mask to simply identify the input message. The minimum value for this parameter is the Rack Base register number plus one. The maximum value is Rack Base plus 2047.

## Register Count

This is the number of registers which will be affected by the data path. If the Start Register is 109 and the Register Count is 4, for instance, registers 109, 110, 111, and 112 are controlled by the data path. Register Count cannot be more than 64.

## Editing

This parameter selects the way a masked message will be converted to a register value before placing in the registers. There are five possible editing modes.

**ASCII** editing simply copies as many characters of the input message as will fit into the registers. If there are more registers than the length of the input, the remaining registers are set to zero. One seven or eight bit character is placed in the lower half of each register and bits 9 through 16 of the register are set to 0.

**PACKED** editing is a special form of ASCII editing. Two characters are placed in each register. The first character is placed in the most significant byte of the first (Start Register) register. This format is compatible with the raw ASCII print mode of a SY/MAX PLC print rung.

**INTEGER** editing converts the masked message (which must be a number) to an integer value (-32768 to 65535) which is placed in the register identified by Start Register. If Register Count is more than one, the other registers will be cleared to zero. Any leading non-numeric characters will be ignored. A - sign, or a + sign may precede the numeric digits in the message. The + and - set the algebraic sign of the edited integer value. Conversion of the message stops when any character other than 0 through 9 is scanned.

**FLOAT** editing converts the masked message to a floating point number. The number may include a sign and a decimal point. Scanning stops on any other character. The result is placed in two registers at Start Register. If more than two registers are specified by Register Count, the others are set to zero. If Register Count is one, only half of the floating point value will be transferred. Although the AIM4 can place FLOAT values on either even or odd register boundaries, the Start Register for FLOAT normally must be odd for subsequent use by SY/MAX PLC processors.

**BCD** editing converts the masked message to a binary coded decimal representation packing 4 digits per register. Up to a 64 digit number may be processed this way. The BCD result is aligned with the least significant digit in bits 1 through 4 of the highest numbered register.

**HEX** editing converts characters from HEX ASCII into binary integers. Valid hex-ASCII digits are 0 through 9 and a through f or A through F.

**OCTAL** editing converts characters from OCTAL ASCII into binary integers before placing them in registers. Valid octal digits are 0 through 7.

**IDEC** editing converts characters from IDEC ASCII into binary integers. This is useful when communicating between a Model 50 or Micro-1 and a SY/MAX processor.

## **Continue**

The Continue parameter is available on paths one, two, and three only. When Continue is set to YES on a given path, each input message is passed on to higher numbered paths whether or not it is matched by the pattern of the given path. When Continue is set to NO, the first path whose pattern matches the input message will process that message and the message will not be processed by (it will not even be tested against the pattern of) higher numbered paths. Path 4 has no Continue parameter since there is no higher numbered path.

## **Global Parameters**

Global parameters control communication between the AIM4 and the PLC processor or the computer running the configuration software. They are:

### **Print/Prog baud**

This is the baud rate of the Print/Prog port. When running the configuration software, this setting must match the setting specified in the SETUP menu of the AIM4SW program. To demultiplex print rungs through the AIM4, or to communicate with the module via the SY/MAX protocol, this parameter must match the setting of the attached processor or NIM SY/MAX port. Usually 9600 is used. The Print/Prog port automatically determines whether to use 11 bit SY/MAX mode or the 10 bit (1 start, 7 data, 1 parity, 1 stop) protocol used by SY/MAX ports when driving a printer.

### **Rack Base**

This parameter is set to match the low register number assigned to the module in the PLC's rack address table. The AIM4 uses this number internally to translate the Start Register parameter to an internal address such that the PLC will transfer the data to its image memory at the register number specified in Start Register. Rack Base is also the number of the register which contains the signaling bits.

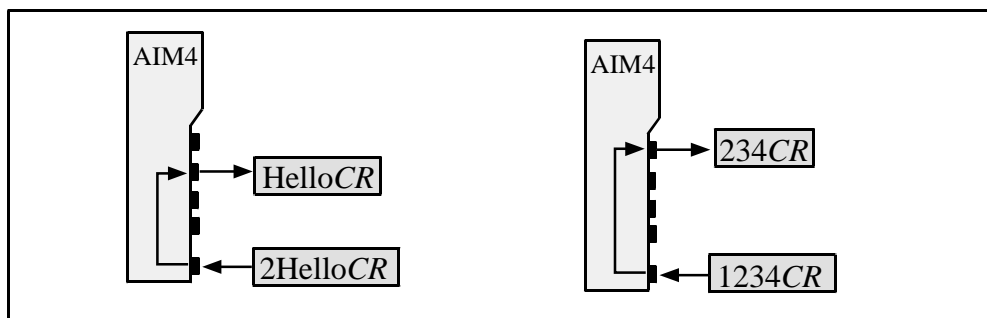


## Output Operations

The AIM4 can output using two different methods, either print rungs from the ladder program or periodic polling without processor intervention. The AIM4-2S and AIM4-2D PLUS units have enhanced output features associated with Periodic Polling. For the polling feature see the Periodic Polling chapter on page 35. Both output methods can be used simultaneously through any combination of ports. The following is an explanation of print rung output through the AIM4.

The AIM4 can be used to demultiplex output data presented to its Print/Prog port to the four data ports. Multiple AIM4 modules can be daisy-chained to direct output to any number of attached peripherals.

Output data is steered from the Print/Prog port to one of the data ports when an ASCII numeric digit 1 through 4 is received on the port. This selection digit is not output, but all subsequent characters are until a carriage return (ASCII 13), line feed (ASCII 10), form feed (ASCII 12) or null (ASCII 0) are received. Any number and combination of these four delimiters will be forwarded to the selected data port until any other character is received at which time the AIM4 resumes waiting for a 1, 2, 3, or 4. A new selection digit may immediately follow the last form control delimiter of the preceding message. To output one of the four delimiting characters in the middle of a message, follow it with another port selection digit.



**Figure 6-1 ASCII Demultiplexing**

To daisy-chain multiple AIM4 modules, connect the RS-422 source to the Print/Prog port of an AIM4 with the RS-422 data port option. Connect one or more of the RS-422 data ports from this module to the Print/Prog port of a second stage AIM4 (which may have RS-232 or RS-422 data ports). In this configuration, two selection digits are required. The first will direct output through one of the data ports of the AIM4 attached to the originating RS-422 source (i.e. the PLC), the second digit will be passed through this AIM4 to one of the four possible second stage AIM4s and will activate one of its ports.

Avoid sending characters other than 1 through 4 to the Print/Prog port when it is in the idle state. Special character codes are sent to the module in this mode by the AIM4SW program to configure the operational parameters.

SY/MAX PLCs include a parity bit in the data output by a print rung. The AIM4 removes this parity bit. Parity of output data is determined by the AIM4 Port Parameters. Print rung specification steps (i.e. baud rate) control the format of data output by the PLC but not data emitted by the AIM4. Print rung specification steps, if present, should match the communication characteristics of the AIM4 Print/Prog port which are the normal SY/MAX values of 1 start bit, 8 data bits, 1 even parity bit, and one stop bit at 300, 1200, 2400, or 9600 baud.

Output devices may inform the AIM4 of their inability to receive data by negating the AIM4 CTS lead. For RS-232 data ports, this means that a positive level must be applied to pin 8 (CTS) of the data port to allow the AIM4 to transmit. For an AIM4 with RS-422 data ports, pin 7 must be more positive than pin 5 in order for the AIM4 to transmit. To disable the handshake feature on RS-232 data ports, strap pin 8 to pin 4. On units with RS-422 data ports, internal pullups will enable CTS when pins 5 and 7 are left unconnected.

The AIM4 provides 4096 bytes of output buffering for each data port. But, if data is continuously sent to the Print/Prog port faster than data port baud rate or external device handshaking will allow it to be passed on, characters can eventually be lost. The AIM4 does not return handshake signals to the RS-422 Print/Prog port. If data is being lost, try using a lower baud rate between the Print/Prog port and the attached data source.

In AIM4 modules with firmware revision earlier than 11Dec92, the Print/Prog port does not recognize SY/MAX protocol messages such as Read rungs, Write rungs, and network Print rungs. If you have an early module and need these features, call the factory for an upgrade kit. SY/MAX operations are further discussed in chapter 7.

## Periodic Polling

The periodic polling feature of the AIM4 is provided to solicit output from devices that must be prompted to output information. The AIM4 will process the input from these devices as set up in its input configuration.

The AIM4 can be used to poll connected devices at user defined intervals with separate messages intended for the device attached to each port. The polling feature of the AIM4 was designed to be used in conjunction with print rungs from the processor as described in the Output Operations chapter on page 33, if they are needed.

The periodic polling screen is reached by pressing **9** in either oNline mode or oFfline mode - Edit configuration in memory. The periodic polling screen can be seen in Figure 7-1 the Periodic Polling Screen on page 36. The first field is the polling period and allows input in seconds, tenths and hundreds of seconds. This is the amount of time that the AIM4 waits between messages.

The next 16 fields are the query message fields. Any message may be from 0 to 64 characters in length. Use the arrow keys to move from message to message or from the message area to the polling period field. When finished with the screen, press the **ENTER** key.

To enter a message, arrow to the message field and type the message characters from the keyboard. If a special or control character is needed, press **F9** to recall the HEX editor screen. The HEX editor is described in the Hex Editor section on page 52. Once finished typing a message, either arrow to the next message or hit **ENTER** to leave the periodic polling screen.

Once messages have been entered for a port, the AIM4 will step through the messages starting with query 1 and ending with the last query that has text, sending each out of the designated port. After sending one message, the AIM4 waits the amount of time required by the polling period before sending the next message. When the AIM4 has reached the last message that contains text, it waits only one polling period before starting through the query list again with query 1, no matter how many blank messages remain.

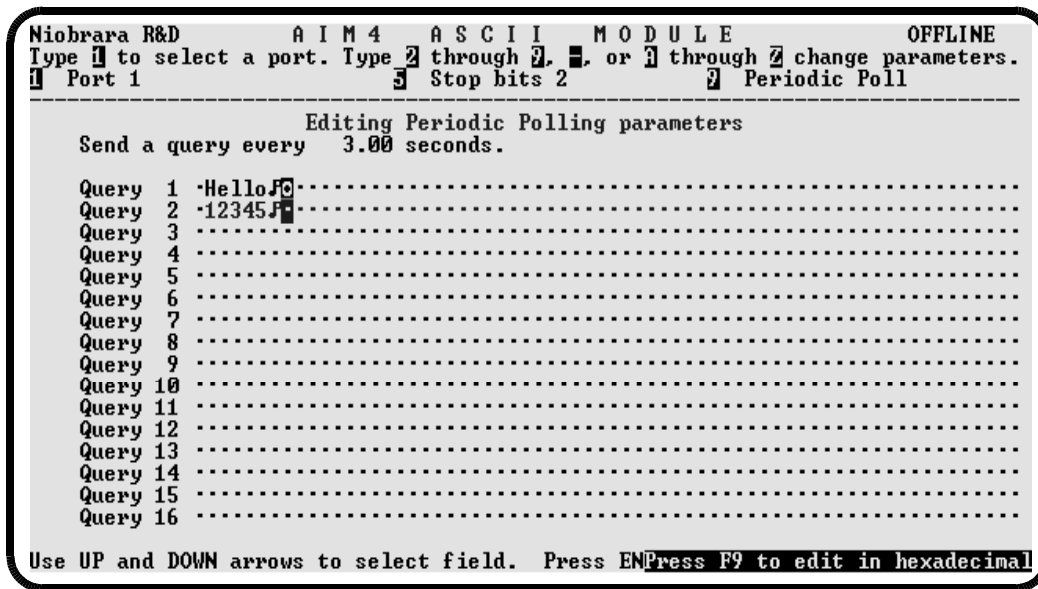


Figure 7-1 Periodic Polling Screen

## Dash 2 PLUS Output Programming

The AIM4-2S, AIM4-2D, and AIM4-2M models have the capability to generate output string under control of rack-addressed registers and therefore without requiring a SY/MAX connection to the processor. Output messages are controlled by programming special character sequences in the Periodic Polling query strings. Output messages can be triggered by the polling timer or, if that timer is set to zero, by toggling bits in a rack-addressable trigger register. If the polling timer is non-zero, this register becomes an inhibit register and can be used to turn polling on or off and to individually inhibit specific queries.

### Output Trigger/Inhibit Register Allocation

The module allocates four trigger/inhibit registers based on the integer value located in registers 7431...7434. The values stored within these registers are the register numbers of the trigger/inhibit registers as defined by the PLC: as specified in the rack base module parameter. For instance, a module has been rack addressed the PLC registers 200 to 350, and the Base Address in the module is set to 200. The Trigger/Inhibit registers to be used in the ladder program are 347, 348, 349, and 350 for ports 1, 2, 3, and 4 respectively. The values stored in the module registers 7431, 7432, 7433, and 7434 would be 347, 348, 349, and 350.

It is a good idea to place these registers at the end of the group of rack addressed registers. The PLC updates the rack sequentially from the lowest assigned register to the highest. The AIM4 will trigger on the data in the Trigger/Inhibit register almost immediately upon the PLC's writing of that register. If the Trigger/Inhibit register is located near the end of the assigned group of registers, all the output registers that may be used in the associated Periodic Poll queries will be written by the PLC before they are triggered. Likewise, if the Trigger/Inhibit registers are near the beginning of the assigned registers, the AIM4 may be triggered and start outputting strings before the PLC is able to write new data to the rest of the module on that scan.

**Note:** It is not necessary for each port to have an independent trigger/inhibit register. Two or more ports may share the same register allowing one condition to govern the output of several ports. In fact the trigger/inhibit register may be assigned to the signaling register of the module (the first register assigned to the module). This way a certain input condition on a particular path on a port could trigger a corresponding query on that or another port without PLC intervention.

## Register Output Substitution Sequences

All register output substitution sequences begin with the @ character. Several formats are available depending on the substitution sequence selected. The possible substitution sequences are as follows:

**Table 7-1 Output Substitution Sequences**

Format	Substitution Sequence
Signed Decimal	@rrrrnnD
Zero Padded Signed Decimal	@rrrrnnE
Unsigned Decimal	@rrrrnnU
Zero Padded Unsigned Decimal	@rrrrnnT
Raw ASCII	@rrrrnnR
Hex	@rrrrnnH
Octal	@rrrrnnO
Unsigned BCD	@rrrrnnB
Floating Point	@rrrrnmF
Zero Padded Floating Point	@rrrrnmG
Literal @ Character	@@

The terminating letter must be upper case. An @ sign which is not part of a valid substitution sequence (is not followed by 6 decimal digits and D, E, U, T, R, H, O, B, F, or G is output as a literal. Although it is unlikely that a poll query not intended to be a register substitution sequence matches a valid substitution sequence format, you should always use the @ sequence to output a single @ character. Note that to output a double @, you would need to include four @ characters in the query. Conversion of @@ to @@@@ is probably the only change required to make an existing AIM4 configuration compatible with the -2 model.

In the substitution sequences enumerated above, the rrrr placeholder is to be replaced by the four digit decimal number of the register whose value is to be substituted. The register number is that of the processor, as specified in the rack base module parameter. If the register number is less than 1000, use leading zeroes to form a four digit number.

For all formats except floating point, the nn placeholder is to be replaced by a two digit decimal number specifying the field width of the substitution. This is the number of characters that will be output for the substitution sequence. In the signed, unsigned, BCD and float formats, the number will be right justified and padded to the specified field width with blanks. In the hex, octal, and zero padded formats, the number will be right justified and padded with zeroes. If the number would be longer than nn characters, the field is extended. Therefore, a field width of 01 can be used when no padding is desired.

In floating point format, the field width has two parts, n and m. the first digit, n specifies the number of places before the decimal point and the second digit (m) specifies the number of places after the point. The field width will be the sum of the n plus m plus one for the point. If the integer part of the number should be longer than n digits, the field is extended. Therefore, a field width of 12 can be used when no leading blanks are desired.

In the RAW ASCII format, the output will be taken from  $nn/2$  registers beginning with register  $rrrr$ . If  $nn$  is an odd number, the last character comes from the most significant byte (bits 9 through 16) of the last register.

## Register Substitution Example

Assume that register 140 of the PLC holds the value 23 (decimal) and register 141 holds the value 640.

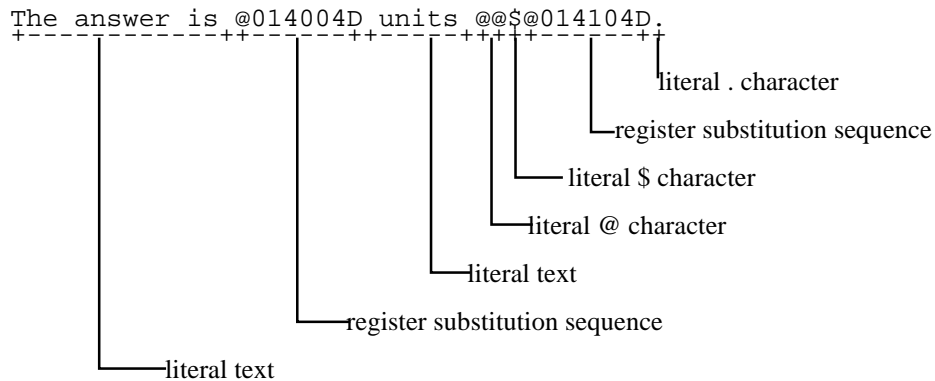
The poll query:

The answer is @014004D units @\$@014104D.

will output:

The answer is 43 units @\$ 640.

The structure of the sample poll is as described below.



## Redirected Port Output

Certain applications require more than the 16 queries available to a single serial port. AIM4 PLUS modules with firmware 15Jun95 or later have the ability use poll queries from one port to be sent out a different physical port. Each port retains its own Trigger/Inhibit register and the bits in these registers are used to trigger the queries. Each port may be redirected to only one other port, but it is possible to redirect multiple ports to a single ports.

The redirection of the ports is controlled by bits 9 through 14 in the Options register R[7435] of the PLUS module. Table 7-2 displays the bit definitions of the Options register. (Notice that bits 1-8 are used on the -2M module only.)

This feature is only applicable for Triggered queries. If the Periodic Poll is enabled on the redirected port, the poll strings will be sent out of the physical port (like normal). Print Demultiplexed strings directed to a port will be sent out that physical port.

**For Example:** Consider a printer connected to Port 1 of an AIM4-2S using an SP9Y splitter cable. (This splitter allows a separate input device to also be connected to Port 1, such as a barcode reader.) The printer application requires a large number of possible strings, say 45 queries, some including PLC register data. Fortunately, Ports 2 and 3 have input only devices connected and are not using their Periodic Poll tables. By setting Register 7435 bits 9 and 10, it is possible to use the queries from Ports 2 and 3 in addition to the queries from Port 1.

Each port will have its own Trigger/Inhibit register, say 150, 151, and 152 for Ports 1, 2, and 3. (All Periodic Poll timers must be set to zero.) When register 150 bit 1 (denoted as R[150].1) is toggled query 1 from Port 1 will be sent out of Port 1. When R[151].4 is toggled, query 4 from Port 2 will be sent out Port 1. When R[152].13 is toggled, query 13 from Port 3 will be sent out Port 1 to the printer.

**Table 7-2 Options Register 7435**

Bit	Function
1	-2M Port 1 Half Duplex when set.
2	-2M Port 2 Half Duplex when set.
3	-2M Port 3 Half Duplex when set.
4	-2M Port 4 Half Duplex when set.
5	-2M Port 1 Tri-state transmit drivers when set.
6	-2M Port 2 Tri-state transmit drivers when set.
7	-2M Port 3 Tri-state transmit drivers when set.
8	-2M Port 4 Tri-state transmit drivers when set.
9	Port 2 Queries sent out Port 1 when set.
10	Port 3 Queries sent out Port 1 when set.
11	Port 4 Queries sent out Port 1 when set.
12	Port 3 Queries sent out Port 2 when set. (Only if bit 10 is clear)
13	Port 4 Queries sent out Port 2 when set. (Only if bit 11 is clear)
14	Port 4 Queries sent out Port 3 when set. (Only if bits 11 and 13 are clear)
15	Reserved
16	Reserved

### Additional Output Comments

It should be noted that any PLC register number used in an AIM4 register output substitution sequence should be assigned to the AIM4 by the PLC. Also those registers used for output must not be the same as a register that is assigned by the module (itself or by the user) as a PLC input. Selecting the register numbers to be used for output substitution should only be done after all input parameters have been determined.





## AIM4-2M Operation

The AIM4-2M module provides the ability to connect to multidrop RS-245 networks. The Options register R[7345] provides for control of the RS-485 transmitter and receiver for each of the 4 ports. The available parameters to be adjusted for each port includes Full and Half duplex on the receiver and Multidrop and Point-to-Point on the transmitter. The AIM4-2M defaults to Full Duplex, Point-to-Point.

**Table 8-1 Options Register 7435**

Bit	Function
1	-2M Port 1 Half Duplex when set.
2	-2M Port 2 Half Duplex when set.
3	-2M Port 3 Half Duplex when set.
4	-2M Port 4 Half Duplex when set.
5	-2M Port 1 Tri-state transmit drivers when set. (Multidrop)
6	-2M Port 2 Tri-state transmit drivers when set. (Multidrop)
7	-2M Port 3 Tri-state transmit drivers when set. (Multidrop)
8	-2M Port 4 Tri-state transmit drivers when set. (Multidrop)
9	Port 2 Queries sent out Port 1 when set.
10	Port 3 Queries sent out Port 1 when set.
11	Port 4 Queries sent out Port 1 when set.
12	Port 3 Queries sent out Port 2 when set. (Only if bit 10 is clear)
13	Port 4 Queries sent out Port 2 when set. (Only if bit 11 is clear)
14	Port 4 Queries sent out Port 3 when set. (Only if bits 11 and 13 are clear)
15	Reserved
16	Reserved

### Full Duplex

When the Options bit for a port is cleared to allow full duplex operation, that means that the port's receiver is enabled at all times. This allows the AIM4 data port to be able to listen even while it is transmitting. This setting is normally used on 4-wire installations.

## Half Duplex

When the Options bit for a port is set to force half duplex operation, that means that the port's receiver is disabled while the port is transmitting. This operation is useful for 2-wire networks because the port's transmit and receive lines are tied together.

## Multidrop

The Options bit is set for a port for Multidrop operation, the state of the transmit driver is determined by the CTS input. When CTS is low (the port is not transmitting), the transmit driver is tri-stated and allowed to float at a high impedance. This allows other transmitters on the same network the ability to transmit. When CTS is driven high, the transmit driver is enabled to allow that port to transmit. This feature is used when the AIM4 port is on a 2-wire network or is acting as a slave on a 4-wire network.

NOTE: It is critical to jumper RTS to CTS (pins 5-6 and 7-8) when the Multidrop option is used. Otherwise, the port will never tri-state and the network will stop.

## Point-to-Point

When the Multidrop Options bit is cleared, the transmit driver is always enabled. This is the normal state in an RS-422 port like those on the AIM4-xD modules. This option is used when the AIM4 port is to be a 4-wire RS-485 master, or when connected to other RS-422 devices.

## SY/MAX Operations

The Print/Prog port of the AIM4 can accept read, write, and print commands formatted according to the SY/MAX protocol. Therefore, an attached SY/MAX device such as a PLC can access the module's rack addressed memory by means of READ or WRITE rungs, can modify the module's configuration parameters dynamically in real time, and can print using a SY/MAX network print message as opposed to a raw print. By modifying the configuration parameters with WRITE rungs, a PLC can change any configuration parameter shown by AIM4SW. These parameters and their processor equivalent register numbers are shown on page 44. One application of dynamic configuration is turning off periodic polling by setting a port's poll interval to 0.

In addition to the configuration parameters, the AIM4's data registers may be accessed through the Print/Prog port. Processor equivalent registers 1 through 2048 are the module's rack addressable data registers. Register 1 is the signaling register, which the PLC reads through the rack as the first register in the module. Note that the data registers may not have the same address through the port that they do through the rack. Data registers need not be rack addressed to be accessed through the Print/Prog port.

Reading the AIM4 data registers through the port allows a SY/MAX device to use the AIM4 without being the controlling PLC in the AIM4 rack and without using one of the PLC's communication ports. Writing the AIM4 data registers can be used to transfer data from an external SY/MAX device to the rack's PLC CPU also without using one of the PLC's communication ports. Because the PLC cannot write the AIM4 data registers through the rack, this technique cannot be used to transfer data from the rack PLC to an external SY/MAX device. (This is of course not true for the PLUS versions.)

Register direction is controlled by the AIM4. On standard units, all rack addressable registers are PLC inputs. On PLUS units, all rack addressed registers are PLC outputs until one of the following occurs:

- A register is configured by means of the start register and register count parameters.
- A register is written to by an external SY/MAX device through the Print/Prog port.

Register 1 is always a PLC input, regardless of the modules configuration. Register direction is reset whenever the power is cycled on the module, or when a start register or register count setting is modified. The direction of a register may be determined by the value of the status register associated with that register. A status of A000 hex means that register is a PLC output. A status of E000 hex means that that register is a PLC input.

### AIM-4 Processor Equivalent Registers

Function	Registers	Value
Signaling register	1	1 bit for each of 16 paths
Rack addressable data registers	2..2048	configuration dependent
Port 1 Setup Registers	2049..3391	
Baud rate	2049	0..15 (see table)
Data bits	2050	0=7 bits, 1=8 bits
Parity	2051	0=none, 1=odd, 2=even
Stop bits	2052	0=1 stop, 1=2 stop
Capitalize	2053	1=yes, 0=no
Accept set	2054..2069	1 bit for each character
Terminate set	2070..2085	1 bit for each character
Path 1	2086..2155	
Pattern	2086..2118	two characters per register
Mask	2119..2151	two characters per register
Start Register	2152	1..8192
Register Count	2153	0..64
Edit Mode	2154	0..7 (see table)
Continue	2155	1=yes, 0=no
Path 2	2156..2225	
Pattern	2156..2188	two characters per register
Mask	2189..2221	two characters per register
Start Register	2222	1..8192
Register Count	2223	0..64
Edit Mode	2224	0..7 (see table)
Continue	2225	1=yes, 0=no
Path 3	2226..2295	
Pattern	2226..2258	two characters per register
Mask	2259..2291	two characters per register
Start Register	2292	1..8192
Register Count	2293	0..64
Edit Mode	2294	0..7 (see table)
Continue	2295	1=yes, 0=no
Path 4	2296..2364	
Pattern	2296..2328	two characters per register
Mask	2329..2361	two characters per register
Start Register	2362	1..8192
Register Count	2363	0..64
Edit Mode	2364	0..7 (see table)
Poll Query 1	2366..2429	two characters per register
Poll Query 2	2430..2493	
Poll Query 3	2494..2557	
Poll Query 4	2558..2621	
Poll Query 5	2622..2685	
Poll Query 6	2686..2749	
Poll Query 7	2750..2813	
Poll Query 8	2814..2877	
Poll Query 9	2878..2941	
Poll Query 10	2942..3005	
Poll Query 11	3006..3069	
Poll Query 12	3070..3133	
Poll Query 13	3134..3197	
Poll Query 14	3198..3261	
Poll Query 15	3262..3325	
Poll Query 16	3326..3389	
Poll Interval	3390..3391	100ths of a second, MSW, LSW

<u>Function</u>	<u>Registers</u>	<u>Values</u>
Port 2 Setup Registers	3392..4734	
Baud rate	3392	0..15 (see table)
Data bits	3393	0=7 bits, 1=8 bits
Parity	3394	0=none, 1=odd, 2=even
Stop bits	3395	0=1 stop, 1=2 stop
Capitalize	3396	1=yes, 0=no
Accept set	3397..3412	1 bit for each character
Terminate set	3413..3428	1 bit for each character
Path 1	3429..3498	
Pattern	3429..3461	two characters per register
Mask	3462..3494	two characters per register
Start Register	3495	1..8192
Register Count	3496	0..64
Edit Mode	3497	0..7 (see table)
Continue	3498	1=yes, 0=no
Path 2	3499..3568	
Pattern	3499..3531	two characters per register
Mask	3532..3564	two characters per register
Start Register	3565	1..8192
Register Count	3566	0..64
Edit Mode	3567	0..7 (see table)
Continue	3568	1=yes, 0=no
Path 3	3569..3638	
Pattern	3569..3601	two characters per register
Mask	3602..3634	two characters per register
Start Register	3635	1..8192
Register Count	3636	0..64
Edit Mode	3637	0..7 (see table)
Continue	3638	1=yes, 0=no
Path 4	3639..3707	
Pattern	3639..3671	two characters per register
Mask	3672..3704	two characters per register
Start Register	3705	1..8192
Register Count	3706	0..64
Edit Mode	3707	0..7 (see table)
Poll Query 1	3709..3772	two characters per register
Poll Query 2	3773..3836	
Poll Query 3	3837..3900	
Poll Query 4	3901..3964	
Poll Query 5	3965..4028	
Poll Query 6	4029..4092	
Poll Query 7	4093..4156	
Poll Query 8	4157..4220	
Poll Query 9	4221..4284	
Poll Query 10	4285..4348	
Poll Query 11	4349..4412	
Poll Query 12	4413..4476	
Poll Query 13	4477..4540	
Poll Query 14	4541..4604	
Poll Query 15	4605..4668	
Poll Query 16	4669..4732	
Poll Interval	4733..4734	100ths of a second, MSW, LSW

<u>Function</u>	<u>Registers</u>	<u>Value</u>
Port 3 Setup Registers	4735..6077	
Baud rate	4735	0..15 (see table)
Data bits	4736	0=7 bits, 1=8 bits
Parity	4737	0=none, 1=odd, 2=even
Stop bits	4738	0=1 stop, 1=2 stop
Capitalize	4739	1=yes, 0=no
Accept set	4740..4755	1 bit for each character
Terminate set	4756..4771	1 bit for each character
Path 1	4772..4841	
Pattern	4772..4804	two characters per register
Mask	4805..4837	two characters per register
Start Register	4838	1..8192
Register Count	4839	0..64
Edit Mode	4840	0..7 (see table)
Continue	4841	1=yes, 0=no
Path 2	4842..4911	
Pattern	4842..4874	two characters per register
Mask	4875..4907	two characters per register
Start Register	4908	1..8192
Register Count	4909	0..64
Edit Mode	4910	0..7 (see table)
Continue	4911	1=yes, 0=no
Path 3	4912..4981	
Pattern	4912..4944	two characters per register
Mask	4945..4977	two characters per register
Start Register	4978	1..8192
Register Count	4979	0..64
Edit Mode	4980	0..7 (see table)
Continue	4981	1=yes, 0=no
Path 4	4982..5050	
Pattern	4982..5014	two characters per register
Mask	5015..5047	two characters per register
Start Register	5048	1..8192
Register Count	5049	0..64
Edit Mode	5050	0..7 (see table)
Poll Query 1	5052..5115	two characters per register
Poll Query 2	5116..5179	
Poll Query 3	5180..5243	
Poll Query 4	5244..5307	
Poll Query 5	5308..5371	
Poll Query 6	5372..5435	
Poll Query 7	5436..5499	
Poll Query 8	5500..5563	
Poll Query 9	5564..5627	
Poll Query 10	5628..5691	
Poll Query 11	5692..5755	
Poll Query 12	5756..5819	
Poll Query 13	5820..5883	
Poll Query 14	5884..5947	
Poll Query 15	5948..6011	
Poll Query 16	6012..6075	
Poll Interval	6076..6077	100ths of a second, MSW, LSW

Function	Registers	Value
Port 4 Setup Registers	6078..7420	
Baud rate	6078	0..15 (see table)
Data bits	6079	0=7 bits, 1=8 bits
Parity	6080	0=none, 1=odd, 2=even
Stop bits	6081	0=1 stop, 1=2 stop
Capitalize	6082	1=yes, 0=no
Accept set	6083..6098	1 bit for each character
Terminate set	6099..6114	1 bit for each character
Path 1	6115..6184	
Pattern	6115..6147	two characters per register
Mask	6148..6180	two characters per register
Start Register	6181	1..8192
Register Count	6182	0..64
Edit Mode	6183	0..7 (see table)
Continue	6184	1=yes, 0=no
Path 2	6185..6254	
Pattern	6185..6217	two characters per register
Mask	6218..6250	two characters per register
Start Register	6251	1..8192
Register Count	6252	0..64
Edit Mode	6253	0..7 (see table)
Continue	6254	1=yes, 0=no
Path 3	6255..6324	
Pattern	6255..6287	two characters per register
Mask	6288..6320	two characters per register
Start Register	6321	1..8192
Register Count	6322	0..64
Edit Mode	6323	0..7 (see table)
Continue	6324	1=yes, 0=no
Path 4	6325..6393	
Pattern	6325..6357	two characters per register
Mask	6358..6390	two characters per register
Start Register	6391	1..8192
Register Count	6392	0..64
Edit Mode	6393	0..7 (see table)
Poll Query 1	6395..6458	two characters per register
Poll Query 2	6459..6522	
Poll Query 3	6523..6586	
Poll Query 4	6587..6650	
Poll Query 5	6651..6714	
Poll Query 6	6715..6778	
Poll Query 7	6779..6842	
Poll Query 8	6843..6906	
Poll Query 9	6907..6970	
Poll Query 10	6971..7034	
Poll Query 11	7035..7098	
Poll Query 12	7099..7162	
Poll Query 13	7163..7226	
Poll Query 14	7227..7290	
Poll Query 15	7291..7354	
Poll Query 16	7355..7418	
Poll Interval	7419..7420	100ths of a second, MSW, LSW

Baud rate of Print/Prog port	7421	0..3 (see table)
Rack address start register	7422	First Addr in PLC rack table

## AIM4-2 Additional Setup Registers

Port 1 Max Count Register	7423	Number of characters
Port 2 Max Count Register	7424	Number of characters
Port 3 Max Count Register	7425	Number of characters
Port 4 Max Count Register	7426	Number of characters
Port 1 Max Silence Register	7427	100ths of a second
Port 2 Max Silence Register	7428	100ths of a second
Port 3 Max Silence Register	7429	100ths of a second
Port 4 Max Silence Register	7430	100ths of a second
Port 1 Signal/Inhibit Register	7431	1..2048
Port 2 Signal/Inhibit Register	7432	1..2048
Port 3 Signal/Inhibit Register	7433	1..2048
Port 4 Signal/Inhibit Register	7434	1..2048
Options Register	7435	Dependent Upon Config.

### AIM4 Setup Register Coded Values

#### Data port baud rate

Register	Baud	Register	Baud
0	50	8	1800
1	75	9	2400
2	110	10	3600
3	134.5	11	4800
4	150	12	7200
5	300	13	9600
6	600	14	14400
7	1200	15	19200

#### Data port edit mode

Register	Mode	Register	Mode
0	ASCII	4	BCD
1	Packed	5	Hexadecimal
2	Integer	6	Octal
3	Float	7	IDEC

#### Print/Prog port baud rate

Register	Baud	Register	Baud
0	300	2	2400
1	1200	3	9600

## Accept and Terminate sets

The accept and terminate character sets are each coded in 16 registers making up a total of 16 \* 16 or 256 bits in each set. Each bit corresponds to one of the possible 8 bit character values 0 through 255. A bit is set if the corresponding character is in the set and clear if the character is not in the character set. The most significant byte of a register is coded before the least significant byte and the least significant bit of a byte is coded before the most significant bit of that byte. Therefore, the character whose code is 0 (ASCII NUL) is represented by bit 9 of the first register in a character set and character 255 (ASCII DEL with high bit) is represented by bit 8 of the last register in the character set. To further illustrate: Bit 1 of the first register represents character code 8. Bit 16 of the first register represents character



code 7. Bit 1 of the last register represents character code 248. And bit 16 of the last register represents character code 247.

## Patterns, Masks, and Queries

These three parameter classes are represented by strings of characters in a contiguous group of registers. Each register holds two characters with the first character in the most significant byte and the second character in the least significant half. This corresponds to the AIM4 Packed edit mode. Each pattern, mask, or query is terminated by an ASCII NUL (all bits 0) character.

## Poll Interval

The poll interval for each port is the time between polls expressed in hundredths of a second. Therefore a value of 100 would cause a one second delay between successive polls. To increase the range of the value beyond 655.35 seconds, it is represented as a 32 bit value in two registers. The lower numbered register is most significant.

For example, a poll interval on port 1 of 1 second calls for a value of 0 in register 3390 and a value of 100 in register 3391. A poll interval of an hour (360000 1/100ths of a second) would call for a value of 5 in register 3390 and 32,320 in register 3391

$(5 * 65536 + 32320 = 360000)$ .

## Start Registers and the Rack address start

The register number stored in each path's Start Register parameter is a PLC register number. The Rack address start register (7422) must be correctly set to the register number of the first register assigned to the module in the PLC's rack address table. The AIM4 internally subtracts this rack base address from the specified Start Register number to determine where to write data values within the AIM4's rack image memory.

## Data Registers

The AIM-4's rack-addressed data registers can be read and written through the Print/Prog port. When addressed in this manner, the signaling register number is always in the range 1 through 2048 regardless of the setting of the rack base address. The signaling register is always found at register 1. To compute the read/write register number of a register from the Start Register programmed in a path, subtract the rack base address from the Start Register and add one to the result. For example, if the AIM-4 is rack addressed at PLC address 100, an external device connected to the Print/Prog port could write PLC register 104 by writing to AIM-4 data register number 5 ( $104 - 100 + 1$ ).

Writing the data registers through the Print/Prog port can be used to communicate data to the PLC from an external device such as an operator interface. Reading the data registers through the Print/Prog port is only useful to obtain the results of AIM-4 inputs because the PLC processor cannot write to the AIM-4 data registers through the rack (these registers are PLC inputs).

## Max Count Message Termination

In addition to the terminate character set provided on all AIM4 models, the AIM4-S2 and AIM4-D2 provide the ability to terminate a message and begin processing based on a specified message length. The length in characters is written to the appropriate register from the table below. Set this parameter to zero (default) to disable maximum count termination. This method is not highly recommended due to the fact that synchronization is easily lost but may be useful in conjunction with maximum silence termination and implemented to resynchronize in the event of a dropped or extra character.

**Table 9-1 Max Count Registers**

Port	Max Count Register
1	7423
2	7424
3	7425
4	7426

## Maximum Silence Message Termination

A message may be terminated by a certain time expiring between characters. The time in hundredths of a second (i.e. the value 100 specifies one second) is written to the appropriate register from the table below. Set this value to zero (default) to disable maximum silence termination.

**Table 9-2 Max Silence Registers**

Port	Max Silence Register
1	7427
2	7428
3	7429
4	7430

## Options Register Values

AIM4 PLUS modules contain an Options register to control the RS-485 line drivers and receivers on the -2M modules and to redirect the triggered periodic poll queries out other physical ports.

**Table 9-3 Options Register 7435**

Bit	Function
1	-2M Port 1 Half Duplex when set.
2	-2M Port 2 Half Duplex when set.
3	-2M Port 3 Half Duplex when set.
4	-2M Port 4 Half Duplex when set.
5	-2M Port 1 Tri-state transmit drivers when set.
6	-2M Port 2 Tri-state transmit drivers when set.
7	-2M Port 3 Tri-state transmit drivers when set.
8	-2M Port 4 Tri-state transmit drivers when set.
9	Port 2 Queries sent out Port 1 when set.
10	Port 3 Queries sent out Port 1 when set.
11	Port 4 Queries sent out Port 1 when set.
12	Port 3 Queries sent out Port 2 when set. (Only if bit 10 is clear)
13	Port 4 Queries sent out Port 2 when set. (Only if bit 11 is clear)
14	Port 4 Queries sent out Port 3 when set. (Only if bits 11 and 13 are clear)
15	Reserved
16	Reserved

## Configuration Software AIM4SW

### AIM4SW

The AIM4SW software program is provided free of charge to AIM4 users. This software is used to configure the operational parameters of the AIM4 as described in the chapter called Configuring the Port Parameters on page 23. AIM4SW can be operated in either online or offline modes or as an emulator of AIM4 operation.

NOTE: Versions prior to 08Oct93 communicate with the AIM4 using a proprietary ASCII protocol and do not fully service *PLUS* modules. Software versions 08Oct93 and later use the SY/MAX protocol for the configuration; thus allowing remote service of the AIM4 through SY/NET, and other SY/MAX devices such as Niobrara's 2496X modems. Full support for the SY/LINK Network Interface Card is now also provided.

In online mode, AIM4SW presents the user with a screen of data describing the operation of a single port. Keystrokes are typed to select the port being displayed and to change the parameters of that port. Changes made in the online mode are transferred to a connected AIM4 module and are effective immediately.

In offline mode, AIM4SW allows a copy of the configuration to be uploaded from the AIM4 to an image in offline memory. The offline copy can be edited without affecting the AIM4 in real time. The offline copy can be saved to a named disk file for later retrieval, or listed on a hard copy printer. An AIM4 need not be connected to perform most offline operations. The offline copy of the configuration parameters may be downloaded to an AIM4 module at any time.

Because the AIM4 has an internal nonvolatile parameter memory, the configuration parameters normally will not need to be reloaded unless a change has been made.

As an emulator, AIM4SW simulates the input operation of a single port of the AIM4. Input comes from the PC keyboard and resulting register values are displayed on the screen. The AIM4 emulator is a useful tool for experimenting with the effects of different port and path parameter configurations. The emulator is entered from the main menu. The configuration of the emulated port can be loaded from offline memory or fetched from a connected AIM4 module.

The startup screen of AIM4SW is shown in Figure <Startup screen> the <Startup screen> on page <Startup screen>. The operational modes are selected by the highlighted menu bar on the fourth line. Selection can be made by moving the cursor to the desired option using the arrow keys and pressing **ENTER**. A short cut is provided, simply type "N" for online, "F" for offline, "E" for emulate, "S" for setup or "Q" to quit.

If offline mode is selected, a second menu appears. This menu presents the available functions for manipulating the offline image of the configuration parameters and for transfer of configuration data to and from the AIM4 module. Use the up and down arrows to move the highlight, **ENTER** to select the highlighted choice, or the initial capital letter to quickly select an option.

The online and offline parameter editing screens are nearly identical. The difference is in operation. Every time a change is made in the online mode, the modified parameter is transmitted to the AIM4. In offline mode, changes are made to the parameter image in offline memory. Changes made in the online screen affect only the configuration stored in the attached AIM4, not the offline copy. Likewise, changes made in offline mode have no effect on a connected AIM4 until they are explicitly sent to it using the "Send memory to AIM4" function.

AIM4SW also contains two convenient utilities for general use, a SY/MAX Register Viewer and a Terminal Emulator. The Register Viewer uses the same setup as the oNline and oFfline functions. The Terminal Emulator has its own setup characteristics.

## Data Entry Keys

Whenever data entry is allowed by the program, certain keys can be used to facilitate data entry. They are:

BACKSPACE	Move cursor left and remove character there
LEFT ARROW	Move cursor to the left one character
RIGHT ARROW	Move cursor to the right one character
DEL	Remove the character under the cursor
INS	Change between insert and overstrike modes of entry
HOME	Move cursor to the left edge of the field
END	Move cursor to the end of the data
Control-F	Move cursor right (Forward) one word
Control-R	Move cursor left (Reverse) one word
Control-D	Delete from the cursor to the end of the field
Control-U	Delete from cursor to the beginning of the field
Control-Y	Delete all characters in the field
ESC	Exit the field without modifying it
ENTER	Accept the contents of the field

When a field is opened for input, the cursor is positioned at the left side of the field. If data is already present in the field, typing any character other than those listed above will cause the field to be blanked allowing entry of new data without first deleting the old. If it is desired to retain the previous data for editing, make sure the first key you type is an editing key such as a left or right arrow.

## Hex Editor

Whenever character entry is allowed by the program, a hexadecimal editor is available for inputting characters not available from the keyboard such as ESC (Hex 1B). The HEX editor may be accessed while inputting messages for output polling if the device being prompted needs special control characters to make it respond. Similarly, the Accept Character and Terminate Character tables are available in HEX as shown in Figure 10-1

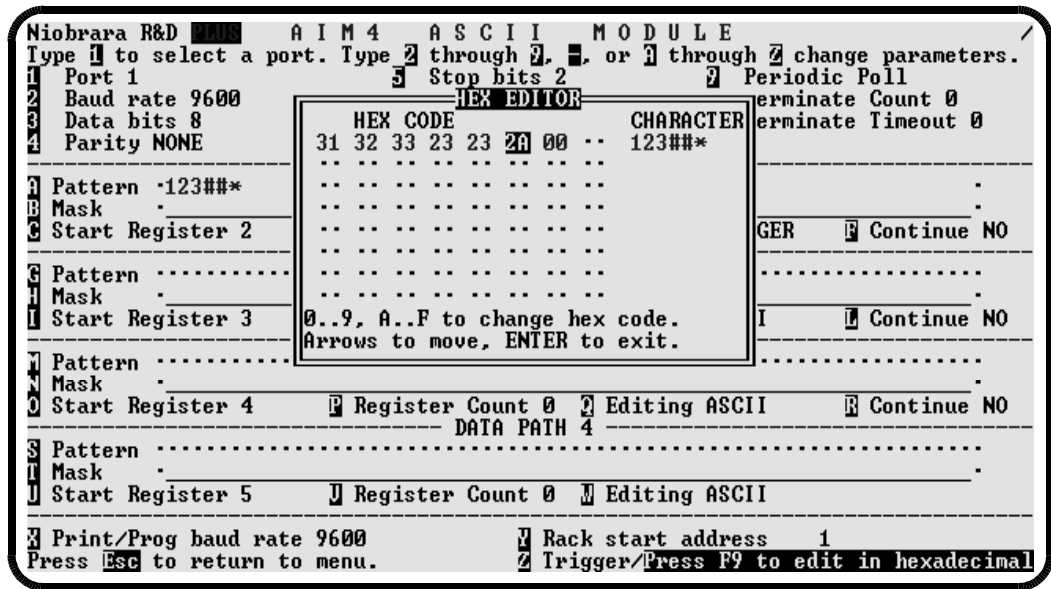


Figure 10-1 Hex Editor screen

Whenever the HEX editor is available, a message appears in the bottom right corner of the screen which says "Press F9 to edit in hexadecimal". Pressing the F9 key on the keyboard will recall the HEX editor screen.

When using the HEX editor, valid key strokes for entering data are 0 through 9 and A through F. The arrow keys move the cursor from character to character and the ENTER key will complete and exit the HEX editing feature. The field currently being edited is highlighted in reverse video (black characters on a white background). The end-of-text marker is 00, which is not used in pattern matching or masking nor, in the case of an output query, transmitted by the AIM4. When characters are typed over the 00, the new characters replace the 00 and the 00 moves to the next available field. Up to 64 characters may be entered in the HEX editor. When 64 characters are used no 00 appears to define the end of text. Attempting to put data into fields past the end of text marker will result in no change in the screen.

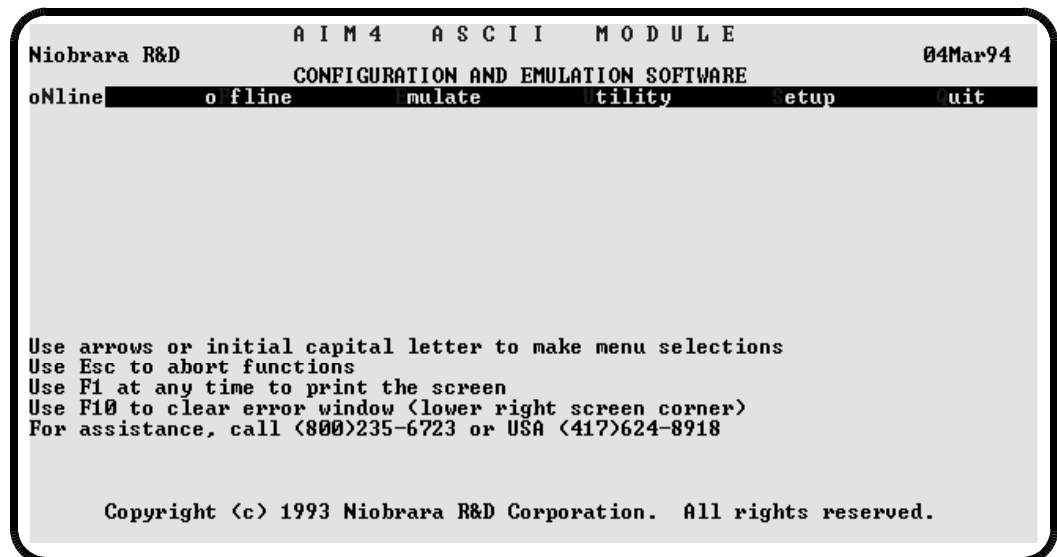


Figure 10-2 AIM4SW Startup Screen

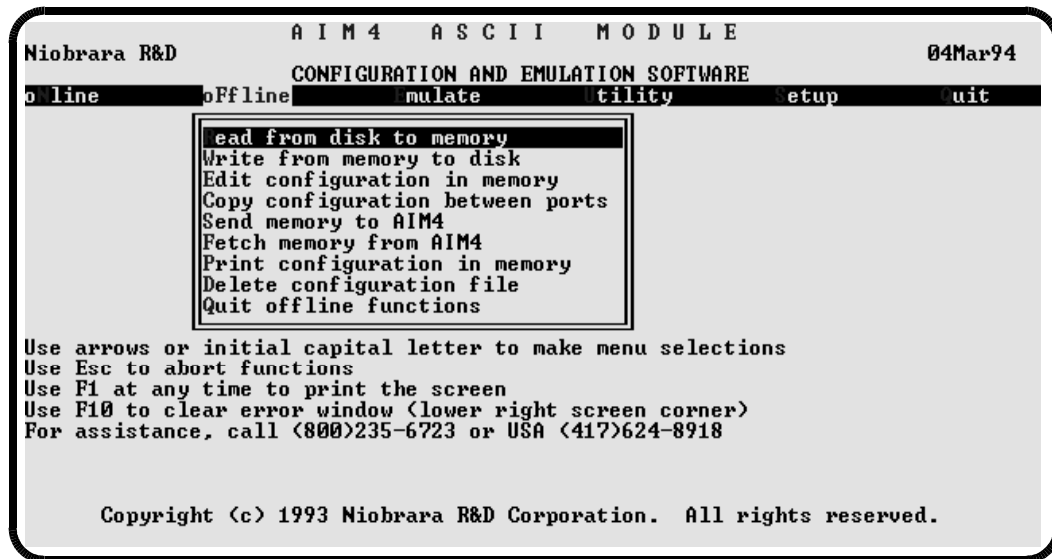


Figure 10-3 AIM4SW oFfline Menu

## Online and Offline Editing Screen

In either the online or offline mode, parameters are changed by typing the key indicated by a high-lighted character just to the left of the parameter to be modified. These keys are 2 through 8 and A through Y. Typing the 1 key selects which of the four groups of port parameters is being displayed and modified. The Print/Prog baud and Rack Base are always displayed and are in common to all ports. These is only one Rack Base for all four ports.

If the Rack Base is modified, AIM4SW will offer to adjust all of the "Start Register" settings by the same amount. This is useful if the rack address of the module must be changed for some reason. If the Rack Base is increased by 100 for example, automatic adjustment of the Start Register parameters will add 100 to each one.

The Pattern and Mask parameters are displayed with small dots to the left and right of the value. This lets you see if there are leading or trailing blanks which would certainly be significant. A Pattern that ends in a blank will only match an input string that ends in a blank. On the offline memory printout, the Pattern and Mask are underscored with ^ characters as an aid to counting and to highlight leading and trailing blanks.

When a Pattern is modified, the program checks to see if it requires characters that are not present in the Accept Character table. If this is the case, a warning is presented.

**NOTE:** The *PLUS* version of the AIM4, (AIM4-2S and AIM4-2D), have additional features that are not supported on the standard AIM4. The AIM4SW software will automatically determine whether the connected AIM4 is a *PLUS* version each time oNline is selected. A highlighted *PLUS* will be presented in the upper left of the display and additional entries will be displayed.

When oFfline editing is selected without first going oNline, the software has no way to determine if the editing is for a *PLUS* or standard unit. Therefore it is assumed that the unit is a *PLUS* and will display the *PLUS* editing screen. If at a later time oNline is selected and the software determines that it is connected to a standard AIM4; further oFfline editing will be done with the standard screens.

It is important to realize that a standard AIM4 unit will ignore data written to setup registers used only in the *PLUS* units. No harm will come from sending a file edited in the *PLUS* screen to a standard AIM4. In fact all files are stored in the *PLUS* format when written to disk.

## Offline Functions

### "Read from disk to memory"

This function reads an AIM4 configuration file into the offline memory. The file should have been created by the "Write from memory to disk" function described below and must have a .AIM extension. When "Read from disk to memory" is selected, a window will open and ask for the name of the file to read. The bottom part of the screen will show a list of all files with the extension .AIM in the current directory. Any subdirectories will be shown in square brackets. Any drive names will also be shown in square brackets, [b: ]. The parent directory (of which the current directory is a subdirectory) is shown by the word "parent" in square brackets. You may either type the name of the file to read or you may use the arrow keys to move the highlight to the desired filename. Pressing **ENTER** with the highlight on a filename will select that file for reading. Pressing **ENTER** with the highlight positioned on a directory (drive) name (either a subdirectory or [parent]) will change the current directory (drive) to that directory (drive) and will show the .AIM files in the new directory (drive). If there are more files than will fit on the screen, pressing the right arrow with the highlight at the right edge of the screen will scroll the display sideways to show more files. Typing the **ESC** key will return to the offline function menu without loading a file or otherwise modifying offline memory.



Figure 10-4 oOffline Read from Disk to Memory

### "Write from memory to disk"

This function saves a copy of the four ports and global configuration parameters from the offline memory to a disk file. "Write from memory to disk" uses the same point and shoot file selection described for "Read from disk to memory" above. To create a new file you must type the name. The name should be a valid MS/DOS filename but should not include any path name or extension. The program will append an extension of .AIM to the name and the file will be placed in the directory which is shown in the bottom half of the screen. To create a file in a directory other than the current one, use the arrow and **ENTER** keys to traverse the directory tree until a listing of the desired directory is shown in the bottom half of the screen. Then type in the file name and press **ENTER**. If you specify (either by typing or by pointing) a file that already exists, you will be prompted for approval before that file is overwritten.

### "Edit configuration in memory"

Just like the online portion of the program, this function displays a screen with the parameters for a single port. As described above, changes here effect the offline memory and not any attached AIM4.

## "Fetch memory from AIM4"

Use this function when you wish to copy the configuration parameters from a connected AIM4 module to the offline memory for editing, printout, or saving on disk. You may copy port setups one at a time or copy all ports in one operation. Fetching any or all ports will also transfer the Print/Prog baud rate and the Rack Base. When the function is invoked, you must specify which AIM4 port (1..4 or All) is to be fetched. Unless you specify "All", you may also choose which port in offline memory is the destination of the transfer which need not be the same port number.

## "Send memory to the AIM4"

This function transfers any one or all four port groups of configuration parameters from offline memory to the AIM4. This function can be used to load configuration back into the AIM4 after it has been Fetched and Edited or if it has been read from a previously saved disk file. Like "Fetch memory from AIM4" you may transfer all ports, or any one port from offline memory to the same or a different port in the AIM4. Any transfer also sets the Print/Prog baud rate and the Rack Base. If you attempt to transfer a Print/Prog baud rate different from the current setting, the program will offer to change the baud rate of the computer to match the new AIM4 setting so the transfer can continue.

## "Print configuration in memory"

This function will produce a report showing the settings of all parameters in offline memory. When this function is selected, you will be prompted for an output filename with the default value of PRN shown. To send the report to the PRN device (normally the parallel printer port), simply press **ENTER**. To send the report to a different port or to a file, type the name and then press **ENTER**. Online configurations may be printed with the **F1** print screen key.

## Emulator

The AIM4 emulator provides a visual simulation of the AIM4's response to data input. The emulator copies a port configuration from offline memory or uploads it from an attached AIM4 module. Input is simulated by accepting characters typed on the computer keyboard. The emulator displays characters as they are entered and accepted. When a message is terminated, the emulator identifies which, if any, Data Path is triggered and which registers are modified. The AIM4SW emulator cannot verify the correctness of the Rack Address assignments to the module nor communicate link parameters such as baud rate, parity, and data bits. Further, the emulator only generates 8 bit no parity ASCII characters with the most significant bit zero. In spite of these limitations, the emulator is a powerful tool for developing and verifying AIM4 configuration setups.

To use the emulator from offline memory, first configure the parameters as desired using the offline editor. Select Emulator from the main menu. You will be prompted for whether the emulated configuration is to come from offline memory or from an online AIM4. Type **F** or **N** or use the arrow and **ENTER** keys. After selecting online or offline, you will be asked to specify which of the four possible ports to emulate as the emulator implements only a single port. Press **ENTER** to select the default of 1 or type **1,2,3**, or **4** and press **ENTER**. When the emulator displays *"Type input characters on keyboard."* at the bottom of the screen, it is ready to accept input. Only printable characters and control codes may be entered. There is no mechanism for typing codes with the most significant bit set. Function key **F10** is used to terminate the simulation. Function key **F1** can be used to obtain a screen printout.



```

Niobrara R&D      A I M 4  A S C I I  M O D U L E      E
Offline port to emulate 1      Type F10 to exit simulator
                               Signal Register <1> = 0000 0000 0000 0001
Received:  -123J-----
Accepted:  -123-----
                               TERMINATOR 0D MATCHED

Capitalize = NO
Pattern 1 MATCHED
Masked:    -123-----
Start Register = 2      Register Count = 1  Edit = INTEGER

Register  Dec  Hex      Dec  Hex      Dec  Hex      Dec  Hex      Dec  Hex      Dec  Hex
   2      123 007B      0 0000      0 0000      0 0000      0 0000      0 0000
   8          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  14          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  20          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  26          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  32          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  38          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  44          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  50          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  56          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  62          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000
  68          0 0000      0 0000      0 0000      0 0000      0 0000      0 0000

                               Ready for more data

```

**Figure 10-5 AIM4 Emulator**

As input characters are typed, they are shown on the line labeled "Received". Characters which match the Accept Character table and are placed in the message buffer are shown on the line labeled "Accepted". In both cases, control characters are displayed as the PC low ASCII graphic character. When a character is entered which is found in the Terminate Character table, the emulator displays the hex value of the terminating character and attempts to match the buffered input message to the four data paths.

On PLUS modules with a terminate character count greater than zero, exceeding this character count will cause the message to terminate. If the inter-character timeout value is set to a number other than zero, pressing F2 will simulate the timeout sequence and cause the message to terminate.

When a Data Path is triggered, the emulator shows the message as modified by the Mask and displays the registers (if any) which will be set by the editing operation. The value of the signal register is displayed in hexadecimal at the upper right corner of the screen.

The emulator displays "Ready for more data" in the lower right corner of the screen when the message has been completely processed. You may begin typing a new message or you can press **F1** to print the screen or **F10** to exit the emulator.

## Utilities

The Utilities menu provides access to useful maintenance and testing functions of the AIM4SW software.

### View registers

Selecting the View registers menu item will invoke a SY/MAX register data viewer/modifier. This viewer continuously performs a block read of 20 registers and displays the contents of those registers in hex, unsigned integer, signed integer, and binary. The status register associated with the data register is also displayed in hex. Like any other oNline or oFfline function involving communication, the register viewer is dependent on the values located in the SETUP Sy/Max menu. Mode, Baud rate, Parity, Route, etc. must be properly set for proper communication.

Niobrara R&D		AIM4 ASCII MODULE							04Mar94
CONFIGURATION AND EMULATION SOFTWARE									
REGSTR	HEX	UNSIGN	SIGNED	BINARY				STAT	
1	0001	1	1	0000	0000	0000	0001	E000	
2	007B	123	123	0000	0000	0111	1011	E000	
3	0000	0	0	0000	0000	0000	0000	A000	
4	0000	0	0	0000	0000	0000	0000	A000	
5	0000	0	0	0000	0000	0000	0000	A000	
6	0000	0	0	0000	0000	0000	0000	E000	
7	0000	0	0	0000	0000	0000	0000	A000	
8	0000	0	0	0000	0000	0000	0000	A000	
9	0000	0	0	0000	0000	0000	0000	A000	
10	0000	0	0	0000	0000	0000	0000	E000	
11	0000	0	0	0000	0000	0000	0000	A000	
12	0000	0	0	0000	0000	0000	0000	A000	
13	0000	0	0	0000	0000	0000	0000	A000	
14	0000	0	0	0000	0000	0000	0000	E000	
15	0000	0	0	0000	0000	0000	0000	A000	
16	0000	0	0	0000	0000	0000	0000	A000	
17	0000	0	0	0000	0000	0000	0000	A000	
18	0000	0	0	0000	0000	0000	0000	A000	
19	0000	0	0	0000	0000	0000	0000	A000	
20	0000	0	0	0000	0000	0000	0000	A000	

AIM4 Sy/Max  
Register Viewer  
  
Press F2 for Help

AIM4 Sy/Max  
Register Viewer  
Press F2 for Help

**Figure 10-6 View Registers**

The Up and Down arrow keys are used to move from register to register.

The Page Up and Page Down keys move in increments of 10 registers.

The Left and Right arrows move from column to column on the same register.

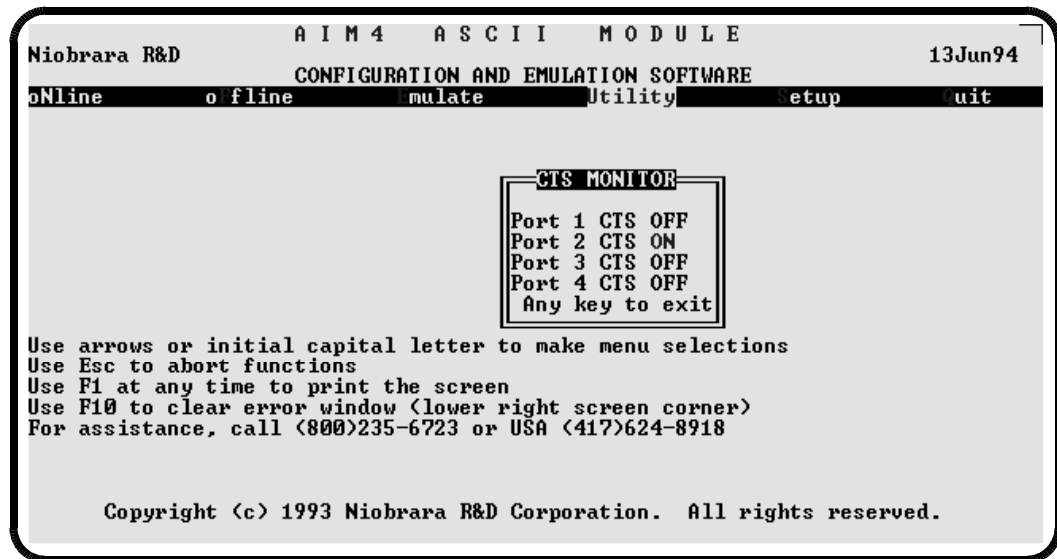
This register viewer is highly useful in that it allows easy editing of the data in the register being viewed. By pressing 0..9 in the decimal fields or 0..9, or A..F in the hex field, an editing mode is entered. New data may be entered at this time. Pressing the Enter key or moving to a new field with the arrow keys will cause the new data to be written to the edited register. If the cursor is located in the REGISTER column the block of registers being viewed may be adjusted by entering a new register number. To edit the binary values, press HOME when on the binary field. Move the cursor to the desired bit and enter a '0' or a '1' and press enter to accept.

Pressing Esc will exit from the Register viewer and return to the main menu. Pressing Esc while editing a data field will result in canceling the edit and the modified data will not be written to the register.

The STAT field displays the status register associated with the data register. The STATUS field is a read only display and can not be modified by the Register Viewer. Two common values are E000 and A000. A000 is the hex representation that the PLC recognizes as a PLC OUTPUT register. E000 is for a PLC INPUT register. This allows easy recognition of registers used by the AIM4 as inputs and used by the PLC as outputs.

## CTS Monitor

Selecting the CTS Monitor from the Utilities menu will invoke a continuous monitoring of the CTS lines of Ports 1 through 4 of the AIM4. When CTS is active, ON will be displayed. When CTS is inactive, OFF will be displayed. See Figure 10-7. CTS must be ON in order for the port to be able to transmit. This is true for Periodic Polling as well as Print demultiplexing operations.



**Figure 10-7 CTS Monitor Screen**

## Terminal Emulator

Selecting the Terminal emulator from the Utilities menu will invoke a terminal emulator according to the setup selected in the Setup menu. The terminal emulator opens as shown in Figure 10-8.

NOTE: The terminal emulator is using COM2:  
Type ctrl-End to return to AIM4SW. Type F7 to begin capture to a file.  
To send hex sequence, type INSERT key, enter hex codes, type ENTER.  
123<0D>  
Hello<0D>  
Now is the time for all good men to come to the aid of their country.<0D>  
<10><11>Enter hex codes to send: 10 15  
  
Capture to file? HELLO.TXT  
Copying output to: "HELLO.TXT". Type F8 to end capture.  
Hello<0D>  
  
Capture file: "HELLO.TXT" closed.  
456<0D>  
█

**Figure 10-8 Terminal Emulator**

The terminal sends the ASCII code for the alpha-numeric characters out the selected COM port. Functions keys F1 through F4 and the keypad arrows send ANSI (i.e. VT100) codes. F7 is reserved for starting a file capture. F8 will close the capture file. The backspace key sends ASCII BS (08 hex). The Delete key sends and ASCII DEL (7F hex). The Insert key allows the transmission of ASCII hexadecimal characters directly from the hex numbers separated by spaces.

The terminal displays printable ASCII characters which are received on the port. Non-printable characters are displayed as the hexadecimal value enclosed in <>, such as <0D><0A> indicates the carriage return, line feed characters. The terminal is always in this "monitor" mode and therefore ANSI output emulation is not provided.

# SETUP

The setup menu accesses the setup parameters for the personal computer to enable it to communicate with the AIM4 and the terminal emulator. The parameters chosen will depend on the exact equipment involved in making the connections.

## SY/MAX SETUP

The connection type is mainly determined by the method of connection to outside world and may be broken into two groups: the personal computer's COM: port and the SY/LINK Network Interface Card.

### Personal Computer COM: port

If the connection from the personal computer is made through one of its serial ports then the Connection type should be one of the following:

- Sy/Max COM:
- Net-to-Net COM:
- Gateway COM:

Sy/Max COM: is the default and most likely will be the one used. In this mode the personal computer will communicate through one of its COM: ports as though it were a SY/MAX device such as a PLC. The full SY/MAX protocol is supported including routing so SY/MAX COM: may be used through SY/MAX mode ports on NIMs and SPE4s with appropriate routing. This mode is to be used when a direct connection from the personal computer COM: port is made to the Print/Prog port of the AIM4. In most cases an RS-232↔RS-422 conversion is required and the Niobrara SC406 cable makes this conversion very convenient.

Net-to-Net COM: is used when connecting to a NIM or SPE4 that is set to Net-to-Net mode. The first drop number in the route will be that of the address of the NIM.

Gateway COM: is used when connecting to an SPE4 port that is in Gateway mode. For more information about Gateway mode see the SPE4 instruction manual.

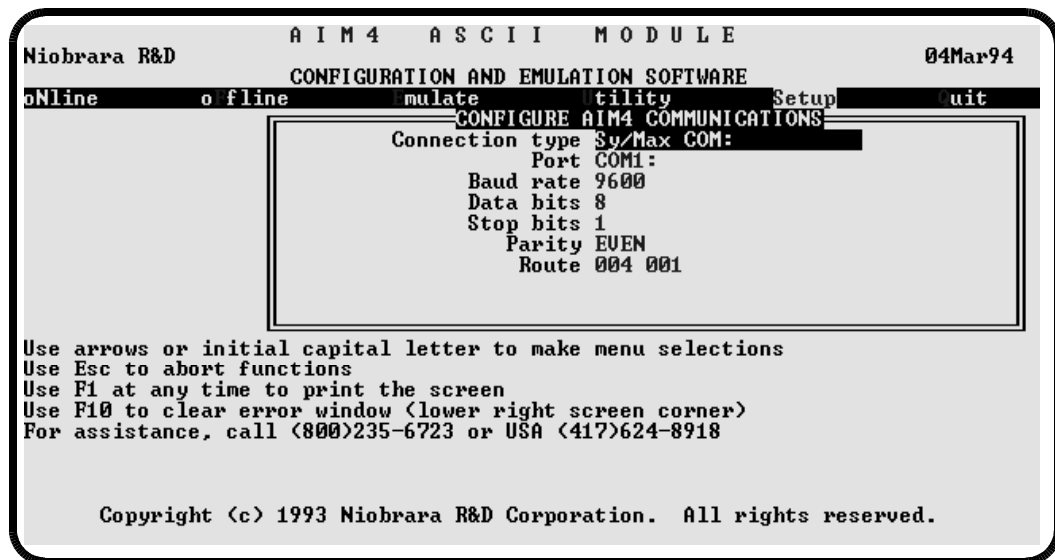


Figure 10-9 SY/MAX Setup Screen

**Port** - When one of the COM: connection types is selected a particular port of the personal computer must be selected. Available choices are: COM1:, COM2:, COM3:, and COM4:. Select the port which will be used to connect to the AIM4.



indirect connection from the RS-422 port of the card is made through other Net-to-Net ports choose Sy/Link Net-to-Net. If the RS-422 port is not used and the connection is made through the SY/NET network to another NIM, the choice does not matter.

**Base address** - This is a hex value that represents the SY/LINK's cards address range selected by DIP switches on the card. Select the same range that is set on the card.

**RS-422 Baud rate** - Select the baud rate to match the external device, normally 9600.

**RS-422 Data bits** - Select the data bits to match the external device, normally 8.

**RS-422 Stop bits** - Select the stop bits to match the external device, normally 1.

**RS-422 Parity** - Select the parity to match the external device, normally EVEN.

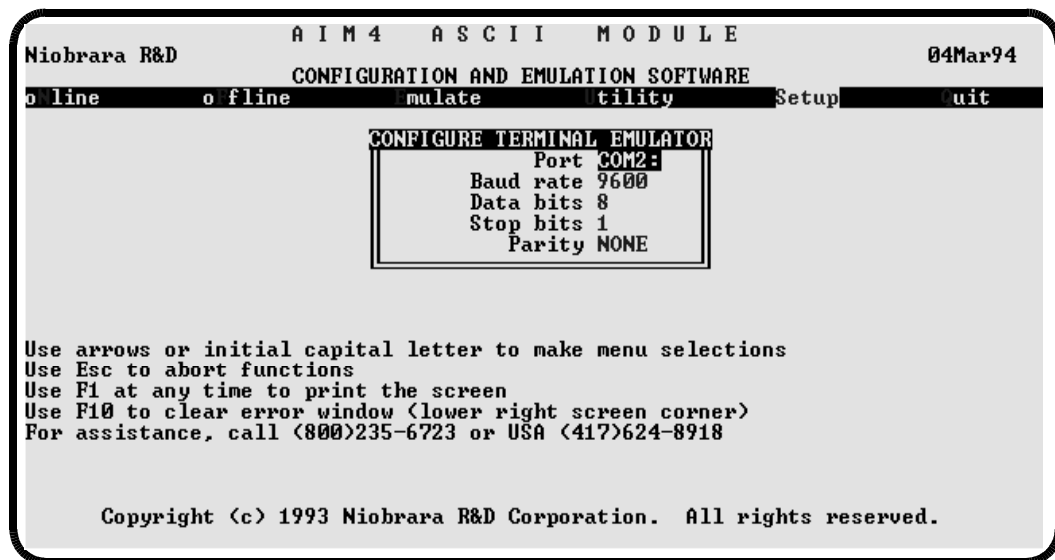
**Sy/Net speed** - Select to match the speed settings of the other devices on the SY/NET.

**Sy/Net size** - Select to match the setting on the other SY/NET devices.

**Route** - The first drop in the route defines the network address of the SY/LINK board. Since the personal computer is connected to the SY/LINK card through the edge connector of the card, port 0, the drop number must start with 0. The remaining two digits of that drop should be selected not to match any other device on the SY/NET. For instance, there are three NIMs on the network addressed 01, 02, and 03. It seems logical to make the SY/LINK card be at address 04 so the first drop in the route field will be 004. The next drop will be that of the NIM port connected to the AIM4, or another Net-to-Net port. If Sy/Link Direct was selected and the AIM4 is connected directly to the RS-422 port of the SY/LINK card the full route statement would be 004 104 as the RS-422 port is considered to be port 1.

## Terminal Emulator SETUP

The Terminal Emulator setup allows an individual setup for the operation of the terminal emulator. For instance, this separate setup will allow COM1 to be used for the AIM4 SY/MAX connection and COM2 to be used for a terminal emulator connection.



**Figure 10-11** Terminal Emulator Setup Screen

**Port** - Available choices are: COM1:, COM2:, COM3:, and COM4:. Select the port which will be used to connect to the external device.

**Baud rate** - The Baud rate selected here is the baud rate of the personal computer serial port selected. This value should be set to match the device connected to the personal computer.

**Data bits** - Selects the number of data bits for the message packet. Choices are 8 or 7 depending on the attached device.

**Stop bits** - The stop bits are normally set to 1 but may be adjusted to 2 for some particular application.

**Parity** - Choices are EVEN, ODD, and NONE.

## Command Line Parameters

AIM4SW (REV 18Jan93 and later) may be started from the DOS command line with a sequence of letters which represent the keystrokes necessary to perform any operation. This allows batch processing of various commands such as downloading of stored setup files. The special characters /R, /D, and /E refer to the Return key, Delete key, and Escape key respectively. The parameters are not case sensitive. The following example changes the Route to 102,055; loads the setup file TEST.AIM into an AIM4, then quits. It is assumed that the AIM4 is connected to the computer and the rest of AIM4SW setup is correct.

```
>AIM4SW SS/R/R/R/R/D102,055/RNFRTEST/RFSAQ
```

The S selects Setup, the second S selects Sy/Max, five Return keys to get to the Route field, /D for Delete to remove any previous route, 102,055 Return, for the new route, Y for Yes save the setup to disk, F for oFfline, R for Read from disk, TEST Return for the filename, F for oFfline, S for Send to module, A for all, and finally Q for Quit.

Another way of changing the Setup is to copy the setup file to another name in DOS, then copy it back to AIM4SW.STP in the batch file before calling AIM4SW.





## AIM4SW Errors

When the AIM4SW program detects an error, a window will display in the lower right corner of the screen. Read the message displayed and then press **F10** to extinguish the error. In some cases the program will attempt to continue but in most cases the function in progress will be aborted. Correct the condition indicated and try the operation again.

SEND Communications timeout.

or

FETCH Communications timeout.

This error occurs when the program transmits a parameter setting or reading message to the AIM4 and receives no response. This can be caused by a wiring problem between the computer and the AIM4, by not connecting the AIM4 Print/Prog port to the COM: port specified in the AIM4SW Setup menu, or by a baud rate mismatch between the AIM4 and the computer. Sometimes resident programs such as mouse drivers or communications programs can also interfere with the operation of AIM4SW. Reboot your computer with these programs disabled.

Communications error sending to AIM4. Invalid acknowledge.

or

FETCH Bad checksum.

The software received a response it did not recognize. See "*SEND Communications timeout*" above, pay special attention to an incorrect baud rate or a resident mouse driver attached to the same port. Verify that you have the latest version of AIM4SW that is compatible with the firmware revision of your AIM4 module.

Attempt to send from memory out of range.

or

Attempt to send to port out of range.

You specified a port number other than 1 through 4 in response to a prompt from "*Send memory to AIM4*".

*Attempt to fetch port out of range.*

or

Attempt to fetch to memory out of range.

You specified a port number other than 1 through 4 in response to a prompt from "*Fetch memory from AIM4*".

Port ww, Path xx: Register (yy) < Rack Base (zz) + 1

AIM4SW has detected a configuration error. The Start Register parameter of the specified port and path is less than the setting of Rack Base plus one. Note the port, path, and register. Type **F10** to let the operation continue, and then use the appropriate function to correct the error.

Port ww, Path xx: Register (yy) > Rack Base (zz) + 2047

AIM4SW has detected a configuration error. The Start Register parameter of the specified port and path is greater than the maximum allowed by the setting of Rack Base. Note the port, path, and register. Type **F10** to let the operation continue, and then use the appropriate function to correct the error.

Register (xx) is less than Rack Base (yy) + 1. Changing.

AIM4SW has detected a configuration error. The Start Register parameter of the specified port and path is less than the setting of Rack Base plus one. AIM4SW has changed the affected register to be equal to Rack Base plus one. You probably need to check your configuration and change it again.

Register (xx) is greater than Rack Base (yy) + 2047. Changing.

AIM4SW has detected a configuration error. The Start Register parameter of the specified port and path is greater than the setting of Rack Base. AIM4SW has changed the affected register to be Rack Base + 2047. You probably need to check your configuration and change it again.

Can't open file "filename" for read.

The disk file specified in "*Read from disk to memory*" cannot be found or cannot be read. Check the file name and try again.

Can't read disk file.

The disk file specified in "*Read from disk to memory*" was found but cannot be read. The file may not be in AIM4 format or may have been created by a more recent version of the program. Refer to the release notes for conversion instructions.

Can't open file "filename" for write.

The file specified in "*Write from memory to disk*", "*Print offline*", or "*Print screen*" cannot be created or the device specified for "*Print offline*" or "*Print screen*" cannot be attached. Your disk or directory may be full or your disk may be write protected. If the error occurs with the default print output file "PRN". Try again with the name "LPT1:".

Can't write to disk file.

The disk file specified in "*Write from memory to disk*" was successfully created but cannot be written. Your disk is probably full.

Can't save setup file.

The setup file "AIM4SW.STP" which holds the programming port and baud rate setting for the program cannot be created or written. Your disk or directory is full or write protected.

Can't locate setup file. Use SETUP option to create it.

The program could not find its setup file "AIM4SW.STP" in the current directory. You may be in the wrong directory. To build a new file, select the Setup option from the main menu and enter the requested data.

Can't read setup file. Use SETUP to change it.

The program setup file "AIM4SW.STP" was found but could not be read. The file has been corrupted or is from a different version of the software. To rebuild the file, select the Setup option from the main menu.

No port name specified at start-up.

An online or transfer operation was requested but no port is selected or the port was not successfully opened. Use Setup to select the COM port attached to the AIM4.

Can't open communications port "COMx:" Status = 133.

The COM port specified in the Setup menu was not detected in your computer's hardware. Verify that the port is installed and set to the proper address and that there is no conflict with a memory resident program or driver.

Can't open communications port "COMx:" Status = 7.

The program could not obtain enough memory from the operating system. Install more memory or remove resident programs.



## Troubleshooting

Finding a problem in a complex system involving the AIM4 can be difficult because so many factors must be just right for it to work at all and any one of them can prevent operation at all. The AIM4 front panel indicators show the progress of input data through the unit and help the user narrow the range of possibilities. By far the most common problems are in selecting the data path parameters. AIM4SW includes an emulation function that can be used to check your Data Path parameters in most applications. To find out why input data does not end up in PLC registers, proceed by tracing the flow of data, step by step, from the data source to the AIM4.

When the AIM4 is receiving characters on a port, the yellow RX light will flicker. This does not mean that the baud rate and other communication parameters are correctly set, simply that the input device is sending data and that its output is correctly connected to the AIM4.

If the RX light never comes on, verify that the connected device is actually sending by connection to a breakout box, computer terminal, or other test instrument. Using a terminal allows you to verify that the communication parameters of baud rate, parity, etcetera are correct and that the message format is what you think it is. Many CRT terminals have a monitor mode that will display all control codes. Verify proper wiring to the AIM4 by consulting the pinout diagrams in the appendix. The most important connections are pin 5 to ground and pin 2 receiving input data.

When the AIM4 has accumulated a message of one or more characters in the Accept Character table (see Figure 12-1 the Accept Character Table on page 71) and one character from the Terminate Character table (see Figure 12-2 on page 71), the red Terminator light for that port will glow for about half a second. A terminate character by itself (unless it is also in the Accept Character table) will not activate the Terminator light.

If the Terminator light fails to illuminate, verify that communications parameters including Baud Rate, Data Bits, Stop Bits, and Parity are correctly set and that the expected characters are properly selected for the Accept and Terminate character tables. If the port is operating in 8 bit mode and the connected device sends characters with the most significant bit set, they will not be recognized unless character codes from 80 to FF hex are enabled in the Accept and Terminate character tables.

When the terminated message matches one of the four Patterns, the green Pattern light will also illuminate. If Terminate lights but Pattern does not, recheck the Patterns for that port's data paths. The Terminator light means that a message is being assembled and passed on to the data path pattern matching, although this does not mean that all received characters are in the message. You may wish to temporarily change a pattern to "\*" just to verify that everything else is working.

If you are not sure exactly what character sequence is being sent to the AIM4, use the **TERMINAL EMULATOR** section of AIM4SW on the personal computer. Connect the attached device to the personal computer and type the polling sequence by hand and watch the reply. Non-printable characters are shown by their hex value within <>. If it is not convenient to use the terminal emulator, it may be helpful to set the pattern to the wildcard \*, set the register count to a large value such as 10 or 20, and set the editing mode to **PACKED**. Include all characters in the **ACCEPT** table, but only one character in the **TERMINATE** table, 0D hex. Now try to slow down the source of data to a point where it only sends a new string every 30 seconds or so. If the attached device is responding to a periodic poll from the AIM4 simply increase the polling interval. This will allow you to look at the raw ASCII data coming into the AIM4 using the register viewer. By selecting only the carriage return for the terminator, the message will only terminate once. If any additional characters follow the carriage return, such as a line feed, these characters will be included in the beginning of the next string. When you recognize a pattern to the data, try setting the terminator to another value that will fit your needs.

Using the register viewer, Square D software, or a programmer, verify that data is arriving in the PLC registers. The PLC ladder program does not need to be running for register data to be transferred. Monitor the signaling register at Rack Base. One bit in this register will toggle any time a Data Path is triggered (Pattern light). If you do not see a bit change in this register, verify that the Rack Base setting matches the PLC's first Rack Address for the AIM4 register slot. If the signaling bit toggles and no data appears in the register check that the register count for the data path is at least one and that the AIM4 Start Register is within the range defined in the PLC rack address table.

Output is simpler than input, as the only parameters that affect operation are the communications parameters of baud rate, parity, data bits, and stop bits. The most common problem with output is failure to provide a signal to AIM4 CTS (RS-232 pin 8). This signal normally comes from a printer's DTR output (usually pin 20, sometimes pin 11). If there is no CTS signal, the TX light will never light up. Some other things to look for:

- 1 If the TX light is flickering but the output device does not respond or responds incorrectly, verify settings of baud rate, parity, data bits, and stop bits. Unless you have a specific reason to the contrary, set the AIM4 port to two stop bits. The port operation must be set from the AIM4SW configuration software. Any specification steps in the print rung must match the settings of the AIM4 Print/Prog port and do not determine that characteristics of the message emitted by the AIM4 data port. The AIM4 strips the parity bit from data entering the Print/Prog port. If parity is desired on the output, it must be selected in the AIM4 Port Parameters.
- 2 If the TX light does not come on and you have verified the presence of a CTS signal, make sure that the first character sent to the Print/Prog port is a digit 1 through 4 identifying the desired output data port. This is normally accomplished with an ASCII step in a print rung. The ASCII step must be the first thing to write and the selection digit must be the first character. Square-D Company's programming software requires programming two characters at a time. The second character can be a blank if it is desired that nothing else print. No TX light can also indicate a baud rate or wiring problem between the PLC and the Print/Prog port. Are all data ports affected?
- 3 If an output message is truncated, it may contain an embedded null (0) or form-feed (ASCII 12) character which the attached output device is not showing or the output device may not be able to accept data fast enough. In the latter case, use a lower baud rate or use the output restraint feature through the AIM4 CTS input.

```

Niobrara R&D      A I M 4      A S C I I      M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 2, 3, or 3 through 2 change parameters.
1 Port 1          5 Stop bits 2          2 Periodic Poll
2 Baud rate 9600  5 Capitalize NO        3 Terminate Count 0
3 Data bits 8     2 Accept characters    4 Terminate Timeout 0
-----
Editing Accept character set
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F      NUL
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F      EVEN
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
Use arrows to move, SPACE to change, RETURN to exit.

```

Figure 12-1 Accept Character Table

```

Niobrara R&D      A I M 4      A S C I I      M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 2, 3, or 3 through 2 change parameters.
1 Port 1          5 Stop bits 2          2 Periodic Poll
2 Baud rate 9600  5 Capitalize NO        3 Terminate Count 0
3 Data bits 8     2 Accept characters    4 Terminate Timeout 0
-----
Editing Terminate character set
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F      NUL
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F      EVEN
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
Use arrows to move, SPACE to change, RETURN to exit.

```

Figure 12-2 Terminator Character Table





These examples are included on the distribution diskette.

## **Example 1**

A barcode scanner reads a label with a variable length number of up to six digits. The processor must store this number and later compare it to a similar number entered on an operator terminal. Valid numbers are in the range of 500 through 100,000. Any other number is a different kind of error.

The barcode scanner outputs an ASCII string consisting of a leading asterisk, 3 to 10 digits, a trailing asterisk, a carriage return and a line feed. The scanner outputs NR if an error occurs.

The operator keyboard outputs ASCII characters 0 through 9 and a carriage return. Additionally, the operator can press a Clear button that sends a C. The clear operation must cancel the input in process and allow the operator to re-enter the number.

Since six digits is longer than can be stored as an integer value, BCD is chosen as the representation of the number.

### **Port 1 Configuration**

The port connected to the scanner is configured as follows:

Baud rate, data bits, parity to match the scanner. Three data paths are used, one to match a valid read, one to match an out of bounds number, and one to match the NR error. (Figure 13-1)

Accept character table includes 0 through 9. Because the asterisk is excluded, it is stripped from the message. (Figure 13-2)

Terminate table has only the carriage return character, 0D. (Figure 13-3)

```

Niobrara R&D      A I M 4      A S C I I      M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 2, 3, or 3 through 4 change parameters.
1 Port 1          5 Stop bits 2      2 Periodic Poll
2 Baud rate 9600  5 Capitalize NO    0 Terminate Count 0
3 Data bits 8     2 Accept characters 1 Terminate Timeout 0
4 Parity NONE     3 Terminate characters
                        DATA PATH 1
1 Pattern -(500-100000).....
2 Mask -.....
3 Start Register 101  0 Register Count 3  0 Editing BCD      0 Continue NO
                        DATA PATH 2
3 Pattern -(-).....
1 Mask -1.....
1 Start Register 101  1 Register Count 3  0 Editing INTEGER  0 Continue NO
                        DATA PATH 3
1 Pattern -NR.....
1 Mask -2.....
0 Start Register 101  2 Register Count 3  2 Editing INTEGER  0 Continue NO
                        DATA PATH 4
3 Pattern .....
1 Mask .....
1 Start Register 104  1 Register Count 0  0 Editing ASCII
3 Print/Prog baud rate 9600      1 Rack start address 100
Press Esc to return to menu.    2 Trigger/Inhibit register 0

```

Figure 13-1 Example 1 - Port 1 - Setup

```

Niobrara R&D      A I M 4      A S C I I      M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 2, 3, or 3 through 4 change parameters.
1 Port 1          5 Stop bits 2      2 Periodic Poll
2 Baud rate 9600  5 Capitalize NO    0 Terminate Count 0
3 Data bits 8     2 Accept characters 1 Terminate Timeout 0

```

Editing Accept character set																
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	NUL
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	EVEN
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF	

Use arrows to move, SPACE to change, RETURN to exit.

Figure 13-2 Example 1 - Port 1 - Accept Table

Niobrara R&D				A I M 4				A S C I I				M O D U L E				O F F L I N E			
Type 1 to select a port. Type 2 through 7, 8, or 9 through 9 change parameters.																			
1	Port 1				5	Stop bits 2				7	Periodic Poll								
2	Baud rate 9600				3	Capitalize NO				8	Terminate Count 0								
3	Data bits 8				7	Accept characters				9	Terminate Timeout 0								
-----																			
Editing Terminate character set																			
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F				
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F				
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	NUL			
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F				
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F				
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	EVEN			
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F				
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F				
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F				
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F				
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF				
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF				
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF				
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF				
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF				
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF				
Use arrows to move, SPACE to change, RETURN to exit.																			

**Figure 13-3 Example 1 - Port 1 - Terminate Table**

When an input message triggers any of these three paths, register 101 is loaded with a value and bit 9 of its status register is toggled. The processor can examine register 101 to determine what event occurred. Register 101 will be 0 on a good read and the value will be packed in registers 102 and 103 as a BCD value. Register 101 will be 1 on an out of bounds label. Register 101 will be 2 on a NR indication. In the latter two cases, registers 102 and 103 will be zeroed. Print rungs may be used to display prompts on the operator terminal, the setup is not affected.

## Port 2 Configuration

The operator interface port is configured as follows:

Baud rate, data bits, parity to match the terminal. Three data paths are used, one to match a valid read, one to match an out of bounds number, and a third to match the Clear command. Capitalize is set to YES in case the operator types a lower case C. (Figure 13-4)

Accept character table includes 0 through 9 and upper and lower case letter C. (Figure 13-5)

Terminate table has the carriage return character, 0D, as well as upper and lower case letter C. (Figure 13-6)

```

Niobrara R&D      A I M 4  A S C I I  M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 4, or 1 through 4 change parameters.
1 Port 2          5 Stop bits 2          2 Periodic Poll
2 Baud rate 9600  5 Capitalize YES       5 Terminate Count 0
3 Data bits 8     2 Accept characters    1 Terminate Timeout 0
4 Parity NONE     3 Terminate characters
                        DATA PATH 1 -----
5 Pattern <(500-100000)>.....
6 Mask           -.....
7 Start Register 105  0 Register Count 3  5 Editing BCD      2 Continue NO
                        DATA PATH 2 -----
8 Pattern <(-)>.....
9 Mask           -1.....
0 Start Register 105  1 Register Count 1  2 Editing INTEGER  1 Continue NO
                        DATA PATH 3 -----
1 Pattern <*C>.....
2 Mask           -2.....
3 Start Register 105  2 Register Count 1  2 Editing INTEGER  2 Continue NO
                        DATA PATH 4 -----
4 Pattern .....
5 Mask           -.....
6 Start Register 108  1 Register Count 0  1 Editing ASCII
                        -----
7 Print/Prog baud rate 9600          2 Rack start address 100
Press Esc to return to menu.        2 Trigger/Inhibit register 0

```

Figure 13-4 Example 1 - Port 2 - Configuration

```

Niobrara R&D      A I M 4  A S C I I  M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 4, or 1 through 4 change parameters.
1 Port 2          5 Stop bits 2          2 Periodic Poll
2 Baud rate 9600  5 Capitalize YES       5 Terminate Count 0
3 Data bits 8     2 Accept characters    1 Terminate Timeout 0

```

Editing Accept character set																
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	NUL
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	EVEN
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF	

Use arrows to move, SPACE to change, RETURN to exit.

Figure 13-5 Example 1 - Port 2 - Accept Table

Niobrara R&D
AIM4
ASCII MODULE
OFFLINE

Type 1 to select a port. Type 2 through 4 or 5 through 7 change parameters.

1 Port 2
5 Stop bits 2
7 Periodic Poll

2 Baud rate 9600
6 Capitalize YES
8 Terminate Count 0

3 Data bits 8
7 Accept characters
8 Terminate Timeout 0

Editing Terminate character set

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	NUL
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	EVEN
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	
D0	D1	D2	D3	D4	D5	D6	D7	D8	DA	DB	DC	DD	DE	DF		
E0	E1	E2	E3	E4	E5	E6	E7	E8	EA	EB	EC	ED	EE	EF		
F0	F1	F2	F3	F4	F5	F6	F7	F8	FA	FB	FC	FD	FE	FF		

Use arrows to move, SPACE to change, RETURN to exit.

Figure 13-6 Example 1 - Port 2 - Terminate Table

When an input message triggers any of these three paths, register 105 is loaded with a value and bit 9 of its status register is toggled. The processor can examine register 105 to determine what event occurred. Register 105 will be 0 on a valid entry and the value will be packed in registers 106 and 107 as a BCD value. Register 105 will be 1 on an out of bounds label. Register 105 will be 2 if the operator presses clear and the processor can use a print rung to send a clear escape sequence to the operator terminal.

Example 2

An automated instrument outputs periodic voltage and current readings. Voltage and current must be loaded into two separate floating point register pairs.

The meter outputs a string intended for a printer. The string includes labels as well as the values which must be transferred to the PLC. The format matches this example:

Voltage        2.34 VDC                                Current        18.2 Amps

The string is followed by a carriage return, line feed sequence.

Baud rate, Data bits, Parity: to match meter. One port of the AIM4 is programmed with two data paths, one for voltage and one for current. (Figure 13-7)

Accept table: digits 0 through 9, decimal point, letters V and A. (Figure 13-8)

Terminate table: letters V and A (carriage and line feed not included) (Figure 13-9)

Note: The extra occurrence of the letter V (in "VDC") poses no problem since a message is not recognized by either path unless it starts with a digit.

```

Niobrara R&D      A I M 4      A S C I I      M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 2, ., or 1 through 2 change parameters.
1 Port 1
2 Baud rate 9600
3 Data bits 8
4 Parity NONE
5 Stop bits 2
6 Capitalize YES
7 Accept characters
8 Terminate characters
          DATA PATH 1
A Pattern -#*U-
B Mask -
C Start Register 101
D Register Count 2
E Editing FLOAT
F Continue NO
          DATA PATH 2
G Pattern -#*A-
H Mask -
I Start Register 101
J Register Count 2
K Editing FLOAT
L Continue NO
          DATA PATH 3
M Pattern -
N Mask -
O Start Register 103
P Register Count 0
Q Editing ASCII
R Continue NO
          DATA PATH 4
S Pattern -
T Mask -
U Start Register 104
V Register Count 0
W Editing ASCII
X Print/Prog baud rate 9600
Y Rack start address 100
Press Esc to return to menu.
Z Trigger/Inhibit register 0
  
```

Figure 13-7 Example 2 - Configuration

```

Niobrara R&D      A I M 4      A S C I I      M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 2, ., or 1 through 2 change parameters.
1 Port 1
2 Baud rate 9600
3 Data bits 8
5 Stop bits 2
6 Capitalize YES
7 Accept characters
8 Terminate Count 0
9 Terminate Timeout 0
          -----
Editing Accept character set
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F      NUL
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F      EVEN
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
Use arrows to move, SPACE to change, RETURN to exit.
  
```

Figure 13-8 Example 2 - Accept Table

Niobrara R&D
AIM4
ASCII MODULE
OFFLINE

Type 1 to select a port. Type 2 through 4, or 5 through 7 change parameters.

1 Port 1
2 Baud rate 9600
3 Data bits 8
5 Stop bits 2
6 Capitalize YES
7 Accept characters
2 Periodic Poll
3 Terminate Count 0
4 Terminate Timeout 0

---

Editing Terminate character set

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	NUL
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	EVEN
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	
D0	D1	D2	D3	D4	D5	D6	D7	D8	DA	DB	DC	DD	DE	DF		
E0	E1	E2	E3	E4	E5	E6	E7	E8	EA	EB	EC	ED	EE	EF		
F0	F1	F2	F3	F4	F5	F6	F7	F8	FA	FB	FC	FD	FE	FF		

Use arrows to move, SPACE to change, RETURN to exit.

Figure 13-9 Example 2 - Terminate Table

Example 3

Three motor drive units are connected to port one of an AIM4-D unit with a 4-wire RS485 multidrop configuration. The system requirements are that the PLC monitor the velocity reading from each of the three drives, and occasionally write a new acceleration value to a drive. The Periodic Poll feature of the AIM4 will be used to prompt the drives for readings every 1 second resulting in a new reading for each particular drive every 3 seconds. The Print/Prog port of the AIM4 will be connected to a COMM port of the PLC with a DC1 cable (CC100). New acceleration values will be sent to the drives by print runs from the PLC ladder program.

Communication consists of a command message sent from the host (AIM4 or PLC) and a reply message returned from the drive. The drive is unable to initiate communication and must be prompted with a command message.

The command message is of the following format:

SOHnnbb####mEOT

Where "SOH" is the ASCII character 01 Hex.

"nn" is the address number of the drive with decimal value 00-99.

"bb" is a two digit integer representing a command. 01 for velocity, 02 for acceleration.

The next four characters "####" are the data padded with leading zeros. On commands requesting information, such as velocity, they should be all zeros. Valid values are within the range: 0000-9999.

The last character "m" is the data format character which defines the decimal point and scaling of the data. Valid values are 0-7.

"EOT" is the terminator, ASCII value 04 Hex.

The replies are of the same format with each unit returning its address, the command that the reply is responding for, the data, and the data format.

The Periodic Poll feature of the AIM4 is used to send a command message to each drive at a specified time interval without PLC intervention. Each drive will only respond to the command with its address. The AIM4 will take each response and mask out the unit address, command nmunomic, data, and data

format, and place each in specific registers in the PLC. The PLC may then copy these registers to another location for further processing before the next message appears at the AIM4 and overwrites the previous message.

## Setup

The AIM4 setup is as follows: (Figure 13-10)

The AIM4 has been rack addressed for registers 200-204 as these are the only registers used by the paths and including the signal register.

Path 1 masks out the unit number of the reply message. This way the PLC will know which unit is responding by the integer value located in register 201.

Path 2 masks out the command. This will be an integer value in register 202.

Path 3 masks out the actual data and places it in register 203.

Path 4 masks out the data format. If the format is always the same, this path is not needed.

Notice that the Continue field is set to YES on paths 1,2, and 3. This allows the AIM4 to reprocess the same message on each path even though it may have matched the pattern of a previous path.

Baud rate, Data bits, Parity, Stop bits are set to match the attached drives.

Accept Characters: SOH (01 Hex), 0 through 9 (Figure 13-11)

Terminate Characters: EOT (04 Hex) (Figure 13-12)

Niobrara R&D		A I M 4		A S C I I		M O D U L E		OFFLINE	
Type 1 to select a port. Type 2 through 2, or 1 through 2 change parameters.									
1	Port 1	5	Stop bits 2	2	Periodic Poll				
2	Baud rate 9600	5	Capitalize NO	3	Terminate Count 0				
3	Data bits 8	7	Accept characters		Terminate Timeout 0				
4	Parity NONE	3	Terminate characters						
----- DATA PATH 1 -----									
3	Pattern -C#####								
B	Mask -								
E	Start Register 201	D	Register Count 1	E	Editing INTEGER	F	Continue YES		
----- DATA PATH 2 -----									
3	Pattern -C#####								
H	Mask -								
I	Start Register 202	J	Register Count 1	K	Editing INTEGER	L	Continue YES		
----- DATA PATH 3 -----									
M	Pattern -C#####								
N	Mask -								
O	Start Register 203	P	Register Count 1	Q	Editing INTEGER	R	Continue YES		
----- DATA PATH 4 -----									
S	Pattern -C#####								
T	Mask -								
U	Start Register 204	V	Register Count 1	W	Editing INTEGER				
-----									
X	Print/Prog baud rate 9600	Y	Rack start address 200						
Press Esc to return to menu.				Z	Trigger/Inhibit register 0				

Figure 13-10 Example 3 - Configuration



```

Niobrara R&D      A I M 4  A S C I I  M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 7, or 8 through 9 change parameters.
1 Port 1          5 Stop bits 2          7 Periodic Poll
2 Baud rate 9600  6 Capitalize NO        8 Terminate Count 0
3 Data bits 8     7 Accept characters    9 Terminate Timeout 0
-----
Editing Accept character set
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F      NUL
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F      EVEN
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
Use arrows to move, SPACE to change, RETURN to exit.

```

Figure 13-11 Example 3 - Accept Table

```

Niobrara R&D      A I M 4  A S C I I  M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 7, or 8 through 9 change parameters.
1 Port 1          5 Stop bits 2          7 Periodic Poll
2 Baud rate 9600  6 Capitalize NO        8 Terminate Count 0
3 Data bits 8     7 Accept characters    9 Terminate Timeout 0
-----
Editing Terminate character set
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F      NUL
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F      EVEN
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
Use arrows to move, SPACE to change, RETURN to exit.

```

Figure 13-12 Example 3 - Terminate Table

```

Niobrara R&D      A I M 4  A S C I I  M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 4, or 5 through 7 change parameters.
1 Port 1          5 Stop bits 2          7 Periodic Poll

-----
Editing Periodic Polling parameters
Send a query every 1.00.. seconds.

Query 1 -0010100002♦-----
Query 2 -0020100002♦-----
Query 3 -0030100002♦-----
Query 4 -----
Query 5 -----
Query 6 -----
Query 7 -----
Query 8 -----
Query 9 -----
Query 10 -----
Query 11 -----
Query 12 -----
Query 13 -----
Query 14 -----
Query 15 -----
Query 16 -----

Use UP and DOWN arrows to select field. Press ENTER to exit.

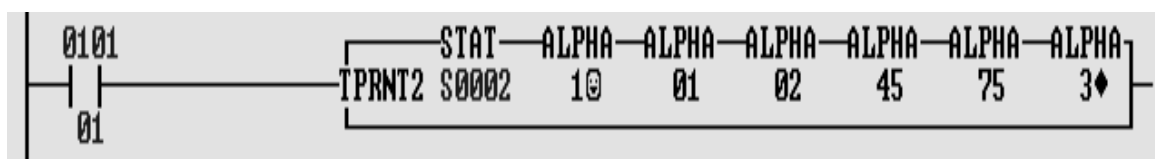
```

**Figure 13-13 Example 3 - Periodic Poll Configuration**

The Query 1 prompts unit #01 for its velocity. One second later Query 2 prompts unit #02. Again one second later Query 3 prompts unit #03. One more second passes and Query 1 is sent again.

The sending of a new acceleration parameter may be handled as follows:

Drive unit #01 needs to have the acceleration set to 4575 with a format 3. The AIM4 is connected to COM2 of the PLC. The leading 1 of the first ALPHA field denotes the port of the AIM4 for the message to be sent. This character will be stripped from the outgoing message by the AIM4.



## Example 4

The setup in Example 3 may be greatly simplified by the use of an AIM4-2D PLUS unit. The use of print rungs in the PLC may be completely eliminated by inserting another periodic poll string with placeholders for registers for the data of the drop number and acceleration. The rack address of the module must be increased to include a few PLC output registers: 205, 206, 207, and 208. Register 205 will contain the unit number. Register 206 will contain the command. Register 207 will contain the data to be written to the drive. Register 208 will be the Trigger/Inhibit register for that port. The periodic poll will be turned off and the Trigger/Inhibit register will be used to trigger each Query. Instead of periodically polling the drives for the velocity, the PLC would load the unit number of the drive into register 205, load the command number into register 206, and the data into register 207. Finally, the PLC would change the state of bit 1 of register 208 and the poll would be sent to the drive.

Since the PLC has complete control of the query, it may load any unit number, command, and data into the query and send the query at will and only one query string is needed for all commands. The E substitution string is used to allow zero padding of the unsigned decimal values.

```

Niobrara R&D      A I M 4  A S C I I  M O D U L E      OFFLINE
Type 1 to select a port. Type 2 through 2. 3. or 1 through 2 change parameters.
1 Port 1          3 Stop bits 2          2 Periodic Poll
-----
Editing Periodic Polling parameters
Send a query every 0.00 seconds.

Query 1 -0020502E0020602E0020704E2-
Query 2 .....
Query 3 .....
Query 4 .....
Query 5 .....
Query 6 .....
Query 7 .....
Query 8 .....
Query 9 .....
Query 10 .....
Query 11 .....
Query 12 .....
Query 13 .....
Query 14 .....
Query 15 .....
Query 16 .....

Use UP and DOWN arrows to select field. Press ENPress F9 to edit in hexadecimal

```

Figure 13-14 Example 4 Periodic Poll



## Throughput and Latency

The exact throughput for a particular AIM4 application involves a number of factors and is difficult to predict. Throughput is defined as the number of registers processed per second. Latency is the delay between the end of the stop bit of the terminating character of an input message and the availability of the data on the PLC bus. Latency between availability of data on the bus and recognition by a ladder rung can be as much as one full scan time.

Latency in the AIM4 can be as low as 0.5 and as high as 20 milliseconds. Most applications will experience latencies in the range of 1 to 3 milliseconds with few exceeding 5 milliseconds. Latencies up to 20 milliseconds may be experienced with the most complex patterns under maximum input and output processing loads. Throughput can be more than 30,000 registers per second. A conservative rule of thumb is not to attempt to process more than 200 messages per second divided by the number of patterns that will be evaluated.

Throughput in most applications is limited only by the 19,200 baud maximum rate of the data ports. Even in complex applications with all four ports being driven at 19,200 baud, the AIM4's internal message buffers allow it to handle bursts of up to sixteen messages per port at the full baud rate provided that there is enough delay between bursts for the pattern matching, masking, and editing process to catch up.

With extremely complex setups involving complex wildcard patterns on multiple paths, FLOAT editing, and all five ports being driven continuously at maximum rates, it is possible for throughput to drop below input data rate at 19,200 baud and, if the RTS restraint feature is not used, for data to be lost. No problems should be experienced when driving all five ports at 9600 baud or slower, no matter what the complexity of the configuration.

Heavy use of SY/MAX I/O through the Print/Prog port such as continuously polling the data registers will also adversely affect throughput. For maximum throughput, only use SY/MAX I/O for printing and dynamic configuration and use the rack interface for input operations.

It should be noted that it is entirely possible for data to be processed by the AIM4, resulting in the signal bit for that path being toggled and another data matching the same pattern and again toggling the signal bit within the period of one scan time of the PLC. If this happens, the first data may be missed by the PLC.

For instance: A bar code reading system is triggered to read bar codes on a box by a photocell. Each box has two bar codes, one of 10 characters and one of 4 characters. The scanner will transmit the first bar code, a + character, then the second bar code, and a carriage return. The orientation of the box may change and sometimes the 4 digit code may be read first or second. If the scanner gets a NO READ for

one of the bar codes, it will transmit a NR instead of the bar code. The PLC needs to know when a NR has occurred so the following setup is tried:

Pattern 1: #####

Pattern 2: #####

Pattern 3: NR

The + and carriage return characters are used as terminators. The port is set for 9600 baud, 8 data bits, no parity, and one stop bit. This makes for a 10 bits per character message.

The message NR+NRCR is sent from the scanner indicating both codes were misread. The AIM4 will treat this as 2 messages of 3 characters each. With the above baud rate and byte size this results in the Pattern 3 signal bit being toggled by the first NR. Approximately 3 to 4mS later that bit is toggled again by the second NR. If the PLC scan time is greater than 3mS then there is a good chance that one or both of these bit toggles may be missed.

The solution to the above could be to slow the baud rate to a value that allows sufficient time to process the data at a rate slower than the scan time. This may not be feasible if the scan time is very large.

Another solution would be to add another pattern to handle the NR+NR occurrence.

Pattern 4: NR+NR

Of course the character + would have to be added to the accept table and the wildcard \* would need to be added to the end of Patterns 1, 2 and 3 to account for the case of the + terminator for those messages. But adding the wildcard \* will force the patterns to be in a specific order to avoid capturing the 10 character message on the 4 character pattern. The continue field must be set to NO for all patterns. The new patterns would be:

Pattern 1: #####\*

Pattern 2: #####\*

Pattern 3: NR+NR

Pattern 4: NR\*

So now bit 1 is toggled when a 10 digit number is entered, bit 2 is toggled by the 4 digit number, bit 3 is toggled by a bad read on both codes, and bit 4 is toggled by a single bad read.

## Connector Pinouts

### RS-232 data ports (DE9P with screw lock posts)

1 CD	input no connection in AIM4
2 RXD	input serial data from external device
3 TXD	output serial data to external device
4 DTR	output pulled up to +10 volts internally
5 SG	signal ground
6 DSR	input no connection in AIM4
7 RTS	output normally high (+10), driven low (-10) when internal buffers are filling faster than data can be processed.
8 CTS	input must be high for multiplexer to be able to transmit. This pin is usually connected to DTR of a printer. Ensure that the attached device can provide this or tie it to pin 4 (DTR).
9 RI	input no connection in AIM4

### RS-232 data port with SD034B adapter attached (DB25P)

2 TXD	output serial data to external device
3 RXD	input serial data from external device
4 RTS	output normally high (+10), driven low (-10) when internal buffers are filling faster than data can be processed.
5 CTS	input must be high for multiplexer to be able to transmit This pin is usually connected to DTR of a printer. Ensure that the attached device can provide this or tie it to pin 20 (DTR).
7 SG	signal ground
20 DTR	output driven to +10 volts internally.

### **RS-422 data port (DE9S with slide lock posts)**

- |                  |                                                            |
|------------------|------------------------------------------------------------|
| 1 TX-            | transmit data (inverted) from AIM4 to output device        |
| 2 TX+            | transmit data (noninverted) from AIM4 to output device     |
| 3 RX-            | receive data (inverted) from data source to AIM4           |
| 4 RX+            | receive data (noninverted) from data source to AIM4        |
| 5 CTS-           | must be more negative than pin 7 to allow AIM4 to transmit |
| 6 RTS-           | driven low (0V) when AIM4 is ready to receive data         |
| 7 CTS+           | must be more positive than pin 5 to allow AIM4 to transmit |
| 8 RTS+           | driven high (+5V) when AIM4 is ready to receive data       |
| 9 Shield ground. | AC coupled to the chassis.                                 |

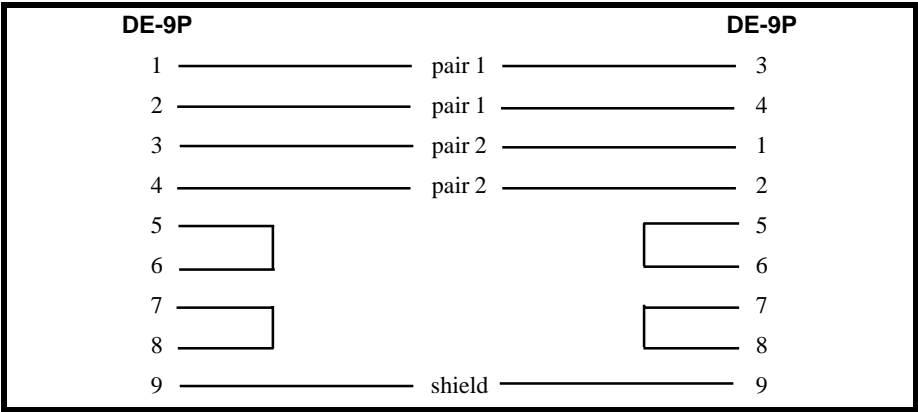
### **RS-422 Print/Prog port (DE9S with slide lock posts)**

- |                                |                                                        |
|--------------------------------|--------------------------------------------------------|
| 1 TX-                          | transmit data (inverted) from AIM4 to SY/MAX device    |
| 2 TX+                          | transmit data (noninverted) from AIM4 to SY/MAX device |
| 3 RX-                          | receive data (inverted) from SY/MAX device to AIM4     |
| 4 RX+                          | receive data (noninverted) from SY/MAX device to AIM4  |
| 5 +5 volts                     | for NR&D SC406 smart cable                             |
| 6 +5 volts                     | for NR&D SC406 smart cable                             |
| 7 Logic ground and +5V return. |                                                        |
| 8 Logic ground.                |                                                        |
| 9 Shield ground.               | AC coupled to the chassis.                             |



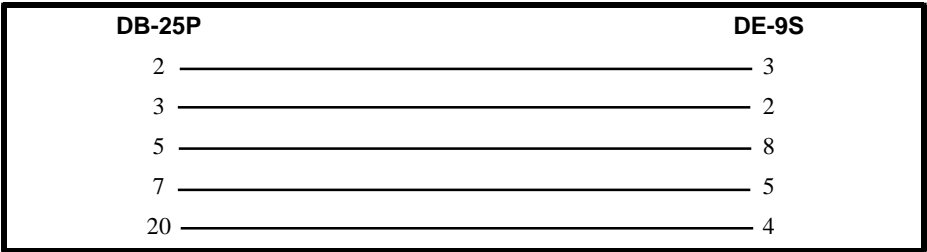
# Recommended Cabling

## AIM4 to SY/MAX RS-422 port

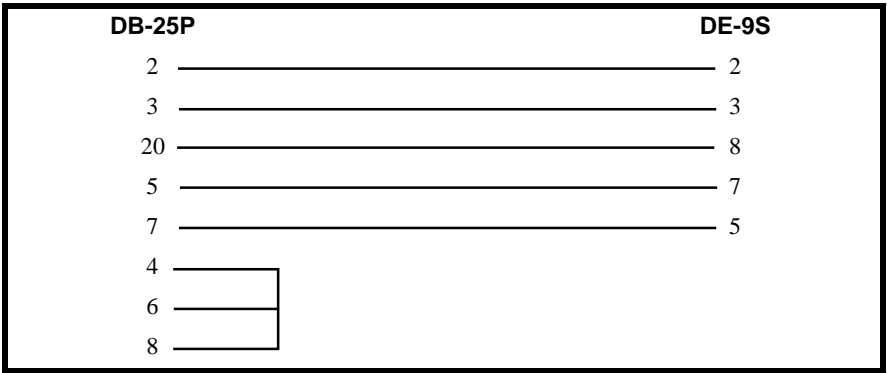


This is a Niobrara **DC1** cable. It may be used to connect the AIM4 Print/Prog port to a Square D PLC such as a Model 300, 400, 500, 600, or 700, or a NIM, SPE4, or EPE5 RS-422 port. This cable is used whenever SY/MAX devices are connected together and handshaking is not required. This cable may be used to connect AIM4-xD and -2M data ports to other AIM4 ports.

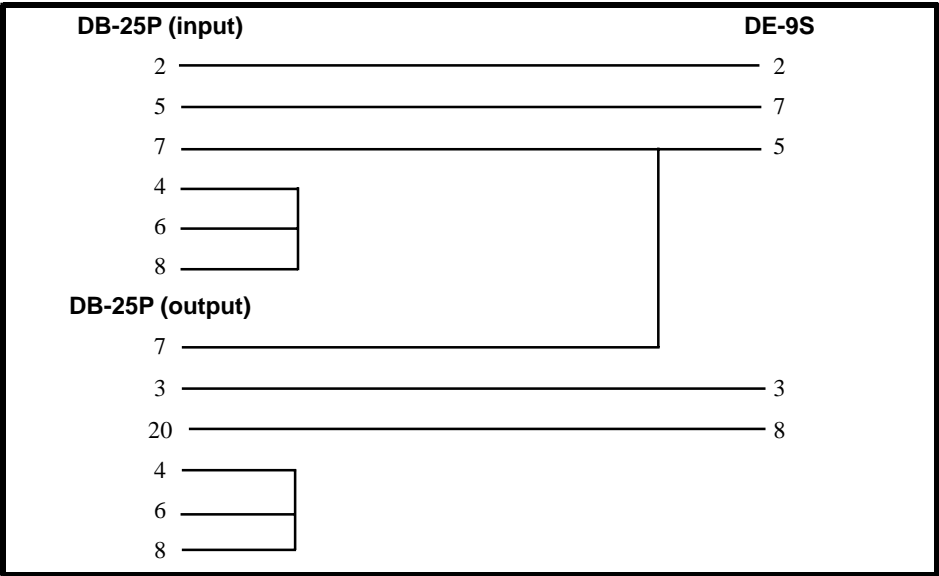
## RS-232 DCE (modem) to AIM4 RS-232 data



RS-232 DTE (terminal) to AIM4 RS-232 data port

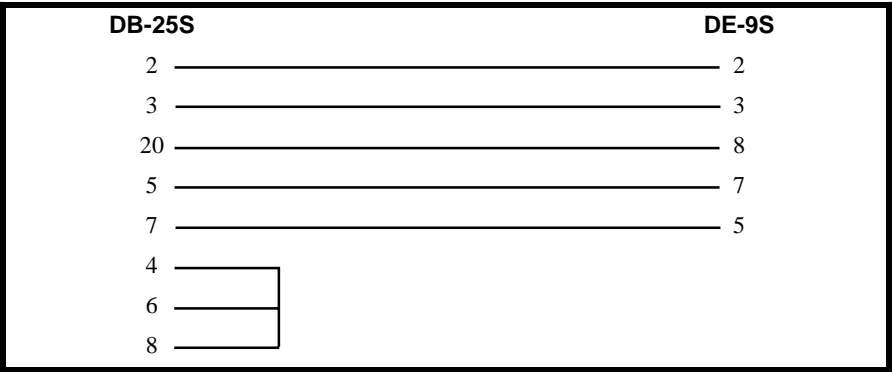


AIM4 data port to separate RS-232 DTE input and output devices



This is a Niobrara **SP25Y** cable.

**Male RS-232 DTE (personal computer) to AIM4 data port**



**Personal Computer to AIM4 Print/Prog port**

Use of a Niobrara SC406 or SC902 smart cable is recommended, although any RS-232 to RS-422 converter may be used.



---

# Appendix A

## Serial Communication Overview

This Appendix is meant to provide a general background for common types of asynchronous serial communication.

### Hardware Overview

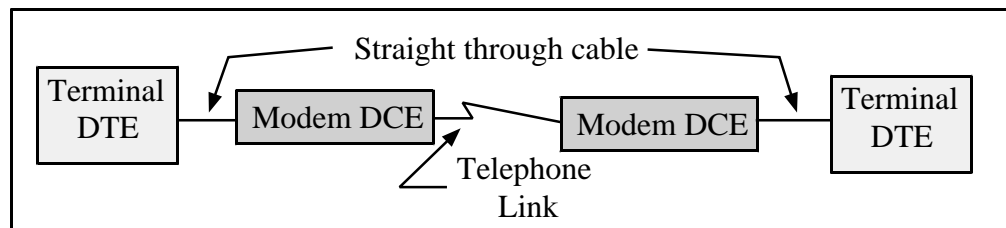
The need for information to be exchanged between independent devices has brought about the development of several serial communication standards. The most commonly encountered are RS-232, RS-422, RS-485, and 20mA current loop.

#### RS-232

RS-232 is intended for connecting two devices together for serial communication for short distances (50 feet or less) and low baud rates (19200 baud or less).

RS-232-C has two main classes of devices: DTE (Data Terminal Equipment) such as terminals and personal computers, and DCE (Data Communication Equipment) such as modems.

The original implementation of the RS232 connection was for connecting terminals to modems as shown in Figure A-1. The standard connectors mounted on the equipment were DB25 females. Straight through cables with 25 pin male connectors were used to connect the DTE to DCE.



**Figure A-1 DTE to Modem connection**

Each pin on the DTE was connected to the same pin on the DCE. The most common pins and their definitions are listed below.

Pin 2: **TD** Transmit Data

This circuit is the path that serial data is sent from the DTE to the DCE.

Pin 3: **RD** Receive Data

This circuit is the path that serial data is sent from the DCE to the DTE.

**Pin 4: RTS** Request to Send

This circuit is the signal that indicates that the DTE wishes to send data to the DCE. In normal operation the RTS line will be OFF (MARK). Once the DTE has data to send it asserts RTS (SPACE) and waits for the DCE to assert CTS. RTS will remain asserted until the data is completely sent. In a full duplex channel, RTS may be asserted at initialization and left in that state.

**Pin 5: CTS** Clear to Send

This circuit is the signal that indicates that the DCE is ready to receive data from the DTE. In normal operation the CTS is not asserted. When the DTE asserts RTS, the DCE will do whatever is necessary to allow data to be sent. (This may mean raising the carrier and waiting until it is stabilized.) When the DCE is ready, it asserts CTS which allows the DTE to send data. When the DTE is finished sending data it will reset the RTS and the DCE will in turn reset its CTS.

Note: Most DTE must have CTS asserted before it will transmit.

**Pin 6: DSR** Data Set Ready

This circuit is the signal that informs the DTE that the DCE is active. It is normally asserted by the DCE at power-up and left that way.

Note: Most DTE must have DSR asserted to operate properly.

**Pin 7: SG** Signal Ground

This circuit is the ground to which all signals are referenced.

**Pin 8: DCD** Data Carrier Detect

This circuit is the signal that the DCE informs the DTE that it has an incoming carrier.

Note: Some DTE must have DCD asserted to operate properly. Also, some personal computer modems always assert DCD.

**Pin 20: DTR** Data Terminal Ready

This circuit provides the signal that informs the DCE that the DTE is alive and well. It is normally asserted by the DTE at power-up and left in that state.

Note: Most DCE must have DTR asserted to operate properly.

**Pin 22: RI** Ring Indicator

This circuit provides the signal from the DCE to indicate that the modem is ringing. The line is asserted by the DCE during each ring cycle.

The full pinout for the standard 25 pin connector is shown in Figure A-1.

With the down-sizing of computers it became necessary to move to a 9 pin port to save room. Only the most commonly used functions were kept for the 9 pin configuration. The TYPE A and TYPE B configurations are shown in Figures A-2 and A-3. The only difference is pins 2 and 3.

**Table A-1 25 pin RS-232 port**

Pin	Name	DTE/DCE	Function
1	CG	<--->	Frame Ground
2	TD	--->	Transmitted Data
3	RD	<---	Received Data
4	RTS	--->	Request to Send
5	CTS	<---	Clear to Send
6	DSR	<---	Data Set Ready
7	SG	<--->	Signal Ground
8	DCD	<---	Data Carrier Detect
9*		<---	Positive DC Test Voltage
10*		<---	Negative DC test Voltage
11*	QM	<---	Equalizer Mode
12+	SDCD	<---	Secondary Data Carrier Detect
13+	SCTS	<---	Secondary Clear to Send
14+	STD	--->	Secondary Transmitted Data
15#	TC	<---	Transmit Clock
16+	SRD	<---	Secondary Receive Data
17#	RC	<---	Receive Clock
18	DCR	<---	Divided Clock Receiver
19+	SRTS	--->	Secondary Request to Send
20	DTR	--->	Data Terminal Ready
21*	SQ	<---	Signal Quality Detect
22	RI	<---	Ring Indicator
23*		<---	Data Rate Selector
24*	SCTE	--->	Data Rate Selector
25*		--->	Busy

In the above table, the character following the pin number means:

- \* rarely used
- + used only if secondary channel implemented
- # used only in synchronous interfaces

Although originally all DB25 RS-232 ports were female, most personal computers which have a DB25 RS-232 connector use a male connector. The female DB25 connector on a personal computer is most likely the parallel printer port and should never be connected to any RS-232 device.

As is indicated above, the 25 RS-232 standard has the option of 2 data channels, each with their own handshake lines, and the option of synchronous link. These functions are rarely used and have been left off of the newer 9 pin ports.

**Table A-2 Type A 9 pin RS-232 port**

Pin	Name	Function
1	DCD	Data Carrier Detect
2	TD	Transmitted Data
3	RD	Received Data
4	DTR	Data Terminal Ready
5	SG	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send
9	RI	Ring Indicator

**Table A-3 Type B 9 pin RS-232 port**

Pin	Name	Function
1	DCD	Data Carrier Detect
2	RD	Received Data
3	TD	Transmitted Data
4	DTR	Data Terminal Ready
5	SG	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send
9	RI	Ring Indicator

The TYPE B connection is the most common on 9 pin personal computer ports. These ports are usually male connectors.

Determining the type of RS-232 port with a voltmeter.

It is possible to determine the type of port with the use of a voltmeter using the following procedure:

- 1 Set the voltmeter for DC volts, 30 volt range. The voltage being read likely be negative and be within the range of +3VDC to -15VDC.
- 2 Power up the equipment.
- 3 Place the negative probe (black) of the voltmeter on the SG pin of the port. (Pin 7 of a DB25 port or pin 5 of a DE9 port)
- 4 Place the positive probe (red) of the voltmeter on pin 2.
- 5 Write down that voltage.
- 6 Place the positive probe (red) of the voltmeter on pin 3.
- 7 Write down that voltage.

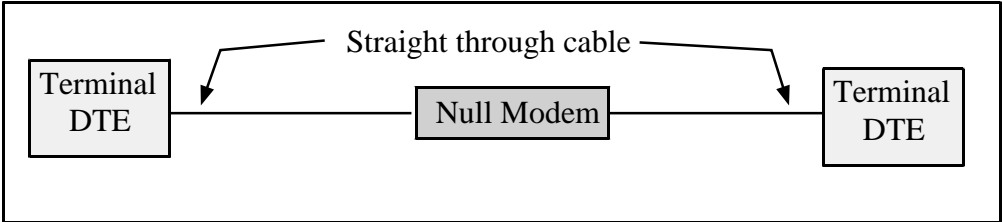
The TX voltage is within the range of -15V to -5V.

The RX voltage is within the range of -3V to +3V.

Therefore, if pin 2 is the more negative voltage of the two, the serial port is **TYPE A**. If pin 3 is the more negative voltage, the serial port is **TYPE B**.



Since almost every device that is not a modem is a DTE, it is quite common to connect DTE to DTE without a modem pair. A "null modem" connection has been established to simplify this situation. It simply reverses the transmit and receive connections from one side of the connector to another and jumpers some of the hardware handshake lines.



**Figure A-2 Null Modem connection**

The null modem is frequently a small enclosure with one male connector and one female connector. Sometimes the null modem is built into a single cable to connect the DTE device to another DTE device. In addition to the crossing of the transmit and receive pins, some additional connections to the hardware handshake pins is usually made the pinouts of null modems for 25 pin and 9 pin connections are shown below.

**Table A-4 DB25 Null Modem**

Male Connector		Female Connector
1	_____	1
2	_____	3
3	_____	2
4	_____	5
5	_____	4
6	_____	20
8	_____	6
20	_____	8
7	_____	7

**Table A-5 DE9 Null Modem**

Male Connector		Female Connector
2	_____	3
3	_____	2
5	_____	5
7	_____	8
8	_____	7
1	_____	4
6	_____	1
4	_____	6

## Electrical characteristics of RS-232

The RS-232 interface is an Single-Ended driver with an open ended receiver. The driver asserts a voltage between -5V and -15V relative to the Signal Ground to represent the MARK state (Logic TRUE). The driver asserts a voltage between +5V and +15V relative to the Signal Ground to represent a SPACE state (Logic FALSE). The fact that there may be a 30 volt swing between MARK and SPACE conditions may lead to problems with the slew rate of the signal due to the capacitance of the cable. If the cable run is long and communication problems are occurring, try lowering the baud rate.

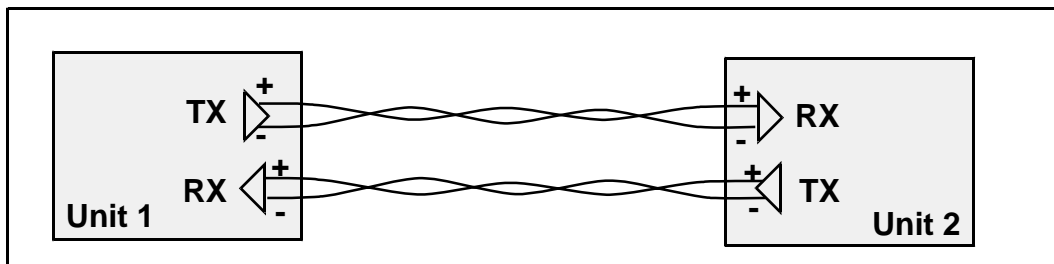
## RS-422

The RS-422 interface uses a closed ended driver and a closed ended receiver. The RS-232 interface is ground referenced. This can cause trouble in situations where a common mode induced noise may severely affect the signal by changing the reference. A better solution for noise immunity is to convert the ground referenced data at the transmission end into a differential signal and transmit this down a balanced, twisted-pair line. At the receiving end any induced noise voltage will appear equally on each line. If the receiver only looks at the differential signal, any induced common mode voltage will be rejected. This is the idea behind the RS-422 and RS-485 standards.

The RS-422 interface typically used in a point to point connection using a pair of wires from the transmitter of Unit 1 to the receiver of the Unit 2. Another pair of wires is from the transmitter of Unit 2 to the receiver of Unit 1. This connection allows for full duplex operation, i.e. the units can transmit messages while they are receiving messages. The Square D SY/MAX family of PLCs us RS-422 communication. Their standard pinout is as follows:

**Table A-6 SY/MAX DE9S RS-422 Pinout**

Pin	Function	Description
1	TX -	Transmit Data from device. (Data OUT) (inverted)
2	TX+	Transmit Data from device. (Data OUT) (non-inverted)
3	RX-	Receive Data (Data IN) (inverted)
4	RX+	Receive Data (Data IN) (non-inverted)
5	CTS-	Clear to Send (inverted)
6	RTS-	Request to Send (inverted)
7	CTS+	Clear to Send (non-inverted)
8	RTS+	Request to Send (non-inverted)
9	Shield	Shield Ground. AC coupled to chassis.



**Figure A-3 RS-422 Setup**

Like RS-485, the RS-422 protocol may be used in a multidrop configuration as shown in Figure A-4. The RS-485 standard requires a minimum capability of 32 receivers on the network. The RS-422 standard places no minimum requirement and therefore is typically used in point to point or as the host of a RS-485 network.

Always connect the + terminals on the TX to the + terminals on the RX. Similarly, connect the - terminals on the TX to the - terminals on the RX.

**Note:** Occasionally the manufacture incorrectly labels the polarity of the connections. If the system is not working try exchanging the polarity of the TX pair and the RX pair on the host.

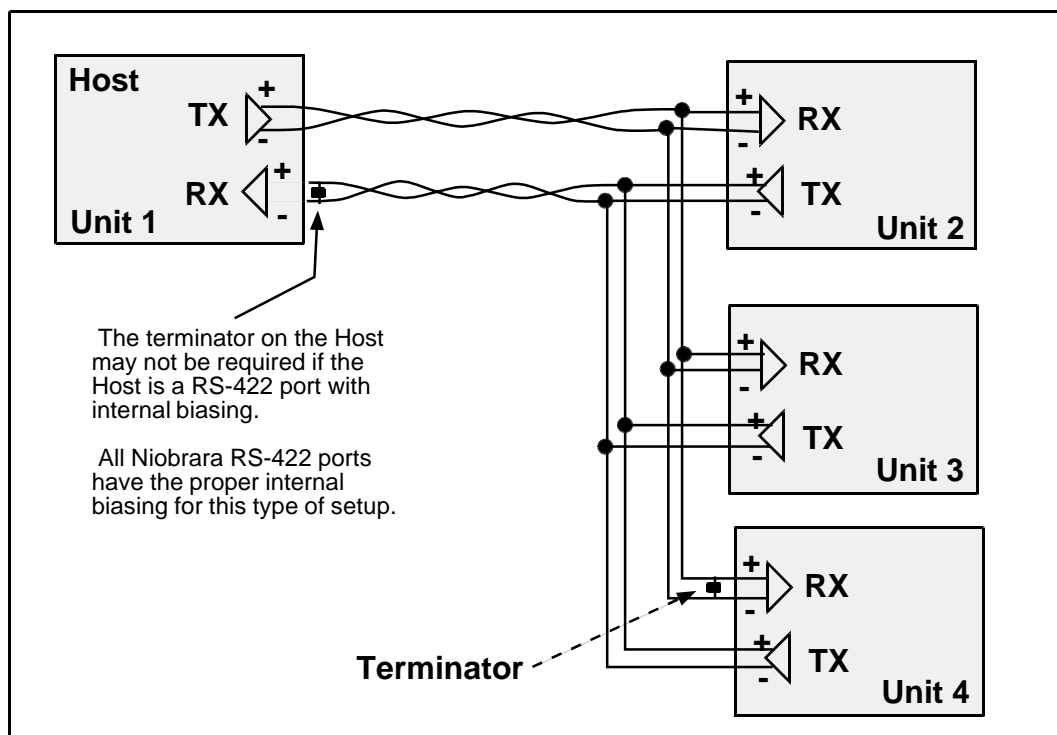
With the high noise immunity and low voltage swings, the RS-422 interface may have long runs of up to 10,000 feet.

Electrical Characteristics of RS-422.

The driver asserts a negative voltage across the receiver to represent a MARK state, a positive voltage to represent the SPACE state. The receiver triggers off of the transition through the zero voltage point.

### RS-485 (four wire)

The RS-485 interface is like the RS-422 interface with the exception that the transmitters are able to tri-state, i.e. float. This allows up to 32 transmitters to be connected to a host receiver and multiple receivers to be connected to a host transmitter. This technique is called multidropping and is shown in Figure A-4. Square D PowerLogic Circuit monitors use this type of RS-485 for communication.



**Figure A-4 RS-485 Four Wire Setup**

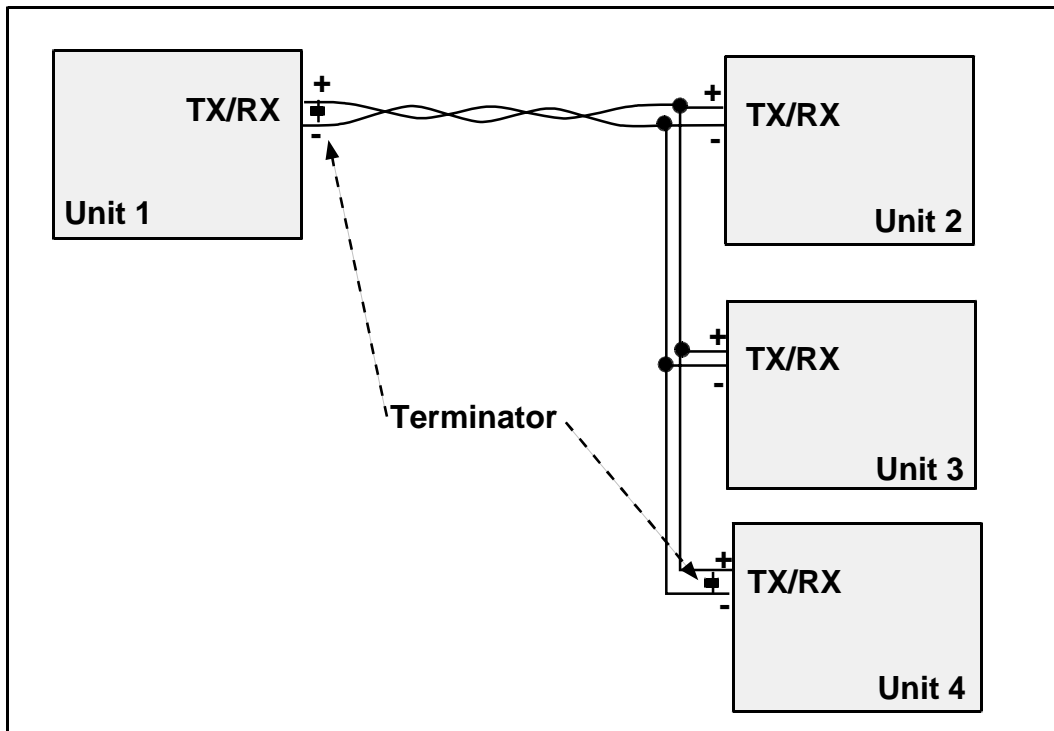
As shown in Figure A-4, it is possible to use a 4 wire RS-422 port to drive a 4 wire RS-485 multidrop network. If the RS-422 port does not have internal biasing on the RX pair it may be necessary to add a terminator to that end of the network. The terminator is a resistor correctly matched to the line which reduces reflections on the network.

### RS-485 (two wire)

Another version of the differential communication system is the RS-485 two wire network. The two wire system is a half-duplex connection where each unit transmits and receives on the same pair of wires. Only one transaction may occur on the network at one time as opposed to the four wire system where the units may transmit while they are receiving. The two wire system is inexpensive to install because only one twisted pair cable is needed. All the + terminals are connected together, all the - terminals are connected together. A terminating resistor is usually required on each end.

Since each unit can listen to the transmissions of every other unit on the network, peer to peer communication is available. The trade off is that the half-duplex connection has half of the throughput of the full duplex four wire system. A typical installation is shown in Figure A-5.

The Allen-Bradley® Data Highway is an example of a two wire RS-485 multidrop network.



**Figure A-5 RS-485 Two wire Multidrop Setup**

**Note:** Occasionally the manufacture incorrectly labels the polarity of the connections. If the system is not working try exchanging the polarity of the TX/RX pair on the unit.

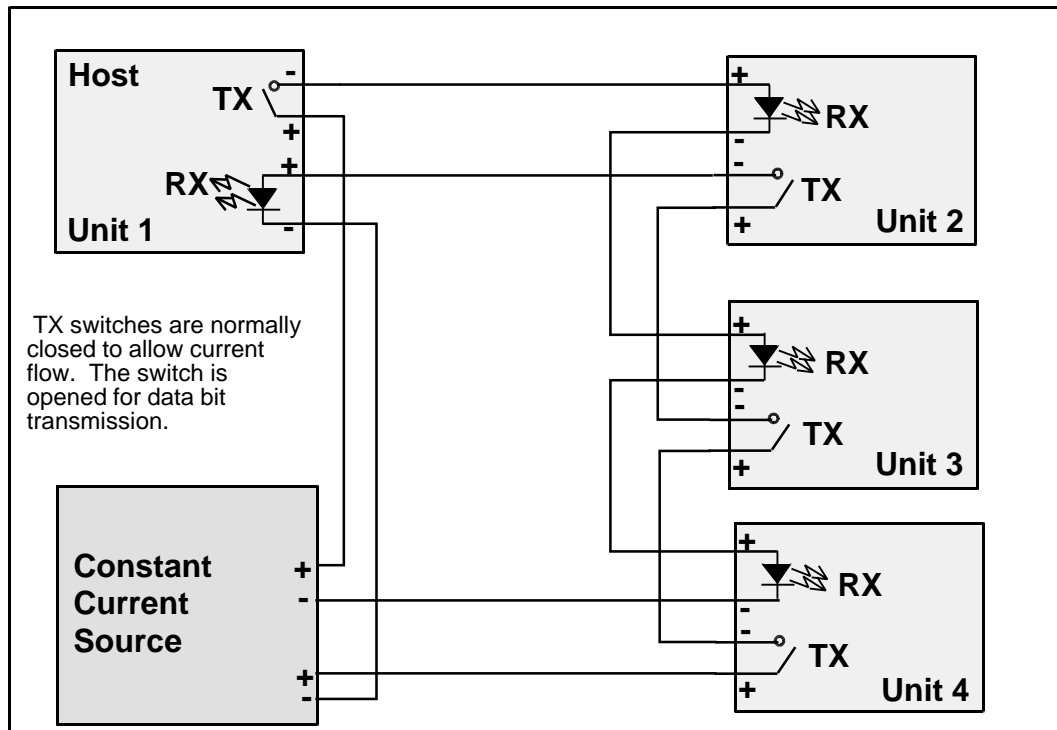
## 20mA Current Loop

The 20mA Current Loop is another multidrop configuration. The transmitting mechanism may be explained as simply opening a normally closed switch for the data bit transmission. The receiver is usually a optical isolator (LED/phototransistor) unit. Each loop is powered by a Constant Current Source. The source may be part of one of the units or may be a separate device as shown in Figure A-6.

As in other multidrop schemes, each unit watches the RX line for its messages with its address. The number of units on the network is dependent upon the addressing scheme as well as the voltage supply of the constant current source. Each RX receiver on the circuit will cause a voltage drop in the supply current. The series sum of these drops must be a value less than the available voltage from the current supply or none of the receivers will work.

To check, add up the voltage drops around the loop and make sure that the sum is less than the compliance voltage of the current source.

Some Red Lion Controls equipment such as the Apollo message centers use a 20mA current loop for serial communication.



**Figure A-6 20mA Current Loop (Full Duplex)**

A half-duplex 20mA current loop may be formed by connecting the a single current supply through all RX and TX connections serially.

## Hardware Handshaking

Sometimes it is necessary for the transmitting and receiving devices to signal each other to control the flow of data. This may be done with special characters in the software (see Software Handshaking on page 104.) or with physical connections to the hardware. This physical signaling system is known as hardware handshaking. The electrical characteristics of the handshaking lines are the same as the TX/RX lines of the system. It may be a single-ended, ground referenced signal in RS-232, or a differential pair in RS-422 or RS-485. The two most common types of hardware handshaking are: Request to Send (RTS) and Clear to Send (CTS).

**Request to Send (RTS)** is employed in two different modes: Request to Send (also known as Push to Talk) and Restraint.

**Request to Send (Push to Talk)** mode is used by the transmitting device to signal the receiving device that it has data to send. This is typically used in modem and radio modem setups. When the transmitting device has data to send, it asserts its RTS. When the modem is ready, it asserts the CTS of the transmitting device to allow it to transmit. The transmitting device will keep RTS asserted until it has sent all of its data.

Push to talk mode is the original mode of the RS-232 standard for connecting DTE to DCE. This mode of RTS usage is not commonly used in most serial communication situations and is usually restricted to modems and radios.

**Restraint** mode is used by the receiving device to signal the transmitting device that it is sending data too quickly for to be processed. When this situation occurs, the receiving device negates its RTS to signal the transmitting device to stop. When the receiving device has "caught up" it asserts its RTS line to signal the transmitting device to continue.

Restraint mode is the most common handshaking mode.

Sometimes if two RS-232 DTE devices are connected together, the DTR pin may be used as the restraint handshake line instead of RTS.

**Clear to Send (CTS)** is used to signal the transmitting device that the receiving device is ready to accept data. The transmitting device will not send data until its CTS is asserted.

In a typical hardware handshaking application RTS on the one device is usually connected to CTS on the other.

The vast majority of cases do not require active hardware handshaking. In this case it is common to jumper the RTS of each unit to its own CTS. This will allow each port to transmit at will.

## Software Overview

### Binary Representation of Data

With the implementation of an appropriate hardware system, a uniform system must be implemented to allow the data to be transferred.

In a synchronous serial system, each bit of data is transferred with a clock signal. This ensures that the receiving device "knows" when each bit is transferred so it can reconstruct the data. The throughput is determined by the number of data bits and the clock frequency. The data synchronization is accomplished by the transmission of a sync character which is an out-of-band character.

The more common type of serial communication is asynchronous. In an asynchronous system the character timing is determined by a local clock at each end. This clock usually runs at 16 times the baud rate. Each bit is to stay in its MARK or SPACE state for the time determined by the baud rate. Data is transmitted in packets of the following form: Start bit; Data Bits, Parity Bit, Stop Bits.

### Start Bit

The Start bit is always a SPACE bit and signifies the beginning of the data packet.

### Data Bits

Typically 7 or 8 bits. On rare occasions may be 5 or 6 bits. The if the data is logic true it is in the MARK state. If the data is false it is in the SPACE state. The data is transmitted LSB first.

### Parity Bit

The parity bit is a bit that follows the last data bit. Its determination is based upon the type of parity selected. The most common types are ODD, EVEN, NONE, MARK, and SPACE. These parity selections are calculated as follows.

**Odd:** The parity bit is set to the MARK state if the number of data bits in the MARK state are even. Therefore the total number of data bits in the MARK state plus the parity bit is an **ODD** number.

**Even:** The parity bit is set to the MARK state if the number of data bits in the MARK state are odd. Therefore the total number of data bits in the MARK state plus the parity bit is an **EVEN** number.

**None:** No parity bit. The stop bit comes immediately after the last data bit.

**Mark:** Always logic high. Like an additional stop bit. The mark parity bit is usually used in conjunction with 7 bit ASCII data.

**Space:** Always logic low. The space parity bit is usually used in conjunction with 7 bit ASCII data.

### Stop Bit

The Stop bit is always a MARK bit. There is always one stop bit, sometimes two stop bits, and on rare occasions 1.5 stop bits. The receiving device starts timing for the next start bit half way through the last stop bit in the message. A message with two stop bits is the same as a message with one stop bit plus one extra character time worth of dead space between packets. Therefore a device set for 1 stop bit can also receive a message with 2 stop bits.

It is worth noting that the total number of bits in a packet is sometimes used to reference the type of communication. For instance, Square D SY/MAX packets have:

1 start bit + 8 data bits + 1 parity bit + 1 stop bit = 11 bits.

Most Hayes modems can only handle a 10 bit protocol. Now this may be:

1 start bit + 8 data + 0 parity + 1 stop = 10 bits  
or 1 start bit + 7 data + 1 parity + 1 stop = 10 bits  
or 1 start bit + 7 data + 0 parity + 2 stop = 10 bits

Since the Hayes modem is designed to handle only 10 bits per character, this explains why it is not possible to send an 11 bit protocol like SY/MAX across these modems. The bit count gets off when the 10th bit arrives at the modem and the modem expects a start bit instead of the stop bit. An 11 bit modem is required for this type of communication.

## Message Determination

With the above descriptions of an asynchronous serial packet, binary data may be sent from one device to another.

### Hexadecimal numbers

Binary representation is somewhat cumbersome to deal with so hexadecimal numbers are often used. Hexadecimal (hex) numbers are base sixteen numbers. There are 16 digits in hex, the ten decimal digits 0 through 9 plus the six letters A through F which represent the decimal numbers 10 through 15. The following is a table of the decimal numbers 0 through 31, their hexadecimal equivalent and their binary equivalent.

**Table A-7 Decimal, Hex, Binary**

Dec	Hex	Binary		Dec	Hex	Binary
0	0	0000		16	10	0001 0000
1	1	0001		17	11	0001 0001
2	2	0010		18	12	0001 0010
3	3	0011		19	13	0001 0011
4	4	0100		20	14	0001 0100
5	5	0101		21	15	0001 0101
6	6	0110		22	16	0001 0110
7	7	0111		23	17	0001 0111
8	8	1000		24	18	0001 1000
9	9	1001		25	19	0001 1001
10	A	1010		26	1A	0001 1010
11	B	1011		27	1B	0001 1011
12	C	1100		28	1C	0001 1100
13	D	1101		29	1D	0001 1101
14	E	1110		30	1E	0001 1110
15	F	1111		31	1F	0001 1111

Notice that it takes four binary digits makes one hex digit. Conversion from binary to hex is straight forward and explains why hex is so popular in PLCs. Since the most PLC devices use 16-bit registers it

takes 4 hex digits to represent one register i.e. 1A2F. The first two characters "1A" make up the most significant byte (8 bits) and the last two characters make up the least significant byte.

## ASCII characters

ASCII is a set of 7-bit characters used in communication, computers and programmable logic controllers. The word ASCII is a acronym for American Standard Code for Information Interchange. ASCII is a way to interpret 7 bits (or 8 bits with a leading 0) as alphanumeric characters. There is an ASCII table following this section for your reference.

In ASCII, the small letter "p" is represented by the binary number 0111 0000. This is the hexadecimal number 70 or the decimal number 112.

Similarly, the capital letter "E" is represented by the binary number 0100 0101 which is the hexadecimal number 45 or the decimal number 69.

## Software Handshaking

It is common for communicating devices to need to exert control over each other. Typically the receiving device will need to force the transmitting device to stop and wait for it to catch up. This may be done at the hardware level (See Hardware Handshaking on page 101.) or at the software level using special characters. A common way for devices communicating with ASCII is to use the X-ON/X-OFF characters.

### X-ON

The X-ON character is the ASCII character DC1, which has the decimal value 17, hex value 11, and may be generated by pressing Ctrl+Q.

When a device receives an X-ON it starts sending data from the position at which it received an X-OFF character. The X-ON is only acted upon if the device is first halted with an X-OFF.

### X-OFF

The X-OFF character is the ASCII character DC3, which has the decimal value 19, the hex value 13, and may be generated by pressing Ctrl+S.

When a device receives an X-OFF it stops sending data. It will remain in this state until an X-ON character is received. At this time, it will resume sending the data.

Obviously software handshaking is only useful when devices are sending characters for the handshake which are out of the normal range of data. Sometimes certain protocols get around this by specifying special escape sequences that the receiving device will recognize. This is done by methods such as sending that special character once if it is to be recognized as a control character and twice if it is to be included as data.



**Table A-8 ASCII Table**

Hex	Dec	Character	Description	Abrv	Hex	Dec	Char.	Hex	Dec	Char.	Hex	Dec	Char.
00	0	[CTRL]@	Null	NUL	20	32	SP	40	64	@	60	96	`
01	1	[CTRL]a	Start Heading	SOH	21	33	!	41	65	A	61	97	a
02	2	[CTRL]b	Start of Text	STX	22	34	"	42	66	B	62	98	b
03	3	[CTRL]c	End Text	ETX	23	35	#	43	67	C	63	99	c
04	4	[CTRL]d	End Transmit	EOT	24	36	\$	44	68	D	64	100	d
05	5	[CTRL]e	Enquiry	ENQ	25	37	%	45	69	E	65	101	e
06	6	[CTRL]f	Acknowledge	ACK	26	38	&	46	70	F	66	102	f
07	7	[CTRL]g	Beep	BEL	27	39	'	47	71	G	67	103	g
08	8	[CTRL]h	Back space	BS	28	40	(	48	72	H	68	104	h
09	9	[CTRL]i	Horizontal Tab	HT	29	41	)	49	73	I	69	105	i
0A	10	[CTRL]j	Line Feed	LF	2A	42	*	4A	74	J	6A	106	j
0B	11	[CTRL]k	Vertical Tab	VT	2B	43	+	4B	75	K	6B	107	k
0C	12	[CTRL]l	Form Feed	FF	2C	44	,	4C	76	L	6C	108	l
0D	13	[CTRL]m	Carriage Return	CR	2D	45	-	4D	77	M	6D	109	m
0E	14	[CTRL]n	Shift Out	SO	2E	46	.	4E	78	N	6E	110	n
0F	15	[CTRL]o	Shift In	SI	2F	47	/	4F	79	O	6F	111	o
10	16	[CTRL]p	Device Link Esc	DLE	30	48	0	50	80	P	70	112	p
11	17	[CTRL]q	Dev Cont 1 X-ON	DC1	31	49	1	51	81	Q	71	113	q
12	18	[CTRL]r	Device Control 2	DC2	32	50	2	52	82	R	72	114	r
13	19	[CTRL]s	Dev Cont 3 X-OFF	DC3	33	51	3	53	83	S	73	115	s
14	20	[CTRL]t	Device Control 4	DC4	34	52	4	54	84	T	74	116	t
15	21	[CTRL]u	Negative Ack	NAK	35	53	5	55	85	U	75	117	u
16	22	[CTRL]v	Synchronous Idle	SYN	36	54	6	56	86	V	76	118	v
17	23	[CTRL]w	End Trans Block	ETB	37	55	7	57	87	W	77	119	w
18	24	[CTRL]x	Cancel	CAN	38	56	8	58	88	X	78	120	x
19	25	[CTRL]y	End Medium	EM	39	57	9	59	89	Y	79	121	y
1A	26	[CTRL]z	Substitute	SUB	3A	58	:	5A	90	Z	7A	122	z
1B	27	[CTRL][	Escape	ESC	3B	59	;	5B	91	[	7B	123	{
1C	28	[CTRL]\	Cursor Right	FS	3C	60	<	5C	92	\	7C	124	
1D	29	[CTRL]]	Cursor Left	GS	3D	61	=	5D	93	]	7D	125	}
1E	30	[CTRL]^	Cursor Up	RS	3E	62	>	5E	94	^	7E	126	~
1F	31	[CTRL]_	Cursor Down	US	3F	63	?	5F	95	_	7F	127	DEL



---

## Appendix B

### NR&D/Online BBS and Internet

Niobrara Research & Development is currently offering a Bulletin Board Service for its customers. This valuable customer service tool makes it easy to bring the user up to date on software revisions, firmware changes, product support news, and more.

This BBS operates on a 24 hour a day basis and is accessible from any personal computer equipped with a Hayes compatible modem. **NR&D/Online** will support communications from 300 to 9600 baud, at 8 data bits, NO parity, and 1 stop bit. Set your communications software for the baud rate of your modem, 8 data bits, NO parity, and 1 stop bit, then dial (417) 624-2028 to connect to **NR&D/Online**.

Once connected, you will find a message center, product bulletins, downloadable files and software, plus other news from the NR&D product support team.

Access and online time for **NR&D/Online** is free! You simply pay for your phone call.

For more information about **NR&D/Online** call Tom Fahrig at (800) 235-6723. He will take your information to allow you to log on to **NR&D/Online**.

Niobrara is now also on the Internet at <http://niobrara.com> so check out our our home page.



## A

AIM4SW, 51

Command Line Parameters, 63

Errors, 65

Installation, 13

SY/LINK Setup, 61

SY/MAX Connection, 18

Terminal Emulator, 18

Trouble Shooting, 69

ASCII

Definition, 104

Input, 9, 19, 23

Output, 9, 21, 33

Table, 105

ASCII Editing Mode, 31, 48

ASCII Substitution Sequence, 37

## B

Baud Rate

Data Port Parameter, 27, 44

Setup Register Values, 48

Print/Prog Port Parameter, 48, 61

Setup Register Values, 48

SY/LINK Board, 62

Terminal Emulator Parameter, 62

Bulletin Board Service, 107

## C

Clear to Send (CTS), 12, 34, 70, 101, 102

Definition, 94

RS-232 Pinout, 87

RS-422 Pinout, 87

CTS Monitor, 58

## D

Data Bits

Data Port Parameter, 27, 44

Definition, 102

Print/Prog Port Parameter, 61

SY/LINK Board, 62

Terminal Emulator Parameter, 62

## H

Hardware Handshaking, 101

## I

INSTALL.EXE Program, 13

## N

NR&D/Online, 107

Null Modem, 97

## O

Options, 39, 41

Output Substitution Sequences, 37

## P

Parity Bit, 34

Data Port Parameter, 28

Data Port Setup Register, 44

Definition, 102

Print/Prog Port Parameter, 61

SY/LINK Board, 62

Terminal Emulator Setup, 63

Periodic Polling, 35

Print/Prog, 21

Baud Rate Selection, 32

SY/MAX Operation, 43

Print/Prog Port, 9, 11, 12, 13, 17, 33, 88, 91

Baud Rate Setting, 48, 61

Output Demultiplexing, 70

SY/MAX Operation, 49, 60, 85

## R

rack address, 13

Rack Start Address, 20

Request to Send (RTS), 101

- Route, 14
  - AIM4SW Setup, 61
  - SY/LINK Setup, 62
- RS-232, 9, 87, 93, 95, 96
- RS-422, 9, 88, 98
- RS-485, 99
  - 2 wire, 99
  - 4 wire, 99
- RS485, 9

## S

- SC406, 9, 13, 17, 20, 60, 88, 91
- SM285, 9. *See also* RS-485
- Software Handshaking, 104
- SPE4, 13, 60, 61
- Start Bit, 102
- Stop Bit
  - Data Port Parameter, 28, 44
  - Definition, 102
  - Print/Prog Port Parameter, 61
  - SY/LINK Board, 62
  - Terminal Emulator Setup, 62
- SY/LINK, 13, 51
  - AIM4SW Setup, 61
- SY/NET, 13, 51, 61

## W

- Wildcards, 30

## X

- X-OFF, 104
- X-ON, 104

## 2

- 20mA Current Loop, 100
- 2496X, 51