3000-SS18
SoftScreen®/Mitsubishi Melsec A Driver

P/N99980-018A

Printed in U.S.A.

Xycom Revision Record

Revision	*Description*	*Date*
A	Manual Released	10/96

Part Number: 99980-018A

Address comments concerning this manual to:

xycom

Technical Publications Department
750 North Maple Road
Saline, Michigan 48176-1292

Table of Contents

SoftScreen/Mitsubishi Melsec A Driver

This *SoftScreen* driver allows the Focal Point™ 3000 family of engines to communicate with Mitsubishi CPUs using the Mitsubishi Melsec A protocol.

The driver is installed separately from *SoftScreen*. However, once it is installed, it becomes a part of *SoftScreen* and is downloaded, along with an application, to the run-time engine.

Supported Devices

This driver supports the following devices:

- AJ71C24 communications modules
- A1SJ71C24-R4 communications modules
- A1SJ71C24-R2 communications modules

When connected to these modules, any Mitsubishi A series CPU can communicate with Xycom's Mitsubishi Melsec A driver.

Installing the Driver

Technical Note

You *must* install *SoftScreen* before you install the driver.

Because *SoftScreen* is a Microsoft Windows® 95 Operating System program, you must install the Mitsubishi Melsec A driver in Windows 95. If you have already installed this driver on your system, this installation will overwrite the current files.

To install the Mitsubishi Melsec A driver...

1. Start Windows 95.

> **Technical Note**
>
> *SoftScreen* must be closed when you install this driver. We also recommend you close all other Windows applications before you install this driver.

2. Insert the Mitsubishi Melsec A Driver Install disk in your local drive (usually drive A).
3. Click the Start button, and then select the Run command.
4. Type A:setup (or B:setup, depending on which local drive you use) in the Open text box, and then click OK or press ENTER to begin the installation.
5. Press the Next button to proceed to the next setup screen.
6. Follow the on-screen prompts to complete the installation.

   As files are being copied to your hard drive, three icons display on the left side of your workstation screen to indicate your progress.

   The far left icon indicates how much of an individual file has been transferred. The middle icon indicates how much of a floppy has been transferred. The far right icon represents the amount of space occupied on the system's hard drive before you install the driver.

Technical Note

To end the installation process at any time, select the Cancel button in the setup dialog boxes. A prompt will inform you that setup is not complete. Select the Exit Setup button if you still want to exit the installation program. If you wish to continue the installation, select the Resume button.

Uninstalling the Driver

To uninstall the Mitsubishi Melsec A driver...

1. From Windows 95, click the Start button. Select the Settings command, then Control Panel.
2. From the Control Panel, double-click on Add/Remove Programs.
3. Double-click on the Melsec A driver entry in the list of removable programs on the Install/Uninstall page.
4. Select Yes in the Confirm File Deletion dialog box.

You will be notified once the driver has been successfully uninstalled.

Connecting to Supported Devices

This section describes the serial port configuration and the cabling pinouts for connecting a 3000 engine to a Mitsubishi CPU.

Configuring the Port

A 3000 engine can communicate with a Mitsubishi CPU via RS-232C and RS-422 protocols.

Cabling

This section provides information on RS-232C and RS-422 cabling.

Electromagnetic Compatibility Warning

The connection of non-shielded equipment interface cables to the Focal Point workstations will invalidate FCC EMI and European Union EMC compliance and may result in interference and/or susceptibility levels which are in violation of relevant regulations. It is the responsibility of the system integrator and/or user to obtain and use shielded interface cables and equipment. If this equipment has more than one connector, do not leave cables connected to unused interfaces. Changes or modifications not expressly approved by the manufacturer could void the user's authority to operate the equipment.

All interface cables must include braid/foil type shields. Communication cable connectors must be metal, ideally zinc die-cast backsheet types, and provide 360° protection about the interface wires. The cable shield braid must be terminated directly to the metal connector shell; ground drain wires alone are not adequate.

RS-232C Pinouts

Figures 1 and 2 depict the 9- and 25-pin RS-232C pinouts to connect a 3000 engine to a Mitsubishi communications module.

Technical Note

When connecting via the RS-232C standard, the cable should be a Belden 9925 or equivalent, maximum length of 50 feet. Keep the cable away from high voltage and current-carrying cables. Refer to the EIA RS-232C specification for more details.

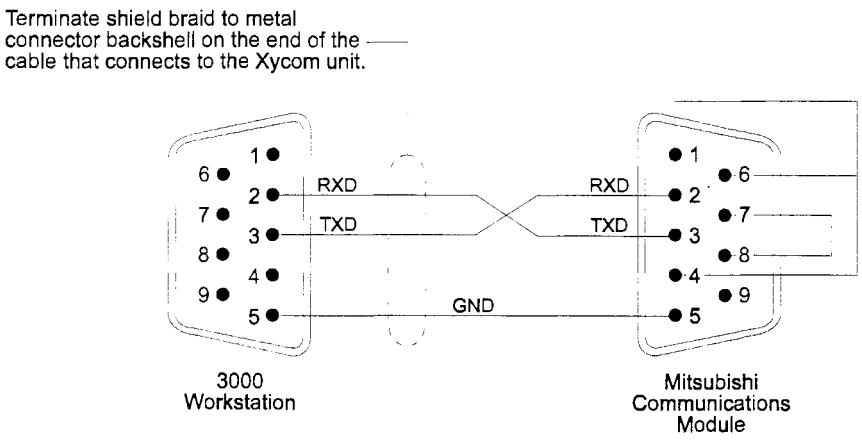

Figure 1. 9-pin RS-232C Pinout

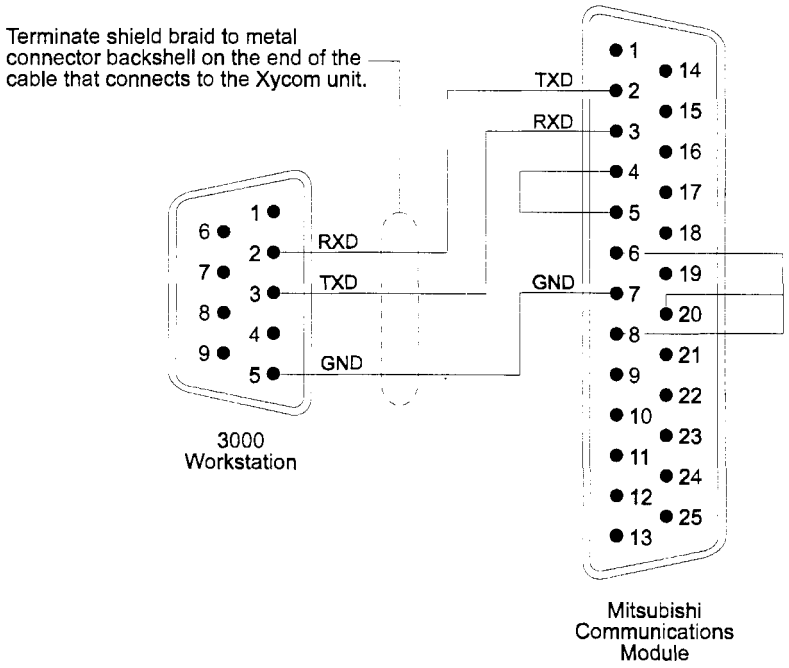

Figure 2. 25-pin RS-232C Pinout

RS-422 Pinout

Figure 3 depicts the RS-422 pinout to connect a 3000 engine to a Mitsubishi communications module.

> **Technical Note**
>
> When connecting via the RS-422 standard, use a Belden 8302 or equivalent cable, maximum length 4000 feet. Keep the cable away from high voltage and current-carrying cables. Refer to the EIA RS-422 specification for more details.

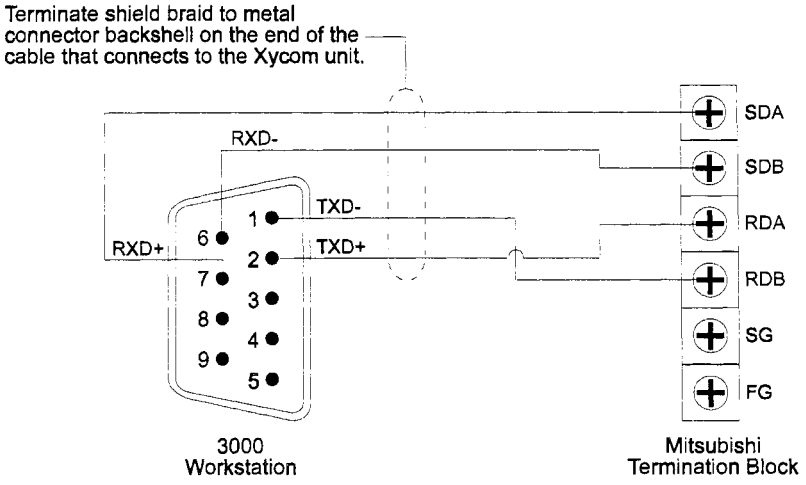

Figure 3. RS-422 Pinout

Development System Configuration

Once you have installed the driver (refer to the *Installing the Driver* section at the beginning of this manual), you must configure it in the *SoftScreen* Development System.

To configure the driver...

1. Open an application in *SoftScreen*. See the *SoftScreen Development System for Windows User's Guide* for information on creating an application.

2. Select the Drivers command on the Configure menu in the Application Navigator. The Configure Physical Drivers dialog box opens, as shown in Figure 4.

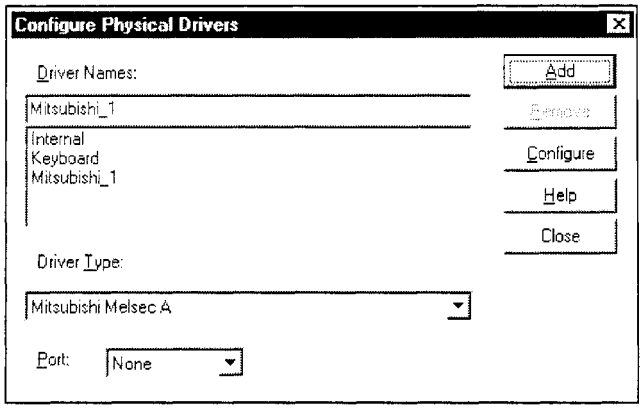

Figure 4. Configure Physical Drivers Dialog Box

3. Select Mitsubishi Melsec A from the Driver Type drop-down list box.
4. Type a unique name in the Driver Names text box, using up to 32 characters. Tag names can begin with a character or a colon, and can contain alphanumeric characters, underscores, and colons. Tag names cannot begin with, or contain, a space.
5. Select the port to which you want to connect the PLC. Choices are None, COM1, COM2, and Special. The default is None.
6. Click the Add button. The driver name is added to the Driver Names list box.
7. Highlight the name in the Driver Names list box, and then click on the Configure button.

   The Mitsubishi Melsec A Configuration dialog box opens, as shown in Figure 5.

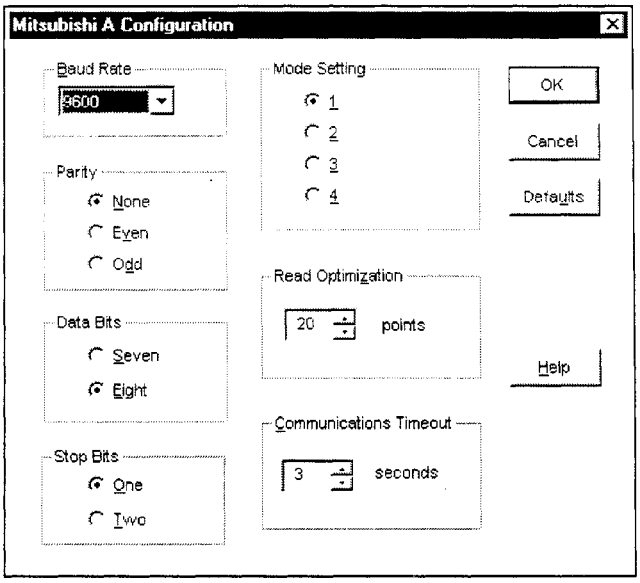

Figure 5. Mitsubishi Melsec A Configuration dialog box

This dialog box reflects the default settings. Table 1 defines the fields in this dialog box.

Table 1. Fields in the Mitsubishi Melsec A Driver Configuration Dialog Box

Field	Definition
Baud Rate	Sets the baud rate at which you will transfer data, from 300 to 19200. The default is 9600.
Parity	Sets error checking to None, Even, or Odd. The default is None.
Data Bits	Sets the number of bits used in transmission to 7 or 8. The default is 8.
Stop Bits	Sets the end of character stop bit to 1 or 2. The default is 1.
Mode Setting	Configures the driver for one of four different communication protocols, allowing communication between the host computer and up to 32 Mitsubishi Melsec A CPUs. The default is 1.
Read Optimization	Optimizes the number of data points read in a single command, from 1 to 64. The default is 20. This number can be changed to affect driver performance.

Field	Definition
Communications Timeout	Sets the time period the engine will wait for a response from the CPU before timing out, from 1 to 30 seconds. The default is 3.

8. Click OK to accept the changes you have made to these settings. If you want to revert to the default settings, click Defaults. Click Cancel to cancel any changes you have made during the current use of the dialog box.

To change settings once you have configured the driver, double-click on the driver name in the Drivers configured list box on the Application Navigator form.

Technical Note

You cannot change the port setting from the Application Navigator form. You must use the Drivers command on the Configure menu in the Application Navigator to change this setting.

Once the driver is configured, you can create tags that address data points on supported devices.

Addressing Supported Devices

SoftScreen uses tags to address data points on supported devices. Tag names can be up to 32 alphanumeric characters. Do not start tag names with a number or a space.

This section describes how to assign these tag names to data points, and defines expressions supported by the Mitsubishi Melsec A protocol.

Assigning Tag Names

To assign a tag name to a data point on a supported device...

1. Select Drivers from the Data drop-down list box on the Application Navigator form.

2. Double-click on the driver name for which you want to configure tags. The data point configuration form opens, as shown in Figure 6.

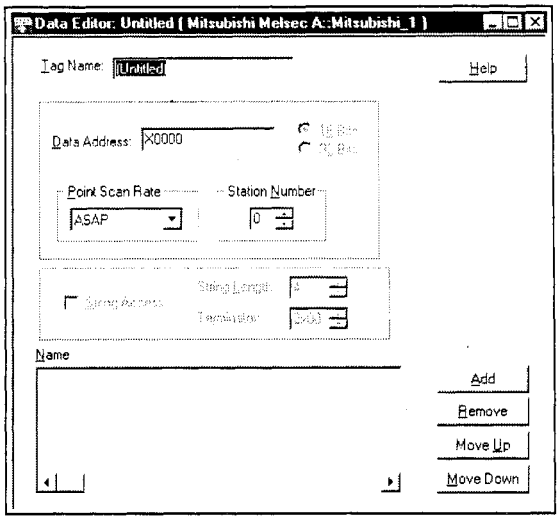

Figure 6. Mitsubishi Melsec A Data Point Configuration Form

Table 2 defines the fields in this form.

Table 2. Fields in the Mitsubishi Melsec A Data Point Configuration Form

Field	Definition
Tag Name	Defines a unique tag name.
Data Address	Links the tag to a valid address. Refer to the *Creating Valid Addresses* section for more information. The default is X0000. If you select a data type that addresses double words, the radio buttons to the right of the data address field will become active, allowing you to select 16 bits or 32 bits as the length. The default is 16 bits.
Point Scan Rate	Sets how often the run-time engine will read from the data point, from ASAP (as soon as possible) to once every eight hours. The default is ASAP.
Station Number	Specifies the address of the CPU on the multi-link network to which you will communicate, from 0 to 31 decimal. The default is 0.

Field	Definition
String Access	When checked, indicates the data address you entered in the Data address field accesses strings. If this field is checked, you must select a string length, from 1 to 128 characters (the default is 4), and a string terminator character, from 0x00 to 0xFF (the default is 0x00). When a string is read from the PLC, the driver will search for the specified terminator, replace it with a null and then store it in the run-time engine. When a string is written to the PLC, the driver will append the specified terminator at the end of the string, and then send it to the PLC. *Note: Refer to Table 3 for information on data types that access strings.*

3. Click Add to add the tag. Click Remove to delete the tag. Click Move Up or Move Down to change the order in which the tags are arranged in the list box.

Creating Valid Addresses

Table 3 defines the valid data types and file ranges for supported devices.

Warning

Do not attempt to read/write 32-bit integer values less than -16,777,216 or greater than +16,777,216. Doing so will cause unpredictable results.

Table 3. Mitsubishi Melsec A Valid Data Types and Addressing Ranges

Data Type	Valid Range	Data Size	String Support	Bit Addressing	Bit Access	Word Access
Input (X)	0-7FFh	Bit	No	No	R/W	N/A
Output (Y)	0-7FFh	Bit	No	No	R/W	N/A
Internal Relay (M)	0-2047d	Bit	No	No	R/W	N/A
Latch Relay (L)	0-2047d	Bit	No	No	R/W	N/A
Link Relay (B)	0-3FFh	Bit	No	No	R/W	N/A
Annunciator (F)	0-255d	Bit	No	No	R/W	N/A
Special Relay (M9)	0-255d	Bit	No	No	R	N/A
Timer Contact (TS)	0-255d	Bit	No	No	R/W	N/A
Timer Coil (TC)	0-255d	Bit	No	No	R/W	N/A
Counter Contact (CS)	0-255d	Bit	No	No	R/W	N/A
Counter Coil (CC)	0-255d	Bit	No	No	R/W	N/A
Timer Value (TN)	0-255d	Signed Single/Double word	No	No	No	R/W
Counter Value (CN)	0-255d	Signed Single/Double word	No	No	No	R/W
Data Register (D)	0-1023d	Signed Single/Double word	Yes	No	No	R/W
Link Register (W)	0-3FFh	Signed Single/Double word	No	No	No	R/W
File Register (R)	0-8191d	Signed Single/Double word	Yes	No	No	R/W
Special Register (D9)	0-255d	Word	No	No	No	R*
Buffer (BU)	0-7FFhh	Word	No	No	No	R/W
Special Function (SF)	SFXX-AAAAh	Byte	No	No	No	R/W (Byte)

**You can write to some special registers.*

Addressing Special Function Units

This driver allows special function unit memories to be read directly from supported communication modules. The special function unit addressing scheme is SF*XX-AAAA* where *XX* is the special function unit number and *AAAA* is the memory location in the particular module.

To calculate the special function unit number for a module, count in hexadecimal how many groups of 16 I/O points precede it in the rack. So, for example, in Figure 7, the special function unit number of the A68AD module would be 07h because one 32-point module and five 16-point modules precede it in the rack.

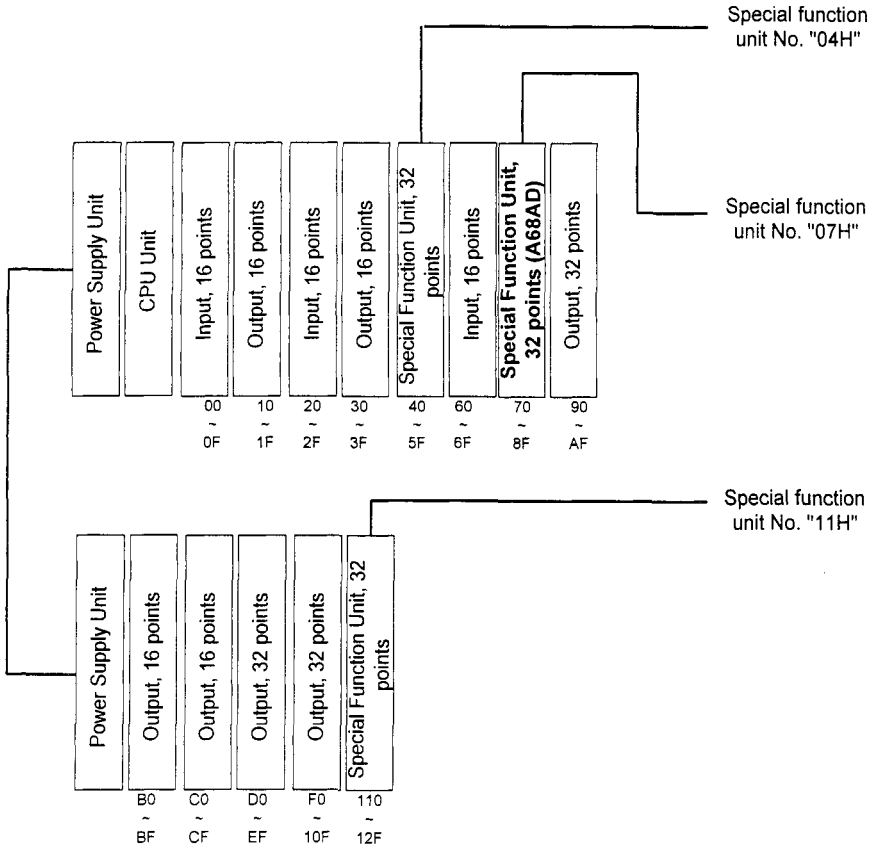

Figure 7. Special Function Unit Example

The value you calculate must correspond with the number and size of the I/O configured by the CPU programming software. If the value does not correspond, use the method specified in Figure 7, but calculate the special function unit based on what is indicated in the programming software. For instance, you could have an input module that is physically 16 points, but is specified in the programming software as 32 points. If you use the above method, you would get a different value than you would in the software.

Buffer memory addresses are unique to the special function unit which is to be accessed. Refer to the module's user documentation for specific information on buffer memory addresses.

For example, to access the buffer memory in the A68AD analog-digital converter unit in the rack configuration illustrated in Figure 7, you would first need to calculate the special unit function number. This would be 07h (the number of 16 bit groups before the A68AD module). The A68AD documentation indicates that its buffer memory starts at address 80h, and that Channel 1 averaging time is in buffer memory location 84h. So this particular module's special function unit address would be SF7-84.

Addressing Examples

Following are examples of tags that address Mitsubishi CPU data points.

Example_1 addresses input file 003F at station number 10 ASAP.

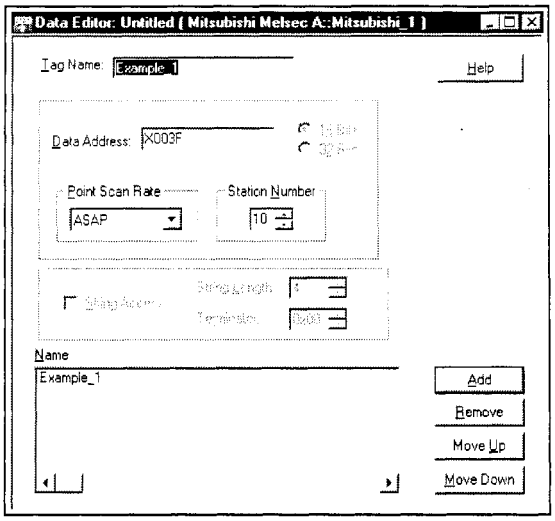

Figure 8. Mitsubishi Melsec A Addressing, Example 1

Example_2 addresses internal relay file 9255 at station number 31 every five seconds.

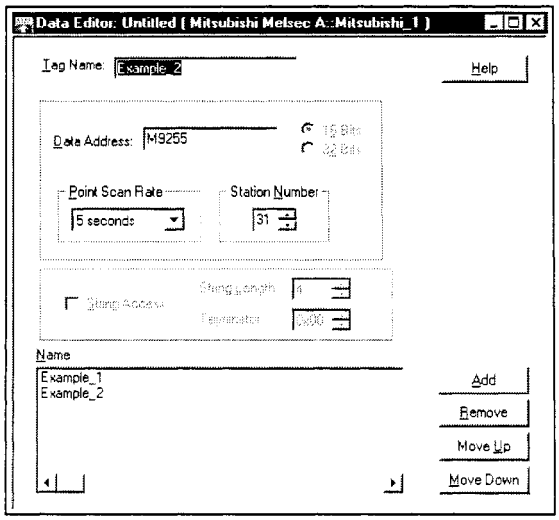

Figure 9. Mitsubishi Melsec A Addressing, Example 2

Example_3 addresses timer value file 255 at station number 16 every 30 minutes.

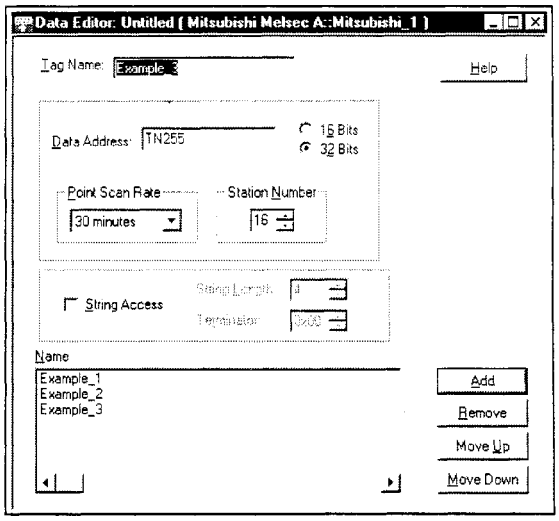

Figure 10. Mitsubishi Melsec A Addressing, Example 3

Retrieving Status Information

Use the strings described in this section to retrieve driver status information.

> **Technical Note**
>
> These strings are not case sensitive.

Driver ID

MitsADriverID returns the following null-terminated string identifying the driver running on the 3000 engine:

```
"Mitsubishi Melsec A Driver"
```

Driver Revision

MitsADriverRev returns a string identifying the driver revision level, such as "1.3."

Error Handling

MitsACommStatus returns a number describing the current communication status of the driver. Table 4 defines these status values.

> **Technical Note**
>
> There are no communication errors if the number is 0.

Table 4. Mitsubishi Melsec A Communication Status Errors

Number	Error	Description
0	No Error	No communications error
1	Timeout	PLC did not respond in the configured amount of time
2	Transmit error	The initial part of the message (before the response data) had an error.
3	Receive error	The response block contained invalid information (station number, header, or data).
4	CS error	The message received had an invalid checksum (CS).
5-7	N/A	Reserved
8	Disable during run	Invalid access has been made during RUN mode.
9	Parity error	Driver parity does not match PLC parity.
10	Sum check error	Sent sum check does not match received sum check.
11	Protocol error	Communications setting different between the driver and the PLC.
12	Framing error	Framing error
13	Overrun error	Overrun error
14	Character area error	The characters in data areas A, B, or C do not match the mode set.
15	Character error	Invalid characters have been received. Valid characters are A to Z, 0 to 9, and "-".
16	PC access error	Buffer memory is unable to make communication with the PC.
24	PC number error	Defined PC number does not exist. Must be FFh.
25	Mode error	Invalid processing mode has been sent.
26	Special function unit specification error	The addressed I/O location does not contain a special function unit.

Number	Error	Description
32	Remote error	Remote RUN/STOP impossible.
41	Special function unit bus error	Memory access to special function unit cannot be made. Faulty hardware.

Each of the data points assigned to the driver can have a different update rate, so on any given scan, some points will be scanned and some will not. When the driver detects an error (either read or write), it will post an alarm if it has not previously posted an alarm. The alarm will be posted at the bottom of the screen for three seconds. During any given scan, only the first error condition in the scan will be posted.

The alarm message that is posted will indicate the data point that caused the error. The number that indicates the type of error that occurred does not appear on screen. However, the number will be logged in the alarm summary along with the date and time of the alarm (refer to the *SoftScreen Development System for Windows User's Guide* for information on the alarm summary).

If the driver is optimizing points, it will read data points in optimized blocks, instead of one at a time. If an error occurs while the driver is reading the block, the alarm message will describe the data point that was at the beginning of the block.

For example, if the driver reads an optimized block of data points (a, b, and c), and an error occurs because data point "c" does not exist in the target device, an alarm message is posted. This message will indicate that there was an error reading data point "a," not data point "c."

Once the driver completes a scan without any errors (after an error has occurred in a previous scan), then the driver will post the following message: `"Mits A: Port(x) NO Errors"`

Communication Status

MitsACommString returns a null-terminated string describing the current communication status of the driver.

Scan Time

MitsAScanTime returns a number (in msecs) describing the amount of time it takes the driver to read the current data points. For example, if all data points are set to ASAP, the system would track the time between the starting point of the scan and the ending point, and then would display the scan time based on these two numbers. However, if one data point is set to an ASAP scan rate, and another is set to an eight-hour scan rate, the system would continue to read the ASAP point until eight hours had passed, then it would read the ASAP point and the eight-hour point, and then provide you with the time period it took for this scan to read both the points.

Index

R

S

T

U