Allen-Bradley

SLC to SCANport™ Communications Module

(Cat. No 1203–SM1)

User Manual

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation, and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss.

Attention statements help you to:

- identify a hazard
- avoid the hazard
- recognize the consequences

Important: Identifies information that is critical for successful application and understanding of the product.

Table of Contents

Using Enhanced Mode

Chapter 4

Troubleshooting

Chapter 5

Specifications

Chapter 6

M0, M1, and G Files

Appendix A

SCANport Messaging

Appendix B

Preface

Read this preface to familiarize yourself with the rest of the manual. This preface covers the following topics:

- who should use this manual
- the purpose of this manual
- safety precautions
- firmware support
- product compatibility
- terms and abbreviations
- conventions used in this manual
- Allen–Bradley support

Who Should Use this Manual?

Use this manual if you design, install, program, or troubleshoot control systems that use the Allen–Bradley SLC to SCANport communications module. You must have previous experience with and a basic understanding of electrical terminology, configuration procedures, equipment, and safety precautions for machinery and control systems.

To efficiently use this communications module, you must be able to program and operate an Allen-Bradley SLC controller.

Purpose of this Manual

This manual provides the information you need to install and use the SLC to SCANport communications module. This manual describes the procedures for installing, configuring, and troubleshooting the SLC to SCANport communications module.

For information on specific product features, refer to the product manual.

Important: Read this manual in its entirety before installing, operating, servicing, or configuring the SLC to SCANport communications module.

Contents of this Manual

This manual contains the following information:

Chapter:	Title:	Contents:
	Preface	Describes the purpose, background, and scope of this manual.
1	Overview	Provides an overview of the SLC to SCANport communications module.
2	Installing the SLC to SCANport Module	Provides the procedures you need to install your SLC to SCANport communications module and attach it to the SCANport network.
3	Using Basic Mode	Provides information that you need to configure your SLC to SCANport communications module for SLC basic mode operation.
4	Using Enhanced Mode	Provides information that you need to configure your SLC to SCANport communications module for SLC enhanced mode operation.
5	Troubleshooting	Provides information about the LED indications and fault descriptions.
6	Specifications	Provides the environmental, electrical, and communications specifications.
A	M0, M1, and G Files	Provides generic information about using the M0, M1, and G files.
B	SCANport Message Index	Provides a listing of some of the most commonly used SCANport message structures.

Safety Precautions

Please read the following safety precautions carefully.

ATTENTION: Only personnel familiar with SCANport devices and the associated machinery should plan or implement the installation, start-up, configuration, and subsequent maintenance of this communications module. Failure to comply may result in personal injury and/or equipment damage.

ATTENTION: The SLC to SCANport module contains ESD (Electrostatic Discharge) sensitive parts and assemblies. Static control precautions are required when installing, testing, or servicing this assembly. Component damage may result if you do not follow ESD control procedures. If you are not familiar with static control procedures, refer to Allen-Bradley Publication 8000-4.5.2, *Guarding against Electrostatic Damage*, or any other applicable ESD protection handbook.

SLC Product Compatibility

The SLC to SCANport module is designed to be used with any SLC processor or adapter capable of supporting SLC rack–based modules.

Terms and Abbreviations

The following terms and abbreviations are specific to this product. For a complete listing of Allen–Bradley terminology, refer to the *Allen–Bradley Industrial Automation Glossary.*

In this manual, we refer to the:

- 1203 SLC to SCANport communications peripheral as the *SLC to SCANport module*.
- Any of the connected SCANport products as the *drive* or *SCANport device*. The current list of SCANport devices include the following: 1305 MICRO, 1336 FORCE, 1336 IMPACT, 1336 PLUS, 1394 digital motion control system, SMC Dialog Plus, SMP–3 smart motor protector, and 1397 DC drive.

Common Techniques Used in this Manual

This manual follows these conventions:

- Bulleted lists provide information, not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- *Italic* type is used for emphasis and chapter names.

► We also use this convention to call attention to helpful information.

Allen–Bradley Support

Allen–Bradley offers support services worldwide, with over 75 Sales/Support Offices, 512 authorized Distributors and 260 authorized Systems Integrators located throughout the United States alone, plus Allen–Bradley representatives in every major country in the world.

Local Product Support

Contact your local Allen–Bradley representative for:

- sales and order support
- product technical training
- warranty support
- support service agreements

Technical Product Assistance

If you need to contact Allen–Bradley for technical assistance, please review the information in the *Troubleshooting* chapter first. If you are still having problems, then call your local Allen–Bradley representative.

Overview

Chapter Objectives

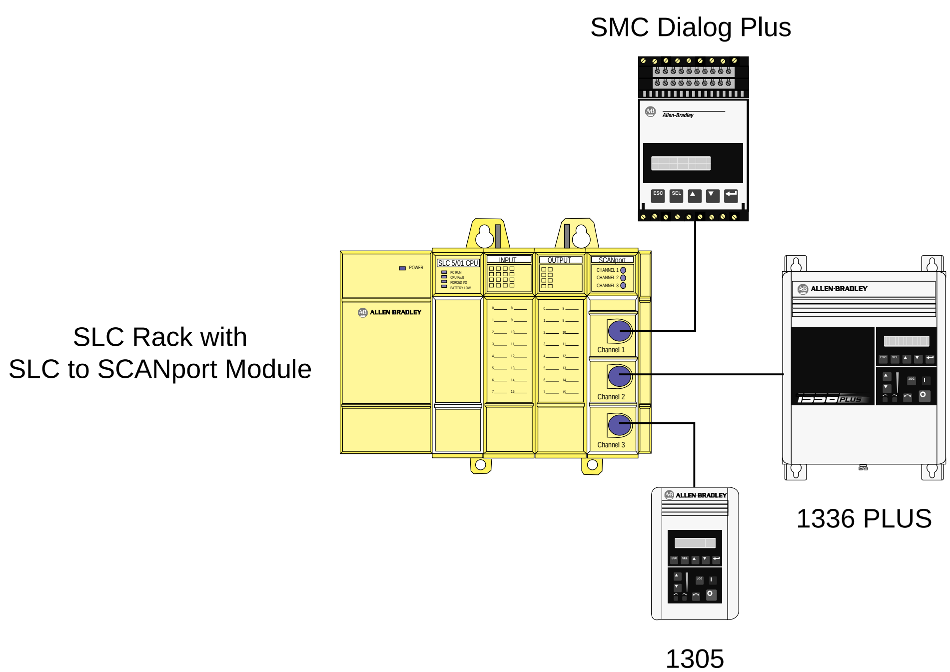
Chapter 1 provides descriptions of the following:

- the SLC to SCANport module
- the available functions

What is the SLC to SCANport Module?

The SLC to SCANport module provides an interface between any SLC processor or other product that can control modules within a SLC rack and up to three SCANport devices as shown in Figure 1.1.

Figure 1.1
Example SLC to SCANport Module Set Up



To connect more than three SCANport devices in a single rack, add additional SLC to SCANport modules to your SLC rack system.

You can use your SLC to SCANport module in a 4, 7, 10, or 13 slot SLC rack or a 2–slot expansion rack available for the fixed I/O configurations of SLC–500 processors.

Important: An SLC rack using this module needs an enclosure of at least 200 mm (8 in) in depth. You cannot place an SLC rack using the SLC to SCANport module in a 150 mm (6 in) deep enclosure.

What Functions Does the SLC to SCANport Module Provide?

Your SLC processor or rack adapter determines what functions are available for the SLC to SCANport module.

If the device in the left hand slot of the SLC chassis is a:	Is basic mode supported?	Is enhanced mode supported?
Fixed style controller (using an expansion rack)	Yes	No①
SLC 5/01 controller	Yes	No
SLC 5/02, 5/03, or 5/04 controller	Yes	Yes
1747–OC open controller	Yes	Yes
Any SLC rack adapter	Yes	No①

① Future SLC product offerings may support enhanced mode communication.

The following table provides information about basic mode and enhanced mode.

Function	Basic Mode	Enhanced Mode
Maximum number of words of I/O per SCANport device	2	10
Total number of words of I/O for module	8	32
16–bit Logic Command (to SCANport device)	Yes	Yes
16–bit Logic Status (from SCANport device)	Yes	Yes
16–bit Reference (to SCANport device)	Yes	Yes
16–bit Feedback (from SCANport device)	Yes	Yes
Datalinks	No	Yes
Safe State Data	No	Yes
Messaging	No	Yes

Installing the SLC to SCANport Module

Chapter Objectives

Chapter 2 covers the following information:

- what you need to do before you install the SLC to SCANport module
- how to install the SLC to SCANport module
- how to remove the SLC to SCANport module

Important: You cannot place an SLC rack unit containing an SLC to SCANport module in an enclosure that is less than 200 mm (8 in) deep.

Before You Install the Module

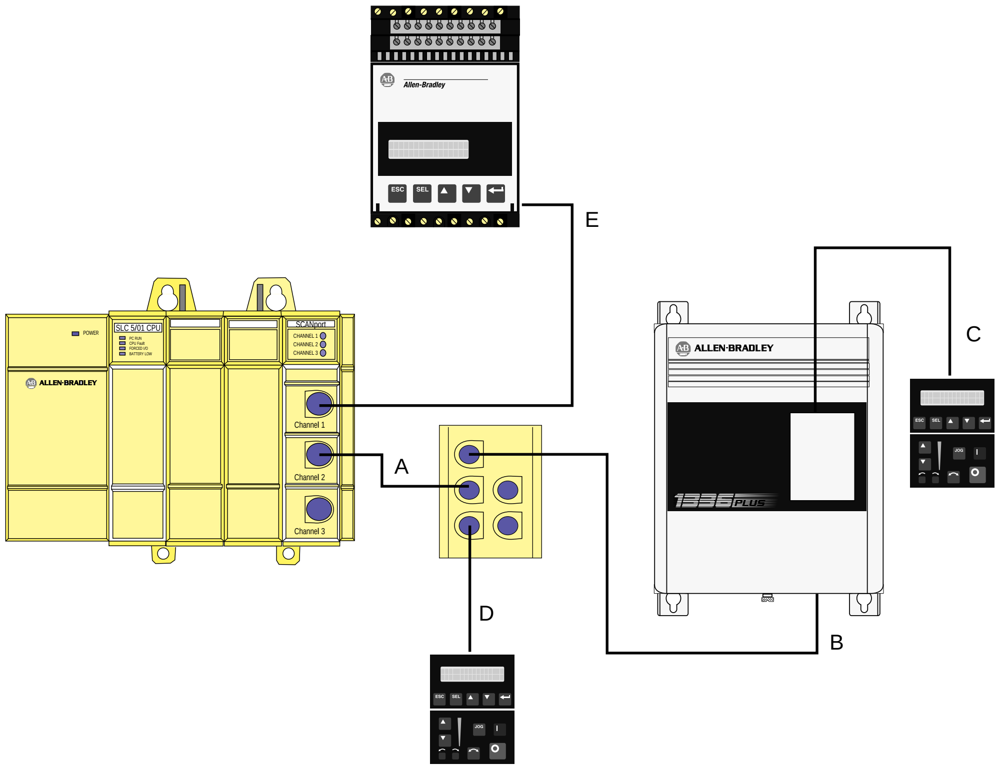
Before you install your SLC to SCANport module, you need to:

- Determine the length of your SCANport cable(s).
- Determine the placement of your SCANport cables.
- Locate the DIP switch on your SLC to SCANport module.

Determine the Length of the SCANport Cable(s)

To connect your SLC to SCANport module to a SCANport device, you need to use an Allen–Bradley SCANport cable. The maximum cable length between any two peripheral devices connected to any SCANport device cannot exceed 10 meters (33 feet). Therefore, in Figure 2.1, A+B+C≤10 meters and D+B+C≤10 meters. However, you would not add the length of cable E to cables A, B, C, or D because it connects to a separate SCANport device (or channel).

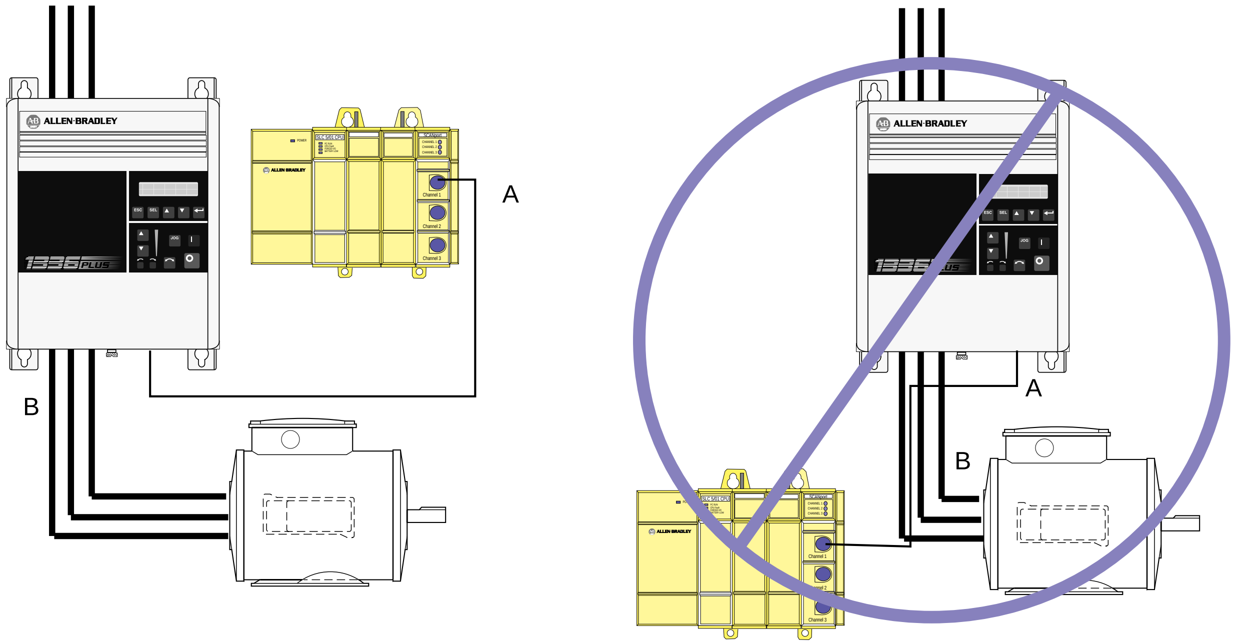
Figure 2.1
Example of SCANport Cable Lengths



Determine the Placement of the SCANport Cables

You must keep the SCANport cables away from high power cables. If your SCANport cables are placed too close to the power cables or run in parallel with power cables, you may introduce noise into the communications system, which can cause problems to your system. Make sure you physically mount and connect SCANport products following the installation guidelines available for each product.

Figure 2.2
Examples of Cable Placements

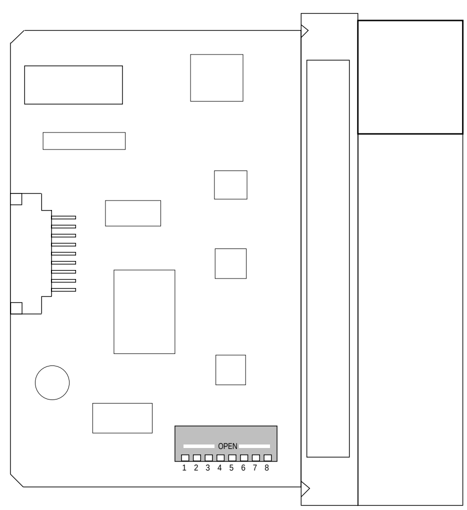


A = Communications wire
B = Power wires

Locate the DIP Switch

You also need to locate a single configuration DIP switch on the module as shown in Figure 2.3.

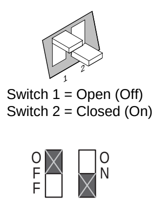
Figure 2.3
Side View of the SLC to SCANport Module Showing DIP Switch Location

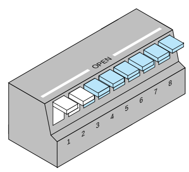
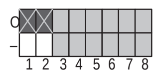
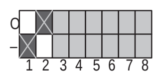
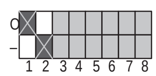
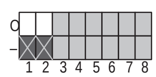
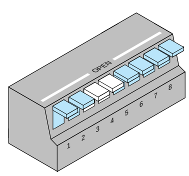
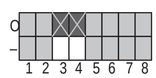
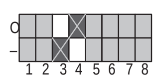
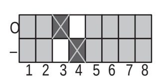
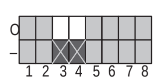
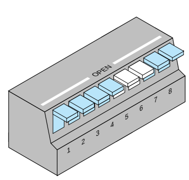
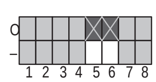
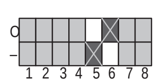
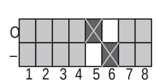
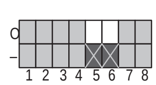


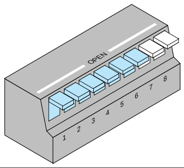
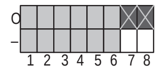
Installing the SLC to SCANport Module

To install the SLC to SCANport module into the chassis:

1. Set the DIP switches. For each SCANport device connected to the SLC to SCANport module, you need to set two DIP switches to select what happens when the SLC processor or rack adapter faults or is placed in program for the appropriate channel.



Set this DIP switch:	To these settings:			To:
Channel 1 Fault/ Program State (Switches 1 and 2) 		SW1 Open (Off)	SW2 Open (Off)	Fault SCANport device (default)
		SW1 Closed (On)	SW2 Open (Off)	Zero data
		SW1 Open (Off)	SW2 Closed (On)	Hold last state
		SW1 Closed (On)	SW2 Closed (On)	Safe state data (enhanced mode only), Fault (basic mode only)
Channel 2 Fault/ Program State (Switches 3 and 4) 		SW3 Open (Off)	SW4 Open (Off)	Fault SCANport device (default)
		SW3 Closed (On)	SW4 Open (Off)	Zero data
		SW3 Open (Off)	SW4 Closed (On)	Hold last state
		SW3 Closed (On)	SW4 Closed (On)	Safe state data (enhanced mode only), Fault (basic mode only)
Channel 3 Fault/ Program State (Switches 5 and 6) 		SW5 Open (Off)	SW6 Open (Off)	Fault SCANport device (default)
		SW5 Closed (On)	SW6 Open (Off)	Zero data
		SW5 Open (Off)	SW6 Closed (On)	Hold last state
		SW5 Closed (On)	SW6 Closed (On)	Safe state data (enhanced mode only), Fault (basic mode only)

Set this DIP switch:	To these settings:			To:
Switches 7 and 8 		SW7 Open (Off)	SW8 Open (Off)	Reserved

▶ If you select a state other than Fault, the channel enable bits located in the first two words of output data will retain their last state values. This ensures that the enabled SCANport connections remain active for those states. The I/O data transferred to the SCANport device will change as configured by the DIP switch.

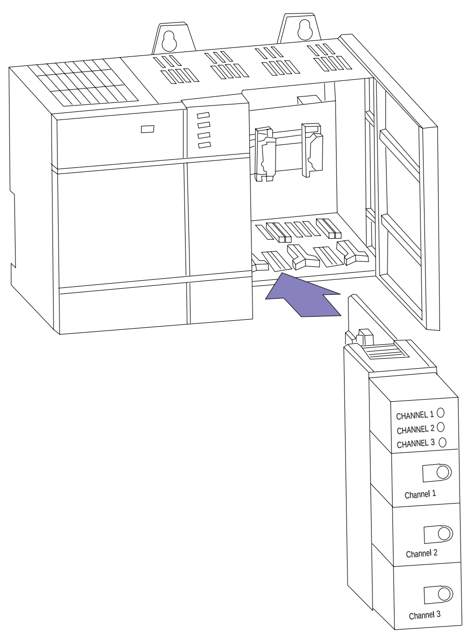
2. Turn off the chassis power supply.

ATTENTION: Do not install the SLC to SCANport module with the chassis power supply on. Inserting or removing the module with the chassis power supply on may damage the module.

---

3. Select a slot for the module in the chassis. You may use any slot except the leftmost slot, which is reserved for the SLC 5/xx processor or rack adapter.

4. Insert the module into the slot you have selected.



5. Apply firm, even pressure to seat the module in the I/O chassis backplane connectors. Make sure the plastic tabs snap into the rack.
6. Connect the SCANport cable(s) from the SCANport device(s) to the SCANport connections in the front of the module.

   Important: You must keep in mind that the maximum cable distance between any two devices connected to a single channel cannot exceed 10 meters (33 feet) of cable. Also, the SCANport cables must not be in close contact with the power cables.

You can insert or remove SCANport cables while a rack is powered. If a cable is removed while the channel is enabled, the connected SCANport device will fault unless otherwise configured at the SCANport device.

Removing the SLC to SCANport Module

To remove the SLC to SCANport module from the chassis, you need to:

1. Remove the SCANport cables.
2. Make sure the rack power is removed.
3. Push in on the hooks on both ends of the module.
4. Gently pull the module from the chassis.

Where Do I Go From Here?

The SLC to SCANport module can operate in either basic mode or enhanced mode. Refer to Chapter 1 for a description of basic mode and enhanced mode.

If you plan to use:	Go to:
Basic mode	Chapter 3
Enhanced mode	Chapter 4

Using Basic Mode

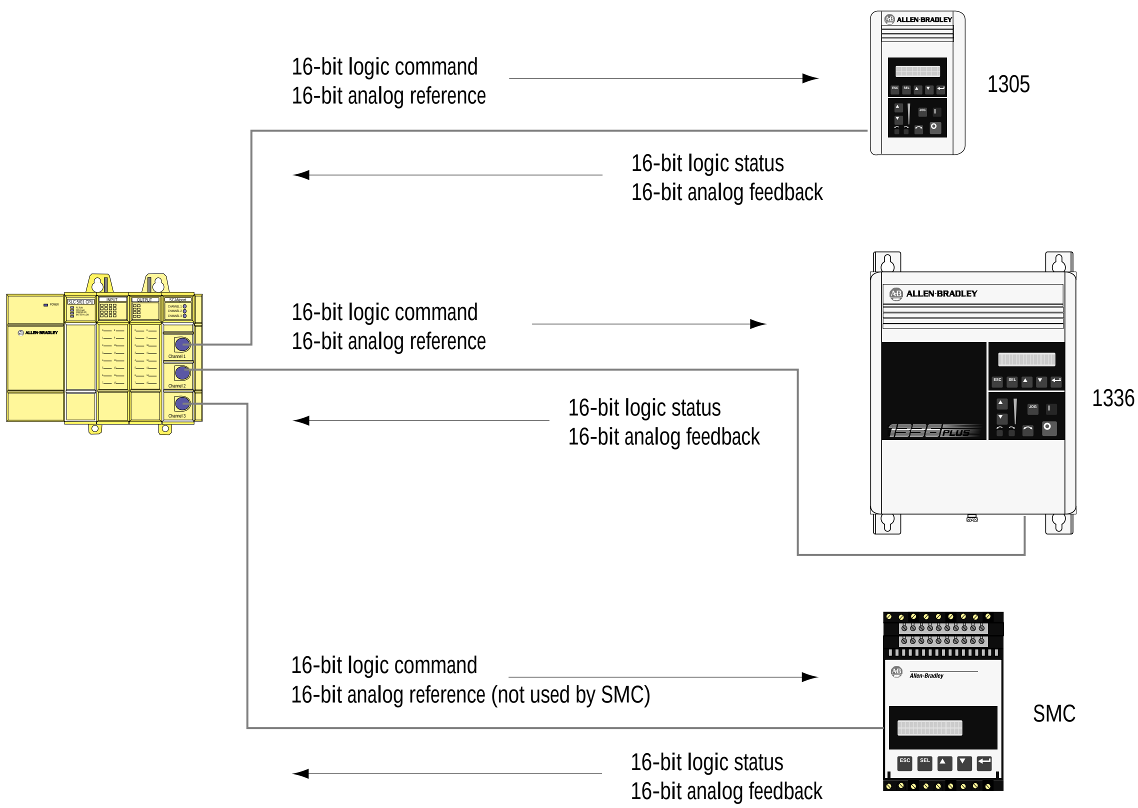
Chapter Objectives

Chapter 3 covers the following information:

- a description of what basic mode provides
- how to configure the SLC to SCANport module for basic mode
- how to transfer data

What Does Basic Mode Provide?

Basic mode sends a 16–bit logic command and a 16–bit analog reference from the module to each SCANport device. It receives a 16–bit logic status and a 16–bit analog feedback signal from each connected SCANport device.



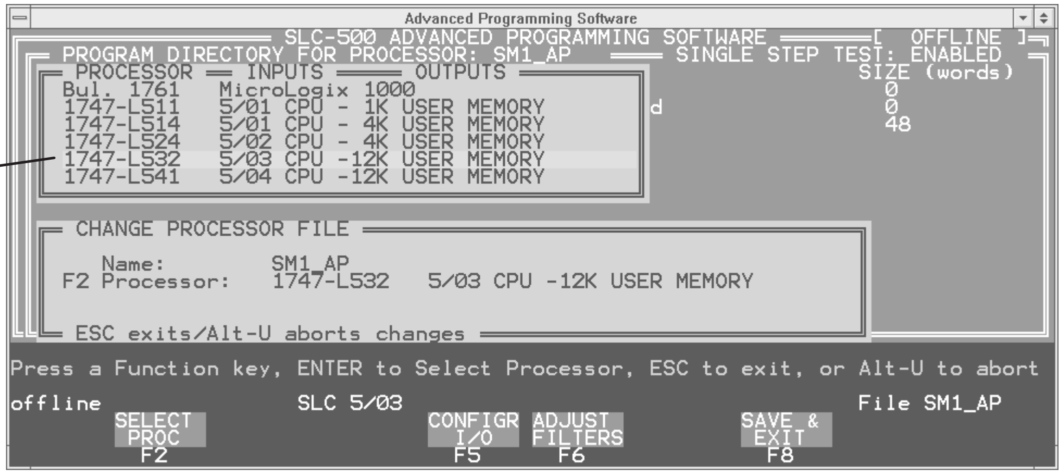
Configuring the SLC to SCANport Module for Basic Mode

To configure the SLC to SCANport module for basic mode using the Advanced Programming Software (APS), you need to:

1. Create a file.
2. Enter a file name. For example purposes, we are using *SM1_AP* as the file name.
3. Highlight the processor as shown in Figure 3.1.

Figure 3.1
Example APS Screen

Highlight the processor you want to use. For example, 1747–L532.



4. Press the *F2* key.

5. Depending on your processor and version of APS, you may be asked to enter the operating system that your processor uses.

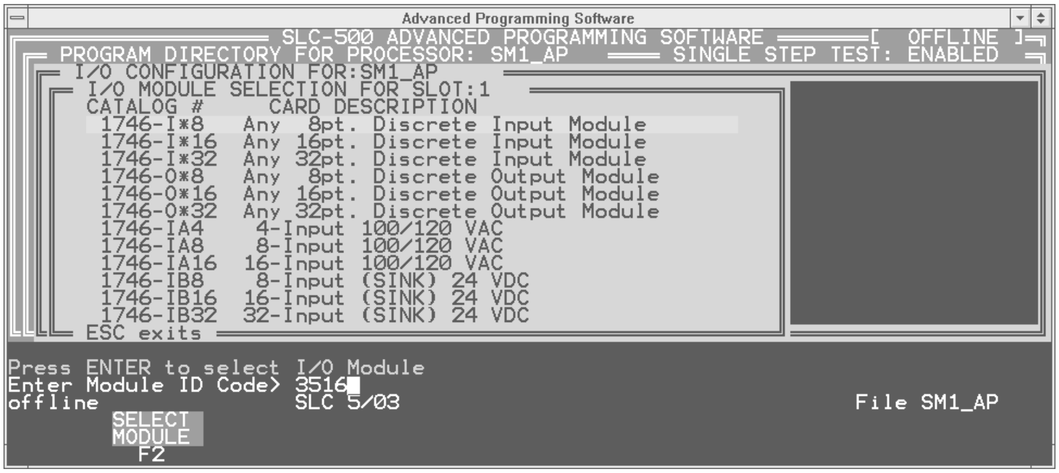
6. Press *F5* to configure the I/O. The screen shown in Figure 3.2 is displayed.

Figure 3.2
Example I/O Configuration Screen

7. Move the cursor to the slot containing the SLC to SCANport module.

8. Press *F5* to modify the slot. The screen shown in Figure 3.3 is displayed.

Figure 3.3
Prompt to Enter the Module ID Code



9. Enter the module ID code. For basic mode, the module ID code is 3516.

10. Press the Enter key.

When you have entered the module ID code, you are returned to the screen shown in Figure 3.2 with the selected module now shown. If you press *F9*, the screen shown in Figure 3.4 shows the configuration information for the SLC to SCANport module. You should not need to change this information for basic mode.

Figure 3.4
Specialty Module Configuration Screen

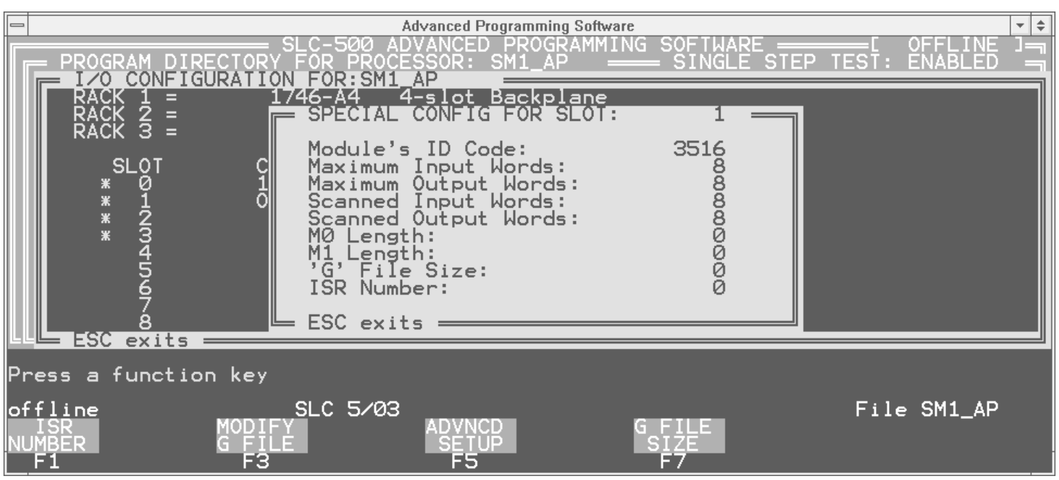
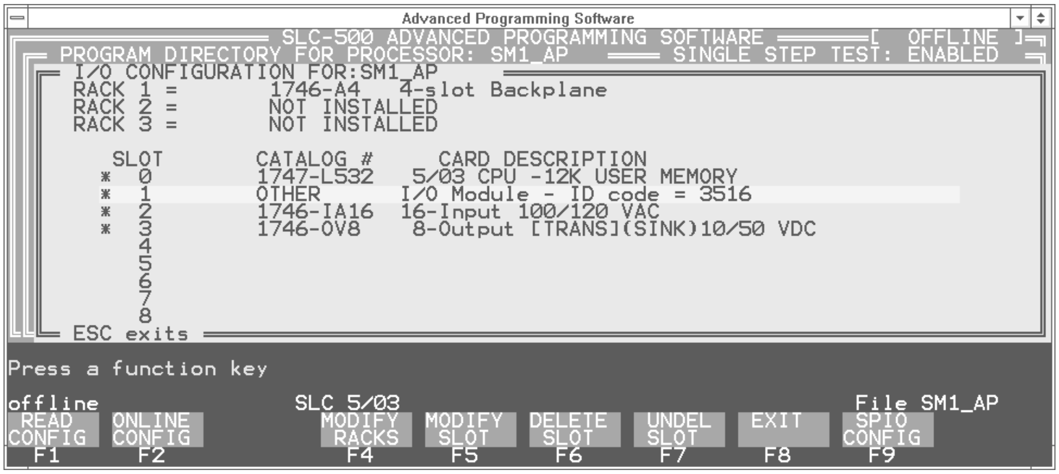


Figure 3.5 shows an example of a completed I/O configuration.

Figure 3.5
An Example of a Completed I/O Configuration



Transferring Data

To transfer data using the SLC to SCANport module, you need to be familiar with how the SLC I/O image table represents the internal data I/O mapping and how the input and output image channel status bits are defined.

When the SLC to SCANport module is configured as a basic mode module, the internal data I/O mapping is represented within the SLC image table as the following:

Output Image			Input Image	
Channel 2 Cmd	Channel 1 Cmd	Word 0	Channel 2 Stat	Channel 1 Stat
Reserved	Channel 3 Cmd	Word 1	Not Used	Channel 3 Stat
Logic Command Channel 1		Word 2	Logic Status Channel 1	
Analog Reference Channel 1		Word 3	Analog Feedback Channel 1	
Logic Command Channel 2		Word 4	Logic Status Channel 2	
Analog Reference Channel 2		Word 5	Analog Feedback Channel 2	
Logic Command Channel 3		Word 6	Logic Status Channel 3	
Analog Reference Channel 3		Word 7	Analog Feedback Channel 3	

Important: Different SCANport devices may define different meanings for the bits in the Logic Command and Logic Status fields. They may also use the Reference and Feedback differently. Refer to the manual for the specific SCANport device for more information.

Channel Status Input Image Definitions

The Input Image Channel Status bits are defined as follows:

Channel 2 Status Channel 1 Status

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Not Used				V2	ID2			Not Used				V1	ID1			Word 0
	Not Used												V3	ID3			Word 1

Channel 3 Status

These bits have the following definitions:

This:	Represents:
ID1 ID2 ID3	SCANport Channel 1, 2, or 3 Connected Adapter Port ID Number. This three bit field contains the adapter port number read from the connector that channel 1, 2, or 3 is connected to on the SCANport device. ID1, ID2, and ID3 should be between 1 and 7. If ID1, ID 2, or ID3 is 7, the channel is not connected to a SCANport device, or the SCANport device may not be powered.
V1 V2 V3	SCANport Channel 1, 2, or 3 Valid Data bit. When high (1), the Logic Status and Analog Feedback values are valid and can be used. The V1, V2, and V3 bit will only go high after the program sets the corresponding data enable bit. When low (0), the values are not valid.

Channel Command Output Image Definitions

The Output Image Channel Command bits are defined as follows:

Channel 2 Command　　　　Channel 1 Command

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td></td><td colspan="7">Not Used</td><td>DE 2</td><td colspan="7">Not Used</td><td>DE 1</td><td>Word 0</td></tr>
<tr><td></td><td colspan="15">Not Used</td><td>DE 3</td><td>Word 1</td></tr>
</table>

Channel 3 Command

These bits have the following definitions:

DE1 DE2 DE3	SCANport Channel 1, 2, or 3 Data Enable bit. While low (0), the channel will not transfer I/O data between the module and the connected SCANport device. When high (1), the channel becomes active to the SCANport device and transfers the appropriate I/O data. When reset to low (0), the channel disconnects from the SCANport device. This usually causes the connected SCANport device to fault.

Example of Basic Mode Data Transfer

This section contains an example program that uses basic mode data transfer. The following portion of the program enables all three SCANport channels on the SLC to SCANport module.

Figure 3.6
Example of Enabling the SCANport Channels

```
|                                                    Channel 1   |
|                                                    SCANport    |
|                                                    Enable      |
|                                                      O:1.0     |
|-----------------------------------------------------( )------|
|                                                   |      0   | |
|                                                   |Channel 2 | |
|                                                   |SCANport  | |
|                                                   |Enable    | |
|                                                   |  O:1.0   | |
|                                                   +---( )----| |
|                                                   |      8   | |
|                                                   |Channel 3 | |
|                                                   |SCANport  | |
|                                                   |Enable    | |
|                                                   |  O:1.1   | |
|                                                   +---( )----+ |
|                                                          0     |
```

The portion of the program shown in Figure 3.7 provides start/stop control and a frequency reference to the 1305 drive connected to SCANport channel 1. The user start is a normally open push button, while the user stop is a normally closed push button.

Figure 3.7
Example of Drive 1 Control and Reference

```
|  Drive 1               Drive 1                          Drive 1    |
|  User                  User                             START      |
|  Momentary             Maintained                       Command    |
|  START                 NOT STOP                         Bit        |
|  Input                 Input                                       |
|    I:2.0                 I:2.0                            O:1.2    |
|-----] [-------------------] [-----------------------------( )-----|
| |      0              |      1                               1    |
| |Drive 1   |Drive 1   |                                           |
| |START     |RUNNING   |                                           |
| |Command   |Status    |                                           |
| |Bit       |Bit       |                                           |
| |  O:1.2      I:1.2   |                                           |
| +---] [---------]/[----+                                          |
|        1         1                                                 |
|  Drive 1                                                Drive 1    |
|  User                                                   STOP       |
|  Maintained                                             Command    |
|  NOT STOP                                               Bit        |
|  Input                                                             |
|    I:2.0                                                  O:1.2    |
|-----]/[---------------------------------------------------( )-----|
| |      1              |                                      0    |
| |Drive 1   |Drive 1   |                                           |
| |STOP      |RUNNING   |                                           |
| |Command   |Status    |                                           |
| |Bit       |Bit       |                                           |
| |  O:1.2      I:1.2   |                                           |
| +---] [---------] [----+                                          |
|        0         1                                                 |
|                                                    Drive 1          |
|                                                    Frequency        |
|                                                    Reference        |
|                                               +MOV---------------+ |
|----------------------------------------------|MOVE              +-|
|                                              |Source      N20:0| |
|                                              |                0| |
|                                              |Dest        O:1.3| |
|                                              |                0| |
|                                              +------------------+ |
```

The portion of the program shown in Figure 3.8 provides start/stop control and a frequency reference to the 1305 drive connected to SCANport channel 2. This section functions the same as that shown in Figure 3.7 except for the changes in addresses.

Figure 3.8
Example of Drive 2 Control and Reference

```
|  Drive 2               Drive 2                          Drive 2    |
|  User                  User                             START      |
|  Momentary             Maintained                       Command    |
|  START                 NOT STOP                         Bit        |
|  Input                 Input                                       |
|    I:2.0                 I:2.0                            O:1.4    |
|-----] [-------------------] [----------------------------( )-----|
| |      2              |      3                             1    |
| |Drive 2   |Drive 2   |                                           |
| |START     |RUNNING   |                                           |
| |Command   |Status    |                                           |
| |Bit       |Bit       |                                           |
| |  O:1.4      I:1.4   |                                           |
| +---] [--------]/[----+                                           |
|        1         1                                                |
|  Drive 2                                                Drive 2    |
|  User                                                   STOP       |
|  Maintained                                             Command    |
|  NOT STOP                                               Bit        |
|  Input                                                             |
|    I:2.0                                                  O:1.4    |
|-----]/[--------------------------------------------------( )-----|
| |      3              |                                    0    |
| |Drive 2   |Drive 2   |                                           |
| |STOP      |RUNNING   |                                           |
| |Command   |Status    |                                           |
| |Bit       |Bit       |                                           |
| |  O:1.4      I:1.4   |                                           |
| +---] [--------] [----+                                           |
|        0         1                                                |
|                                                    Drive 2        |
|                                                    Frequency      |
|                                                    Reference      |
|                                               +MOV---------------+ |
|----------------------------------------------|MOVE              +-|
|                                              |Source       N20:1| |
|                                              |                 0| |
|                                              |Dest         O:1.5| |
|                                              |                 0| |
|                                              +------------------+ |
```

The portion of the program shown in Figure 3.9 provides start/stop control and a frequency reference to the 1305 drive connected to SCANport channel 3. This section functions the same as that shown in Figure 3.7 and Figure 3.8 except for the changes in address.

Figure 3.9
Example of Drive 3 Control and Reference

```
|  Drive 3              Drive 3                        Drive 3    |
|  User                 User                           START      |
|  Momentary            Maintained                     Command    |
|  START                NOT STOP                       Bit        |
|  Input                Input                                     |
|    I:2.0                I:2.0                          O:1.6    |
|-----] [-------------------] [----------------------------( )-----|
| |      4             |      5                              1    |
| |Drive 3   |Drive 3   |                                         |
| |START     |RUNNING   |                                         |
| |Command   |Status    |                                         |
| |Bit       |Bit       |                                         |
| |  O:1.6      I:1.6   |                                         |
| +---] [--------]/[----+                                         |
|        1         1                                              |
|  Drive 3                                             Drive 3    |
|  User                                                STOP       |
|  Maintained                                          Command    |
|  NOT STOP                                            Bit        |
|  Input                                                          |
|    I:2.0                                               O:1.6    |
|-----]/[--------------------------------------------------( )-----|
| |      5             |                                     0    |
| |Drive 3   |Drive 3   |                                         |
| |STOP      |RUNNING   |                                         |
| |Command   |Status    |                                         |
| |Bit       |Bit       |                                         |
| |  O:1.6      I:1.6   |                                         |
| +---] [--------] [----+                                         |
|        0         1                                              |
|                                                 Drive 3         |
|                                                 Frequency       |
|                                                 Reference       |
|                                              +MOV---------------+ |
|----------------------------------------------|MOVE             +-|
|                                              |Source      N20:2| |
|                                              |                0| |
|                                              |Dest        O:1.7| |
|                                              |                0| |
|                                              +-----------------+ |
```

The following data table shows the input data read from the SLC to SCANport module via the SLC backplane.

```
address    15      data       0
I:1         0000 0000 0000 0000 Drives 1 & 2 SCANport Channel Status
I:1.1       0000 0000 0000 0000 Drive 3 SCANport Channel Status
I:1.2       0000 0000 0000 0000 Drive 1 Logic Status
I:1.3       0000 0000 0000 0000 Drive 1 Feedback
I:1.4       0000 0000 0000 0000 Drive 2 Logic Status
I:1.5       0000 0000 0000 0000 Drive 2 Feedback
I:1.6       0000 0000 0000 0000 Drive 3 Logic Status
I:1.7       0000 0000 0000 0000 Drive 3 Feedback
```

The following data table shows the data to be sent to the SLC to SCANport module via the SLC backplane.

```
address   15      data       0
O:1        0000 0000 0000 0000  Drives 1 & 2 SCANport Channel Enables
O:1.1      0000 0000 0000 0000  Drive 3 SCANport Channel Enable
O:1.2      0000 0000 0000 0000  Drive 1 Logic Command
O:1.3      0000 0000 0000 0000  Drive 1 Reference
O:1.4      0000 0000 0000 0000  Drive 2 Logic Command
O:1.5      0000 0000 0000 0000  Drive 2 Reference
O:1.6      0000 0000 0000 0000  Drive 1 Logic Command
O:1.7      0000 0000 0000 0000  Drive 1 Reference
```

Using Enhanced Mode

Chapter Objectives

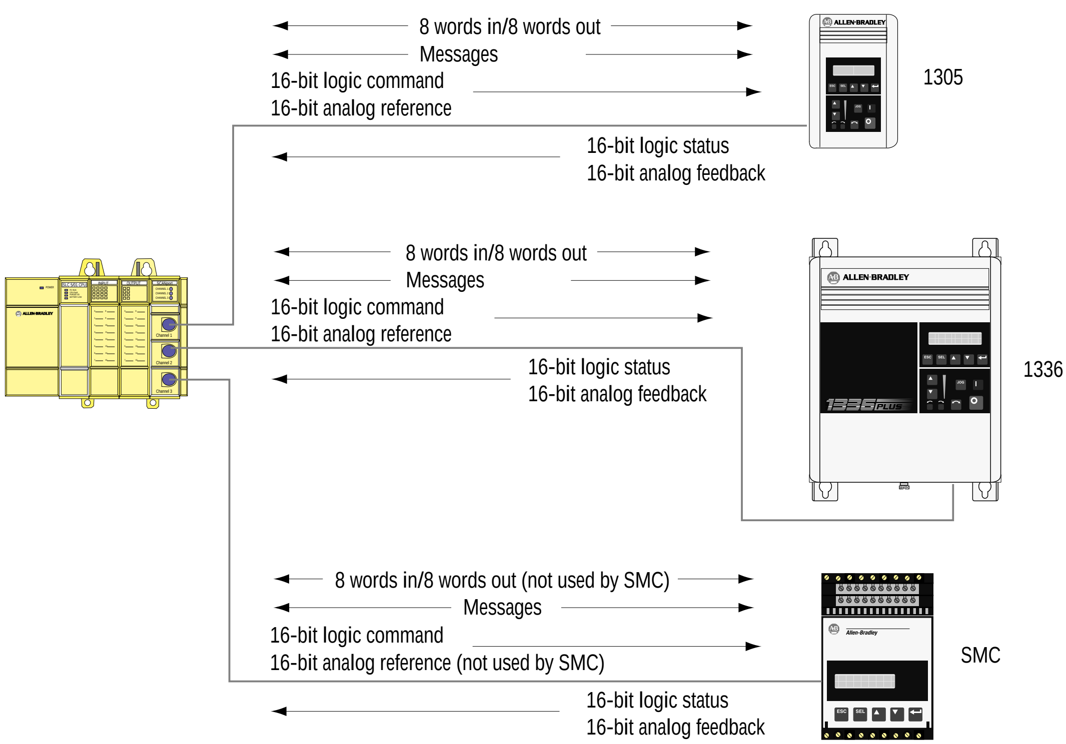
Chapter 4 covers the following information:

- a description of what enhanced mode provides
- how to configure the SLC to SCANport module for enhanced mode
- how to use the I/O image
- how to configure G files
- how to use M files

What Does Enhanced Mode Provide?

Enhanced mode supports the basic mode features which include a 16–bit logic command and a 16–bit analog reference from the module to each SCANport device as well as a 16–bit logic status and a 16–bit analog feedback signal back from each connected SCANport device.

In addition, enhanced mode optionally provides datalinks, safe state data, and messaging.



What Are Datalinks?

Datalinks let you cyclically transfer parameter values to and from a SCANport device (provided that the SCANport device supports datalinks). By using datalinks, you can change the value of a parameter without using the SLC to SCANport messaging function. Each datalink consists of two 16–bit words of input and two 16–bit words of output when enabled. Up to 8 words in and 8 words out of data are available if supported in the connected SCANport device.

SCANport devices that support this function have a group of parameters for datalink configuration. These parameters are identified as *Data In A1–D2* and *Data Out A1–D2*. To use datalinks, you need to:

1. Set up a configuration file, called a G file, to enable the datalinks from the SLC to SCANport module side.
2. Configure or link the *Data In A1–D2* and *Data Out A1–D2* parameters in the SCANport device.

Setting up the G file is covered in more detail later in this chapter.

What Is Safe State Configuration Data?

You can select constant values that your SLC to SCANport module will maintain in the event of an SLC processor mode change or error. These constant values are referred to as safe state data. When the SLC is placed in program mode or an SLC fault occurs, the control outputs can be set to automatically switch to the constant values set in the safe state data words. This lets you define a safe operating state for controlled devices that depend on a pre–programmed output from the module.

ATTENTION: Use the G file to configure your safe state values based on your knowledge of how the SCANport devices connected on each channel operate. Refer to the manual for your SCANport device for additional information.

Refer to Chapter 2 for the DIP switch configuration for fault/program state.

What Is Messaging?

Messaging lets you get and modify SCANport device parameters as well as providing access to other internal SCANport–related information or services. To use messaging, you need to configure the M file mechanism of the SLC processor. The M file mechanism is covered later in this chapter. Appendix B, *SCANport Messaging*, provides examples of SCANport messages.

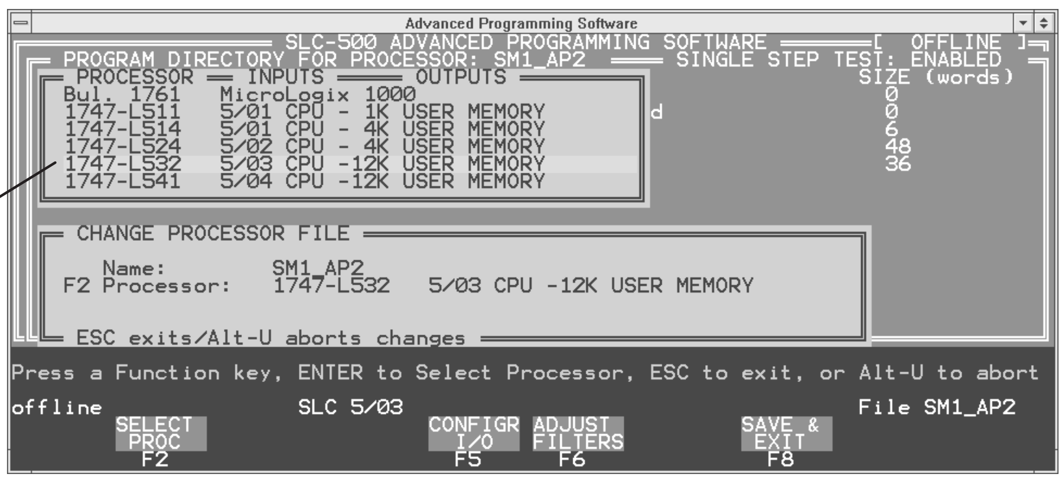
Configuring the SLC to SCANport Module for Enhanced Mode

To configure the SLC to SCANport module for enhanced mode using the Advanced Programming Software (APS), you need to:

1. Create a file.
2. Enter a file name. For example purposes, we are using *SM1_AP2* as the file name.
3. Highlight the processor as shown in Figure 4.1.

Figure 4.1
Example APS Screen

Highlight the processor you want to use. For example, 1747–L532.



4. Press the *F2* key.
5. Depending on your processor and version of APS, you may be asked to enter the operating system that your processor uses.

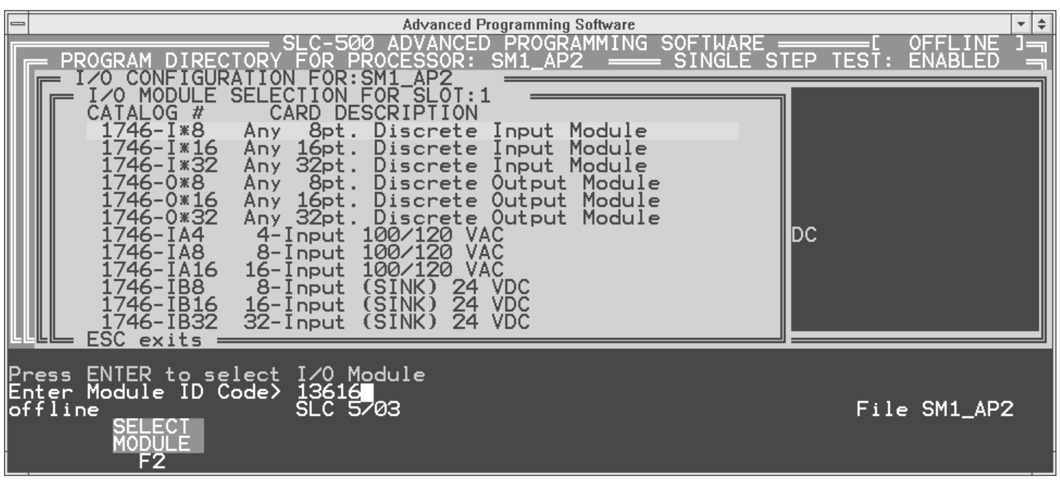
6. Press *F5* to configure the I/O using the screen shown in Figure 4.2.

Figure 4.2
An Example of the I/O Configuration Screen

7. Move the cursor to the slot containing the SLC to SCANport module.

8. Press *F5* to modify the slot using the screen shown in Figure 4.3.

Figure 4.3
Prompt to Enter the Module ID Code



9. Enter the module ID code. For enhanced mode, the module ID code is 13616.

10. Press the Enter key.

11. The slot is now configured for the SLC to SCANport module to be used in enhanced mode. This is shown in Figure 4.4

Figure 4.4
Example I/O Configuration Screen

12. Press *F9* to add the information to configure the specialty I/O using the screen shown in Figure 4.5.

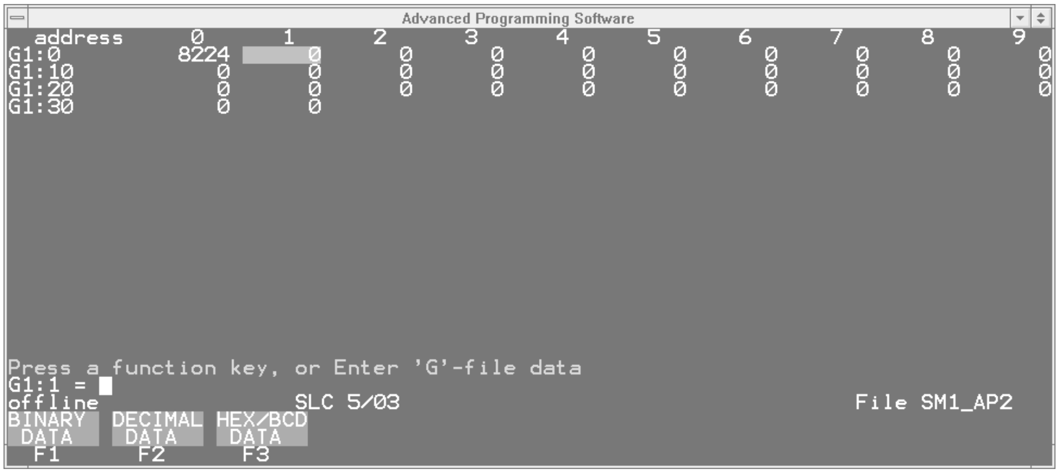
Figure 4.5
Specialty I/O Configuration Screen

▶ If you are not familiar with G files and M0/M1 files, you should read the sections that are provided later in this chapter that pertain to these files before continuing.

13. Press *F7* to set the G file size a value from 2 to 32 words using the screen shown in Figure 4.5.

14. Press *F3* to configure the G file data using the screen shown in Figure 4.6. Figure 4.8 provides the definition of the G file data.

Figure 4.6
Screen Used to Configure G File Data



15. When you return to the screen shown in Figure 4.5, press *F5* to access the advanced set up menu to configure the size of the messaging buffers.

16. Press *F5* to enter the size of the M0 file. For this module, the maximum file size is 800.

17. Press *F6* to enter the size of the M1 file. For this module, the maximum file size is 400.

Using the I/O Image

The following SLC I/O image table represents the internal data I/O mapping for the SLC to SCANport module when configured as an enhanced mode module.

Figure 4.7
SLC I/O Image Table

Output Image			Input Image	
Channel 2 Cmd	Channel 1 Cmd	Word 0	Channel 2 Status	Channel 1 Status
Reserved	Channel 3 Cmd	Word 1	Message Status	Channel 3 Status
Logic Command Channel 1		Word 2	Logic Status Channel 1	
Analog Reference Channel 1		Word 3	Analog Feedback Channel 1	
Logic Command Channel 2		Word 4	Logic Status Channel 2	
Analog Reference Channel 2		Word 5	Analog Feedback Channel 2	
Logic Command Channel 3		Word 6	Logic Status Channel 3	
Analog Reference Channel 3		Word 7	Analog Feedback Channel 3	
Channel 1 Datalink A1 Input		Word 8	Channel 1 Datalink A1 Output	
Channel 1 Datalink A2 Input		Word 9	Channel 1 Datalink A2 Output	
Channel 1 Datalink B1 Input		Word 10	Channel 1 Datalink B1 Output	
Channel 1 Datalink B2 Input		Word 11	Channel 1 Datalink B2 Output	
Channel 1 Datalink C1 Input		Word 12	Channel 1 Datalink C1 Output	
Channel 1 Datalink C2 Input		Word 13	Channel 1 Datalink C2 Output	
Channel 1 Datalink D1 Input		Word 14	Channel 1 Datalink D1 Output	
Channel 1 Datalink D2 Input		Word 15	Channel 1 Datalink D2 Output	
Channel 2 Datalink A1 Input		Word 16	Channel 2 Datalink A1 Output	
Channel 2 Datalink A2 Input		Word 17	Channel 2 Datalink A2 Output	
Channel 2 Datalink B1 Input		Word 18	Channel 2 Datalink B1 Output	
Channel 2 Datalink B2 Input		Word 19	Channel 2 Datalink B2 Output	
Channel 2 Datalink C1 Input		Word 20	Channel 2 Datalink C1 Output	
Channel 2 Datalink C2 Input		Word 21	Channel 2 Datalink C2 Output	
Channel 2 Datalink D1 Input		Word 22	Channel 2 Datalink D1 Output	
Channel 2 Datalink D2 Input		Word 23	Channel 2 Datalink D2 Output	
Channel 3 Datalink A1 Input		Word 24	Channel 3 Datalink A1 Output	
Channel 3 Datalink A2 Input		Word 25	Channel 3 Datalink A2 Output	
Channel 3 Datalink B1 Input		Word 26	Channel 3 Datalink B1 Output	
Channel 3 Datalink B2 Input		Word 27	Channel 3 Datalink B2 Output	
Channel 3 Datalink C1 Input		Word 28	Channel 3 Datalink C1 Output	
Channel 3 Datalink C2 Input		Word 29	Channel 3 Datalink C2 Output	
Channel 3 Datalink D1 Input		Word 30	Channel 3 Datalink D1 Output	
Channel 3 Datalink D2 Input		Word 31	Channel 3 Datalink D2 Output	

Enhanced Mode Interface

The Channel Status and Message Status bits provide additional status information pertaining to the validity of certain pieces of data. This information includes the port that the particular channel is connected to on the SCANport device, the state of the I/O data, and the status of each message buffer.

These status fields are defined as follows:

Channel 2 Status | Channel 1 Status

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	D2	C2	B2	A2	V2	ID2			D1	C1	B1	A1	V1	ID1			Word 0
	X	M0 STA	MSTAT CH3		MSTAT CH2		MSTAT CH1		D3	C3	B3	A3	V3	ID3			Word 1

Message Status Ch 1–3 | Channel 3 Status

These bits have the following definitions:

This:	Represents the:
ID1 ID2 ID3	SCANport Channel 1, 2, or 3 Connected Adapter Port ID Number. This three bit field contains the adapter port number that channel 1, 2, or 3 is connected to on the SCANport device. ID1, ID2, and ID3 should be between 1 and 7. If ID1, ID2, or ID3 is 7, the channel is not connected to the SCANport device, or the SCANport device may not be powered.
V1 V2 V3	SCANport Channel 1, 2, or 3 Valid Data bit. When high (1), the Logic Status and Analog Feedback values are valid and can be used. When low (0), the values are not valid.
A–D1(–3)	SCANport Channel 1, 2, or 3 Datalink A–D Valid Data bit. When high (1), the data associated Datalink A–D of the corresponding channel is valid and can be used. When low (0), the values are not valid.
M0 STA	M0 File Status bit. When high (1), the SLC program can enable any previously written M0 file message to the module. When low (0), the SLC to SCANport module is either actively reading the last sent M0 file data changes, or the SLC program has not loaded any M0 file data into the module. If any changes are made to the M0 file message buffers, the SLC program should check this status bit before enabling that message to be sent out any SCANport channel.
MSTAT CH1–3	Message Status bits for the message buffer of the corresponding channel. These two bit fields contain the status of each of the message buffers. There is one message response buffer for each channel. The first bit (word 1, bits 9, 11, and 13) contains the READY bit. The READY bit is active (high=1) when a message request can be enabled to the SCANport device. The second bit (word 1, bits 8, 10, and 12) contains the DONE bit. The DONE bit is active (high=1) when an M1 file message buffer contains response data to a message request. When both the READY and the DONE bits are 0, the buffer is in a BUSY state. This is the state during which the module is actually requesting the data from the SCANport device. These status bits should never be high at the same time.

Important: Do not enable messages while writing to the message buffer.

The Channel Command bytes contain I/O Data Enable and Message Enable bits that are used to activate I/O Data and Messages to a SCANport device. These functions are independent; you can use either function by itself or use both functions together.

The eight unique message buffers in the M0 file can contain a message for any of the three SCANport channels. You can select which message buffer is to be used for each channel with the corresponding MSG ID bits. This lets a message be written into a buffer only once and used as many times as needed.

To transmit a message, the MSG ID bits are set to select the desired buffer and a channel's Message Enable bit is set to 1. When the message status bits indicate the message is done, the response can be read from the M1 message buffer area for the corresponding SCANport channel used. When the Message Enable bit is subsequently cleared to 0, the DONE status bit will be reset and the READY bit set to allow for another message sequence.

The definition for the channel command output image is as follows:

Channel 2 Command

Channel 1 Command

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td></td><td colspan="3">Not Used</td><td colspan="3">MSG ID</td><td>ME 2</td><td>DE 2</td><td colspan="3">Not Used</td><td colspan="3">MSG ID</td><td>ME 1</td><td>DE 1</td><td>Word 0</td></tr>
<tr><td></td><td colspan="8">Reserved</td><td colspan="3">Not Used</td><td colspan="3">MSG ID</td><td>ME 3</td><td>DE 3</td><td>Word 1</td></tr>
</table>

Channel 3 Command

These bits have the following definitions:

This:	Represents the:
DE1 DE2 DE3	SCANport Channel 1, 2, or 3 Data Enable bit. While low (0), the channel is not transferring I/O data (including datalink data) between the connected SCANport device. When high (1), the channel becomes active to the SCANport device and transfers the appropriate I/O data. When cleared to low (0), the channel disconnects from the SCANport device, which usually causes the connected SCANport device to fault.
ME1 ME2 ME3	SCANport Channel 1, 2, or 3 Message Enable bit. When set high (1), the message selected by the Message ID field is transmitted through the appropriate channel to the SCANport device. This bit should be held high during the duration of the request until the status DONE bit is asserted (1). The clearing (0) of this bit clears the DONE status and returns the message status to the READY state.
MSG ID	Message Identifier field. This field allows for the selection of one of the eight message buffers in the M0 file area for each channel. Multiple channels can use the same message ID buffer simultaneously.

Configuring G Files

You can use G files to enable additional I/O (datalinks) between the SCANport device and the SLC to SCANport module. G files also hold the safe state values for the output data to be transferred to the SCANport devices when it is switch–configured to use it. The G file data is specified as follows: G=g–file, s=slot

Figure 4.8
G File Image

G File Image		
Reserved	Gs:0	
Datalink Enables	Gs:1	Datalink Configuration Field
Logic Command Channel 1	Gs:2	
Analog Reference Channel 1	Gs:3	
Logic Command Channel 2	Gs:4	
Analog Reference Channel 2	Gs:5	
Logic Command Channel 3	Gs:6	
Analog Reference Channel 3	Gs:7	
Channel 1 Datalink A1 Input	Gs:8	
Channel 1 Datalink A2 Input	Gs:9	
Channel 1 Datalink B1 Input	Gs:10	
Channel 1 Datalink B2 Input	Gs:11	
Channel 1 Datalink C1 Input	Gs:12	
Channel 1 Datalink C2 Input	Gs:13	
Channel 1 Datalink D1 Input	Gs:14	
Channel 1 Datalink D2 Input	Gs:15	
Channel 2 Datalink A1 Input	Gs:16	
Channel 2 Datalink A2 Input	Gs:17	Safe State Configuration Data
Channel 2 Datalink B1 Input	Gs:18	
Channel 2 Datalink B2 Input	Gs:19	
Channel 2 Datalink C1 Input	Gs:20	
Channel 2 Datalink C2 Input	Gs:21	
Channel 2 Datalink D1 Input	Gs:22	
Channel 2 Datalink D2 Input	Gs:23	
Channel 3 Datalink A1 Input	Gs:24	
Channel 3 Datalink A2 Input	Gs:25	
Channel 3 Datalink B1 Input	Gs:26	
Channel 3 Datalink B2 Input	Gs:27	
Channel 3 Datalink C1 Input	Gs:28	
Channel 3 Datalink C2 Input	Gs:29	
Channel 3 Datalink D1 Input	Gs:30	
Channel 3 Datalink D2 Input	Gs:31	

The G file datalink configuration field is defined as follows:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Not Used				ED 3	EC 3	EB 3	EA 3	ED 2	EC 2	EB 2	EA 2	ED 1	EC 1	EB 1	EA 1

Gs.1

These bits have the following definitions:

This:	Represents:
EA1 EA2 EA3	SCANport Channel 1, 2, or 3 Datalink A Enable bit. When set high (1), datalink A is enabled for the corresponding channel. **Note**: The datalink is only active while the channel's Data Enable bit is also set.
EB1 EB2 EB3	SCANport Channel 1, 2, or 3 Datalink B Enable bit. When set high (1), datalink B is enabled for the corresponding channel. **Note**: The datalink is only active while the channel's Data Enable bit is also set.
EC1 EC2 EC3	SCANport Channel 1, 2, or 3 Datalink C Enable bit. When set high (1), datalink C is enabled for the corresponding channel. **Note**: The datalink is only active while the channel's Data Enable bit is also set.
ED1 ED2 ED3	SCANport Channel 1, 2, or 3 Datalink D Enable bit. When set high (1), datalink D is enabled for the corresponding channel. **Note**: The datalink is only active while the channel's Data Enable bit is also set.

You may enable datalinks with or without providing safe state data. You can configure G files that are between 2 and 32 words in length.

ATTENTION: Configure your safe state values based on your knowledge of how the SCANport devices connected on each channel operate. Refer to the manual for your SCANport device for additional information.

Using M Files

To transfer messages between the SLC processor and any of the SCANport devices connected through the SLC to SCANport module, you need to use M files. The SLC to SCANport module uses M0 and M1 files. The M0 file is a module output file (write–only), while the M1 file is a module input file (read–only). Messages to the SCANport module from the SLC processor are sent using the M0 file, while the M1 file contains the responses.

The M0 file image contains multiple transmit message buffers set up as shown in Figure 4.9.

Figure 4.9
M0 File Structure

e = slot location

Message Buffer 0

Service		**M0:e.000**
Class		**M0:e.001**
Instance		**M0:e.002**
Attribute		**M0:e.003**
Request Length = n		**M0:e.004**
Req Data 2	Req Data 1	**M0:e.005**
Req Data n	...	**M0:e.0xx**

Message Buffer 1

Service		**M0:e.100**
Class		**M0:e.101**
Instance		**M0:e.102**
Attribute		**M0:e.103**
Request Length = n		**M0:e.104**
Req Data 2	Req Data 1	**M0:e.105**
Req Data n	...	**M0:e.1xx**

Message Buffer 2

Service		**M0:e.200**
Class		**M0:e.201**
Instance		**M0:e.202**
Attribute		**M0:e.203**
Request Length = n		**M0:e.204**
Req Data 2	Req Data 1	**M0:e.205**
Req Data n	...	**M0:e.2xx**

Message Buffer 3

Service		**M0:e.300**
Class		**M0:e.301**
Instance		**M0:e.302**
Attribute		**M0:e.303**
Request Length = n		**M0:e.304**
Req Data 2	Req Data 1	**M0:e.305**
Req Data n	...	**M0:e.3xx**

Message Buffer 4

Service		**M0:e.400**
Class		**M0:e.401**
Instance		**M0:e.402**
Attribute		**M0:e.403**
Request Length = n		**M0:e.404**
Req Data 2	Req Data 1	**M0:e.405**
Req Data n	...	**M0:e.4xx**

Message Buffer 5

Service		**M0:e.500**
Class		**M0:e.501**
Instance		**M0:e.502**
Attribute		**M0:e.503**
Request Length = n		**M0:e.504**
Req Data 2	Req Data 1	**M0:e.505**
Req Data n	...	**M0:e.5xx**

Message Buffer 6

Service		**M0:e.600**
Class		**M0:e.601**
Instance		**M0:e.602**
Attribute		**M0:e.603**
Request Length = n		**M0:e.604**
Req Data 2	Req Data 1	**M0:e.605**
Req Data n	...	**M0:e.6xx**

Message Buffer 7

Service		**M0:e.700**
Class		**M0:e.701**
Instance		**M0:e.702**
Attribute		**M0:e.703**
Request Length = n		**M0:e.704**
Req Data 2	Req Data 1	**M0:e.705**
Req Data n	...	**M0:e.7xx**

The M0 file contains eight separate buffer areas. Each area can be used to send messages to any of the three channels. You need to access a unique area of the M0 files for each message buffer used. Individual message request status bits (located in Word 1 of the I/O input image) are used to monitor the progress of each message request enabled. The Message ID selection field and Message Enable bits (located in Words 0 and 1 of the I/O output image) initiate each message transaction and subsequently free the receive buffer for the next message.

The M1 file image contains the contents of the multi–position DIP switch, an echo of the contents of the G file data, and three receive message buffers as shown in Figure 4.10.

Figure 4.10
M1 File Structure

Echo of G File Contents and DIP Switch Read Out

e = slot location

Reserved	DIP SW	**M1:e.000**
Datalink Enables		**M1:e.001**
Logic Command Channel 1		**M1:e.002**
Analog Reference Channel 1		**M1:e.003**
Logic Command Channel 2		**M1:e.004**
Analog Reference Channel 2		**M1:e.005**
Logic Command Channel 3		**M1:e.006**
Analog Reference Channel 3		**M1:e.007**
Channel 1 Datalink A1 In		**M1:e.008**
Channel 1 Datalink A2 In		**M1:e.009**
Channel 1 Datalink B1 In		**M1:e.010**
Channel 1 Datalink B2 In		**M1:e.011**
Channel 1 Datalink C1 In		**M1:e.012**
Channel 1 Datalink C2 In		**M1:e.013**
Channel 1 Datalink D1 In		**M1:e.014**
Channel 1 Datalink D2 In		**M1:e.015**
Channel 2 Datalink A1 In		**M1:e.016**
Channel 2 Datalink A2 In		**M1:e.017**
Channel 2 Datalink B1 In		**M1:e.018**
Channel 2 Datalink B2 In		**M1:e.019**
Channel 2 Datalink C1 In		**M1:e.020**
Channel 2 Datalink C2 In		**M1:e.021**
Channel 2 Datalink D1 In		**M1:e.022**
Channel 2 Datalink D2 In		**M1:e.023**
Channel 3 Datalink A1 In		**M1:e.024**
Channel 3 Datalink A2 In		**M1:e.025**
Channel 3 Datalink B1 In		**M1:e.026**
Channel 3 Datalink B2 In		**M1:e.027**
Channel 3 Datalink C1 In		**M1:e.028**
Channel 3 Datalink C2 In		**M1:e.029**
Channel 3 Datalink D1 In		**M1:e.030**
Channel 3 Datalink D2 In		**M1:e.031**

Channel 1 Receive Buffer

Service (echo or error)		**M1:e.100**
Class (echo)		**M1:e.101**
Instance (echo)		**M1:e.102**
Attribute (echo)		**M1:e.103**
Response Length = n		**M1:e.104**
Resp Data 2	Resp Data 1	**M1:e.105**
Resp Data n	...	**M1:e.1xx**

Channel 2 Receive Buffer

Service (echo or error)		**M1:e.200**
Class (echo)		**M1:e.201**
Instance (echo)		**M1:e.202**
Attribute (echo)		**M1:e.203**
Response Length = n		**M1:e.204**
Resp Data 2	Resp Data 1	**M1:e.205**
Resp Data n	...	**M1:e.2xx**

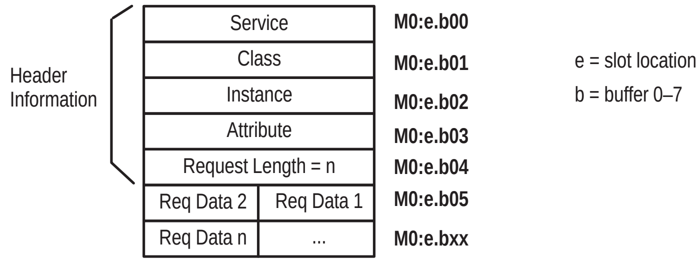
Channel 3 Receive Buffer

Service (echo or error)		**M1:e.300**
Class (echo)		**M1:e.301**
Instance (echo)		**M1:e.302**
Attribute (echo)		**M1:e.303**
Response Length = n		**M1:e.304**
Resp Data 2	Resp Data 1	**M1:e.305**
Resp Data n	...	**M1:e.3xx**

SCANport messages access data structures within the SCANport device. These data structures are called objects. An object contains information for a particular purpose. For example, a parameter object can contain information such as parameter values, parameter names, scaling information, and units.

Figure 4.11 shows the first M0 message buffer structure.

Figure 4.11
SLC to SCANport Module M0 Buffer 0 Message Structure



Where:

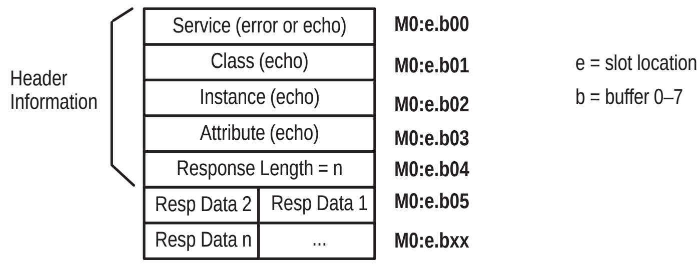
This field:	Specifies:
Service	The action or service requested. The following service values are available: **Enter this value:** — **To request this service:** 0001H (1 decimal) — Read Parameter Full/All Info 0005H (5 decimal) — Reset to Default ① 000eH (14 decimal) — Get Attribute Single 0010H (16 decimal) — Set Attribute Single 0015H (21 decimal) — Restore from Storage ① 0016H (22 decimal) — Save to Storage ① 0032H (50 decimal) — Get Attribute Scattered 0034H (52 decimal) — Set Attribute Scattered 004bH (75 decimal) — Read Enum String ① ① Not all SCANport devices support these services. For example, if you enter a service value of 0001H (1 decimal), you are requesting that the SCANport device provide all available information about a particular object. A service value of 000eH is a request for only one piece of information about a particular object.
Class	The type of object to access within the SCANport device. The class is the first index into the SCANport device's database. It directs the message to the desired functional database. For example, a class value of 000fH (15 decimal) indicates that the message is intended to access the parameter database.
Instance	A particular occurrence of an object in the SCANport device. The instance provides an index into the referenced functional database. For example, when accessing the parameter database, the instance value is the parameter number. If you want to access information about all instances of the object, specify an instance of 0.
Attribute	A specific piece of information about an object. Values are always less than 256. For example, in a parameter object, an attribute value of 0001H (1 decimal) indicates that the message is accessing the parameter value. An attribute value of 0007H (7 decimal) indicates that the message is accessing the parameter name text string.
Request length	The length, in bytes, in this request. This value is normally less than or equal to 96 bytes. However, Get/Set Attribute Scattered messages can be longer.

This field:	Specifies:
Req Data	The actual data portion of the request message. Up to a maximum of 96 bytes of message data is available. This field is optional depending on the type of message sent.

The response from the SCANport device appears in the message buffer of the channel used (Channel 1 = M1:e.100–199, Channel 2 = M1:e.200–299, Channel 3 = M1:e.300–399).

Figure 4.12 shows the structure of the message response buffers inside the SLC M1 file.

Figure 4.12
M1 File Message Response Buffer Structures



Where:

This field:	Contains:
Service	The same value as the service field of the request message if the message transaction was successful. If an error occurred, the service will be 0014H (20 decimal) and additional error information will be placed in the response data field. The error codes are provided at the end of this appendix.
Class	The same value that was used for the class field in the request message.
Instance	The same value that was used for the instance field in the request massage.
Attribute	The same value that was used for the attribute field in the request message.
Response Length[1]	The amount of data, in bytes, in this response. Most messages contain 96 or fewer bytes of data. However, Get/Set Attribute Scattered messages can be longer.
Resp Data[1]	The actual data portion of this response. This field varies in length depending on the message. If an error occurred, this field contains the SCANport error code.

[1] The size of the returned packets determines the response length. Some SCANport devices may return lengths greater than the actual amount of data in the response. These products always return a length which is a multiple of six. For example, a 1336 PLUS drive may reply to a Read Number of Parameters message with a length of six in the response. The first two bytes contain the complete response data. The extra data bytes should be ignored.

Examples of Enhanced Mode Data Transfer

The following examples are portions of the same program that is used to transfer data using the enhanced mode mechanism.

Figure 4.13 shows an example configuration of the G file for the SLC to SCANport module. The Advanced Programming Software (APS) sets the first word; do not modify this word. Each bit in the second word enables a datalink. Refer to the manual for your SCANport device for more information about datalinks.

Each additional word in the G file contains safe state data. This is the data that is sent to the attached SCANport device(s) if the SLC to SCANport module DIP switch is configured to use safe state data and the SLC becomes faulted or is changed to program mode.

Figure 4.13
Example G File Configuration

```
address       15      data       0
G1:0            0000 0000 0000 0000 Reserved (Used by SLC)
G1:1            0000 0000 0000 0000 Datalink Enables
                |||| |||| |||| ||||
                ++++ |||| |||| |||+--- Channel 1 Datalink A Enable
                  |  |||| |||| ||+---- Channel 1 Datalink B Enable
                  |  |||| |||| |+----- Channel 1 Datalink C Enable
                  |  |||| |||| +------ Channel 1 Datalink D Enable
                  |  |||| ||||
                  |  |||| |||+-------- Channel 2 Datalink A Enable
                  |  |||| ||+--------- Channel 2 Datalink B Enable
                  |  |||| |+---------- Channel 2 Datalink C Enable
                  |  |||| +----------- Channel 2 Datalink D Enable
                  |  ||||
                  |  |||+------------- Channel 3 Datalink A Enable
                  |  ||+-------------- Channel 3 Datalink B Enable
                  |  |+--------------- Channel 3 Datalink C Enable
                  |  +---------------- Channel 3 Datalink D Enable
                  |
                  +------------------- Not Used

G1:2            0000 0000 0000 0000 Safe State Data – Channel 1 Logic Command
G1:3            0000 0000 0000 0000 Safe State Data – Channel 1 Reference
G1:4            0000 0000 0000 0000 Safe State Data – Channel 2 Logic Command
G1:5            0000 0000 0000 0000 Safe State Data – Channel 2 Reference
G1:6            0000 0000 0000 0000 Safe State Data – Channel 3 Logic Command
G1:7            0000 0000 0000 0000 Safe State Data – Channel 3 Reference
G1:8            0000 0000 0000 0000 Safe State Data – Channel 1 Datalink A1
G1:9            0000 0000 0000 0000 Safe State Data – Channel 1 Datalink A2
G1:10           0000 0000 0000 0000 Safe State Data – Channel 1 Datalink B1
G1:11           0000 0000 0000 0000 Safe State Data – Channel 1 Datalink B2
G1:12           0000 0000 0000 0000 Safe State Data – Channel 1 Datalink C1
G1:13           0000 0000 0000 0000 Safe State Data – Channel 1 Datalink C2
G1:14           0000 0000 0000 0000 Safe State Data – Channel 1 Datalink D1
G1:15           0000 0000 0000 0000 Safe State Data – Channel 1 Datalink D2
G1:16           0000 0000 0000 0000 Safe State Data – Channel 2 Datalink A1
G1:17           0000 0000 0000 0000 Safe State Data – Channel 2 Datalink A2
G1:18           0000 0000 0000 0000 Safe State Data – Channel 2 Datalink B1
G1:19           0000 0000 0000 0000 Safe State Data – Channel 2 Datalink B2
G1:20           0000 0000 0000 0000 Safe State Data – Channel 2 Datalink C1
G1:21           0000 0000 0000 0000 Safe State Data – Channel 2 Datalink C2
G1:22           0000 0000 0000 0000 Safe State Data – Channel 2 Datalink D1
G1:23           0000 0000 0000 0000 Safe State Data – Channel 2 Datalink D2
G1:24           0000 0000 0000 0000 Safe State Data – Channel 3 Datalink A1
G1:25           0000 0000 0000 0000 Safe State Data – Channel 3 Datalink A2
G1:26           0000 0000 0000 0000 Safe State Data – Channel 3 Datalink B1
G1:27           0000 0000 0000 0000 Safe State Data – Channel 3 Datalink B2
G1:28           0000 0000 0000 0000 Safe State Data – Channel 3 Datalink C1
G1:29           0000 0000 0000 0000 Safe State Data – Channel 3 Datalink C2
G1:30           0000 0000 0000 0000 Safe State Data – Channel 3 Datalink D1
G1:31           0000 0000 0000 0000 Safe State Data – Channel 3 Datalink D2
```

Figure 4.14 shows a portion of the program that enables all three SCANport channels on the SLC to SCANport module.

Figure 4.14
Example of Enabling the SCANport channels

```
|                                                     Channel 1   |
|                                                     SCANport    |
|                                                     Enable      |
|                                                       O:1.0     |
|--------------------------------------------------------( )------|
|                                                    |      0   | |
|                                                    |Channel 2 | |
|                                                    |SCANport  | |
|                                                    |Enable    | |
|                                                    |  O:1.0   | |
|                                                    +---( )----| |
|                                                    |      8   | |
|                                                    |Channel 3 | |
|                                                    |SCANport  | |
|                                                    |Enable    | |
|                                                    |  O:1.1   | |
|                                                    +---( )----+ |
|                                                           0     |
```

The portion of the program shown in Figure 4.15 provides start/stop control and a frequency reference to the 1305 drive connected to SCANport channel 1. The user start is a normally open push button, while the user stop is a normally closed push button.

Figure 4.15
Example of Drive 1 Control and Reference

```
|  Drive 1               Drive 1                         Drive 1    |
|  User                  User                            START      |
|  Momentary             Maintained                      Command    |
|  START                 NOT STOP                        Bit        |
|  Input                 Input                                      |
|    I:2.0                 I:2.0                           O:1.2    |
|-----] [-------------------] [----------------------------( )-----|
| |      0              |      1                               1    |
| |Drive 1   |Drive 1   |                                           |
| |START     |RUNNING   |                                           |
| |Command   |Status    |                                           |
| |Bit       |Bit       |                                           |
| |  O:1.2      I:1.2   |                                           |
| +---] [---------]/[----+                                          |
|        1          1                                               |
|  Drive 1                                               Drive 1    |
|  User                                                  STOP       |
|  Maintained                                            Command    |
|  NOT STOP                                              Bit        |
|  Input                                                            |
|    I:2.0                                                 O:1.2    |
|-----]/[--------------------------------------------------( )-----|
| |      1              |                                      0    |
| |Drive 1   |Drive 1   |                                           |
| |STOP      |RUNNING   |                                           |
| |Command   |Status    |                                           |
| |Bit       |Bit       |                                           |
| |  O:1.2      I:1.2   |                                           |
| +---] [---------] [----+                                          |
|        0          1                                               |
|                                                   Drive 1          |
|                                                   Frequency        |
|                                                   Reference        |
|                                              +MOV----------------+ |
|----------------------------------------------|MOVE              +-|
|                                              |Source      N20:0| |
|                                              |                0| |
|                                              |Dest        O:1.3| |
|                                              |                0| |
|                                              +------------------+ |
```

The portion of the program shown in Figure 4.16 provides start/stop control and a frequency reference to the 1305 drive connected to SCANport channel 2. This section functions the same as that shown in Figure 4.15 except for the changes in addresses.

Figure 4.16
Example of Drive 2 Control and Reference

```
|  Drive 2              Drive 2                        Drive 2    |
|  User                 User                           START      |
|  Momentary            Maintained                     Command    |
|  START                NOT STOP                       Bit        |
|  Input                Input                                     |
|    I:2.0                I:2.0                          O:1.4    |
|-----] [-------------------] [----------------------------( )-----|
| |      2              |      3                           1      |
| |Drive 2   |Drive 2   |                                         |
| |START     |RUNNING   |                                         |
| |Command   |Status    |                                         |
| |Bit       |Bit       |                                         |
| |  O:1.4      I:1.4   |                                         |
| +---] [--------]/[----+                                         |
|        1          1                                             |
|  Drive 2                                             Drive 2    |
|  User                                                STOP       |
|  Maintained                                          Command    |
|  NOT STOP                                            Bit        |
|  Input                                                          |
|    I:2.0                                               O:1.4    |
|-----]/[--------------------------------------------------( )-----|
| |      3              |                                  0      |
| |Drive 2   |Drive 2   |                                         |
| |STOP      |RUNNING   |                                         |
| |Command   |Status    |                                         |
| |Bit       |Bit       |                                         |
| |  O:1.4      I:1.4   |                                         |
| +---] [--------] [----+                                         |
|        0          1                                             |
|                                                 Drive 2          |
|                                                 Frequency        |
|                                                 Reference        |
|                                             +MOV---------------+ |
|---------------------------------------------|MOVE              +-|
|                                             |Source       N20:1| |
|                                             |                 0| |
|                                             |Dest         O:1.5| |
|                                             |                 0| |
|                                             +------------------+ |
```

The portion of the program shown in Figure 4.17 provides start/stop control and a frequency reference to the 1305 drive connected to SCANport channel 3. This section functions the same as that shown in Figure 4.15 and Figure 4.16 except for the changes in address.

Figure 4.17
Example of Drive 3 Control and Reference

```
|  Drive 3              Drive 3                        Drive 3    |
|  User                 User                           START      |
|  Momentary            Maintained                     Command    |
|  START                NOT STOP                       Bit        |
|  Input                Input                                     |
|    I:2.0                I:2.0                          O:1.6    |
|-----] [--------------------] [----------------------------( )-----|
| |      4              |      5                           1      |
| |Drive 3   |Drive 3   |                                         |
| |START     |RUNNING   |                                         |
| |Command   |Status    |                                         |
| |Bit       |Bit       |                                         |
| |  O:1.6      I:1.6   |                                         |
| +---] [--------]/[----+                                         |
|        1          1                                             |
|  Drive 3                                             Drive 3    |
|  User                                                STOP       |
|  Maintained                                          Command    |
|  NOT STOP                                            Bit        |
|  Input                                                          |
|    I:2.0                                               O:1.6    |
|-----]/[-------------------------------------------------( )-----|
| |      5              |                                  0      |
| |Drive 3   |Drive 3   |                                         |
| |STOP      |RUNNING   |                                         |
| |Command   |Status    |                                         |
| |Bit       |Bit       |                                         |
| |  O:1.6      I:1.6   |                                         |
| +---] [--------] [----+                                         |
|        0          1                                             |
|                                                  Drive 3         |
|                                                  Frequency       |
|                                                  Reference       |
|                                             +MOV---------------+ |
|---------------------------------------------|MOVE              +-|
|                                             |Source       N20:2| |
|                                             |                 0| |
|                                             |Dest         O:1.7| |
|                                             |                 0| |
|                                             +------------------+ |
```

The following data table shows the input data read from the SLC to SCANport module via the SLC backplane.

```
address        15       data        0
I:1               0000 0000 0000 0000   SCANport Channels 1 & 2 Channel Status
I:1.1             0000 0000 0000 0000   SCANport Channel 3 Channel Status
I:1.2             0000 0000 0000 0000   SCANport Channel 1 Logic Status
I:1.3             0000 0000 0000 0000   SCANport Channel 1 Feedback
I:1.4             0000 0000 0000 0000   SCANport Channel 2 Logic Status
I:1.5             0000 0000 0000 0000   SCANport Channel 2 Feedback
I:1.6             0000 0000 0000 0000   SCANport Channel 3 Logic Status
I:1.7             0000 0000 0000 0000   SCANport Channel 3 Feedback
I:1.8             0000 0000 0000 0000   SCANport Channel 1 Datalink A1 Out
I:1.9             0000 0000 0000 0000   SCANport Channel 1 Datalink A2 Out
I:1.10            0000 0000 0000 0000   SCANport Channel 1 Datalink B1 Out
I:1.11            0000 0000 0000 0000   SCANport Channel 1 Datalink B2 Out
I:1.12            0000 0000 0000 0000   SCANport Channel 1 Datalink C1 Out
I:1.13            0000 0000 0000 0000   SCANport Channel 1 Datalink C2 Out
I:1.14            0000 0000 0000 0000   SCANport Channel 1 Datalink D1 Out
I:1.15            0000 0000 0000 0000   SCANport Channel 1 Datalink D2 Out
I:1.16            0000 0000 0000 0000   SCANport Channel 2 Datalink A1 Out
I:1.17            0000 0000 0000 0000   SCANport Channel 2 Datalink A2 Out
I:1.18            0000 0000 0000 0000   SCANport Channel 2 Datalink B1 Out
I:1.19            0000 0000 0000 0000   SCANport Channel 2 Datalink B2 Out
I:1.20            0000 0000 0000 0000   SCANport Channel 2 Datalink C1 Out
I:1.21            0000 0000 0000 0000   SCANport Channel 2 Datalink C2 Out
I:1.22            0000 0000 0000 0000   SCANport Channel 2 Datalink D1 Out
I:1.23            0000 0000 0000 0000   SCANport Channel 2 Datalink D2 Out
I:1.24            0000 0000 0000 0000   SCANport Channel 3 Datalink A1 Out
I:1.25            0000 0000 0000 0000   SCANport Channel 3 Datalink A2 Out
I:1.26            0000 0000 0000 0000   SCANport Channel 3 Datalink B1 Out
I:1.27            0000 0000 0000 0000   SCANport Channel 3 Datalink B2 Out
I:1.28            0000 0000 0000 0000   SCANport Channel 3 Datalink C1 Out
I:1.29            0000 0000 0000 0000   SCANport Channel 3 Datalink C2 Out
I:1.30            0000 0000 0000 0000   SCANport Channel 3 Datalink D1 Out
I:1.31            0000 0000 0000 0000   SCANport Channel 3 Datalink D2 Out
```

The following data table shows the data to be sent to the SLC to SCANport module via the SLC backplane.

```
address       15      data       0
O:1               0000 0000 0000 0000   SCANport Channels 1 & 2 Channel Command
O:1.1             0000 0000 0000 0000   SCANport Channel 3 Channel Command
O:1.2             0000 0000 0000 0000   SCANport Channel 1 Logic Command
O:1.3             0000 0000 0000 0000   SCANport Channel 1 Reference
O:1.4             0000 0000 0000 0000   SCANport Channel 2 Logic Command
O:1.5             0000 0000 0000 0000   SCANport Channel 2 Reference
O:1.6             0000 0000 0000 0000   SCANport Channel 3 Logic Command
O:1.7             0000 0000 0000 0000   SCANport Channel 3 Reference
O:1.8             0000 0000 0000 0000   SCANport Channel 1 Datalink A1 In
O:1.9             0000 0000 0000 0000   SCANport Channel 1 Datalink A2 In
O:1.10            0000 0000 0000 0000   SCANport Channel 1 Datalink B1 In
O:1.11            0000 0000 0000 0000   SCANport Channel 1 Datalink B2 In
O:1.12            0000 0000 0000 0000   SCANport Channel 1 Datalink C1 In
O:1.13            0000 0000 0000 0000   SCANport Channel 1 Datalink C2 In
O:1.14            0000 0000 0000 0000   SCANport Channel 1 Datalink D1 In
O:1.15            0000 0000 0000 0000   SCANport Channel 1 Datalink D2 In
O:1.16            0000 0000 0000 0000   SCANport Channel 2 Datalink A1 In
O:1.17            0000 0000 0000 0000   SCANport Channel 2 Datalink A2 In
O:1.18            0000 0000 0000 0000   SCANport Channel 2 Datalink B1 In
O:1.19            0000 0000 0000 0000   SCANport Channel 2 Datalink B2 In
O:1.20            0000 0000 0000 0000   SCANport Channel 2 Datalink C1 In
O:1.21            0000 0000 0000 0000   SCANport Channel 2 Datalink C2 In
O:1.22            0000 0000 0000 0000   SCANport Channel 2 Datalink D1 In
O:1.23            0000 0000 0000 0000   SCANport Channel 2 Datalink D2 In
O:1.24            0000 0000 0000 0000   SCANport Channel 3 Datalink A1 In
O:1.25            0000 0000 0000 0000   SCANport Channel 3 Datalink A2 In
O:1.26            0000 0000 0000 0000   SCANport Channel 3 Datalink B1 In
O:1.27            0000 0000 0000 0000   SCANport Channel 3 Datalink B2 In
O:1.28            0000 0000 0000 0000   SCANport Channel 3 Datalink C1 In
O:1.29            0000 0000 0000 0000   SCANport Channel 3 Datalink C2 In
O:1.30            0000 0000 0000 0000   SCANport Channel 3 Datalink D1 In
O:1.31            0000 0000 0000 0000   SCANport Channel 3 Datalink D2 In
```

Datalinks

A simple datalink application on a 1336 PLUS drive is to set a parameter number into one of the Data In parameters. The SLC output image word for that datalink will then control the value of that parameter.

For example, on a 1336 PLUS drive connected to channel 1 of an SLC to SCANport module installed in slot 1, use datalink A1 to control the value of parameter 27. To do this, you need to:

1. Set the lowest bit of the second word of the G file to a 1. This enables Datalink A on channel 1).

2. Use a Human Interface Module (HIM) to set parameter 111 (*Data In A1*) to 27.

The value in O:1.8 now controls the value of parameter 27 in the 1336 PLUS drive.

A similar datalink application is to set a parameter number into one of the Data Out parameters. The value of that parameter is then displayed in the SLC input image word for that parameter.

For example, on a 1336 PLUS drive connected to channel 1 of a SLC to SCANport module installed in slot 1, use Datalink A1 to monitor the value of parameter 27. To do this, you need to:

1. Set the lowest bit of the second word of the G file to a 1. This enables Datalink A on channel 1).

2. Use a Human Interface Module (HIM) to set parameter 119 (*Data Out A1*) to 27.

The value in I:1.8 now monitors the value of parameter 27 in the 1336 PLUS drive.

ATTENTION: If you are using a 1336 FORCE or 1336 IMPACT, the datalink operations work differently. In these products, you need to link other parameters to the datalink parameters rather than program an index value. Refer to your user manual for these SCANport devices for information on creating links.

Troubleshooting

Chapter Objectives

Chapter 5 provides information about the LED states.

LED States

The following table provides information about the LED states.

LED	State	Description	Suggested action
Off	No module power	The rack is not receiving power.	Check the power supply connections to the rack.
Solid Red	Channel connection or power problem	The SCANport connection is not operational, or the SCANport device is not powered	1 Check to make sure that power is applied to the SCANport device. 2 Check the cable connections. 3 Change the SCANport cable. 4 Try using another channel. 5 Replace either the SLC to SCANport module or the SCANport device.
Flashing Red	Channel communication problem	The module cannot maintain or establish communications with the SCANport device.	1 Verify the configuration. 2 Remove the SCANport cable. 3 Re–insert the SCANport cable into the channel to reset the condition.
Solid Green	Channel operational	I/O signals are being passed between the module and the SCANport device.	None
Flashing Green	Channel not enabled for I/O operations	The enable bit for the channel has not been set. Only messaging operations are functional.	Program the controller to set the data enable bit for the appropriate channel for I/O operation.
Solid Orange	Connected device problem	SCANport device incompatibility.	Consult the factory.

Specifications

Chapter Objectives

Chapter 6 provides the specifications that you may need to install, repair, or use your SLC to SCANport communications module.

Product Specifications

The following are the product specifications.

This category:	Has these specifications
Operating temperature	0 – +60°C (+32 – +140°F)
Storage temperature	–40 – +85°C (–104 – +185°F)
Relative humidity	5 – 95% non–condensing
Shock and vibration	Category I – less than 9 kilograms (20 pounds) per A–B guidelines
Power consumption	300 mA@5V — SLC Backplane 60 mA@12V — SCANport load (from each channel)
ESD susceptibility	IEC 801–2 to Level 3 (4KV contact, 8KV open air)
Regulatory agencies	UL 508C and CUL

European Union Directive Compliance

If this product is installed within the European Union or EEA regions and has the CE mark, the following regulations apply.

EMC Directive

This apparatus is tested to meet Council Directive 89/336/EEC Electromagnetic Compatibility (EMC) using a technical construction file and the following standards, in whole or in part:

- EN 50081–2 EMC—Generic Emission Standard, Part 2 —Industrial Environment
- EN 50082–2 EMC—Generic Immunity Standard, Part 2 —Industrial Environment

The product described in this manual is intended for use in an industrial environment.

Low Voltage Directive

This apparatus is also designed to meet Council Directive 73/23/EEC Low Voltage, by applying the safety requirements of EN61131–2 Programmable Controllers, Part 2—Equipment Requirements and Tests.

For specific information that the above norm requires, see the appropriate sections in this manual, as well as the following Allen–Bradley publications:

- Industrial Automation Wiring and Grounding Guidelines, publication 1770–4.11
- Guidelines for Handling Lithium Batteries, publication AG–5.4
- Automation Systems Catalog, publication B111

M0, M1, and G Files

Appendix Objectives

This appendix provides information about M0–M1 files and G files. The information is general in nature and supplements specific information contained in earlier chapters of this manual. Topics include:

- M0–M1 files
- G files

▶ The SLC to SCANport module is considered to be a specialty I/O module.

M0–M1 Files

M0 and M1 files are data files that reside only in specialty I/O modules, such as the SLC to SCANport module. There is no image for these files in the processor memory. The application of these files depends on the function of the particular specialty I/O module. The M0 file is a module output file (a write only file), and the M1 file is a module input file (a read only file).

M0 and M1 files can be addressed in your ladder program and they can also be acted upon by the specialty I/O module, independent of the processor scan. Keep the following in mind when creating and applying your ladder logic.

Important: During the processor scan, the ladder program can address M0 and M1 data with bit, word, or file instructions. Each time an M0–M1 file address is encountered in the program, an immediate data transfer to or from the specialty I/O module occurs. The impact these immediate data transfers have on processor scan time is described in appendix A of the *Advanced Programming Software User Manual*, Publication Number 1747–6.4.

Configuring M0–M1 Files Using APS Software

M0 and M1 files are configured as part of the I/O configuration procedure for the processor file. After you have assigned the specialty I/O module to a slot (the procedure is the same as assigning other modules), the following functions are displayed at the bottom of the APS screen:

READ CONFIG	ONLINE CONFIG		MODIFY RACK	MODIFY SLOT	DELETE SLOT	UNDEL SLOT	EXIT	SPIO CONFIG
F1	F2		F4	F5	F6	F7	F8	F9

To configure the M0 and M1 files:

1. Press *F9* for Specialty I/O Configuration. The following functions are displayed.

ISR NUMBER	MODIFY G FILE	ADVNCD SETUP	G FILE SIZE
F1	F3	F5	F7

2. Press *F5* for Advanced Setup. The following functions are displayed:

INPUT SIZE	OUTPUT SIZE	SCANNED INPUT	SCANNED OUTPUT	M0 FILE SIZE	M1 FILE SIZE
F1	F2	F3	F4	F5	F6

3. Press *F5* for M0 File Size.

4. Enter the number of M0 file words required (the required number is listed in the user manual for the specific specialty I/O module). For the SLC to SCANport module, enter 800.

5. Press *F6* for M1 File Size.

6. Enter the number of M1 file words required (the required number is listed in the user manual for the specific specialty module). For the SLC to SCANport module, enter 400.

The specialty I/O module may require that you also configure the G file and specify an ISR (interrupt subroutine) number. The SLC to SCANport module requires you to configure the G file; you do not need to specify an ISR number. These tasks are accomplished with function keys F1, F3, and F7 shown in step 1. G files are covered later in this appendix.

Addressing M0–M1 Files

M0 and M1 files use the following address format:

Mf:e.s/b

Where	M	=	module
	f	=	file type (0 or 1)
	e	=	slot (1–30)
	s	=	word (0 to maximum supplied by module)
	b	=	bit (0–15)

Restrictions on Using M0–M1 Data File Addresses

You can use M0 and M1 data file addresses in all instructions except the OSR instruction and the following instruction parameters:

Instruction	Parameter (uses file indicator #)
BSL, BSR	File (bit array)
SQO, SQC, SQL	File (sequencer file)
LFL, LFU	LIFO (stack)
FFL, FFU	FIFO (stack)

Monitoring Bit Addresses

For SLC 5/02 processors, the M0/M1 monitoring option is always disabled. (This processor does not let you monitor the actual state of each addressed M0/M1 address.) For SLC 5/03 and SLC 5/04 processors, you can choose to disable or enable the monitoring option by selecting *F6*, System Config, from the APS main menu.

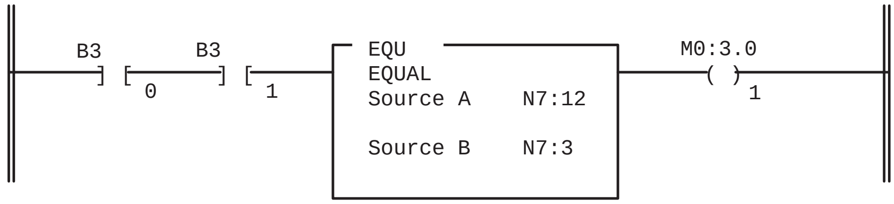
M0/M1 Monitoring Option Disabled

When you monitor a ladder program in the Run or Test mode with the M0/M1 monitoring option disabled, the following bit instructions, addressed to an M0 or M1 file, are indicated as false regardless of their actual true/false logical state.

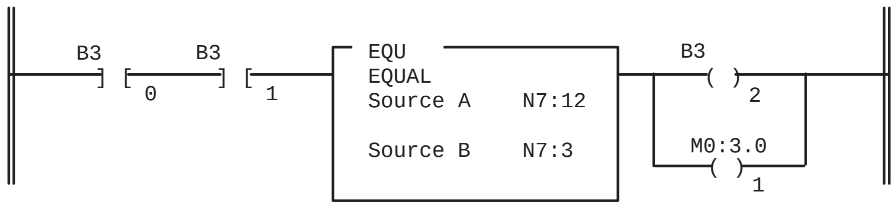
```
  Mf:e.s      Mf:e.s      Mf:e.s       Mf:e.s      Mf:e.s
 ---] [--   ---]/[--     ---( )--     --(L)--    ---(U)--
       b          b            b           b           b

f= file (0 or 1)
```

When you are monitoring the ladder program in the Run or Test mode, the APS or HHT display does not show these instructions as being true when the processor evaluates them as true.

If you need to show the state of the M0 or M1 addressed bit, you can transfer the state to an internal processor bit. This is shown below, where an internal processor bit is used to indicate the true/false state of a rung.



This rung will not show its true rung state because the EQU instruction is always shown as true and the M0 instruction is always shown as false.



OTE instruction B3/2 has been added to the rung. This instruction shows the true or false state of the rung.

M0/M1 Monitoring Option Enabled

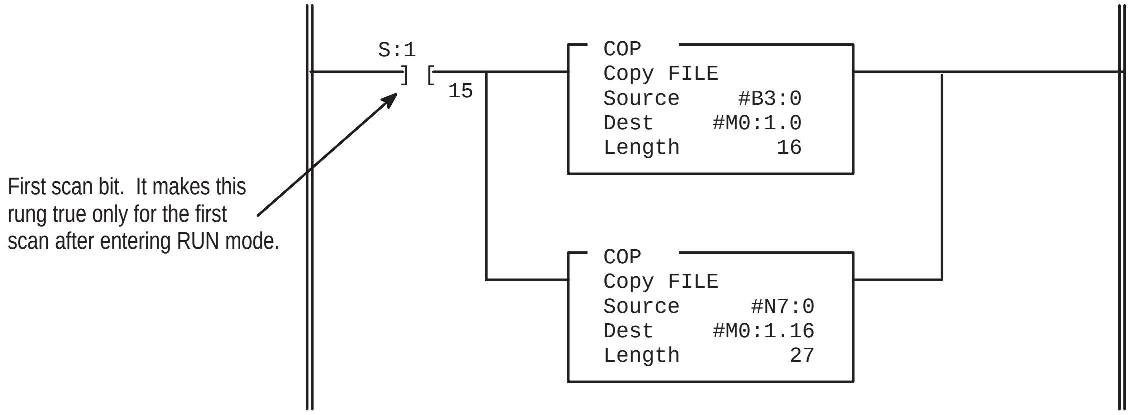
Important: The SLC 5/02 processor does not support this option.

The SLC 5/03 and SLC 5/04 processors let you monitor the actual state of each addressed M0/M1 address (or data table). The highlighting appears normal when compared to the other processor data files. The processor's performance is degraded to the degree of M0/M1 referenced screen data. For example, if your screen has only one M0/M1 element, degradation is minimal. If your screen has 69 M0/M1 elements, degradation is significant.

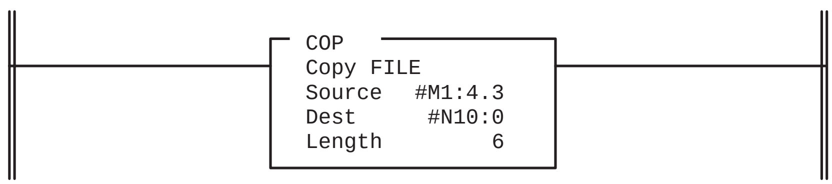
Transferring Data Between Processor Files and M0 and M1 Files

The processor does not contain an image of the M0 or M1 file. As a result, you must edit and monitor M0 and M1 file data via instructions in your ladder program. For example, you can copy a block of data from a processor data file to an M0 or M1 data file or vice versa using the COP instruction in your ladder program.

The following COP instructions copy data from a processor bit file and integer file to an M0 file. For the example, assume the data is configuration information that affects how the specialty I/O module operates.



The following COP instruction copies data from an M1 data file to an integer file. This technique is used to monitor the contents of an M0 or M1 data file indirectly, in a processor data file.



Access Time

During the program scan, the processor must access the specialty I/O card to read/write M0 or M1 data. You need to add this access time to the execution time of each instruction referencing M0 or M1 data. For the SLC 5/03 and SLC 5/04 processors, the instruction types vary in their execution times.

The following table shows approximate access times per instruction or word of data for the SLC 5/02, SLC 5/03, and SLC 5/04 processors.

Processor	Instruction Type	Access Time per Bit Instruction or Word of Data	Access Time per Multi–Word Instruction
SLC 5/02 Series B	All types [1]	1930 µs	1580 µs plus 670 µs per word
SLC 5/02 Series C	All types [1]	1160 µs	950 µs plus 400 µs per word
SLC 5/03 (All Series)	XIC or XIO	782 µs	—
	OTU, OTE, or OTL	925 µs	—
	COP to M file	—	772 µs plus 23 µs per word
	COP from M file	—	760 µs plus 22 µs per word
	FLL	—	753 µs plus 30 µs per word
	MVM to M file	894 µs	—
	any source or destination M file address	730 µs	—
SLC 5/04 OS400	XIC or XIO	743 µs	—
	OTU, OTE, or OTL	879 µs	—
	COP to M file	—	735 µs plus 23 µs per word
	COP from M file	—	722 µs plus 22 µs per word
	FLL	—	716 µs plus 30 µs per word
	MVM to M file	850 µs	—
	any source or destination M file address	694 µs	—

[1] Except the OSR instruction and the instruction parameters noted on page A–3.

SLC 5/02 Processor Example

```
  M0:2.1      M1:3.1      M0:2.1
---] [--    ---]/[--     ---( )--
      1           1           10
```

If you are using an SLC 5/02 Series B processor, add 1930 µs to the program scan time for each bit instruction addressed to an M0 or M1 data file. If you are using an SLC 5/03 Series C processor, add 1160 µs.

```
      ┌ COP ──────────────┐
──────┤ Copy FILE         ├──────────║
      │ Source      #B3:0 │          ║
      │ Dest       #M:1.0 │          ║
      │ Length         34 │          ║
      └───────────────────┘          ║
```

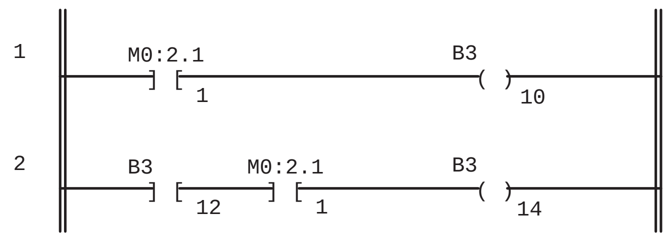
If you are using an SLC 5/02 Series B processor, add 1580 µs plus 670 µs per word of data addressed to the M0 or M1 file. As shown above, 34 words are copied from #B3:0 to M0:1.0. Therefore, this adds 24360 µs to the scan time of the COP instruction. If you are using an SLC 5/02 Series C processor, add 950 µs plus 400 µs per word. This adds 14550 µs to the scan time of the COP instruction.

SLC 5/03 Processor Example

The SLC 5/03 access times depend on the instruction type. Consult the previous table for the correct access times to add. As an example, if you use a COP to M file instruction like the one shown, add 772 µs plus 23 µs per word. This adds 1554 µs to the SLC 5/03 scan time due to the COP instruction.

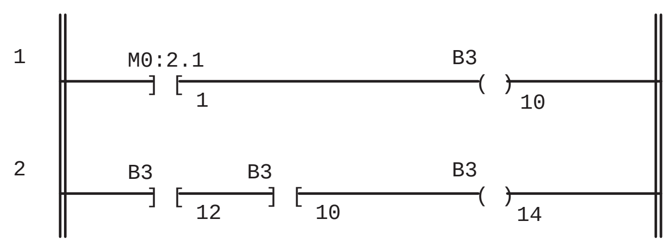
Minimizing the Scan Time

To keep the processor scan time to a minimum, reduce the use of instructions addressing the M0 or M1 files. For example, XIC instruction M0:2.1/1 is used in rungs 1 and 2 of the following example, adding approximately 2 ms to the scan time if you are using a SLC 5/02 Series B processor.



XIC instructions in rungs 1 and 2 are addressed to the M0 data file. Each of these instructions adds approximately 1 ms to the scan time (SLC 5/02 Series B Processor).

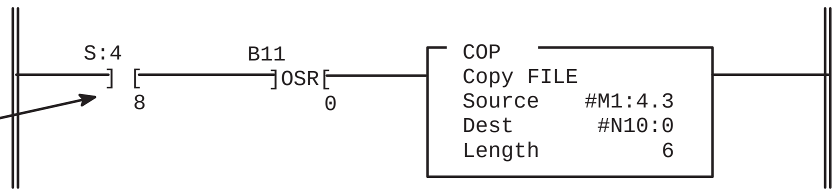
In the equivalent rungs shown below, XIC instruction M0:2.1/1 is used only in rung 1, reducing the SLC 5/02 scan time by approximately 1 ms.



These rungs provide equivalent operation to those of figure A by substituting XIC instruction B3/10 for XIC instruction M0:2.1/1 in rung 2. Scan time is reduced by approximately 1 ms (Series B processor).

The following figure shows another economizing technique. The COP instruction addresses an M1 file, adding approximately 4.29 ms to the scan time if you are using a SLC 5/02 Series B processor. Scan time economy is realized by making this rung true only periodically, as determined by clock bit S:4/8. (Clock bits are covered in chapter 1 of the *Advanced Programming Software Reference Manual*, Publication 1747–6.11.) A rung such as this might be used when you want to monitor the contents of the M1 file, but monitoring need not be on a continuous basis.

S:4/8 causes the #M1:4.3 file to update the #N10:0 file every 2.56 seconds.



Capturing M0–M1 File Data

The first two ladder diagrams in the previous section show a technique that lets you capture and use M0 or M1 data as it exists at a particular time. In the first figure, bit M0:2.1/1 could change state between rungs 1 and 2. This could interfere with the logic applied in rung 2. The second figure avoids the problem. If rung 1 is true, bit B3/10 captures this information and places it in rung 2.

In the second example of the last section, a COP instruction is used to monitor the contents of an M1 file. When the instruction goes true, the six words of data in file #M1:4.3 is captured as it exists at that time and placed in file #N10.0.

G Files

Some specialty I/O modules use G (confiGuration) files (indicated in the specific specialty I/O module user's manual). You can think of the files as the software equivalent of DIP switches.

You can access and edit the content of G files offline from the I/O Configuration function. You cannot access G files under the Monitor File function. Data you enter into the G file is passed to the specialty I/O module when you download the processor file and enter the REM Run or any one of the REM Test modes.

Configuring G Files Using APS Software

The G file is configured as part of the I/O configuration procedure for the processor file. After you have assigned the specialty I/O module to a slot (the procedure is the same as assigning other modules except that you must specify the ID code of the specialty I/O module), the following functions are displayed at the bottom of the APS screen:

READ CONFIG	ONLINE CONFIG		MODIFY RACK	MODIFY SLOT	DELETE SLOT	UNDEL SLOT	EXIT	SPIO CONFIG
F1	F2		F4	F5	F6	F7	F8	F9

This is the starting point for configuring the G file and other parameters of the specialty I/O module.

To create and monitor the G file.

1. Press *F9* for Specialty I/O Configuration. The following functions are displayed:

ISR NUMBER	MODIFY G FILE	ADVNCD SETUP	G FILE SIZE
F1	F3	F5	F7

2. Press *F7* for G File Size.

3. Specify the number of words required for the specialty I/O module. For the SLC to SCANport module, enter 32.

4. Press *F3* for Modify G File. The content of the G file is displayed in the display area. Data is shown in the default form, decimal:

```
--------------------------------------------------------
address   0    1    2    3    4    5    6    7    8    9
G1:0     xxxx   0    0    0    0    0    0    0    0    0
G1:10       0   0    0    0    0    0
-------------------------------------------------------------
```

The function keys displayed below the data table indicate the three data formats available to you, binary data, decimal data, and hex/bcd data:

BINARY DATA	DECIMAL DATA	HEX/BCD DATA
F1	F2	F3

The following figure shows the three G file data formats that you can select. Word addresses begin with the file identifier G and the slot number you have assigned to the specialty I/O module. In this case, the slot number is 1. Sixteen words have been created (addresses G1:0 through G1:15).

16–word G file, I/O slot 1, decimal format

```
address   0    1    2    3    4    5    6    7    8    9
G1:0     xxxx   0    0    0    0    0    0    0    0    0
G1:10       0   0    0    0    0    0
```

16–word G file, I/O slot 1, hex/bcd format

```
address   0    1    2    3    4    5    6    7    8    9
G1:0    xxxx 0000 0000 0000 0000 0000 0000 0000 0000 0000
G1:10   0000 0000 0000 0000 0000 0000
```

16–word G file, I/O slot 1, binary format

```
address 15      data      0
G1:0    xxxx xxxx xxxx xxxx
G1:1    0000 0000 0000 0000
G1:2    0000 0000 0000 0000
G1:3    0000 0000 0000 0000
G1:4    0000 0000 0000 0000
G1:5    0000 0000 0000 0000
G1:6    0000 0000 0000 0000
G1:7    0000 0000 0000 0000
G1:8    0000 0000 0000 0000
G1:9    0000 0000 0000 0000
G1:10   0000 0000 0000 0000
G1:11   0000 0000 0000 0000
G1:12   0000 0000 0000 0000
G1:13   0000 0000 0000 0000
G1:14   0000 0000 0000 0000
G1:15   0000 0000 0000 0000
```

Editing G File Data

Edit the data in the G file according to your application and the requirements of the specialty I/O module. You edit the data offline under the I/O configuration function only. With the decimal and hex/bcd formats, edit data at the word level:

G1:1 = 234 (decimal format)
G1:1 = 00EA (hex/bcd format)

With the binary format, edit data at the bit level:

G1/19 = 1

Important: The processor automatically configures word 0 of the G file according to the particular specialty I/O module. You cannot edit word 0.

SCANport Messaging

Appendix Objectives

This appendix contains information to help you use SCANport. The following topics are covered:

- message and reply structures
- examples of SCANport message structures

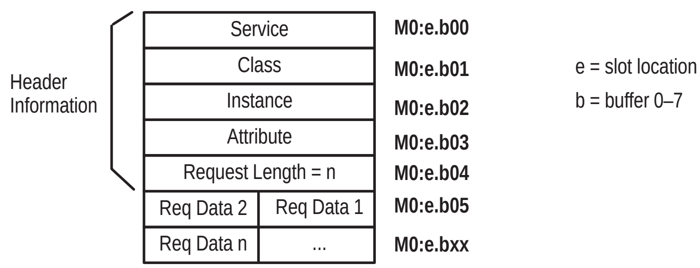
Message and Reply Structures

Before you can send a message, you need to copy the message into one of the M0 message buffers.

SCANport messages access data structures within the SCANport device. These data structures are called objects. An object contains information for a particular purpose. For example, a parameter object can contain information such as parameter values, parameter names, scaling information, and units.

Figure B.1 shows the first M0 message buffer structure.

Figure B.1
SLC to SCANport Module M0 Buffer 0 Message Structure



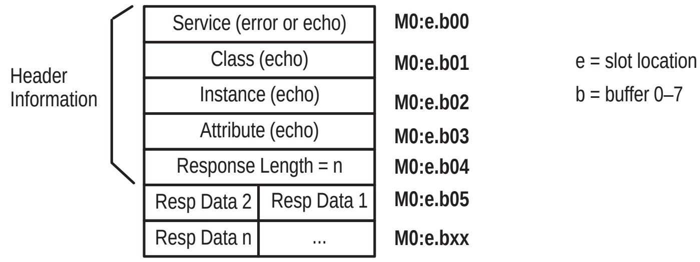
Where:

This field:	Specifies:
Service	The action or service requested. The following service values are available: **Enter this value:** — **To request this service:** 0001H (1 decimal) — Read Parameter Full/All Info 0005H (5 decimal) — Reset to Default ① 000eH (14 decimal) — Get Attribute Single 0010H (16 decimal) — Set Attribute Single 0015H (21 decimal) — Restore from Storage ① 0016H (22 decimal) — Save to Storage ① 0032H (50 decimal) — Get Attribute Scattered 0034H (52 decimal) — Set Attribute Scattered 004bH (75 decimal) — Read Enum String ① ① Not all SCANport devices support these services. For example, if you enter a service value of 0001H (1 decimal), you are requesting that the SCANport device provide all available information about a particular object. A service value of 000eH is a request for only one piece of information about a particular object.
Class	The type of object to access within the SCANport device. The class is the first index into the SCANport device's database. It directs the message to the desired functional database. For example, a class value of 000fH (15 decimal) indicates that the message is intended to access the parameter database.
Instance	A particular occurrence of an object in the SCANport device. The instance provides an index into the referenced functional database. For example, when accessing the parameter database, the instance value is the parameter number. If you want to access information about all instances of the object, specify an instance of 0.
Attribute	A specific piece of information about an object. Values are always less than 256. For example, in a parameter object, an attribute value of 0001H (1 decimal) indicates that the message is accessing the parameter value. An attribute value of 0007H (7 decimal) indicates that the message is accessing the parameter name text string.
Request length	The length, in bytes, in this request. This value is normally less than or equal to 96 bytes. However, Get/Set Attribute Scattered messages can be longer.
Req Data	The actual data portion of the request message. Up to a maximum of 96 bytes of message data is available. This field is optional depending on the type of message sent.

The response from the SCANport device appears in the message buffer of the channel used (Channel 1 = M1:e.100–199, Channel 2 = M1:e.200–299, Channel 3 = M1:e.300–399).

Figure B.2 shows the structure of the message response buffers inside the SLC M1 file.

Figure B.2
M1 File Message Response Buffer Structures



Where:

This field:	Contains:
Service	The same value as the service field of the request message if the message transaction was successful. If an error occurred, the service will be 0014H (20 decimal) and additional error information will be placed in the response data field. The error codes are provided at the end of this appendix.
Class	The same value that was used for the class field in the request message.
Instance	The same value that was used for the instance field in the request massage.
Attribute	The same value that was used for the attribute field in the request message.
Response Length[1]	The amount of data, in bytes, in this response. Most messages contain 96 or fewer bytes of data. However, Get/Set Attribute Scattered messages can be longer.
Resp Data[1]	The actual data portion of this response. This field varies in length depending on the message. If an error occurred, this field contains the SCANport error code.

[1] The size of the returned packets determines the response length. Some SCANport devices may return lengths greater than the actual amount of data in the response. These products always return a length which is a multiple of six. For example, a 1336 PLUS drive may reply to a Read Number of Parameters message with a length of six in the response. The first two bytes contain the complete response data. The extra data bytes should be ignored.

Available SCANport Messages

You can use the following SCANport messages:

This message:	Shown on page:	Lets you:
Read Number of Parameters	B–7	Request how many parameters the SCANport device supports.
Read Parameter Value	B–8	Request the value for a specific parameter.
Read Parameter Name Text	B–9	Request the name of a specific parameter.
Write a Value to Parameter	B–10	Write a value to a specific parameter.
Read Full Parameter	B–11	Request the name and value of a specific parameter.
Set Default Parameter Values ①	B–15	Reset the values of all parameters to the factory default values.
Restore Parameter Values from Non–volatile Storage ①	B–16	Restores the values of all parameters to the values stored in non–volatile storage.
Save Parameter Values to Non–volatile Storage ①	B–17	Saves the values of all parameters to non–volatile storage.
Read Enum String for a Value in Parameter ①	B–18	Request the text string that corresponds to a specific bit in a specific parameter.
Read Product Number	B–19	Request the product number from a device.
Read Product Text	B–20	Request the product text from a device.
Read Product Series Number	B–21	Request the product series number from a device.
Read Product Software Version	B–22	Request the product software version from a device.
Scattered Read	B–23	Request the values of multiple parameters, not necessarily starting from parameter 1.
Scattered Write	B–25	Write the values of multiple parameters, not necessarily starting from parameter 1.
Read Parameter Link from Parameter Number①	B–27	Request the parameter link information for a specific parameter.
Write Parameter Link from Parameter Number①	B–28	Write the parameter link information for a specific parameter.

① Not all SCANport devices support these messages.

SLC SCANport Messaging Ladder Program

When B3:0/0 is set to a value of 1, the program shown in Figure B.3 sends the message structure contained in N10:0 to the SCANport device connected to Channel 1 of the SLC to SCANport module. When B3:0/0, B3:0/1, and B3:0/2 have all been reset to zero, the message response has been received in N11:0.

Figure B.3
SLC Ladder Program Example

```
Rung 2:0
| Wait for  |SCANport  |SCANport                                       |
| Reply to  |Channel 1 |Channel 1                                      |
| Message   |READY     |DONE                                           |
|     B3:0     I:1.1      I:1.1           +COP---------------+    |
|----] [---------]/[---------] [------------+-+COPY FILE         +-+-|
|        2         9          8           | |Source   #M1:1.100| | |
|                                         | |Dest       #N11:0| | |
|                                         | |Length        100| | |
|                                         | +------------------+ | |
|                                         | Wait for             | |
|                                         | Reply to             | |
|                                         | Message              | |
|                                         |     B3:0             | |
|                                         +----(U)----------------+ |
|                                         |       2              | |
|                                         | Send                 | |
|                                         | SCANport             | |
|                                         | Channel 1            | |
|                                         | Message              | |
|                                         |   O:1.0              | |
|                                         +----(U)----------------+ |
|                                                 1                 |

Rung 2:1
| Send      |M0 File   |SCANport  |SCANport          Send         |
| Message   |Changed   |Channel 1 |Channel 1         SCANport     |
|                      |READY     |DONE              Channel 1    |
|                                                    Message      |
|     B3:0     I:1.1      I:1.1      I:1.1             O:1.0      |
|----] [---------] [---------] [---------]/[-----------+----(L)-----+-|
|        1         14         9          8           |      1    | |
|                                                   | Send      | |
|                                                   | Message   | |
|                                                   |     B3:0  | |
|                                                   +----(U)-----+ |
|                                                   |      1    | |
|                                                   | Wait for  | |
|                                                   | Reply to  | |
|                                                   | Message   | |
|                                                   |     B3:0  | |
|                                                   +----(L)-----+ |
|                                                          2       |
```

```
Rung 2:2
| Setup     |SCANport  |SCANport                                      |
| Send      |Channel 1 |Channel 1                                     |
| Message   |READY     |DONE                                          |
| Buffer    |                                                         |
|     B3:0     I:1.1      I:1.1            +COP---------------+   |
|----] [--------] [--------]/[-----------+-+COPY FILE         +-+-|
|       0         9         8            | |Source      #N10:0| | |
|                                        | |Dest     #M0:1.100| | |
|                                        | |Length         100| | |
|                                        | +------------------+ | |
|                                        | Message              | |
|                                        | Buffer               | |
|                                        | ID#                  | |
|                                        |   O:1.0              | |
|                                        +----(L)---------------+ |
|                                        |      2               | |
|                                        | Message              | |
|                                        | Buffer               | |
|                                        | ID#                  | |
|                                        |   O:1.0              | |
|                                        +----(U)---------------+ |
|                                        |      3               | |
|                                        | Message              | |
|                                        | Buffer               | |
|                                        | ID#                  | |
|                                        |   O:1.0              | |
|                                        +----(U)---------------+ |
|                                        |      4               | |
|                                        | Setup                | |
|                                        | Send                 | |
|                                        | Message              | |
|                                        | Buffer               | |
|                                        |     B3:0             | |
|                                        +----(U)---------------+ |
|                                        |      0               | |
|                                        | Send                 | |
|                                        | Message              | |
|                                        |     B3:0             | |
|                                        +----(L)---------------+ |
|                                              1                  |

Rung 2:3
|                                                                 |
|-------------------------------+END+-----------------------------|
|                                                                 |
```

Example Messages and Replies

This section provides examples of SCANport messages that you can send using the SLC to SCANport module. Each example contains two parts. The first part provides information about the SCANport message. Buffer 0 is used to send messages to the SCANport device connected to Channel 1. The second part provides examples that can be used with the SLC ladder program shown in Figure B.3. The message is contained in file N10, and the response is contained in file N11. All data file values are shown in hexadecimal. Some example messages also show file N11 in ASCII. You should note that in ASCII mode, the string is shown in a byte–swapped fashion.

Read Number of Parameters

The Read Number of Parameters message lets you request how many parameters the SCANport device supports. The following is an example of this request:

Read Number of Parameters

Value		
000e	**M0:e.000**	Service=Get Attribute Single
000f	**M0:e.001**	Parameter Class
0000	**M0:e.002**	Instance
0002	**M0:e.003**	Attribute=Last Parameter Number
0000	**M0:e.004**	Data Length = 0

Response

Value		
000e	**M1:e.100**	Service=Get Attribute Single
000f	**M1:e.101**	Parameter Class
0000	**M1:e.102**	Instance
0002	**M1:e.103**	Attribute=Last Parameter Number
0006	**M1:e.104**	Data Length=6 Bytes
00d8	**M1:e.105**	Data=216 Parameters
0000	**M1:e.106**	Discard
0000	**M1:e.107**	Discard

Figure B.4 shows an example of a Read Number of Parameters request. The reply to the Read Number of Parameters request message indicates that the SCANport device contains 216 (D8h) parameters.

Figure B.4
Example of Read Number of Parameters

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        000e   000f   0000   0002   0000   0000   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        000e   000f   0000   0002   0006   00d8   0000   0000   0000   0000
```

Read Parameter Value

The Read Parameter Value message lets you request the value for a specific parameter. The following is an example of this request:

Read Parameter Value

Value		
000e	**M0:e.000**	Service=Get Attribute Single
000f	**M0:e.001**	Parameter Class
0005	**M0:e.002**	Instance=Parameter #5
0001	**M0:e.003**	Attribute=Value
0000	**M0:e.004**	Data Length=0

Response

Value		
000e	**M1:e.100**	Service=Get Attribute Single
000f	**M1:e.101**	Parameter Class
0005	**M1:e.102**	Instance=Parameter #5
0001	**M1:e.103**	Attribute=Value
0002	**M1:e.104**	Data Length=2 Bytes
0007	**M1:e.105**	Data=Value of 7

Figure B.5 shows an example of a Read Parameter Value request. The value of parameter number 5 is 7.

Figure B.5
Example of Read Parameter Value

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        000e   000f   0005   0001   0000   0000   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        000e   000f   0005   0001   0002   0007   0000   0000   0000   0000
```

Read Parameter Name Text

The Read Parameter Name Text message lets you request the name of a specific parameter. The following is an example of this request:

Read Parameter Name Text

Value		
000e	**M0:e.000**	Service=Get Attribute Single
000f	**M0:e.001**	Parameter Class
0005	**M0:e.002**	Instance=Parameter #5
0007	**M0:e.003**	Attribute=Parameter Name
0000	**M0:e.004**	Data Length=0

Response

Value		
000e	**M1:e.100**	Service=Get Attribute Single
000f	**M1:e.101**	Parameter Class
0005	**M1:e.102**	Instance=Parameter #5
0007	**M1:e.103**	Attribute=Parameter Name
0011	**M1:e.104**	Data Length=17 Bytes
4610	**M1:e.105**	Lo Byte=Character Count of Parameter Name Hi Byte=First Character of Parameter Name (1st)
6572	**M1:e.106**	Parameter Name (Chars Lo=2nd, Hi=3rd)
2071	**M1:e.107**	Parameter Name (Chars Lo=4th, Hi=5th)
6553	**M1:e.108**	Parameter Name (Chars Lo=6th, Hi=7th)
656c	**M1:e.109**	Parameter Name (Chars Lo=8th, Hi=9th)
7463	**M1:e.110**	Parameter Name (Chars Lo=10th, Hi=11th)
3120	**M1:e.111**	Parameter Name (Chars Lo=12th, Hi=13th)
2020	**M1:e.112**	Parameter Name (Chars Lo=14th, Hi=15th)
0020	**M1:e.113**	Lo Byte=Last Character of Parameter Name (16th) Hi Byte=Discard

Figure B.6 shows an example of a Read Parameter Name Text request. The parameter name text for parameter number 5 is *Freq Select 1*. Note that the low byte of word N11:5 indicates that the name text contains 16 (10h) bytes.

Figure B.6
Example of Read Parameter Name Text

```
address      0      1      2      3      4      5      6      7      8      9
N10:0         000e   000f   0005   0007   0000   0000   0000   0000   0000   0000

  address      0      1      2      3      4      5      6      7      8      9
N11:0         000e   000f   0005   0007   0011   4610   6572   2071   6553   656c
N11:10        7463   3120   2020   0020   0000   0000   0000   0000   0000   0000

  address      0      1      2      3      4      5      6      7      8      9
N11:0        \00\0E \00\0F \00\05 \00\07 \00\11  F \10  e  r      q   e  S   e  l
N11:10        t  c   1             \00    \00\00 \00\00 \00\00 \00\00 \00\00 \00\00
```

Write Value to Parameter

The Write Value to Parameter message lets you write a value to a specific parameter. The following is an example of this request:

Write a Value of 6 to Parameter 5

Value		
0010	**M0:e.000**	Service=Set Attribute Single
000f	**M0:e.001**	Parameter Class
0005	**M0:e.002**	Instance=Parameter #5
0001	**M0:e.003**	Attribute=Value
0002	**M0:e.004**	Data Length=2 Bytes
0006	**M0:e.005**	Data=Value of 6

Response

Value		
0010	**M1:e.100**	Service=Set Attribute Single
000f	**M1:e.101**	Parameter Class
0005	**M1:e.102**	Instance=Parameter #5
0001	**M1:e.103**	Attribute=Value
0000	**M1:e.104**	Data Length=0 Bytes

Figure B.7 shows an example of a Write Value to Parameter request. The drive accepted the message, and parameter number 5 now has a value of 6.

Figure B.7
Example of Write Value to Parameter

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        0010   000f   0005   0001   0002   0006   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        0010   000f   0005   0001   0000   0000   0000   0000   0000   0000
```

Read Full Parameter

The Read Full Parameter message lets you request the name and value of a specific parameter. The following is an example:

Read Full Parameter 7

Value		
0001	**M0:e.000**	Service=Get Attribute All
000f	**M0:e.001**	Parameter Class
0007	**M0:e.002**	Instance=Parameter #7
0000	**M0:e.003**	Attribute
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value		
0001	**M1:e.100**	Service=Get Attribute All
000f	**M1:e.101**	Parameter Class
0007	**M1:e.102**	Instance=Parameter #7
0000	**M1:e.103**	Attribute
0035	**M1:e.104**	Data Length=53 Bytes
0064	**M1:e.105**	Value
6400	**M1:e.106**	Lo Byte=Link Path Size = 0 Hi Byte=Descriptor Lo Byte
0200	**M1:e.107**	Lo Byte=Descriptor Hi Byte; Hi Byte=Data Type
1002	**M1:e.108**	Lo Byte=Data Size = 2 Bytes Hi Byte=Character Count of Parameter Name
6341	**M1:e.109**	Parameter Name (Chars Lo=1st, Hi=2nd)
6563	**M1:e.110**	Parameter Name (Chars Lo=3rd, Hi=4th)
206c	**M1:e.111**	Parameter Name (Chars Lo=5th, Hi=6th)
6954	**M1:e.112**	Parameter Name (Chars Lo=7th, Hi=8th)
656d	**M1:e.113**	Parameter Name (Chars Lo=9th, Hi=10th)
3120	**M1:e.114**	Parameter Name (Chars Lo=11th, Hi=12th)
2020	**M1:e.115**	Parameter Name (Chars Lo=13th, Hi=14th)
2020	**M1:e.116**	Parameter Name (Chars Lo=15th, Hi=16th)
5304	**M1:e.117**	Lo Byte=Character Count of Units String Hi Byte=First Character of Units String (1st)
6365	**M1:e.118**	Units String (Chars Lo=2nd, Hi=3rd)
0073	**M1:e.119**	Lo Byte=Last Character of Units String (4th) Hi Byte=Character Count of Help String (Always 0)
0000	**M1:e.120**	Minimum Value
ea60	**M1:e.121**	Maximum Value
03e8	**M1:e.122**	Default Value
0001	**M1:e.123**	Scaling Multiplier
000a	**M1:e.124**	Scaling Divisor
0001	**M1:e.125**	Scaling Base
0000	**M1:e.126**	Scaling Offset
0000	**M1:e.127**	Multiplier Link (Parameter Used as Multiplier Value)
0000	**M1:e.128**	Divisor Link (Parameter Used as Divisor Value)
0000	**M1:e.129**	Base Link (Parameter Used as Base Value)
0000	**M1:e.130**	Offset Link (Parameter Used as Offset Value)
0001	**M1:e.131**	Lo Byte=Decimal Precision; Hi Byte=Discard

The Descriptor and Scaling fields are described on the following pages.

Figure B.8 shows an example of a Read Full Parameter request. This message reads all the information about parameter 7 from the SCANport device. It is encoded as shown in the following tables. You should note that the data is not word aligned.

Figure B.8
Example of Read Full Parameter

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        0001   000f   0007   0000   0000   0000   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        0001   000f   0007   0000   0035   0064   6400   0200   1002   6341
N11:10       6563   206c   6954   656d   3120   2020   2020   5304   6365   0073
N11:20       0000   ea60   03e8   0001   000a   0001   0000   0000   0000   0000
N11:30       0000   0001   0000   0000   0000   0000   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0      \00\01 \00\0F \00\07 \00\00 \00 5  \00 d   d \00 \02\00 \10\02  c  A
N11:10      e  c      l   i  T   e  m   1                    S \04  c  e  \00 s
N11:20     \00\00 \EA ‘  \03\E8 \00\01 \00\0A \00\01 \00\00 \00\00 \00\00 \00\00
N11:30     \00\00 \00\01 \00\00 \00\00 \00\00 \00\00 \00\00 \00\00 \00\00 \00\00
```

The following table shows the starting address of individual attributes that can be read using the Read Full Parameter request.

Attribute	Starting Address	Size (Bytes)	Description
1 (01h)	N11:5	2	Parameter value
2 (02h)	N11:6 (Lo Byte)	1	Link path size (always 0)
4 (04h)	N11:6 (Hi Byte)	2	Descriptor — See descriptor table
5 (05h)	N11:7 (Hi Byte)	1	Data type — See Data Types table
6 (06h)	N11:8 (Lo Byte)	1	Parameter value data size in bytes
7 (07h)	N11:8 (Hi Byte)	17	Parameter name string: *Accel Time 1* The Hi byte of N11:8 is the number of characters in the string and is always 16 (10H)
8 (08h)	N11:17	5	Units string: *Secs* The Lo byte of N11:17 is the number of characters in the string and is always 4 (04H)
9 (09h)	N11:19 (Hi Byte)	1	Help string (always 0 indicating no help string)
10 (0Ah)	N11:20	2	Minimum value
11 (0Bh)	N11:21	2	Maximum value
12 (0Ch)	N11:22	2	Default Value
13 (0Dh)	N11:23	2	Scaling multiplier — see scaling formula
14 (0Eh)	N11:24	2	Scaling divisor — see scaling formula
15 (0Fh)	N11:25	2	Scaling base — see scaling formula
16 (10h)	N11:26	2	Scaling offset — see scaling formula
17 (11h)	N11:27	2	Multiplier link — parameter containing multiplier value
18 (12h)	N11:28	2	Divisor link — parameter containing divisor value
19 (13h)	N11:29	2	Base link — parameter containing base value
20 (14h)	N11:30	2	Offset link — parameter containing offset value
21 (15h)	N11:31 (Lo Byte)	1	Decimal precision (see scaling formula)

The following table shows how the data type field is coded:

Data Type Value	Description of Data Type of Parameter Value Field
1	16–bit word
2	16–bit unsigned integer
3	16–bit signed integer
4	Boolean
5	Short integer
6	Double integer
7	Long integer
8	Unsigned short integer

The descriptor bits are defined as follows:

This descriptor bit:	Has the following definition when set to 1:
0	Not used. This bit should always be 0.
1	Supports ENUM strings.
2	Supports scaling.
3	Supports scaling links.
4	Read only parameter.
5	Monitor parameter (parameter is continuously updated by SCANport device).
6	Supports extended precision scaling.

Scaling Formulas

Four scaling formulas are provided. Two scaling formulas are for use with extended precision scaling and two are for normal scaling. The decimal precision variable is always used to locate the decimal point for a display by counting from the rightmost digit. In extended precision scaling, the decimal precision variable is also used in the scaling formula.

The four formulas are shown here. The first two formulas are used when descriptor bit 6 is set to 1.

$$\text{Engineering Value} = \frac{(\text{Internal Value} + \text{Offset}) \times \text{Multiplier} \times \text{Base}}{\text{Divisor} \times 10^{\text{Decimal Precision}}}$$

$$\text{Internal Value} = \frac{\text{Engineering Value} \times \text{Divisor} \times 10^{\text{Decimal Precision}}}{\text{Multiplier} \times \text{Base}} - \text{Offset}$$

$$\text{Engineering Value} = \frac{(\text{Internal Value} + \text{Offset}) \times \text{Multiplier} \times \text{Base}}{\text{Divisor}}$$

$$\text{Internal Value} = \frac{\text{Engineering Value} \times \text{Divisor}}{\text{Multiplier} \times \text{Base}} - \text{Offset}$$

Set Default Parameter Values

The Set Default Parameter Values message lets you reset the values of all parameters to the factory default values. The following is an example of this request:

Set Default Parameter Values

Value	Address	Description
0005	**M0:e.000**	Service=Reset All to Factory Defaults
000f	**M0:e.001**	Parameter Class
0000	**M0:e.002**	Instance
0000	**M0:e.003**	Attribute
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value	Address	Description
0005	**M1:e.100**	Service=Reset All to Factory Defaults
000f	**M1:e.101**	Parameter Class
0000	**M1:e.102**	Instance
0000	**M1:e.103**	Attribute
0000	**M1:e.104**	Data Length=0 Bytes

Figure B.9 shows a Set Default Parameter Values request. This message has set all parameter values in the SCANport device's EEPROM and RAM to the factory default values.

Figure B.9
Example of Set Default Parameter Values

```
address       0      1      2      3      4      5      6      7      8      9
N10:0          0005   000f   0000   0000   0000   0000   0000   0000   0000   0000

  address       0      1      2      3      4      5      6      7      8      9
N11:0          0005   000f   0000   0000   0000   0000   0000   0000   0000   0000
```

Restore Parameter Values from Non–volatile Storage

The Restore Parameter Values from Non–volatile Storage message lets you restore the values of all parameters to the values stored in non–volatile storage. The following is an example of this request:

Restore Parameter Values from Non–Volatile Storage

Value		
0015	**M0:e.000**	Service=Restore from Storage
000f	**M0:e.001**	Parameter Class
0000	**M0:e.002**	Instance
0000	**M0:e.003**	Attribute
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value		
0015	**M1:e.100**	Service=Restore from Storage
000f	**M1:e.101**	Parameter Class
0000	**M1:e.102**	Instance
0000	**M1:e.103**	Attribute
0000	**M1:e.104**	Data Length=0 Bytes

Figure B.10 shows a Restore Parameter Values from Non–volatile Storage request. This message has successfully restored all SCANport device parameters in RAM from non–volatile storage.

Figure B.10
Example of Restore Parameter Values from Non–volatile Storage

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        0015   000f   0000   0000   0000   0000   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        0015   000f   0000   0000   0000   0000   0000   0000   0000   0000
```

Save Parameter Values to Non–volatile Storage

The Save Parameter Values to Non–volatile Storage message lets you save the values of all parameters to non–volatile storage. The following is an example of this request:

Save Parameter Values to Non–Volatile Storage

Value		
0016	**M0:e.000**	Service=Save to Storage
000f	**M0:e.001**	Parameter Class
0000	**M0:e.002**	Instance
0000	**M0:e.003**	Attribute
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value		
0016	**M1:e.100**	Service=Save to Storage
000f	**M1:e.101**	Parameter Class
0000	**M1:e.102**	Instance
0000	**M1:e.103**	Attribute
0000	**M1:e.104**	Data Length=0 Bytes

Figure B.11 shows a Save Parameter Values to Non–volatile Storage request. This message has successfully saved all SCANport device parameters from RAM to non–volatile storage.

Figure B.11
Example of Save Parameter Values to Non–volatile Storage

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        0016   000f   0000   0000   0000   0000   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        0016   000f   0000   0000   0000   0000   0000   0000   0000   0000
```

Read Enum String for Value in Parameter

The Read Enum String for Value in Parameter message lets you request the text string that corresponds to a specific bit in a specific parameter. The following is an example of this request:

Read Enum String for a Value of 1 in Parameter #5

Value		
004b	**M0:e.000**	Service=Get Enum String
000f	**M0:e.001**	Parameter Class
0005	**M0:e.002**	Instance=Parameter #5
0001	**M0:e.003**	Attribute=Value/Bit# = 1
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value		
004b	**M1:e.100**	Service=Get Enum String
000f	**M1:e.101**	Parameter Class
0005	**M1:e.102**	Instance=Parameter #5
0001	**M1:e.103**	Attribute=Value/Bit# = 1
000c	**M1:e.104**	Data Length=12 Bytes
6552	**M1:e.105**	Enum String (Chars Lo=1st, Hi=2nd)
6f6d	**M1:e.106**	Enum String (Chars Lo=3rd, Hi=4th)
6574	**M1:e.107**	Enum String (Chars Lo=5th, Hi=6th)
5020	**M1:e.108**	Enum String (Chars Lo=7th, Hi=8th)
746f	**M1:e.109**	Enum String (Chars Lo=9th, Hi=10th)
2020	**M1:e.110**	Enum String (Chars Lo=11th, Hi=12th)

Figure B.12 shows a Read Enum String for Value in Parameter request. In this example, parameter 5 has an Enum string of *Remote Pot* associated with a value of 1. Note that enum strings are all 12 characters long.

Figure B.12
Example of Read Enum String for Value in Parameter

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        004b   000f   0005   0001   0000   0000   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        004b   000f   0005   0001   000c   6552   6f6d   6574   5020   746f
N11:10       2020   0000   0000   0000   0000   0000   0000   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0      \00 K  \00\0F \00\05 \00\01 \00\0C  e  R   o  m   e  t   P      t  o
N11:10            \00\00 \00\00 \00\00 \00\00 \00\00 \00\00 \00\00 \00\00 \00\00
```

Read Product Number

The Read Product Number message lets you request the product number from the SCANport device. The following is an example of this request:

Read Product Number

Value		
000e	**M0:e.000**	Service=Get Attribute Single
0092	**M0:e.001**	Internal A–B Vendor Specific Class
0000	**M0:e.002**	Instance
0000	**M0:e.003**	Attribute
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value		
000e	**M1:e.100**	Service=Get Attribute Single
0092	**M1:e.101**	Internal A–B Vendor Specific Class
0000	**M1:e.102**	Instance
0000	**M1:e.103**	Attribute
0006	**M1:e.104**	Data Length=6 Bytes
0003	**M1:e.105**	Product Number (Value)
0000	**M1:e.106**	Discard
0000	**M1:e.107**	Discard

Figure B.13 shows a Read Product Number request. In this example, the product number is 3.

Figure B.13
Example of Read Product Number

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        000e   0092   0000   0000   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        000e   0092   0000   0000   0006   0003   0000   0000
```

Read Product Text

The Read Product Text message lets you request the product text from the SCANport device. The following is an example of this request:

Read Product Text

Value		
000e	**M0:e.000**	Service=Get Attribute Single
0092	**M0:e.001**	Internal A–B Vendor Specific Class
0000	**M0:e.002**	Instance
0001	**M0:e.003**	Attribute
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value		
000e	**M1:e.100**	Service=Get Attribute Single
0092	**M1:e.101**	Internal A–B Vendor Specific Class
0000	**M1:e.102**	Instance
0001	**M1:e.103**	Attribute
0012	**M1:e.104**	Data Length=18 Bytes
7542	**M1:e.105**	Product Name String (Chars Lo=1st, Hi=2nd)
206C	**M1:e.106**	Product Name String (Chars Lo=3rd, Hi=4th)
3331	**M1:e.107**	Product Name String (Chars Lo=5th, Hi=6th)
3633	**M1:e.108**	Product Name String (Chars Lo=7th, Hi=8th)
5020	**M1:e.109**	Product Name String (Chars Lo=9th, Hi=10th)
554c	**M1:e.110**	Product Name String (Chars Lo=11th, Hi=12th)
2053	**M1:e.111**	Product Name String (Chars Lo=13th, Hi=14th)
2020	**M1:e.112**	Product Name String (Chars Lo=9th, Hi=10th)
0000	**M1:e.113**	Discard

Figure B.14 shows a Read Product Text request.

Figure B.14
Example of Read Product Text

```
address      0      1      2      3      4      5      6      7      8      9
N10:0         000e   0092   0000   0001   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0         000e   0092   0000   0001   0012   7542   206c   3331   3633   5020
N11:10        554c   2053   2020   0000
```

Read Product Series Number

The Read Product Series Number message lets you request the product series number from a SCANport device. The following is an example of this request:

Read Product Series Number

Value		
000e	**M0:e.000**	Service=Get Attribute Single
0092	**M0:e.001**	Internal A–B Vendor Specific Class
0000	**M0:e.002**	Instance
0003	**M0:e.003**	Attribute
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value		
000e	**M1:e.100**	Service=Get Attribute Single
0092	**M1:e.101**	Internal A–B Vendor Specific Class
0000	**M1:e.102**	Instance
0003	**M1:e.103**	Attribute
0006	**M1:e.104**	Data Length=6 Bytes
0001	**M1:e.105**	Product Series Number (Value 1=A, 2=B, ...)
0000	**M1:e.106**	Discard
0000	**M1:e.107**	Discard

Figure B.15 shows a Read Product Series Number request. The product series is A (1=A, 2=B, and so forth).

Figure B.15
Example of Read Product Series Number

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        000e   0092   0000   0003   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        000e   0092   0000   0003   0006   0001   0000   0000
```

Read Product Software Version

The Read Product Software Version message lets you request the product software version from a SCANport device. The following is an example of this request:

Read Product Software Version

Value		
000e	**M0:e.000**	Service=Get Attribute Single
0092	**M0:e.001**	Internal A–B Vendor Specific Class
0001	**M0:e.002**	Instance
0001	**M0:e.003**	Attribute
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value		
000e	**M1:e.100**	Service=Get Attribute Single
0092	**M1:e.101**	Internal A–B Vendor Specific Class
0001	**M1:e.102**	Instance
0001	**M1:e.103**	Attribute
0006	**M1:e.104**	Data Length=6 Bytes
00ca	**M1:e.105**	Product Software Version (00ca=202=FRN2.02)
0000	**M1:e.106**	Discard
0000	**M1:e.107**	Discard

Figure B.16 shows a Read Product Software Version request. The software version for microprocessor 1 is FRN2.02. The instance number, set into N10:2, determines the microprocessor being accessed by this message. Some products have multiple microprocessors, and all products have at least one.

Figure B.16
Example of Read Product Software Version

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        000e   0092   0001   0001   0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        000e   0092   0001   0001   0006   00ca   0000   0000
```

Scattered Read

The Scattered Read message lets you request the values of multiple parameters, not necessarily starting from parameter 1. The following is an example of this request:

Scattered Read

Value		
0032	**M0:e.000**	Service=Get Attribute Scattered
0093	**M0:e.001**	Internal A–B Vendor Specific Class
0000	**M0:e.002**	Instance
0000	**M0:e.003**	Attribute
000c	**M0:e.004**	Data Length=12 Bytes
0001	**M0:e.005**	Parameter 1
0000	**M0:e.006**	Always 0
0002	**M0:e.007**	Parameter 2
0000	**M0:e.008**	Always 0
0003	**M0:e.009**	Parameter 3
0000	**M0:e.010**	Always 0

Note: The **Scattered Read** can continue on in this pattern for up to 47 parameters.

Response

Value		
0032	**M1:e.100**	Service=Get Attribute Scattered
0093	**M1:e.101**	Internal A–B Vendor Specific Class
0000	**M1:e.102**	Instance
0000	**M1:e.103**	Attribute
000c	**M1:e.104**	Data Length=12 Bytes
0001	**M1:e.105**	Parameter 1*
0078	**M1:e.106**	Value
0002	**M1:e.107**	Parameter 2*
0000	**M1:e.108**	Value
0003	**M1:e.109**	Parameter 3*
0000	**M1:e.110**	Value

Note: The **Scattered Read** can continue on in this pattern for up to 47 parameters.

* If an error has occurred while reading to this parameter, the MSB of the parameter number will be set to 1 and the value field will contain an error code (see the error table at the end of this appendix).

Figure B.17 shows a Scattered Read request. This example reads the values of three parameters.

N10:4 contains the length of the data in bytes (three parameters require 12 bytes). A pair of words are required for each parameter being read (starting at N10:5). The first word of each pair is the parameter number. The second word is a place holder.

The response message (N11 file) has the same structure as the request message with a few changes. If an error occurred while reading one of the parameters, the high bit of that parameter number is set and the second word of the pair contains an error code. If the high bit of the parameter number is not set, the second word of the pair contains the parameter value.

Figure B.17
Example of Scattered Read

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        0032   0093   0000   0000   000c   0001   0000   0002   0000   0003
N10:10       0000

  address     0      1      2      3      4      5      6      7      8      9
N11:0        0032   0093   0000   0000   000c   0001   0078   0002   0000   0003
N11:10       0000
```

Scattered Write

The Scattered Write message lets you write the values of multiple parameters, not necessarily starting from parameter 1. The following is an example of this request:

Scattered Write

Value		
0034	**M0:e.000**	Service=Set Attribute Scattered
0093	**M0:e.001**	Internal A–B Vendor Specific Class
0000	**M0:e.002**	Instance
0000	**M0:e.003**	Attribute
000c	**M0:e.004**	Data Length=12 Bytes
0005	**M0:e.005**	Parameter 5
0001	**M0:e.006**	Value
0006	**M0:e.007**	Parameter 6
0002	**M0:e.008**	Value
0007	**M0:e.009**	Parameter 7
0258	**M0:e.010**	Value

Note: The **Scattered Write** can continue on in this pattern for up to 47 parameters.

Response to Scattered Write

Value		
0034	**M1:e.100**	Service=Set Attribute Scattered
0093	**M1:e.101**	Internal A–B Vendor Specific Class
0000	**M1:e.102**	Instance
0000	**M1:e.103**	Attribute
000c	**M1:e.104**	Data Length=12 Bytes
0005	**M1:e.105**	Parameter 5*
0000	**M1:e.106**	Status (see error codes)
0006	**M1:e.107**	Parameter 6*
0000	**M1:e.108**	Status (see error codes)
0007	**M1:e.109**	Parameter 7*
0000	**M1:e.110**	Status (see error codes)

Note: The **Scattered Write** can continue on in this pattern for up to 47 parameters.

* If an error has occurred while writing to this parameter, the MSB of the parameter number will be set to 1.

Figure B.18 shows a Scattered Write request. This example writes three parameters.

N10:4 contains the length of the data in bytes (three parameters require 12 bytes). A pair of words are required for each parameter being read (starting at N10:5). The first word of each pair is the parameter number. The second word is the value to be written.

The response message (N11 file) has the same structure as the request message with a few changes. If an error occurred while writing one of the parameters, the high bit of that parameter number is set and the second word of the pair contains an error code. If the high bit of the parameter number is not set, the second word of the pair contains a zero.

Figure B.18
Example of Scattered Write

address	0	1	2	3	4	5	6	7	8	9
N10:0	0034	0093	0000	0000	000c	0001	0000	0002	0000	0003
N10:10	0000									

address	0	1	2	3	4	5	6	7	8	9
N11:0	0034	0093	0000	0000	000c	0001	0078	0002	0000	0003
N11:10	0000									

Read Parameter Link from Parameter Number

The Read Parameter Link from Parameter Number message lets you request the parameter link information for a specific parameter. The following is an example of this request:

Read Parameter Link from Parameter Number 5

Value	Address	Description
000e	**M0:e.000**	Service=Get Attribute Single
0099	**M0:e.001**	Internal A–B Vendor Specific Class
0005	**M0:e.002**	Instance=Parameter #5
0000	**M0:e.003**	Attribute=Link
0000	**M0:e.004**	Data Length=0 Bytes

Response

Value	Address	Description
000e	**M1:e.100**	Service=Get Attribute Single
0099	**M1:e.101**	Internal A–B Vendor Specific Class
0005	**M1:e.102**	Instance=Parameter #5
0000	**M1:e.103**	Attribute=Link
0006	**M1:e.104**	Data Length=6 Bytes
0006	**M1:e.105**	Link Number
0000	**M1:e.106**	Discard
0000	**M1:e.107**	Discard

Figure B.19 shows a Read Parameter Link from Parameter Number request. This example is a read of the link value of parameter 5 (the parameter number is in N10:2). The link value is 6.

Figure B.19
Example of Read Parameter Link from Parameter Number

```
address      0      1      2      3      4      5      6      7      8      9
N10:0         000e   0099   0005   0000   0000

  address      0      1      2      3      4      5      6      7      8      9
N11:0         000e   0099   0005   0000   0006   0006   0000   0000
```

Write Parameter Link from Parameter Number

The Write Parameter Link from Parameter Number message lets you write the parameter link information for a specific parameter. The following is an example of this request:

Write Parameter Link to Parameter Number 5

Value		
0010	**M0:e.000**	Service=Set Attribute Single
0099	**M0:e.001**	Internal A–B Vendor Specific Class
0005	**M0:e.002**	Instance=Parameter #5
0000	**M0:e.003**	Attribute=Link
0002	**M0:e.004**	Data Length=2 Bytes
0007	**M0:e.005**	Link Number

Response

Value		
0010	**M1:e.100**	Service=Set Attribute Single
0099	**M1:e.101**	Internal A–B Vendor Specific Class
0005	**M1:e.102**	Instance=Parameter #5
0000	**M1:e.103**	Attribute=Link
0000	**M1:e.104**	Data Length=0

Figure B.20 shows a Write Parameter Link from Parameter Number request. This example is a write to the link value of parameter 5 (the parameter number is in N10:2). The link value being written is 6.

Figure B.20
Example of Write Parameter Link from Parameter Number

```
address     0      1      2      3      4      5      6      7      8      9
N10:0        0010   0099   0005   0000   0002   0006

  address     0      1      2      3      4      5      6      7      8      9
N11:0        0010   0099   0005   0000   0000
```

Error Codes

The following error codes are possible:

If you get this number:	Then:
0	No error occurred. The operation was successful.
1	The service failed. The SCANport device could not perform this request.
2	Service not supported.
3	Class not supported.
4	Instance not supported.
5	Attribute not supported.
6	Value out of range.
7	SCANport device conflict — cannot perform this request while the SCANport device is in the current state (usually while the drive is running).
0fdH (253)	Invalid message length. The message is too long or too short to transmit.
0feH (254)	The message timed out before the response was given. Check the SCANport connection.

B

C

D

E

G

L

M

S

T

We Want Our Manuals to be the Best!

You can help! Our manuals must meet the needs of you, the user. This is your opportunity to make sure they do just that. By filling out this form you can help us provide the most useful, thorough, and accurate manuals available. Please take a few minutes to tell us what you think. Then mail this form, FAX it, or send comments via E-Mail.

FAX: to your local Allen-Bradley Sales Office or 414/242-8579

E-Mail: via Internet to "SEPATTER@ABPOST.remnet.ab.com"

PUBLICATION NAME ______________________________

PUBLICATION NUMBER, DATE AND PART NUMBER (IF PRESENT) ______________________________

✔ **CHECK THE FUNCTION THAT MOST CLEARLY DESCRIBES YOUR JOB.**

❏ SUGGEST / RESPONSIBLE FOR THE PURCHASE OF EQUIPMENT
❏ MAINTAIN / OPERATE PROGRAMMABLE MACHINERY
❏ DESIGN / IMPLEMENT ELECTRICAL SYSTEMS
❏ TRAIN / EDUCATE MACHINE USERS
❏ SUPERVISE FLOOR OPERATIONS

✔ **WHAT LEVEL OF EXPERIENCE DO YOU HAVE WITH EACH OF THE FOLLOWING PRODUCTS?**

	NONE	LITTLE	MODERATE	EXTENSIVE
PROGRAMMABLE CONTROL	❏	❏	❏	❏
AC / DC DRIVES	❏	❏	❏	❏
PERSONAL COMPUTERS	❏	❏	❏	❏
NC / CNC CONTROLS	❏	❏	❏	❏
DATA COMMUNICATIONS / LAN	❏	❏	❏	❏

✔ **RATE THE OVERALL QUALITY OF THIS MANUAL BY CIRCLING YOUR RESPONSE BELOW. (1) = POOR (5) = EXCELLENT**

HELPFULNESS OF INDEX / TABLE OF CONTENTS	1	2	3	4	5
CLARITY	1	2	3	4	5
EASE OF USE	1	2	3	4	5
ACCURACY AND COMPLETENESS	1	2	3	4	5
QUALITY COMPARED TO OTHER COMPANIES' MANUALS	1	2	3	4	5
QUALITY COMPARED TO OTHER ALLEN-BRADLEY MANUALS	1	2	3	4	5

✔ **WHAT DID YOU LIKE MOST ABOUT THIS MANUAL?**

✔ **WHAT DID YOU LIKE LEAST ABOUT THIS MANUAL?**

✔ **PLEASE LIST ANY ERRORS YOU FOUND IN THIS MANUAL (REFERENCE PAGE, TABLE, OR FIGURE NUMBERS).**

✔ **DO YOU HAVE ANY ADDITIONAL COMMENTS?**

✔ **COMPLETE THE FOLLOWING.**

NAME ______________ COMPANY ______________

TITLE ______________ DEPARTMENT ______________

STREET ______________ CITY ______________ STATE ________ ZIP ______________

TELEPHONE ______________ DATE ______________

CUT ALONG DOTTED LINE

FOLD HERE

FOLD HERE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 413 MEQUON, WI

POSTAGE WILL BE PAID BY ADDRESSEE

ALLEN-BRADLEY
Attn: Marketing Communications
P.O. Box 760
Mequon, WI 53092-9907

Rockwell Automation

Allen-Bradley

Allen-Bradley, a Rockwell Automation Business, has been helping its customers improve productivity and quality for more than 90 years. We design, manufacture and support a broad range of automation products worldwide. They include logic processors, power and motion control devices, operator interfaces, sensors and a variety of software. Rockwell is one of the world's leading technology companies.

Worldwide representation.

Argentina • Australia • Austria • Bahrain • Belgium • Brazil • Bulgaria • Canada • Chile • China, PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Ecuador • Egypt • El Salvador • Finland • France • Germany • Greece • Guatemala • Honduras • Hong Kong • Hungary • Iceland • India • Indonesia • Ireland • Israel • Italy • Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Malaysia • Mexico • Netherlands • New Zealand • Norway • Pakistan • Peru • Philippines • Poland • Portugal • Puerto Rico • Qatar • Romania • Russia–CIS • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • United Arab Emirates • United Kingdom • United States • Uruguay • Venezuela • Yugoslavia

Allen-Bradley Headquarters, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414 382-2000 Fax: (1) 414 382-4444

Publication 1203–5.9 — October 1996

PN 74103–824–01