FISHER-ROSEMOUNT

RS3 ™

PeerWay Interfaces Manual

Performance Series 1, Release 4.0

September 1999
U.S. Manual PN: 1984-2650-2100
A-4 Size Manual PN: 1984-2650-2110

1984-2650-5007
1984-2650-5017

Printed in the U.S.A.

Fisher-Rosemount Systems, Inc.
8301 Cameron Road
Austin, Texas 78754–3895 U.S.A.

Telephone: (512) 835–2190
FAX: (512) 834–7313

Comment Form
RS3™ Manuals

PeerWay Interfaces Manual 1984-2650-21x0

Please give us your feedback to help improve this manual.

	Never	Rarely	Sometimes	Usually	Always
1. Do you actually use this manual when you are:					
• configuring	______	______	______	______	______
• making changes or enhancements	______	______	______	______	______
• operating the system	______	______	______	______	______
• troubleshooting	______	______	______	______	______
• other ______________________	______	______	______	______	______
2. Does this manual answer your questions?	______	______	______	______	______

3. What could be changed in this manual to make it more useful?

Errors and Problems: Please note errors or problems in this manual, including chapter and page number of affected information, if applicable; or send a marked-up copy of the affected page(s).

May we contact you about your comments? ☐ Yes ☐ No

Name ______________________________

Company ______________________________

Phone ______________________________

Date ______________________________

Fisher-Rosemount FAX #: (512) 834-7200

Thank you!

Name ______________________________

Company ______________________________

Address ______________________________

____

____

Place
stamp
here

Fisher-Rosemount Systems, Inc.
Technical Documentation
8301 Cameron Road
Austin, TX 78754-3895
U.S.A.

Seal with tape.

FISHER·ROSEMOUNT

RS3™
PeerWay Interfaces Manual

About This Manual

Changes for This Release

-
-
-
-
-

Revision Level for This Manual

For This Software Version:	Refer to This Document:		
	Title	Date	Part Number
P1R4.0	PeerWay Interfaces Manual	September 1999	1984-2650-
P1R1.0	PeerWay Interfaces Manual	May 1996	1984–2650–21x0
18R2	PeerWay Interfaces Manual	August 1994	1984-2650-18x1
18R1	PeerWay Interfaces Manual	October 1993	1984-2650-18x1
17	PeerWay Interfaces Manual	August 1992	1984-2650-17x1
16	PeerWay Interfaces Manual	January 1992	1984-2650-16x1
15	PeerWay Interfaces Manual	January 1991	1984-2650-15x1

References to Other Manuals

References to other RS3 user manuals list the manual, chapter, and sometimes the section as shown below.

Sample Entries:

For ..., see CC: 3.

Manual Title — Chapter

For ..., see CC: 1-1.

Manual Title — Chapter-Section

Abbreviations of Manual Titles

AL = Alarm Messages
BA = ABC Batch
CB = ControlBlock Configuration
CC = Console Configuration
DT = Disk and Tape Functions
IO = I/O Block Configuration
OP = Operator's Guide
OV = System Overview and Glossary
PW = PeerWay Interfaces
RB = Rosemount Basic Language
RI = RNI Release Notes and Installation Guide
RP = RNI Programmer's Reference Manual
SP = Site Preparation and Installation
SV = Service

Reference Documents

Prerequisite Documents

You should be familiar with the information in the following documents before using this manual:

System Overview Manual and Glossary	1984-2640-21x0
Software Release Notes, Performance Series 1	1984-2818-0110

Related Documents

You may find the following documents helpful when using this manual:

ABC Batch Software Manual	1984-2654-21x0
Alarm Messages Manual	1984-2657-19x1
ABC Batch Quick Reference Guide	1984-2818-1103
Configuration Quick Reference Guide	1984-2812-0808
Console Configuration Manual	1984-2643-21x0
ControlBlock Configuration Manual	1984-2646-21x0
I/O Block Configuration Manual	1984-2645-21x0
Operator's Guide	1984-2647-19x1
Rosemount Basic Language Manual	1984-2653-21x0
RNI Programmer's Reference Manual	1984-3356-02x1
RNI Release Notes and Installation Guide	1984-3357-02x1
Service Manual, Volume 1	1984-2648-21x0
Service Manual, Volume 2	1984-2648-31x0
Site Preparation and Installation Manual	1984-2642-21x0
Software Discrepancies for Performance Series 1	1984-2818-0311
User Manual Master Index	1984-2641-21x0

Contents

FISHER·ROSEMOUNT

RS3 ™
PeerWay Interfaces Manual

Chapter 1:
Supervisory Computer Interface (SCI)

List of Figures

List of Tables

Section 1:
SCI Protocols

This section describes how information moves between the Supervisory Computer Interface (SCI) and the host computer.

You can configure the SCI to communicate using one of three protocols:

- Asynchronous terminator
- Asynchronous length
- X.25

Asynchronous Terminator Protocol

Asynchronous terminator protocol is a character-oriented protocol that allows low-speed access to the RS3 control system. Most half-duplex and full-duplex ASCII asynchronous devices, such as personal computers and small minicomputers, can communicate with the SCI using asynchronous terminator protocol.

Table 1.1.1 summarizes the characteristics of the asynchronous terminator protocol. Figure 1.1.1 shows the general format of asynchronous terminator protocol messages.

Table 1.1.1. Asynchronous Terminator Protocol Characteristics

Area	Characteristics
Message content	• Messages consist of ASCII characters.
Message termination	• **Host to SCI**--The host must terminate each message with at least one unprintable ASCII character (hex 00 to 1F, and 7F). The SCI interprets an unprintable character as the end of a message. • The SCI interprets the next printable ASCII character (hex 20 to 7E) as the beginning of the next message. • **SCI to host**--The SCI terminates each message with configurable trailer characters ("Message Trailer String in Hex Format" field on the SCI CONFIGURATION screen).
Error handling	• **Host to SCI**--If the SCI receives a message incorrectly, it ignores the message and waits for another one. • **SCI to Host**--If the host receives a message incorrectly or if no reply is received after a user-determined period of time, then the host should retransmit the request message.
Applications	• Useful for low-speed access to RS3 using devices such as personal computers. • Compatible with most ASCII asynchronous interfaces.

Host to SCI Communication

ASCII Printable Character	ASCII Printable Character		ASCII Unprintable Character

The SCI interprets the next ASCII printable character as the start of the next message.

SCI to Host Communication

Configurable Header Characters	ASCII Printable Character	ASCII Printable Character		Configurable Trailing Characters

Figure 1.1.1. Asynchronous Terminator Protocol General Message Formats

Asynchronous Length Protocol

Asynchronous length protocol provides fast throughput of data by allowing messages to be transmitted in ASCII or binary format. You must program the host computer to transmit and receive messages in asynchronous length format. Table 1.1.2 summarizes the characteristics of the asynchronous length protocol. Figure 1.1.2 shows the general format of asynchronous length protocol messages.

Table 1.1.2. Asynchronous Length Protocol Characteristics

Area	Characteristics
Message content	• Messages consist of ASCII characters and binary data. • Each message contains two 2-byte binary integers before the message identifier. – The first integer contains the message length. – The second integer contains the negation of the message length. • If the message lengths add up to zero, then the message is OK.
Message termination	• The message length and negation of message length binary integers are used to determine the length of the message. • If these integers do not add up to zero, the configurable transmission gap ("TX Gap" field on the SCI Configuration screen) can be used to interpret the end of the message.
Error handling	• **Host to SCI**--If the SCI receives a message incorrectly, it waits until the end of the message and then discards it. • **SCI to Host**--If the host receives a message incorrectly or if no reply is received after a user-determined period of time, then the host should retransmit the request message.
Applications	• Useful for applications that need fast throughput.

Host to SCI Communication

Message Length (binary)	Negation of message length (binary)	ASCII Character or binary data		ASCII Character or binary data

SCI to Host Communication

Header Characters	Message Length (binary)	Negation of message length (binary)	ASCII Character or binary data		ASCII Character or binary data	Trailer Characters

Figure 1.1.2. Asynchronous Length Protocol General Message Format

X.25 Protocol

X.25 protocol is fully compatible with the 1985 CCITT Recommendation X.25. The host computer must be able to initiate and participate in an X.25 session. Table 1.1.3 summarizes the X.25 protocol characteristics.

Table 1.1.3. X.25 Protocol Characteristics

Area	Characteristics
X.25 Compatibility	• Fully compatible with 1985 CCITT X.25 specification. • Supports LAPB (Link Access Protocol-Balanced). • Supports up to 50 virtual connections with the SCI.
Protocol interface	• Synchronous interface. • The host computer initiates all X.25 sessions. • The SCI can process up to 10 messages at one time.
Message handling	• X.25 protocol handles error detection and recovery.
Message content	• The content of X.25 messages can be ASCII characters or binary data.
Applications	• Useful for applications that require X.25 support.

Section 2:
SCI Configuration

This section describes how to use the SCI Configuration screen.

SCI Configuration Screen

The purpose of the SCI Configuration screen is to match the operation of the SCI to your host computer. Different fields appear on the screen, depending on the entries selected. Figure 1.2.1 shows the SCI Configuration screen. Table 1.2.1 alphabetically lists all of the fields that can appear on the screen.

❑ **To call up the SCI Configuration screen, type:**

[C] [S] [*node #*] [ENTER] at the command line

```
                           SCI CONFIGURATION SCREEN         27-Jul-92  11:27:58
Node Address⇒ xx  Program Version X.XX   Boot X.XX      Host Comm  ACTIVE
Selected Comm Port⇒ RS232  Comm Port Baud Rate⇒ 9600     Checksum⇒ No
Data Bits and Parity⇒ 7M    XON/XOFF Control⇒ No       EIA Option⇒ None
Block Update Disable⇒ Yes    Log All Changes to Change Log⇒ Yes
Reply Header String in Hex format⇒ 0A 28 00 00     Mode⇒ Async Terminator
Reply Trailer String in Hex format⇒ 29 0D 00 00
Extended Write Permissions⇒No
Periodic Alarm Generation⇒Multiple      Periodic Read Alarm Gen⇒No
SCI Login Password⇒                     SCI Configuration Filename⇒
                      Press <Enter> To:
Send Message Pairs⇒                     Save Configuration⇒
Save SCI Program⇒                       Load SCI Program⇒
                                                                CONFIG 1
```

Figure 1.2.1. SCI Configuration Screen

Table 1.2.1. SCI Configuration Screen Field Descriptions

Field	Description	Allowable Entries
Block Update Disable	Specifies whether or not all supervisory computer attempts to write to ControlBlocks are denied. The ability to write to an individual ControlBlock is configured on the ControlBlock CB Continuous Faceplate screen. To write to ControlBlocks, enter **No**. To disable writing to ControlBlocks, enter **Yes**. For more information, see the “Writing to ControlBlocks” heading in this section.	No, Yes
Boot	Shows the software version of the ROM on the processor board.	Display only
Checksum	Specifies whether or not a CRC checksum is used in request and reply messages. For more information, see the “Checksum Field” heading in this section.	No, Yes
Clock	(X.25 protocol only) Specifies the source of the clock signals for the transmitted and received data. If you configure the SCI as a terminal and use the RS-422 port, then enter **CLK EXT** and set the clock jumpers in the EXTERNAL position on the Nonvolatile Memory Card. If you configure the SCI as a modem and use the RS-422 port, then enter **CLK INT** and set the clock jumpers in the INTERNAL position on the Nonvolatile Memory Card. If you use the RS-232 port, enter **CLK NRZI**, **CLK FM0**, or **CLK FM1**, depending on the clock source. For information about Nonvolatile Memory Card clock jumper positions, see SV: 4.	CLK INT, CLK EXT, CLK NRZI, CLK FM0, CLK FM1
Comm Port Baud Rate	Specifies the baud rate of the link between the SCI and the host computer.	Asynchronous terminator and asynchronous length protocols: 1200, 2400, 4800, 9600, 19200, 38400 X.25 protocol: 300, 1200, 1800, 2400, 3600, 4800, 9600, 19200, 38400, 76800, 204800
Connect Timeout	(X.25 protocol only) Specifies the Recommendation X.25 link level idle transmit timeout (parameter T3). This is the length of time (in seconds) that a virtual connection can be idle before the host terminates the session. An entry of **0** indicates that no timeout value is in effect.	0 to 5000

(continued on next page)

Table 1.2.1. SCI Configuration Screen Field Descriptions (continued)

Field	Description	Allowable Entries
Data Bits and Parity	The number of data bits per character and the type of parity: **N**(none), **M**(mark), **E**(even), **O**(odd).	7N, 7M, 7E, 7O, 8N
Data Format	(Asynchronous length protocol only) Specifies the character format of certain message data. Most message data is in ASCII format; however, some messages allow data to be in binary format. **ASCII** specifies that message data is in ASCII character format. **DEC** specifies that message data is in DEC single-precision floating point format with LSB ordering. **IEEE** specifies that message data is in IEEE single-precision floating point format with MSB ordering.	ASCII, DEC, IEEE
EIA Option	Specifies an EIA communication option. **Modem** specifies that the SCI provides modem communication support. Modem can be used when the SCI is connected to a modem. When the SCI is configured as a terminal, the SCI monitors the modem CD signal to determine when the host has called. The SCI then asserts DTR, the modem answers the phone, and the modem establishes the carrier. When the SCI drops DTR, the modem hangs up. **CTS** specifies that the SCI and host use RTS and CTS flow control. CTS can be used when the SCI is directly connected to the host. When the SCI is configured as a modem and the host asserts DTR, then the SCI can transmit. When the SCI is configured as a terminal and the host asserts DSR, then the SCI can transmit. **None** specifies that no EIA options are used.	None, CTS, Modem
Extended Write Permissions	Determines whether or not extended write permissions are allowed. For information about standard and extended write permissions, see the “Writing to ControlBlocks” heading in this section.	No, Yes
Host Comm	Shows the active/inactive status of the SCI.	Display only
Load SCI Program	Loads the SCI operating program $$SCIPROG from disk to the SCI nonvolatile memory. However, the SCI does not run with the new program until you power off and then power on the SCI.	Cursor to the field and press [ENTER].

(continued on next page)

Table 1.2.1. SCI Configuration Screen Field Descriptions (continued)

Field	Description	Allowable Entries
Log All Changes to Change Log	Specifies whether or not all block changes made by the SCI are logged in the Operator Log. If **No** is specified, block changes may still be logged if requested by individual messages. For more information, see the "Log All Changes to Change Log Field" heading in this section.	No, Yes
Max Number of UNACKED Packets	(X.25 protocol only) Specifies the Recommendation X.25 parameter K, which is the maximum number of unacknowledged packets that can exist at one time.	1 to 7
Max Retries	(X.25 protocol only) Specifies the Recommendation X.25 parameter N2, which is the maximum number of times that a host attempts to send a packet without receiving an acknowledgement.	1 to 100
Message Header String in Hex Format	Specifies up to four optional characters with which the SCI begins reply messages. The characters are entered as hex values. A value of **00** indicates no character. All four character places must be entered. For example, to configure header characters hex 34 and 38, you must enter "34380000".	Any ASCII character hex value (01 to 7F); 00 indicates no character
Message Trailer String in Hex Format	Specifies up to four optional characters that the SCI appends to reply messages. The characters are entered as hex values. A value of **00** indicates no character. All four character places must be entered. For example, to configure trailer characters hex 34 and 38, you must enter "34380000".	Any ASCII character hex value (01 to 7F); 00 indicates no character
Mode	Specifies the mode of communication between the SCI and the host computer. **Async Terminator** specifies that the communication mode is asynchronous terminator protocol. **Async Length LSB** specifies that the communication mode is asynchronous length protocol, and that the message length, message length negation, and CRC characters use LSB ordering. **Async Length MSB** specifies that the communication mode is asynchronous length protocol, and that the message length, message length negation, and CRC characters use MSB ordering. **X.25 DCE** specifies that the communication mode is X.25 protocol, and that the SCI is functionally configured as a DCE. **X.25 DTE** specifies that the communication mode is X.25 protocol, and that the SCI is functionally configured as a DTE.	Asynch Terminator, Asynch Length LSB, Asynch Length MSB, X.25 DCE, X.25 DTE

(continued on next page)

Table 1.2.1. SCI Configuration Screen Field Descriptions (continued)

Field	Description	Allowable Entries
Node Address	Specifies the node number of the SCI.	Any node number
Packet Size	(X.25 protocol only) Specifies the recommendation X.25 parameter N1, which is the maximum number of bytes in an X.25 packet.	16 to 1024
Periodic Alarm Generation	Specifies how the“SCI Periodic Data Lost” and “SCI Failed Periodic Read of xxx” alarm messages are generated. An entry of **Multiple** specifies that these alarms are generated each time that the alarm condition occurs. An entry of **Single** specifies that the alarms are generated only when the alarm condition occurs. These messages indicate an unusual alarm condition. In most cases, you should configure **Multiple** to generate the alarms each time that they occur. However, if you read SCI table data through a slow HIA (Highway Interface Adapter), you may want to configure **Single** to reduce the number of alarm messages.	Multiple, Single
Periodic Read Alarm Gen	It is strongly recommended that you configure **No** for this field. Configuring **Yes** can result in a flood of alarms from the SCI.	No, Yes
Program Version	Shows the software version of the SCI operating program.	Display only
Retransmission Timeout	(X.25 protocol only) Specifies the Recommendation X.25 parameter T1, which is the length of time (in milliseconds) that the host waits before retransmitting a packet if it does not receive an acknowledgement.	10 to 9999
Save Configuration	Saves the SCI configuration from this screen to disk in the file specified in the “SCI Configuration Filename” field.	Cursor to the field and press [ENTER].
Save SCI Program	Saves the SCI operating program $$SCIPROG from the SCI nonvolatile memory to disk.	Cursor to the field and press [ENTER].
SCI Configuration Filename	Specifies a file name to which the SCI configuration can be saved.	Any valid file name

(continued on next page)

Table 1.2.1. SCI Configuration Screen Field Descriptions (continued)

Field	Description	Allowable Entries
SCI Login Password	Specifies a password that can be used in SCI/host communication. A password can be used as a security measure on dial-up and multi-use communication lines. If you configure a password, then communication between the SCI and host must begin with the Host Connect message, which includes the password. If you configure a password, then the host must send the SCI at least one valid message per minute, or the SCI disconnects the line. If communication ends for any reason, then the host must send the Host Connect message again to reestablish communication. If you are using a password with X.25 protocol, each virtual session with the SCI must begin with the Host Connect message. Each session must communicate with the SCI once a minute to stay connected.	Any string up to 16 characters long.
Selected Comm Port	Specifies the communication port that is used to connect the SCI to the host computer.	RS232, RS422
Send Message Pairs	Downloads the configured message pairs from the console at which you are configuring to the SCI nonvolatile memory. Some discrete data can be sent to the host as message pair text. The configured message pairs from a console are used to determine the message pair text.	Cursor to the field and press [ENTER].
TX Gap	(Asynchronous length protocol only) Specifies the smallest possible time period between host transmissions (in milliseconds). If the SCI receives a message and the message length and negation of message length fields do not add up to zero, the SCI terminates and discards the message after receiving no characters for the continuous time period in the “TX Gap” field.	1 to 1000
XON / XOFF Control	Specifies whether or not XON/XOFF flow control is used in SCI/host communications.	No, Yes

Log All Changes to Change Log Field

The "Log All Changes to Change Log" field on the SCI Configuration screen is used with the Log field in individual request messages to determine whether or not block changes made by the SCI are entered in the Operator Log.

Figure 1.2.2 shows the decision process the SCI uses to determine whether or not block changes made by a message are entered in the OperatorLog.

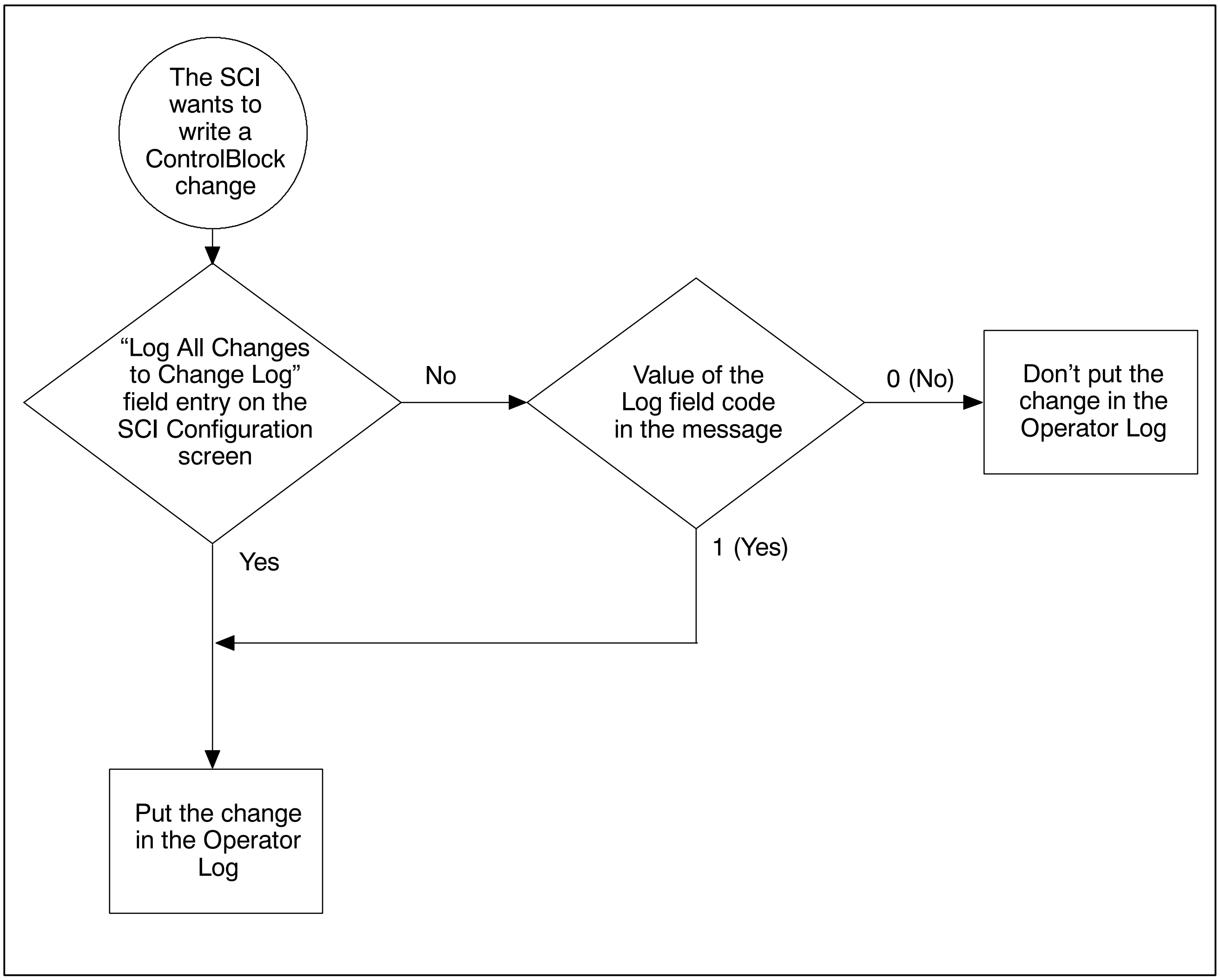


Figure 1.2.2. SCI Decision Process for Operator Log Entries

Writing to ControlBlocks

Table 1.2.2 summarizes configuration of the Continuous Faceplate screen and the SCI Configuration screen that is required to write to ControlBlock values.

Table 1.2.2. Summary of the Configuration Required to Write ControlBlocks

To write to this value:	These conditions must be met:
ControlBlock mode	• You must configure “SC⇒Yes” on the Continuous Faceplate screen. • In addition, to write Remote mode: – the ControlBlock must be a PID block. – you must link a value into input C. • In addition, to write Comp SP mode: – the ControlBlock must be a PID block. – the “Shed Mode” value on the Continuous Faceplate screen must be Manual, Auto, or Remote. • In addition, to write DDC mode: – the “Shed Mode” value on the Continuous Faceplate screen must be Manual, Auto, or Remote.
ControlBlock analog output	If you configure “Extended Write Permissions⇒No” on the SCI Configuration screen: • The “Shed Mode” value on the Continuous Faceplate screen must be Manual, Auto, or Remote. • You should configure “DDC” on the Continuous Faceplate screen. • The ControlBlock must be in DDC mode. If you configure “Extended Write Permissions⇒Yes” on the SCI Configuration screen: • The ControlBlock must be in DDC, Local, or Manual mode. • If the ControlBlock is in DDC mode: – the “Shed Mode” value on the Continuous Faceplate screen must be Manual, Auto, or Remote. – you should configure “DDC” on the Continuous Faceplate screen. • If the ControlBlock is in Local or Manual mode: – you must configure “SC⇒Yes” on the Continuous Faceplate screen.

(continued on next page)

Table 1.2.2. Summary of the Configuration Required to Write ControlBlocks (continued)

To write to this value:	These conditions must be met:
PID ControlBlock setpoint (analog input B)	If you configure “Extended Write Permissions⇒No” on the SCI Configuration screen: • The ControlBlock cannot have a value linked into input B. • The “Shed Mode” value on the Continuous Faceplate screen must be Manual, Auto, or Remote. • You should configure “CSP” on the Continuous Faceplate screen. • The ControlBlock must be in Comp SP mode. If you configure “Extended Write Permissions⇒Yes” on the SCI Configuration screen: • The ControlBlock cannot have a value linked into input B. • If the ControlBlock is in Comp SP mode: – the “Shed Mode” value on the Continuous Faceplate screen must be Manual, Auto, or Remote. – You should configure “CSP” on the Continuous Faceplate screen. • If the ControlBlock is in any mode except Comp SP: – You must configure “SC⇒Yes” on the Continuous Faceplate screen.
PID ControlBlock analog inputs A and C–O; non-PID ControlBlock analog inputs A–O	• The ControlBlock cannot have a value linked into the input. • You must configure “SC⇒Yes” on the Continuous Faceplate screen.
ControlBlock discrete inputs	• The ControlBlock cannot have a value linked into the input. • You must configure “SC⇒Yes” on the Continuous Faceplate screen. • REMINDER: If the discrete value is scaled as one-half of a message pair, then the 8-character value that the host writes must exactly match one of the messages assigned to that discrete value.

Section 3:
Message Formats

This section provides formats for SCI/host communication messages, including general formats for request and reply messages, followed by specific formats for individual message types.

Data Types

SCI messages provided in this section can use the following data types, shown in Table 1.3.1.

Table 1.3.1. SCI Message Data Types

Format	Meaning
b	Binary byte.
c	ASCII printing characters.
d	Decimal number.
f	Unscaled floating point value.
h	Hexadecimal number.
m	Discrete value (0 or 1) or message pair text.
p	Scaled floating point number, unscaled floating point number, discrete value, or message pair text.
s	Scaled or unscaled floating point number.
v	Variable format data described elsewhere.

How Do Request and Reply Messages Function?

This heading describes the function of request and reply messages and their general format.

How Are Request and Reply Messages Used?

The host initiates all communication between the host and the SCI with a request message. The SCI answers the request message with a reply message. Figure 1.3.1 shows how request and reply messages are used.

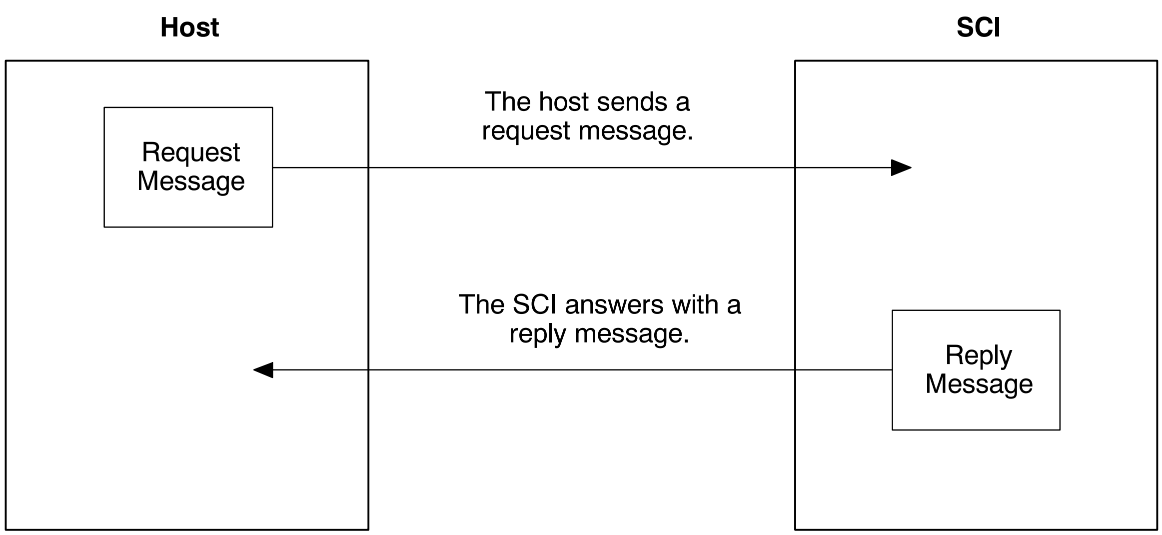


Figure 1.3.1. Request and Reply Messages

Request and Reply Message Formats

This section describes the general formats of request and reply messages. The specific format of a message depends on the message type. The message types are described later in this section.

Asynchronous Terminator and X.25 Protocols

Table 1.3.2 shows the request message format for asynchronous terminator and X.25 protocols. Table 1.3.3 shows the reply message format for asynchronous terminator and X.25 protocols.

Table 1.3.2. Request Message Format for Asynchronous Terminator and X.25 Protocols

Format	Field	Description
2h	Message ID	An alphanumeric code that identifies the message. For more information, see the “Message ID” heading in this section.
v	Body	The variable format message body. The request message bodies are described in the rest of this section.
4h	CRC	A hexadecimal CRC used to detect transmission errors. If you configure “Checksum⇒no” on the SCI Configuration screen, this field does not appear.
v	EOM	ASCII non-printable characters that signal the end of the message.

Table 1.3.3. Reply Message Format for Asynchronous Terminator and X.25 Protocols

Format	Field	Description
v	Header	A user-defined sequence of characters that tells the host this is the start of a message. You configure the header in the "Message Header String in Hex Format" field on the SCI Configuration screen.
2h	Message ID	An alphanumeric code that identifies the message. For more information, see the "Message ID" heading in this section.
1d	Alarm	A field that identifies whether or not an SCI alarm exists. You can use the Alarm Information message (type 42) to find out more information about SCI alarms. This field can have the following values: 0=no alarms exist 1=alarms exist
v	Body	The variable format message body. The reply message bodies are described in the rest of this section.
4h	CRC	A hexadecimal CRC used to detect transmission errors. If you configure "Checksum⇒no" on the SCI Configuration screen, this field does not appear.
v	Trailer	A user-defined sequence of characters that tells the host this is the end of a message. You configure the trailer in the "Message Trailer String in Hex Format" field on the SCI Configuration screen.

Asynchronous Length Protocol

Table 1.3.4 shows the general request message format for asynchronous length protocol. Table 1.3.5 shows the reply message format for asynchronous length protocol.

Table 1.3.4. Request Message Format for Asynchronous Length Protocol

Format	Field	Description
2b	Message Length	The number of bytes in the message.
2b	Message Length Negation	The negation of the number of bytes in the message.
2h	Message ID	An alphanumeric code that identifies the message. For more information, see the “Message ID” heading in this section.
v	Body	The variable format message body. The request message bodies are described in the rest of this section.
2b	CRC	A hexadecimal CRC used to detect transmission errors. If you configure “Checksum⇒no” on the SCI Configuration screen, this field does not appear.

Table 1.3.5. Reply Message Format for Asynchronous Length Protocol

Format	Field	Description
v	Header	A user-defined sequence of ASCII characters that tells the host this is the start of a message. You configure the header in the “Message Header in Hex Format” field on the SCI Configuration screen. The header is not typically used with asynchronous length protocol.
2b	Message Length	The number of bytes in the message.
2b	Message Length Negation	The negation of the number of bytes in the message.
2h	Message ID	An alphanumeric code that identifies the message. For more information, see the “Message ID” heading.
1d	Alarm	A field that identifies whether or not an SCI alarm exists. You can use the Alarm Information message (type 42) to find out more information about SCI alarms. This field can have the following values: 0=no alarms exist 1=alarms exist
v	Body	The variable format message body. The reply message bodies are described in the rest of the section.
2b	CRC	A hexadecimal CRC used to detect transmission errors. If you configure “Checksum⇒no” on the SCI Configuration screen, this field does not appear.
v	Trailer	A user-defined sequence of ASCII characters that tells the host this is the end of a message. You configure the trailer in the “Message Trailer in hex Format” field on the SCI Configuration screen. The trailer is not typically used with asynchronous length protocol.

Message ID

Each message contains a message identifier (ID). When the host sends a request message, it assigns a message identifier to the message. When the SCI sends back the reply message, it copies the same message identifier into the reply message. When the host sends the next request message, it assigns a different message ID to the message. Figure 1.3.2 shows how the message identifier is used.

NOTE: The SCI ignores the message identifier in X.25 protocol.

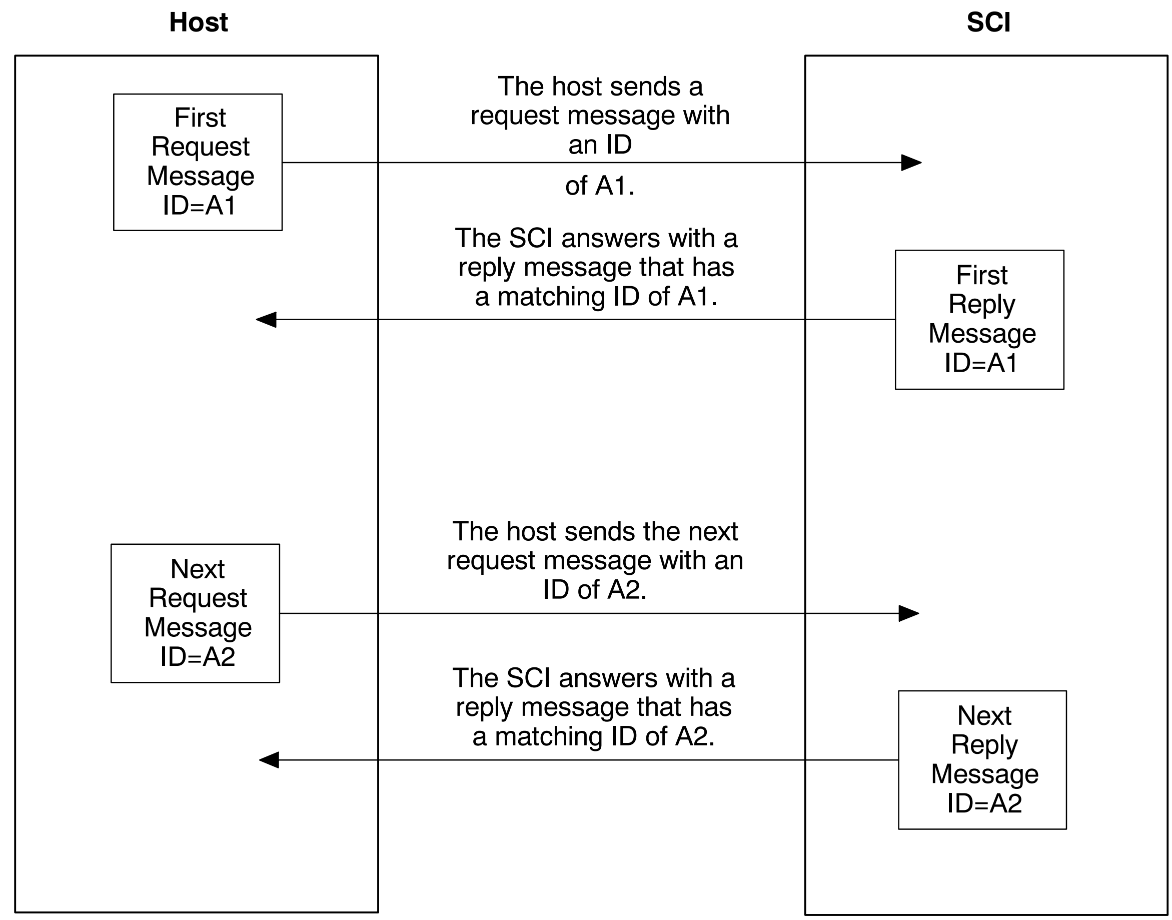


Figure 1.3.2. Message ID in Sample Request and Reply Messages

Duplicate Message IDs in the SCI

If the SCI receives two consecutive request messages with the same message ID, it assumes that the request messages are identical and sends the previous reply message instead of building a new reply message.

Figure 1.3.3 shows an example of a transmission failure resulting in duplicate message identifiers in the SCI.

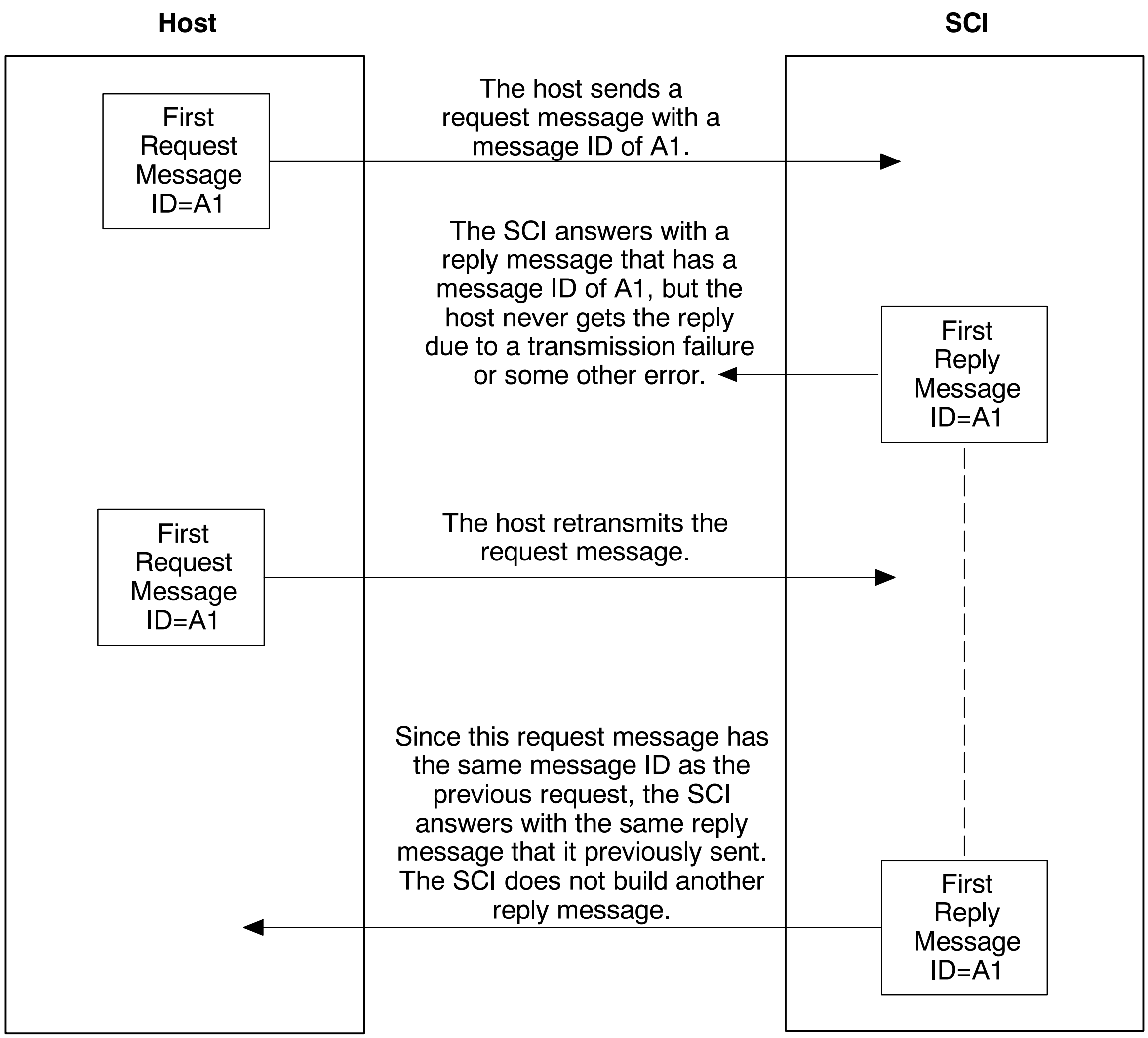


Figure 1.3.3. Example of Transmission Failure Resulting in Duplicate Message IDs in the SCI

Figure 1.3.4 shows an example of incorrect use of message IDs by the host. If the request messages are different, then the message identifiers in consecutive request messages must be different. Otherwise, the reply messages will be identical, as shown in Figure 1.3.4.

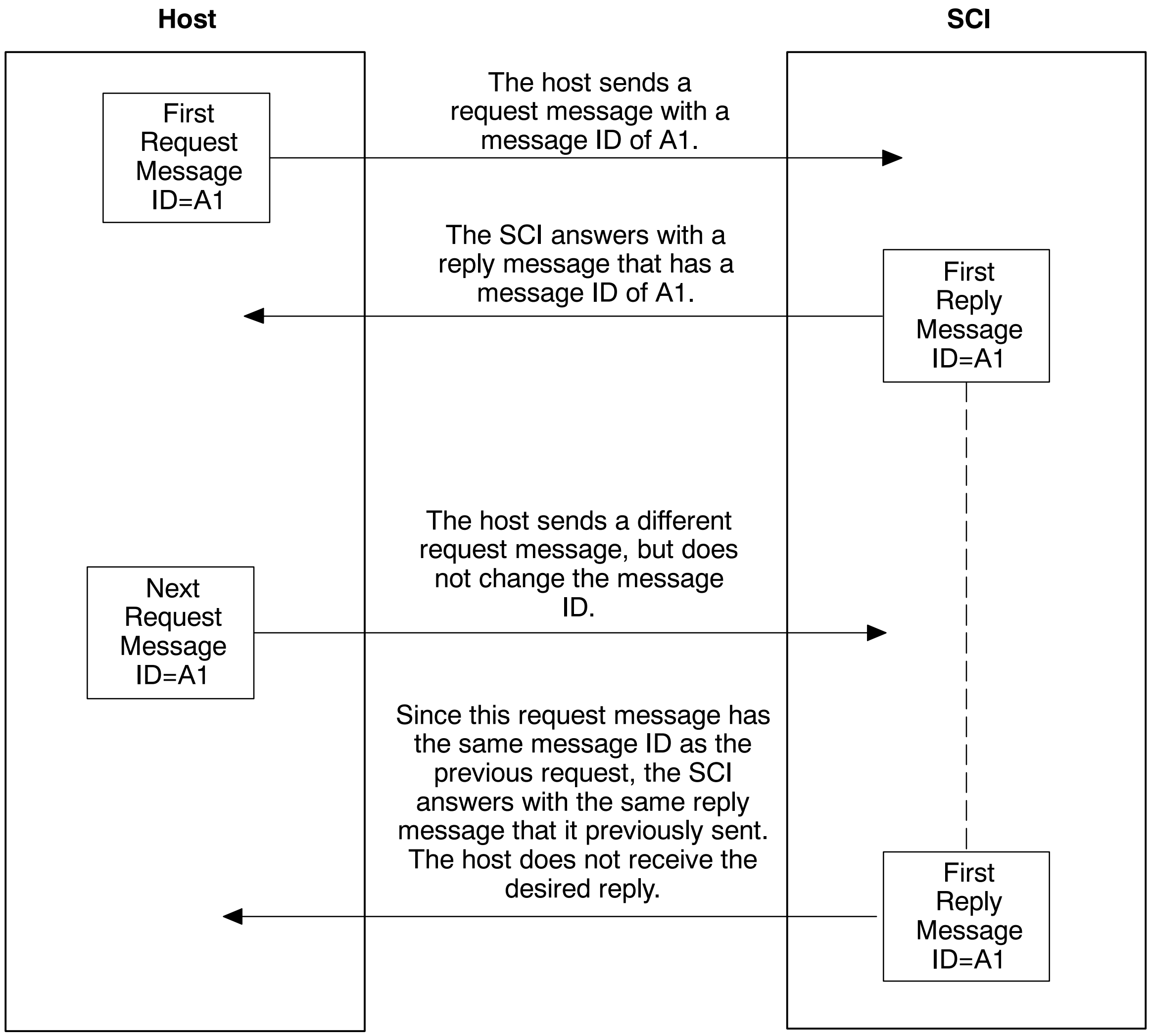


Figure 1.3.4. Example of Incorrect Use of Message IDs by the Host Resulting in Duplicate Message IDs in the SCI

Message Summary

Table 1.3.6. SCI Message Summary

Message Number	Description
	Password Messages
1	Host Connect Establishes communications when the SCI when passwords are configured.
2	Host Disconnect Ends communication when the SCI when passwords are configured.
	Raw Data Messages
11	Read Raw Controller Reads raw block data from a configuration on a Controller Processor.
12	Write Raw Controller Writes raw block data to a configuration on a Controller Processor.
13	Read Raw Disk Reads raw block data from a configuration on a disk volume.
14	Write Raw Disk Writes raw block data from a configuration on a disk volume.
	Field Code Messages
21	Read Controller Field Code Data Reads block field code values from a Controller Processor.
22	Write Controller Field Code Data Writes block field codes from a Controller Processor.
23	Read Disk Field Code Data Reads block field code values from a disk volume.

(continued on next page)

Table 1.3.6. SCI Message Summary (continued)

Message Number	Description
11	**Table Data Messages**
31	Host Time Synchronizes host time and SCI table time; also enables and disables polling.
32	Clear Table Location Clears SCI table locations.
33	Configure Table Entry Configures SCI table entries.
34	Read Table Data Reads SCI table data.
35	Write Table Data Writes data to Controller Processors through the SCI table.
36	Write Table Data Status Alternate method of getting the status of the Write Table Data Message (type 35).
	Node Alarm Messages
41	Node Alarm Mask Specifies the PeerWay nodes that have generated alarms stored in the SCI.
42	Alarm Information Retrieves a PeerWay node alarm message from the SCI alarm queue.
	Trend Data Message
61	Trend Data Retrieves data values from trend files.
	Report Transfer Message
62	Report Transfer Transfers a generated report.
	Raw File Transfer Messages
63	Raw File Transfer Read Reads raw data from a file.
64	Raw File Transfer Write Writes raw data to a file.

(continued on next page)

Table 1.3.6. SCI Message Summary (continued)

Message Number	Description
	RS3 Information Messages
71	Set RS3 Time Sets the RS3 time from the host.
72	Look Up Block Address Obtains a block address from a block tag.
73	ControlFile Status Retrieves information from a ControlFile Status screen.
	SCI Information Messages
81	SCI Status Retrieves SCI status information.
82	Read Message Pairs Reads the message pairs that are loaded into the SCI.
83	Message Format Specifies the format of message data for X.25 and asynchronous length protocols.
	Diagnostic Messages
91	Loopback Performs a loopback test on host/SCI communications.
92	Debug Copies a request or reply message from the SCI buffer.

Message Formats

Purpose

Password messages are used with an SCI password to control host access to the SCI.

When Required

If you configure a password in the “SCI Login Password” field on the SCI Configuration screen, then the host needs to use the Host Connect and Host Disconnect messages.

Establishing Communication

If you configure a password, then the host must begin communication between the host and SCI with the Host Connect message, which includes the password.

If communication is ended for any reason, then the host must send the Host Connect message again to reestablish communications.

Periodic Communication

If you configure a password, then the host must send the SCI at least one valid message per minute, or the SCI disconnects the line.

X.25 Protocol

If you configure a password when using X.25 protocol, each virtual session with the SCI must begin with the Host Connect message. Each session must communicate with the SCI once a minute to stay connected.

Messages

Message type 1—Host Connect message

- used to establish communication with the SCI.

Message type 2—Host Disconnect message

- used to stop communications with the SCI.

Host Connect Message Formats (Message Type 1)

Table 1.3.7 shows the Host Connect request message format.
Table 1.3.8 shows the Host Connect reply message format.

Table 1.3.7. Host Connect *Request* Message--Message Type 1

Format	Field	Definition
2d	Type	01—message type 1
16c	Password	The 16-character connect / password string.

Table 1.3.8. Host Connect *Reply* Message--Message Type 1

Format	Field	Definition
2d	Type	01—message type 1
3d	Error	Error code. For a list of error codes, see Section 6.

Host Disconnect Message Formats (Message Type 2)

Table 1.3.9 shows the host Disconnect request message format.
Table 1.3.10 shows the Host Disconnect reply message format.

Table 1.3.9. Host Disconnect *Request* Message--Message Type 2

Format	Field	Definition
2d	Type	02—message type 2

Table 1.3.10. Host Disconnect *Reply* Message--Message Type 2

Format	Field	Definition
2d	Type	02—message type 2
3d	Error	Error code. For a list of error codes, see Section 6.

Raw Data Messages

Purpose
Raw Data messages are used to back up block configurations from a Controller Processor or disk to the host computer and to write the configurations back to the Controller Processor or disk.

Data Format
Raw Data messages transfer block data in raw internal format as a stream of hexadecimal characters. The host cannot interpret or modify the raw data.

To prevent modification, the Controller Processor appends a checksum to the data. If the host writes the data back, then the Controller Processor verifies the checksum.

Hunks of Data
The raw data for one block may contain several thousand characters. As a result, the raw data for a block is divided into hunks. The host sends a request message and the SCI sends a reply message for each hunk of data.

Hunks must be transferred in numeric order.

Hunk Size
The host specifies the size of the hunks in the request message. The hunk size must be the same for all messages used for a single block. The maximum hunk size is 1024 characters.

Raw Data Transfer
Only one raw data transfer can be active at any one time.

SCI to Host Data Transfer
The host requests a hunk of data, receives the reply, requests the next hunk, receives that reply, and so on, until the entire block of data has been transferred to the host.

Host to SCI Data Transfer
The host writes a hunk of data, receives the reply, writes the next hunk of data, receives that reply, and so on, until the entire block of data has been transferred to the SCI. Then, the SCI writes the data to the Controller Processor or disk.

Accessing Disk Data
Accessing disk data can be a slow process. The SCI may take a long time to reply when accessing the disk.

Messages

Message type 11—Read Raw Controller message

- used to read raw block data to the configuration on a Controller Processor.

Message type 12—Write Raw Controller message

- used to write raw block data to the configuration on a Controller Processor.

Message type 13—Read Raw Disk message

- used to read raw block data from a configuration on a disk volume.

Message type 14—Write Raw Disk message

- used to write raw block data to a configuration on a disk volume.

Read Raw Controller Message Formats (Message Type 11)

Table 1.3.11 shows the Read Raw Controller request message format. Table 1.3.12 shows the Read Raw Controller reply message format.

Table 1.3.11. Read Raw Controller *Request* Message --Message Type 11

Format	Field	Definition
2d	Type	11—message type 11.
10c	Block Address	RS3 block address.
4d	Hunk Size	Number of hexadecimal digits per hunk. (Must be a multiple of 4.)
2d	Hunk Number	Requested hunk number. The first hunk number is 1.

Table 1.3.12. Read Raw Controller *Reply* Message --Message Type 11

Format	Field	Definition
2d	Type	11—message type 11.
3d	Error code	Error code. For more information, see Section 6.
2d	Hunk Number	Current hunk number.
4d	Total Size	Total number of hexadecimal digits to be transferred for this block address.
vh	Data	The hexadecimal block data. The number of digits transferred is the smaller of the Hunk Size field in the request message and the number of digits remaining for the block.

Write Raw Controller Message Formats (Message Type 12)

Table 1.3.13 shows the Write Controller request message format.
Table 1.3.14 shows the Write Controller reply message format.

Table 1.3.13. Write Raw Controller *Request* Message--Message Type 12

Format	Field	Definition
2d	Type	12—message type 12.
10c	Block Address	RS3 block address.
4d	Total Size	Total number of hexadecimal digits to be transferred for this block address.
4d	Hunk Size	Number of hexadecimal digits per hunk. (Must be a multiple of 4.)
2d	Hunk Number	Current hunk number. The first packet number is 1.
vh	Data	The hexadecimal block data. The number of digits transferred is the smaller of the Hunk Size field and the number of digits remaining for this block.

Table 1.3.14. Write Raw Controller *Reply* Message--Message Type 12

Format	Field	Definition
2d	Type	12—message type 12.
3d	Error	Error code. For a list of error codes, see Section 6.
2d	Hunk Number	Current hunk number.

Read Raw Disk Message Formats (Message Type 13)

Table 1.3.15 and Table 1.3.16 show the Read Raw Disk request message formats. Table 1.3.15 is the recommended format. Table 1.3.16 is an alternate format, but it can only be used on PeerWay 1.

Table 1.3.17 shows the Read Raw Disk reply message format.

Table 1.3.15. Read Raw Disk *Request* Message--All PeerWays--Message Type 13

Format	Field	Definition
2d	Type	13—message type 13.
3d	Error	Error code. For a list of error codes, see Section 6.
2d	Hunk Number	Current hunk number.
4d	Total Size	Total number of hexadecimal digits to be transferred for this block address.
vh	Data	The hexadecimal block data. The number of digits transferred is the smaller of the Hunk Size field in the request message and the number of digits remaining for this block.

Table 1.3.16. Read Raw Disk *Request* Message--On PeerWay 1 Only--Message Type 13

Format	Field	Definition
2d	Type	13—message type 13
2d	PeerWay	PeerWay number where the disk is mounted.
10c	Volume	Disk volume name.
2d	Node	Node number where the volume is inserted. If zero, the SCI searches the PeerWay for the volume.
1d	Drive	Drive number (if node is nonzero).
9c	File name	Disk file name.
10c	Block address	RS3 block address.
4d	Hunk size	Number of hexadecimal digits per hunk. (Must be a multiple of 4.)
2d	Hunk number	Requested hunk number. The first hunk number is 1.

Table 1.3.17. Read Raw Disk *Reply* Message--Message Type 13

Format	Field	Definition
2d	Type	13—message type 13.
3d	Error	Error code. For a list of error codes, see Section 6.
2d	Hunk Number	Current hunk number.
4d	Total Size	Total number of hexadecimal digits to be transferred for this block address.
vh	Data	The hexadecimal block data. The number of digits transferred is the smaller of the Hunk Size field in the request message and the number of digits remaining for this block.

Write Raw Disk Message Formats (Message Type 14)

Table 1.3.18 and Table 1.3.19 show the Write Raw Disk request message formats. Table 1.3.18 is the recommended format. Table 1.3.19 is an alternative format, but it can be used only on PeerWay 1.

Table 1.3.20 shows the Write Raw Disk reply message format.

Table 1.3.18. Write Raw Disk *Request* Message--All PeerWays--Message Type 14

Format	Field	Definition
2d	Type	14—message type 14.
10c	Volume	Disk volume name.
3d	Node	Node number where the volume is inserted. If zero, the SCI searches the PeerWay for the volume.
1d	Drive	Drive number if node is nonzero.
9c	File name	Disk file name.
10c	Block Address	RS3 block address.
4d	Total Size	Total number of hexadecimal digits to be transferred for this block address.
4d	Hunk Size	Number of hexadecimal digits per hunk. (Must be a multiple of 4.)
2d	Hunk Number	Requested hunk number. The first hunk number is 1.
vh	Data	The hexadecimal block data. The number of digits transferred is the smaller of the Hunk Size field and the number of digits remaining for this block.

Table 1.3.19. Write Raw Disk *Request* Message--PeerWay 1 Only--Message Type 14

Format	Field	Definition
2d	Type	14—message type 14.
2d	PeerWay	PeerWay number where the disk is mounted.
10c	Volume	Disk volume name.
2d	Node	Node number where the volume is inserted. If zero, the SCI searches the PeerWay for the volume.
1d	Drive	Drive number if node is nonzero.
9c	File name	Disk file name.
10c	Block Address	RS3 block address.
4d	Total Size	Total number of hexadecimal digits to be transferred for this block address.
4d	Hunk Size	Number of hexadecimal digits per hunk. (Must be a multiple of 4.)
2d	Hunk Number	Requested hunk number. The first hunk number is 1.
v	Data	The hexadecimal block data. The number of digits transferred is the smaller of the Hunk Size field and the number of digits remaining for this block.

Table 1.3.20. Write Raw Disk *Reply* Message--Message Type 14

Format	Field	Definition
2d	Type	14—message type 14.
3d	Error	Error code. For a list of error codes, see Section 6.
2d	Hunk Number	Current hunk number.

Field Code Messages

Purpose

Field code messages are used to read from and write to specific block data fields.

Field Codes

Block fields are identified by field codes. For descriptions of field codes, see Section 4.

Scaling

All request messages have a Scaling field that determines how the field code data is formatted.

Analog values can have:

- internal scaling= values are scaled from 0 to 1.

  or

- display scaling= values are scaled according to ControlBlock “Eng Zero” and “Eng Max” fields.

Discrete values can be:

- 0 or 1

  or

- one part of a message pair (e.g., ON or OFF)

Messages

Message type 21—Read Controller Field Code Data message

- used to read field code values from a Controller Processor.

Message type 22—Write Controller Field Code Data message

- used to write field code values to a Controller Processor.

Message type 23—Read Disk Field Code Data message

- used to read field code values from a disk volume.

Read Controller Field Code Data Message Formats (Message Type 21)

Table 1.3.21 shows the Read Controller Field Code Data request message format. Table 1.3.22 shows the Read Controller Field Code Data reply message format.

Table 1.3.21. Read Controller Field Code Data *Request* Message--Message Type 21

Binary Format	ASCII Format	Field	Definition
2d	2d	Type	21—message type 21.
10c	10c	Block Address	RS3 block address.
1b	1d	Scaling	Determines the scaling in the reply message for scalable analog values and discrete values. Field Entry / Analog Values / Discrete Values 0 internal scaling 0 or 1 1 display scaling 0 or 1 2 internal scaling message pairs 3 display scaling message pairs
v	v	Entry List	List of request entries. The format of each entry is shown below.
			Each entry in the list has the following format:
2b	4d	Code	Field code. For descriptions of field codes, see Section 4.
0	1c	Comma	A comma follows each field code, including the last one (not used in binary format).

Table 1.3.22. Read Controller Field Code Data *Reply* Message--Message Type 21

Binary Format	ASCII Format	Field	Definition
2d	2d	Type	This is message type 21.
2b	3d	Error	Error code. For a list of error codes, see Section 6.
v	v	Entry List	List of return entries. The format of each entry is shown below.
			Each entry in the list has the following format:
v	v	Data Item	Data item that corresponds to the field code in the request message.
0	1c	Comma	A comma follows each field code, including the last one (not used in binary format).

Write Controller Field Code Data Message Formats (Message Type 22)

Table 1.3.23 shows the Write Controller Field request message format. Table 1.3.24 shows the Write Controller Field reply message format.

Table 1.3.23. Write Controller Field Code Data *Request* Message--Message Type 22

Binary Format	ASCII Format	Field	Definition
2d	2d	Type	22—message type 22.
10c	10c	Block Address	RS3 block address.
1b	1d	Scaling	Determines the scaling in the reply message for scalable analog values and discrete values. Field Entry / Analog Values / Discrete Values 0 internal scaling 0 or 1 1 display scaling 0 or 1 2 internal scaling message pairs 3 display scaling message pairs
1b	1d	Log	Specifies whether or not to log this change to the Operator Log. 0=do not log this change 1=log this change If you configure “Log All Changes to Change Log⇒yes” on the SCI Configuration screen, this change is logged regardless of this field. For more information about logging changes, see Section 2 of this chapter.
v	v	Entry list	List of items to write to the Controller Processor. The format of each entry is listed below.
			Each entry in the list has the following format:
2b	4d	Field Code	Field code. For descriptions of field codes, see Section 4.
0	1c	=	An ASCII equals sign (“=”) (not used in binary format).
v	v	Data Item	Data item that corresponds to the field code.
0	1c	Comma	A comma follows all entries, including the last one (not used in binary format).

Table 1.3.24. Write Controller Field Code Data *Reply* Message--Message Type 22

Binary Format	ASCII Format	Field	Definition
2d	2d	Type	22—message type 22.
2b	3d	Error	Error code. For a list of error codes, see Section 6.

Read Disk Field Code Data Message Formats (Message Type 23)

Table 1.3.25 and Table 1.3.26 show the Read Disk Field Code Data request message formats. Table 1.3.25 is the recommended format. Table 1.3.26 is an alternative format, but can only be used on PeerWay 1.

Table 1.3.27 shows the Read Disk Field Code Data reply message format.

NOTE: Only static values are stored on the disk. Dynamic values are always returned as zero.

Table 1.3.25. Read Disk Field Code Data *Request* Message--All PeerWays--Message Type 23

Binary Format	ASCII Format	Field	Definition
2d	2d	Type	23—message type 23.
10c	10c	Volume	Disk volume name.
2b	3d	Node	Node number where the volume is inserted. If zero, the SCI searches the PeerWay for the volume.
1b	1d	Drive	Drive number if node is nonzero.
9c	9c	File name	Disk file name.
10c	10c	Block Address	RS3 block address.
1b	1d	Scaling	Determines the scaling in the reply message for scalable analog values and discrete values. Field Entry / Analog Values / Discrete Values 0 / internal scaling / 0 or 1 1 / display scaling / 0 or 1 2 / internal scaling / message pairs 3 / display scaling / message pairs
v	v	Entry List	List of request entries. The format of each entry is shown below.
			Each entry in the list has the following format:
2b	4d	Code	Field code. For descriptions of field codes, see Section 4.
0	1c	Comma	A comma follows each field code, including the last one (not used in binary format).

Table 1.3.26. Read Disk Field Code Data *Request* Message--PeerWay 1 Only--Message Type 23

Binary Format	ASCII Format	Field	Definition
2d	2d	Type	23—message type 23.
1b	2d	PeerWay	PeerWay number where the disk is mounted.
10c	10c	Volume	Disk volume name.
1b	2d	Node	Node number where the volume is inserted. If zero, the SCI searches the PeerWay for the volume.
1d	1d	Drive	Drive number if node is nonzero.
9c	9c	File name	Disk file name.
10c	10c	Block Address	RS3 block address.
1b	1d	Scaling	Determines the scaling in the reply message for scalable analog values and discrete values. Field Entry / Analog Values / Discrete Values 0 internal scaling 0 or 1 1 display scaling 0 or 1 2 internal scaling message pairs 3 display scaling message pairs
v	v	Entry List	List of request entries. The format of each entry is shown below.
			Each entry in the list has the following format:
2b	4d	Code	Field code. For descriptions of field codes, see Section 4.
0	1c	Comma	A comma follows each field code, including the last one (not used in binary format).

Table 1.3.27. Read Disk Field Code Data *Reply* Message--Message Type 23

Binary Format	ASCII Format	Field	Definition
2d	2d	Type	23—message type 23.
2b	3d	Error	Error code. For a list of error codes, see Section 6.
v	v	Entry List	List of return entries. The format of each entry is shown below.
			Each entry in the list has the following format:
v	v	Data Item	Data item that corresponds to the field code in the request message.
0	1c	Comma	A comma follows each field code, including the last one (not used in binary format).

Table Data Messages

Purpose

Table data messages are used to control execution of the SCI table. Table data messages are useful for acquiring or writing large amounts of block variable data.

Terminology

Host time	The time reference used by the host for table data messages. The host time is not associated with RS3 time.
Polling	The act of the SCI acquiring data from the RS3 control system and putting it into the SCI table.
SCI table	A collection of block variable data from the RS3 control system. The SCI table contains the last four data values collected for each table entry.
SCI table time	The time reference used by the SCI table for table data messages. The SCI table time is not associated with RS3 time.
Table entry	The configuration information for one source of data. You configure table entries for the SCI table. Each entry contains a block variable address, scaling option, and update interval. The SCI table contains the last four data values for each table entry.
Update interval	Specifies how often new block variable data is added to the SCI table for a table entry. Each table entry has its own update interval.

SCI Table Size

If the SCI has the 1/2 Meg Processor Card (1984-2120-000x), then the SCI table has 1000 entries (0-999).

If the SCI has the 1 Meg Processor Card (1984-2137-000x), then the SCI table has 1500 entries (0-1499).

Table Entry Fields

Messages 32 through 36 contain fields that indicate entry numbers. These fields can accommodate 3-digit or 4-digit entries. However, all entries in a single request must be the same size.

Messages

Message Type 31—Host Time message

- used to synchronize the host time and the SCI table time; also used to enable and disable polling.

Message Type 32—Clear Table Location message

- used to clear a range of SCI table locations.

Message Type 33—Configure Table Entry message

- used to configure a table entry.

Message Type 34—Read Table Data message

- used to read SCI table data.

Message Type 35—Write Table Data message

- used to write data to the RS3 control system.

Message Type 36—Write Table Data Status message

- used as an alternate way to get a status about a Write Table Data request message (type 35).

Typical Operating Sequence

Figure 1.3.5 and Figure 1.3.6 show a typical operating sequence for initially using the SCI table to acquire block variable data.

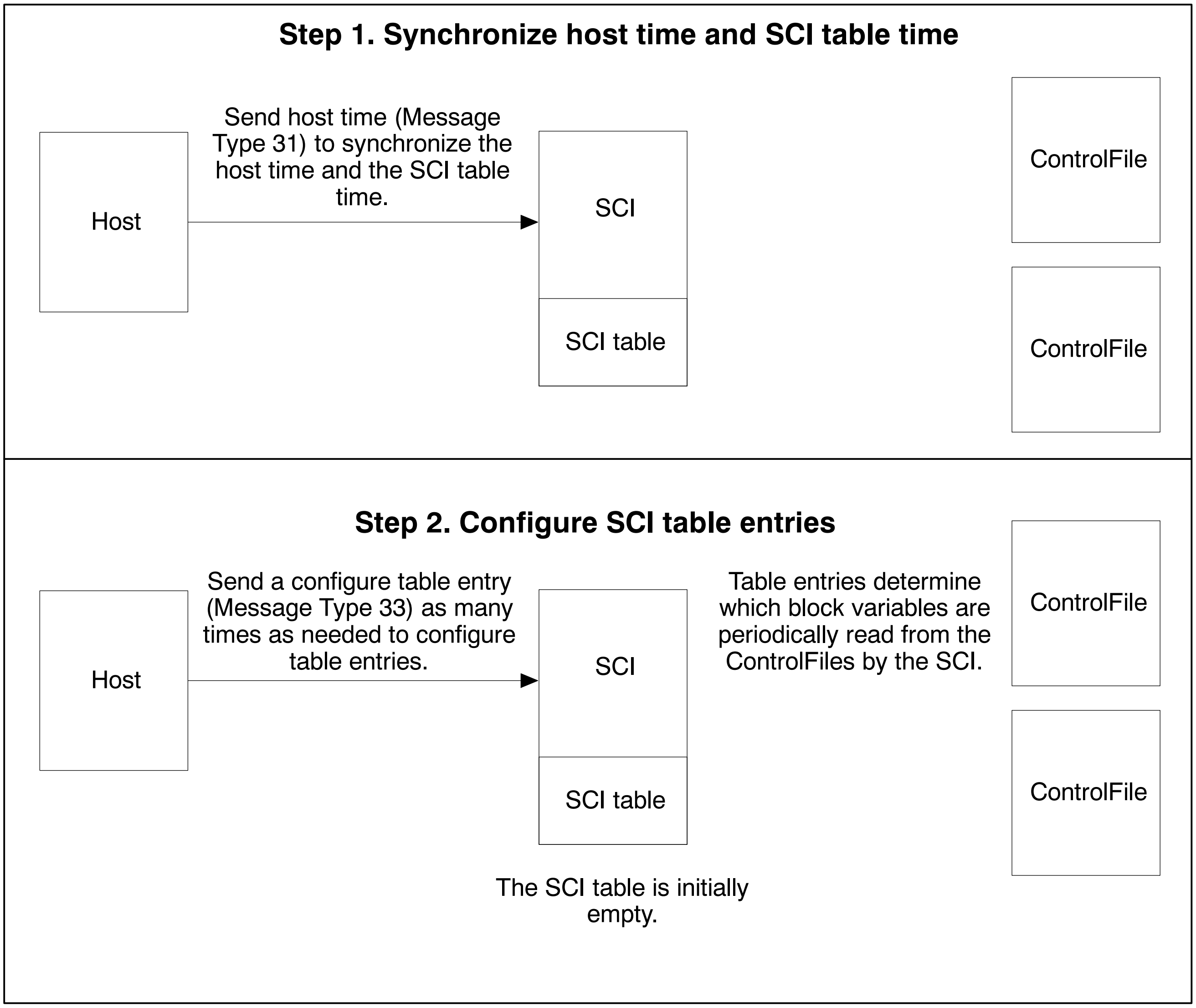


Figure 1.3.5. Typical Sequence for Using the SCI Table to Acquire Block Variable Data

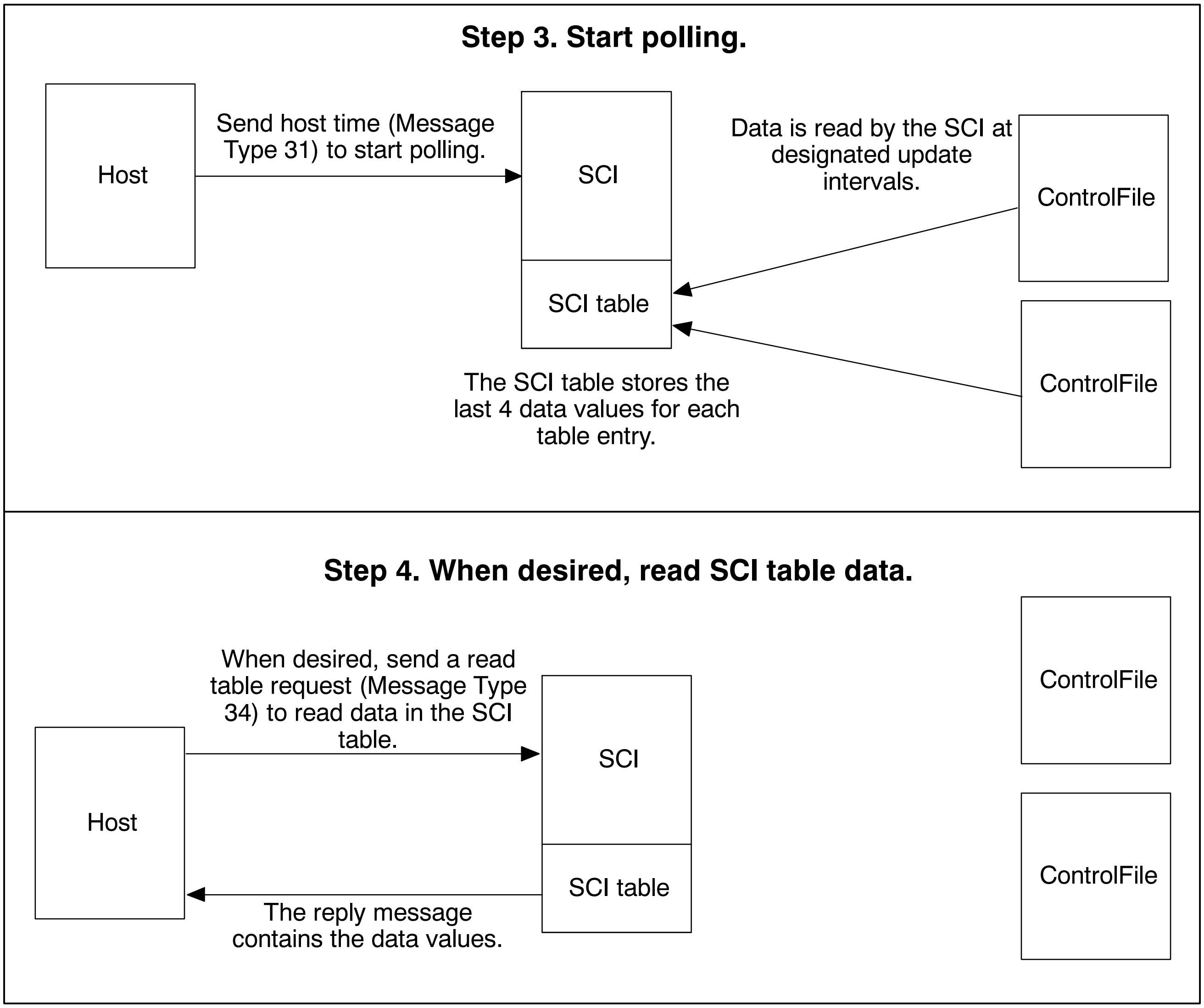


Figure 1.3.6. Typical Sequence for Using the SCI Table to Acquire Block Variable Data

Timing Issues

Several timing issues must be considered when using the SCI table.

Host/SCI Timing

The Host Time message (Message Type 31) is used to synchronize the host time and the SCI table time so that read requests from the host for timed information are correctly executed.

Host time and SCI table time are not associated with RS3 time.

NOTE: Do not attempt to use the Set Rosemount System Time (Message Type 71 or SCI Status Message (Message Type 81) to set host time.

Sending Host Time Messages

After a while, the host time and SCI table time may drift apart. To prevent this drift from becoming large enough to cause timing problems, you should periodically send Host Time messages (Message Type 31) to the SCI. Host Time messages do not have to be sent very often to keep the host/SCI timing adequate.

You can try this method for sending timing messages: Send the Host Time message (Message Type 31) once every 12 to 24 hours. If you get timing problems, send Host Time messages twice as often. Keep shortening the time between Host Time messages until the timing problems go away.

NOTE: Host Time messages 50 and 60 are indications that you should reset the Host Time Message.

Timing of Read Table Data messages

You must be careful when sending Read Table Data request messages. At any given update, the SCI needs a time interval to obtain all of the data values.

In Figure 1.3.7, this period of time is shown by the dotted lines. For example, to get data values from time 00:00, the SCI starts gathering data values a short time before 00:00 and continues a short time after 00:00.

To be sure that the SCI has received the data values that you want, you should wait a short while after the desired sample time before sending the Read Request message. The longest you should have to wait is 5 seconds, and normally you need to wait much less than that.

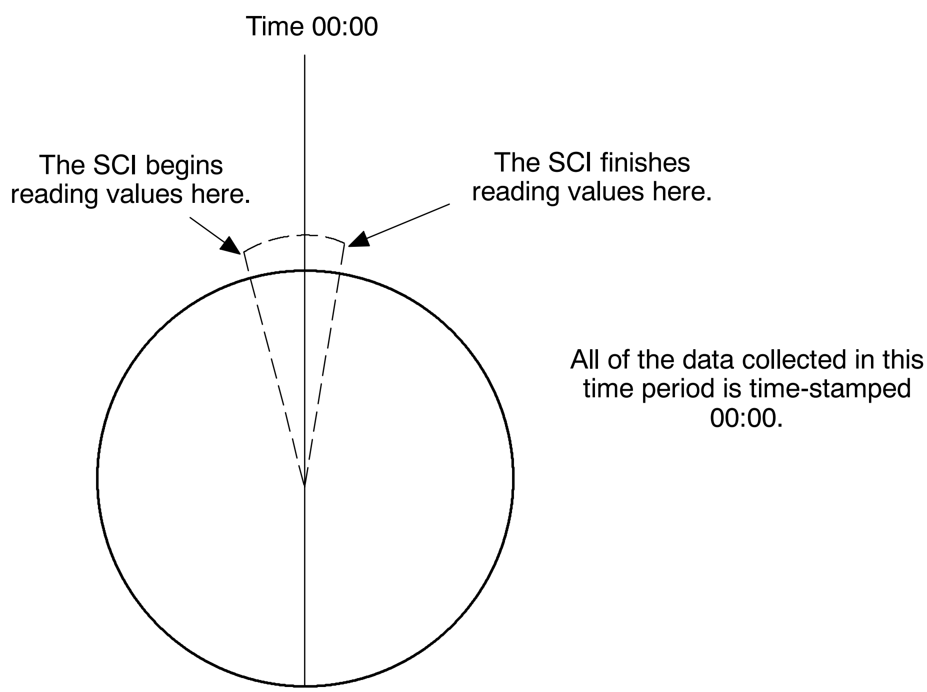


Figure 1.3.7. Timing of SCI Table Data

Writing to Blocks Using the SCI Table

You can write to ControlBlock variables using the SCI table with message type 35. If you have configured a table entry for a ControlBlock variable, you can write to that variable. For more information about requirements for writing to ControlBlocks, see Section 2.

There are two methods that can be used to get status information about Write Table Data request messages.

Typical method for getting status information

The typical method of getting status information consists of sending a Write Table Data request message, waiting for a complete reply, sending the next Write Table Data request message, waiting for a complete reply, and so on, until all of the writes are completed. The upper portion of Figure 1.3.8 shows the typical method of getting status information.

Alternate method for getting status information

The alternate method of getting status information can be used when the host needs to perform a large number of writes in a short amount of time. In this method, Write Table Data reply messages are sent without waiting for information from the ControlFiles.

After all of the writes are complete, multiple Write Table Data Status request messages (type 36) are sent to the SCI. These messages request that the SCI put the status information associated with the Write Table Data request messages (type 35) into the Write Table Data Status reply messages (type 36). The lower portion of Figure 1.3.8 shows the alternate method of getting status information.

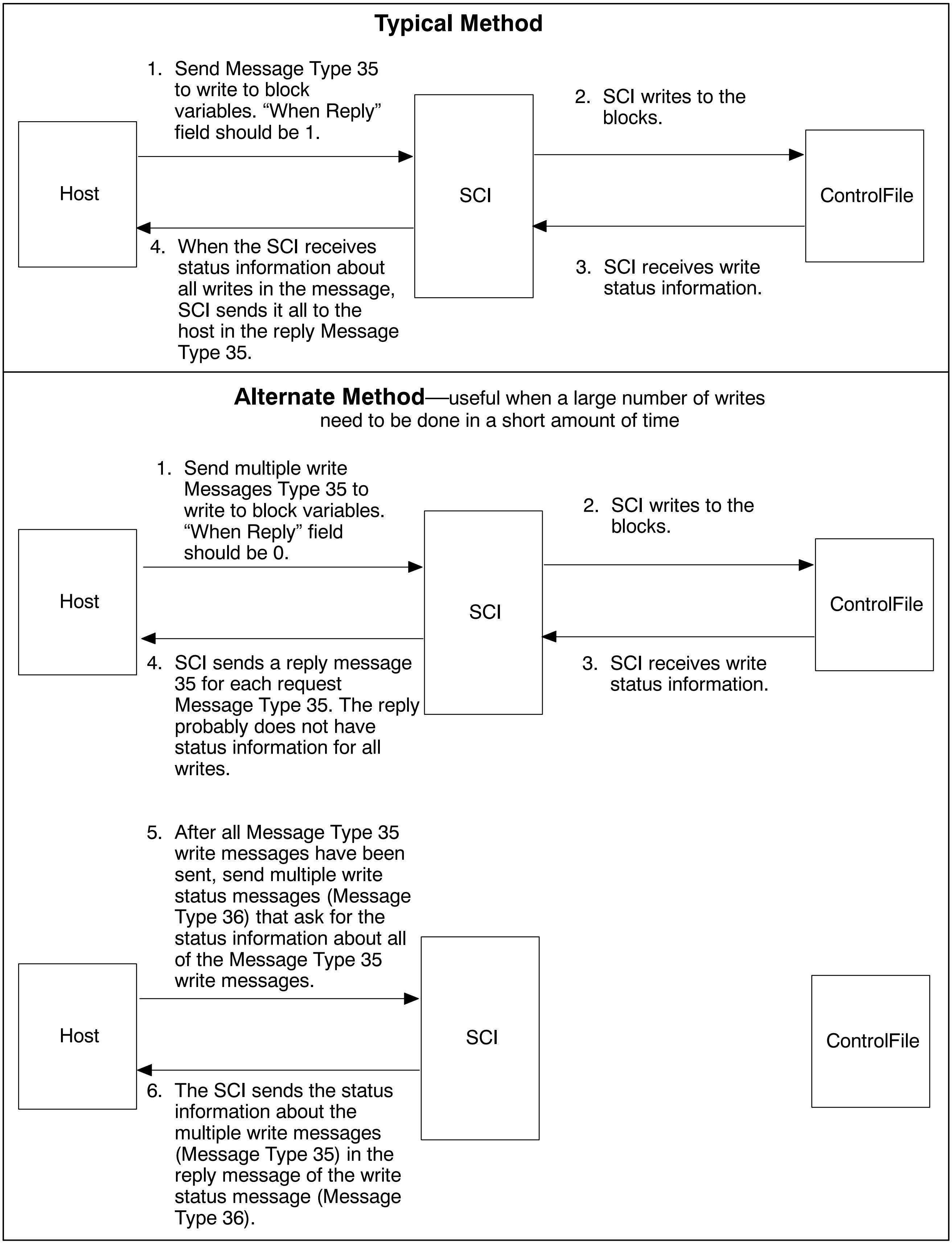


Figure 1.3.8. Methods of Getting Write Table Data Request Message Status Information

Host Time Message Formats (Message Type 31)

Table 1.3.28 shows the Host Time request message format.
Table 1.3.29 shows the Host Time Reply message format.

Table 1.3.28. Host Time *Request* Message--Message Type 31

Format	Field	Definition
2d	Type	31—Message Type 31.
1d	Polling	0—disable polling. 1—enable polling.
6d	Time	Current time of day in HHMMSS format.

Table 1.3.29. Host Time *Reply* Message--Message Type 31

Format	Field	Definition
2d	Type	31—Message Type 31.
3d	Error	Error code. For a list of error codes, see Section 6.

Clear Table Location Message Formats (Message Type 32)

Table 1.3.30 shows the Clear Table Location request message format. Table 1.3.31 shows the Clear Table Location reply message format.

NOTE: If you clear a table entry that has a configured block variable, then all table entries with the same block variable are also cleared.

Table 1.3.30. Clear Table Location *Request* Message--Message Type 32

Format	Field	Definition
2d	Type	32—message type 32.
3d or 4d	Low Location	Low SCI table location number to be cleared.
3d or 4d	High Location	High SCI table location number to be cleared.

Table 1.3.31. Clear Table Location *Reply* Message--Message Type 32

Format	Field	Definition
2d	Type	32—message type 32.
3d	Error	Error code. For a list of error codes, see Section 6.

Configure Table Entry Message Formats (Message Type 33)

Table 1.3.32 shows the Configure Table Entry request message. Table 1.3.33 shows the Configure Table Entry reply message.

NOTE: If you want to configure a table entry in an SCI table location that already contains a table entry, then you must first clear the SCI table location with a Clear Table Location message (type 32).

Table 1.3.32. Configure Table Entry *Request* Message--Message Type 33

Format	Field	Definition
2 d	Type	33—message type 33.
2 d	# of Entries	Number of entries to be configured in this message.
v	Entry List	List of entries. The format of each entry is shown below.
		Each entry in the list has the following format:
3 d or 4 d	Entry Number	SCI table entry location number.
16 c	Block Variable	RS3 block variable.
1 d	Scaling	0= analog values have internal scaling (0 to 1) discrete values are 0 or 1 1= analog values have display scaling (“Eng Zero” and “Eng Max” fields) discrete values are one half of a message pair
1 d	Update Interval	How often you want to update table data from the RS3. 0=15 seconds 1=30 seconds 2=60 seconds 3=120 seconds 4=300 seconds 5=5 seconds 6=10 seconds 7=continuous
1 c	Comma	A comma is placed after each entry in the list, including the last entry.

Table 1.3.33. Configure Table Entry *Reply* Message--Message Type 33

Format	Field	Definition
2 d	Type	33—message type 33.
3 d	Error	Error code. For a list of error codes, see Section 6.
2 d	# of Replies	Number of replies returned in this reply.
v	Entry Reply List	List of entry replies. There is an entry reply for each entry written in the request message. The format of each entry reply is shown below.
		Each entry in the list has the following format:
3 d	Error	Error code. For a list of error codes, see Section 6.
1 c	Comma	A comma is placed after each reply in the list, including the last reply.

Read Table Entry Message Formats (Message Type 34)

Table 1.3.34 shows the Read Table Data request message format. Table 1.3.35 shows the Read Table Data reply message format.

NOTE:

- The Host Time message request and reply are in minutes and seconds from the start of the host hour.

  For example, if the hour is set to midnight (000000 using Message Type 31), then the start of the host hour is every successive hour; that is 010000, 020000, 030000, and so on. Message 34 measures time in minutes and seconds from the start of the host hour.
- The Host Time from the start of the host hour must be evenly divisible by the “Update Interval” in the Configure Table Entry Message (Message Type 33).

Table 1.3.34. Read Table Data *Request* Message--Message Type 34

Binary Format	ASCII Format	Field	Definition
2 d	2 d	Type	34—message type 34.
2 b*	4 d	Time	Desired time of each read in MMSS format. An MM field entry of 60 represents the most recent valid sample available.
1 b	2 d	# of Ranges	Number of table entry ranges to read.
v	v	Entry List	List of table entry ranges to read. The format of each entry range is shown below.
			Each entry in the list has the following format:
2 b	3 d or 4 d	Low Entry	Low table entry number.
2 b	3 d or 4 d	High Entry	High table entry number.
0	1 c	Comma	A comma follows all entry range items, including the last item (not used in binary format).

* If the “Time” request message is in the binary form, the first byte is minutes and the second byte is seconds.

Table 1.3.35. Read Table Data *Reply* Message--Message Type 34

Binary Format	ASCII Format	Field	Definition
2 d	2 d	Type	34—message type 34.
2 b	3 d	Error	Error code. For a list of error codes, see Section 6.
1 b	2 d	# of Entries	Number of entries returned in this message.
v	v	Entry List	List of entry values. The format of each entry is shown below.
			Each entry in the list has the following format:
4 b	8 p	Value	Data value.
1 b	1 d	Block Mode	Current block mode. To determine the block mode for the number returned, see field code 0005 in Section 4 of this manual. Match the number returned against the mode numbers for field code 0005 for the block type you are reading. A number of 9 means that the the block mode is not available. If the block mode is not available, an alarm code is generated.
1 b	2 d	Alarm Code	Alarm value associated with this point. If multiple codes can exist for an entry, the highest code number is returned. 00= no special conditions 01= at high limit 02= at low limit 03= at rate limit 04= hold forward active 05= value affected by tracking 06= value affected by logic 10= miscellaneous system alarm 20= low advisory alarm 21= high advisory alarm 22= miscellaneous advisory alarm 23= low critical alarm 24= high critical alarm 25= miscellaneous critical alarm 30= low hardware alarm 31= high hardware alarm 32= miscellaneous hardware alarm 33= miscellaneous PLC alarm 34= miscellaneous SIB alarm 40= input invalid or no longer configured 41= caution: bad scaling 42= Block type of origin block unknown 50= message came too early—SCI has not yet read data for most recent valid sample 51= error in data collection 52= entry update of value is late (sample lost) or the controller is not responding to the request. 60= link value unavailable (time stamp not found) or SCI no longer has data associated with requested time 61= periodic point subsystem disabled—turn SCI polling on 62= item not configured in SCI table
0	1 c	Comma	A comma appears after each item, including the last item (not used in binary format).

Write Table Data Message Formats (Message Type 35)

Table 1.3.36 shows the Write Table Data request message format. Table 1.3.37 shows the Write Table Data reply message format.

Table 1.3.36. Write Table Data *Request* Message--Message Type 35

Binary Format	ASCII Format	Field	Definition
1 b	2 d	Type	25—Message Type 35.
1 b	1 d	Scaling	0= analog values have internal scaling (0 to 1) discrete values are 0 or 1 1= analog values have display scaling ("Eng Zero" and "Eng Max") discrete values are one half of a message pair
1 b	1 d	Log	0= Do not log this change to the Operator Change Log 1= Log this change to the Operator Change Log If you configure "Log All Changes to Change Log⇒yes" on the SCI Configuration screen, this change is logged regardless of this field. For more information about logging changes, see Section 2 of this chapter.
1 b	1 d	When reply	0= Reply as soon as all of the write requests have been issued 1= Wait for completion of all the writes specified in this message before replying
1 b	2 d	# of Entries	Number of entries to be written in this message (maximum of 32).
v	v	Entry List	List of entries to modify and the new values. The format of each entry is shown below.
			Each entry in the list has the following format:
2 b	3 d or 4 d	Entry	SCI table entry number to modify.
0	1 c	=	An ASCII equals sign ("=") (not used in binary format).
4 b	8 f	Value	The value to write to the block variable.
0	1 c	Comma	A comma follows all entries, including the last entry (not used in binary format).

Table 1.3.37. Write Table Data *Reply* Message--Message Type 35

Binary Format	ASCII Format	Field	Definition
1 b	2 d	Type	35—message type 35.
2 b	3 d	Error	Error code. For a list of error codes, see Section 6.
1 b	2 d	# of Returns	Number of return entries in this message.
v	v	Entry List	List of return entries. The format of each entry is shown below.
			Each entry in the list has the following format:
2 b	3 d	Error	Error code. For a list of error codes, see the *Alarm Messages Manual*. If the When Reply field in the request message is 0, then an error code of 51 indicates that the write is still active.
0	1 c	Comma	A comma appears after each item, including the last item (not used in binary format).

Write Table Data Status Message Formats (Message Type 36)

Table 1.3.38 shows the Write Table Data Status request message format. Table 1.3.39 shows the Write Table Data Status reply message format.

Table 1.3.38. Write Table Data Status *Request* Message--Message Type 36

Binary Format	ASCII Format	Field	Definition
1 b	2 d	Type	36—message type 36.
1 b	2 d	# of Entries	Number of entries for which to return status values (maximum of 32).
v	v	Entry List	List of entries. The format of each entry is shown below.
			Each entry in the list has the following format:
2 b	3 d or 4 d	Entry Number	Entry number for which to return a status values.
0	1 c	Comma	A comma follows all entries, including the last entry (not used in binary format).

Table 1.3.39. Write Table Data Status *Reply* Message--Message Type 36

Binary Format	ASCII Format	Field	Definition
1 b	2 d	Type	36—message type 36.
2 b	3 d	Error	Error code. For a list of error codes, see Section 6.
1 b	2 d	# of Returns	Number of return entries returned in this message.
v	v	Entry List	List of entries. The format of each entry is shown below.
			Each entry in the list has the following format:
2 b	3 d	Error	Error code. For a list of error codes, see Section 6.
0	1 c	Comma	A comma appears after each item, including the last entry (not used in binary format).

Node Alarm Messages

Purpose

Node alarm messages allow the host to receive alarm information from RS3 PeerWay nodes.

Messages

Message 41—Node Alarm Mask message

- used to specify which PeerWay nodes have alarms stored in the SCI alarm queue.

Message 42—Alarm Information message

- used to retrieve an alarm message from the SCI alarm queue.

SCI Alarm Queue

The SCI alarm queue stores up to 200 alarm messages from RS3 nodes.

Alarm Queue Overflow

If the SCI alarm queue has filled up, the SCI puts alarm #320 “Queue Full — Alarms Lost” into the SCI alarm queue. This alarm message indicates that the SCI alarm queue has filled up and that some alarms will be lost. The alarm has a transaction field entry of 1 (alarm occurrence).

When the alarm overflow condition has cleared, the SCI puts another alarm #320 into the SCI alarm queue. This time the alarm has a transaction field entry of 3 (alarm clear).

All alarms that were time-stamped between these two messages (the alarm occurrence and the alarm clear) did not go into the SCI alarm queue and are lost.

Node Alarm Mask Message Formats (Message Type 41)

The Node Alarm Mask message specifies which PeerWay nodes have alarms stored in the SCI. Alarms generated by the specified nodes are stored in the SCI alarm queue.

Table 1.3.40 shows the Node Alarm Mask request message formats. Table 1.3.41 shows the Node Alarm Mask reply message formats.

Default

The default configuration specifies that no PeerWay node alarms are put into the SCI alarm queue.

Multiple PeerWays

If you want to receive alarms from more than one PeerWay, you must send a Node Alarm Mask message for each PeerWay.

Table 1.3.40. Alarm Mask *Request* Message--Message Type 41

Format	Field	Definition
2d	Type	41—message type 41
2d	PeerWay	PeerWay node number 00=PeerWay number on which the SCI resides
32d	Mask	32 digit mask that determines whether or not alarms generated by each node on the PeerWay are stored in the SCI alarm queue. Each digit represents a PeerWay node. The first digit represents node 1, the second digit represents node 2, and so on. 0= alarms generated by this node are not stored in the SCI alarm queue 1= alarms generated by this node are stored in the SCI alarm queue

Table 1.3.41. Alarm Mask *Reply* Message--Message Type 41

Format	Field	Definition
2d	Type	41—message type 41
3d	Error	Error code. For a list of error codes, see Section 6.

Alarm Information Message Formats (Message Type 42)

The Alarm Information message is used to retrieve information about a PeerWay node alarm from the SCI alarm queue. Figure 1.3.9 describes how to retrieve alarm information from the SCI alarm queue.

Table 1.3.42 shows the Alarm Reporting request message format. Table 1.3.43 shows the Alarm Reporting reply message format.

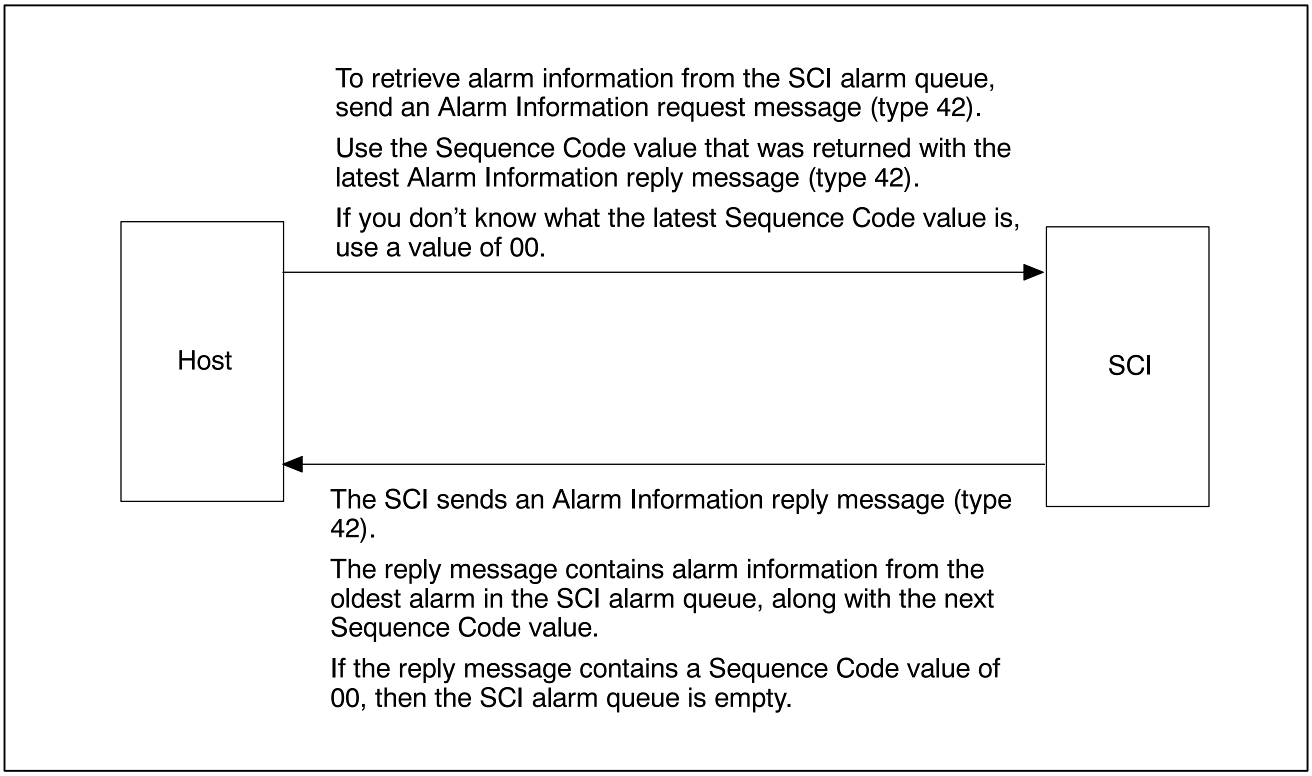


Figure 1.3.9. Retrieving Alarm Information from the SCI Alarm Queue

Table 1.3.42. Alarm Reporting *Request* Message--Message Type 42

Format	Field	Definition
2d	Type	42—message type 42.
2d	Sequence Code	Sequence code associated with a PeerWay node alarm.

Table 1.3.43. Alarm Reporting *Reply* Message--Message Type 42

Format	Field	Definition
2d	Type	42—message type 42.
3d	Error	Error code. For a list of error codes, see Section 6.
2d	Sequence Code	Sequence code returned by the SCI. If no items remain in the SCI alarm queue, this field is 00.
3d	Number of Items	Number of items remaining in the SCI alarm queue.
v	Alarm Item	Alarm data item. The format of the data item is shown below. If no items remain in the SCI alarm queue, this field is zero characters in length.
		The alarm data item has the following format:
1d	Transaction	Type of alarm transaction 0= operator change log entry 1= alarm / event occurrence 2= alarm acknowledge 3= alarm / event clear 4= alarm message
1d	Alarm Type	Type of alarm 0=event 1=advisory alarm 2=critical alarm 3=hardware alarm 4=system alarm 5=disk message 6=operator change log 7=report alarm 8=batch alarm
2d	Alarm Priority	Alarm priority or console key number.
3d	Event Queue #	Event queue number.
16c	Block Address	Address of block in alarm.
3d	Alarm Number	Alarm number or Operator Log ID number.

(continued on next page)

Table 1.3.43. Alarm Reporting *Reply* Message--Message Type 42 (continued)

Format	Field	Definition
2d	Year	Year time-stamped for alarm.
2d	Month	Month time-stamped for alarm.
2d	Date	Day of month time-stamped for alarm.
2d	Hour	Hour time-stamped for alarm.
2d	Minute	Minute time-stamped for alarm.
2d	Second	Second time-stamped for alarm.
vc	Text	0 to 80 characters of formatted alarm text. For any nonzero alarm “Transaction” field value, the alarm text contains a block tag and the alarm message. For the operator change log (“Transaction” field value equals 0), the text contains the following data: • Block tag or address of the alarm source • Change description • Old value • New value • Node type • User key type

Trend Data Message (Message Type 61)

Purpose

The Trend Data message is used to retrieve data values from console trend files.

Amount of Data

Up to 60 data samples are retrieved from the trend file.

Start Date and Time

The date and time specified in the Trend Data request message determine the first data sample retrieved from the trend file. The trend data samples are then retrieved backwards in time until 60 samples have been retrieved.

The date and time specified in the request message must have already occurred.

Period

The Period field in the request message determines the time between data samples that are retrieved from the trend file. The trend data samples are retrieved backwards in time from the start date and time according to the frequency of the Period field.

The Period field value is not associated with the trend file sampling frequency.

Message Formats

Table 1.3.44 and Table 1.3.45 show the Trend Data request message formats. Table 1.3.44 is the recommended format. Table 1.3.45 is an alternative format, but it can only be used on PeerWay 1.

Table 1.3.46 shows the Trend Data reply message format.

Table 1.3.44. Trend Data *Request* Message--All PeerWays--Message Type 61

Format	Field	Definition
2d	Type	61—message type 61
3d	Node	Node number of console to get trend data from.
1d	File Number	Trend file number.
16c	Block Variable	RS3 block variable.
1d	Scaling	0= analog values have internal scaling (0 to 1) discrete values are 0 or 1 1= analog values have display scaling (“Eng Zero” and “Eng Max” fields) discrete values are one-half of a message pair
2d	Year	Year to start retrieving trend data for the block variable.
2d	Month	Month to start retrieving trend data for the block variable.
2d	Date	Day to start retrieving trend data for the block variable.
2d	Hour	Hour to start retrieving trend data for the block variable.
2d	Minute	Minute to start retrieving trend data for the block variable.
2d	Second	Second to start retrieving trend data for the block variable.
6d	Period	Number of seconds between data samples.

Table 1.3.45. Trend Data *Request* Message--PeerWay 1 Only--Message Type 61

Format	Field	Definition
2d	Type	61—message type 61
2d	Node	Node number of console to get trend data from.
1d	File Number	Trend file number.
16c	Block Variable	RS3 block variable.
1d	Scaling	0= analog values have internal scaling (0 to 1) discrete values are 0 or 1 1= analog values have display scaling ("Eng Zero" and "Eng Max" fields) discrete values are one-half of a message pair
2d	Year	Year to start retrieving trend data for the block variable.
2d	Month	Month to start retrieving trend data for the block variable.
2d	Date	Day to start retrieving trend data for the block variable.
2d	Hour	Hour to start retrieving trend data for the block variable.
2d	Minute	Minute to start retrieving trend data for the block variable.
2d	Second	Second to start retrieving trend data for the block variable.
6d	Period	Number of seconds between data samples.

Table 1.3.46. Trend Data *Reply* Message--Message Type 61

Format	Field	Definition
2d	Type	61—message type 61
3d	Error	Error code. For a list of error codes, see Section 6.
3d	Trend Status	Current console status of the point being trended. 0=good 1=new trend, no data yet 2=configuration error 3=no room in Controller Processor for trend 4=data missing 5=bad type, analog / contact conflict 6=initial, trend config change 7=trend not in this file 8=trend not in this node 9=unable to communicate with console
2d	Year	Year of first sample.
2d	Month	Month of first sample.
2d	Date	Day of first sample.
2d	Hour	Hour of first sample.
2d	Minute	Minute of first sample.
2d	Second	Second of first sample.
2d	# of Samples	Number of samples returned.
v	Reply List	List of replies. Each reply is one data sample for the block variable. The format of each entry is shown below.
		Each entry in the list has the following format:
8p	Value	Data sample value.
2d	Alarm Code	Alarm value associated with this point. If multiple codes can exist for an entry, the highest code number is returned. 41= caution: bad scaling 63= null, ignore this data point 64= low, data point below −12.5% 65= high, data point above 112.5% 66= unable to trend value for data point
1c	Comma	A comma appears after each item, including the last item.

Report Transfer Message (Message Type 62)

Purpose

The Report Transfer message is used to transfer a generated report to the host. The report is typically transferred one line at a time using a series of request and reply messages, but more sophisticated transfer options can be specified in the request message.

Sequence Numbers

Sequence numbers are used in the request and reply messages to properly transfer the report.

Number of active report transfers

Up to 10 report transfers can be active at any one time.

Typical operating sequence

Figure 1.3.10 shows the typical operating sequence for a report transfer.

Completion of report transfer

If the reply contains the final portion of a report, the returned sequence number is 0.

Specifying a generated report

You can specify which generated report you want to have transferred from a report file in two ways:

- Specify the generated report number in the Report Number field in the request message. The SCI then ignores the date and time fields. This is the preferred method.
- Specify the date and time of the generated report in the Year, Month, Day, Hour, Minute, and Second fields. Enter a 0 in the Report Number field. The first generated report in the report file on or after this date and time is transferred. This method requires extra disk accesses and can take a long time to complete.

Message Formats

Table 1.3.47 shows the Report Transfer request message format. Table 1.3.48 shows the Report Transfer reply message format.

Typical Operating Sequence

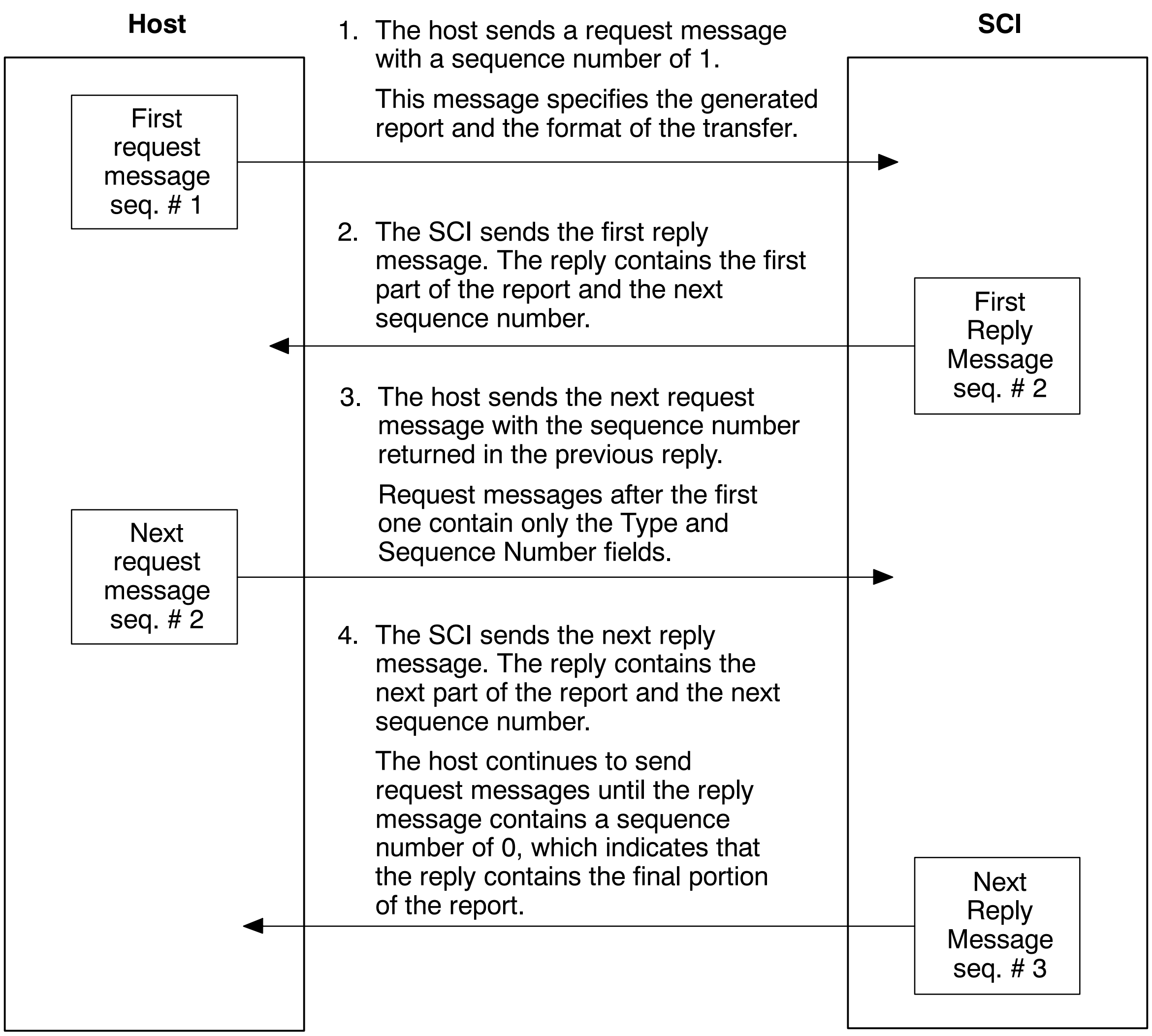


Figure 1.3.10. Typical Operating Sequence of Report Transfer

Table 1.3.47. Report File Transfer *Request* Message--Message Type 62

Format	Field	Definition
2d	Type	62—message type 62
4d	Sequence Number	Sequence number of the report transfer message. The initial message should have a sequence number of 1. Subsequent messages should use the sequence number provided in the previous reply message.
		The fields below are required only if the sequence number is 1.
10c	Volume	Disk volume name.
3d	Node	PeerWay node number of disk.
1d	Drive	Disk drive number.
9c	Filename	Report file name.
5d	Report Number	Generated report number. If non-zero, this value determines which generated report in the report file is transferred. The date and time fields are ignored. If zero, the date and time fields determine which generated report in the report file is transferred. The first generated report on or after the specified date and time is transferred.
2d	Year	Year of desired generated report.
2d	Month	Month of desired generated report.
2d	Day	Day of desired generated report.
2d	Hour	Hour of desired generated report.
2d	Minute	Minute of desired generated report.
2d	Second	Seconds of desired generated report.
4d	Max Length	The maximum length of a reply message. Valid entries are 32 to 1024.
3d	Timeout	This timeout specifies the maximum amount of time, in seconds, allowed from the last reply message to the next request message. Once the timeout is reached, the report transfer is aborted. Valid entries are 0 to 300 seconds. In general, enter 0 if only one report transfer is to be active at a time. Enter a non-zero value if more than one report transfer is to be active at a time.
1d	Generation Checking	Frequency of report generation checking. If the report file that is being accessed generates a report during the transfer, the status code of the reply message indicates an error and the report transfer may be corrupt. This field specifies how often the report file is checked to see if a new report has been generated. 0= check when transfer is complete. If a report generation has occurred during the transfer, an error code will be returned with the last reply message. 1= check with each request. This entry slows down the transfer time but gives a quicker indication of the error condition.

(continued on next page)

Table 1.3.47. Report File Transfer *Request* Message--Message Type 62 (continued)

Format	Field	Definition
1d	Line Transmission	Determines whether lines of a report can be transmitted in whole or in part. 0= partial lines can be transmitted in a message. 1= only complete lines can be transmitted in a message. If this entry is selected, the Max Length field entry must be sized to handle the longest possible report line.
1d	Multiple Lines	Determines whether or not multiple report lines can be transmitted in a message. 0= multiple report lines can be transmitted in a message. 1= only one report line can be transmitted in a message.
1d	Blank Lines	Determines whether or not blank report lines are included in a message. 0= ignore blank report lines. 1= include blank report lines.
1d	Space Compression	Determines whether or not messages contain space compression. 0= use space compression in messages. If 5 or more consecutive space characters appear in a report line, they are replaced by the space compression character (”compress char” field) and a three-digit compression code (”compress code” field). For more information, see the “Space Compression” heading in this section. 1= don’t use space compression (include space characters).
1c	Compression Character	Space compression character. (Used if “Space Compression” entry is 0.) This character is used in replies to signify space compression and is followed by the three-digit compression code. It is recommended that this character not appear in the report text.
1d	Compression Code	Meaning of three-digit compression code. (Used if “Space Compression” entry is 0.) 0=code represents the number of space characters. 1=code represents the column position of the next non-space character.
1d	Repeat	Action to take when the space compression character (“compress char” field) occurs in the report text. (Used if “Space Compression” entry is 0.) 0=replace the character in the report text with a space character. 1=repeat the character in the report text.

Table 1.3.48. Report File Transfer *Reply* Message--Message Type 62

Format	Field	Definition
2d	Type	62—message type 62
3d	Error	Error code. For a list of error codes, see Section 6.
4d	Sequence #	The sequence number that should be used in the next request message. A value of 0 indicates that transfer of the report is complete after this reply message.
3d	# of Entries	The number of entries included in this reply.
v	List of Entries	List of report entries. The format of each entry is shown below.
		Each entry in the list has the following format:
5d	Row Number	The report row number. The row number does not change when transmitting the remainder of a partial line.
3d	Line Length	The length of the transmitted report line.
v	Text	The ASCII characters of the report line.

Simple Report Transfer

The Report Transfer request message contains several transfer option fields. Most users will only perform simple report transfers. Table 1.3.49 shows the values that you can use for the transfer option fields to perform simple report transfers.

Table 1.3.49. Transfer Option Field Values to Perform Simple Report Transfers

Transfer Option Field	Value	Description of Simple Report Transfer
Timeout	0	No timeout is specified between a reply message and the next request message.
Generation Checking	1	The report file is checked after each request to see if a report was generating during the transfer.
Line Transmission	1	Only complete report lines are transferred. The Max Length field must be sized to handle the longest possible report line.
Multiple Lines	1	Only one report line is transmitted in a reply message.
Blank Lines	1	Blank lines are included in the report.
Space Compression	1	No space compression is used.
Compression Character	1	Value is ignored. Has no effect.
Compression Code	1	Value is ignored. Has no effect.
Repeat	1	Value is ignored. Has no effect.

Space Compression

Space compression can be used to reduce the number of characters in a report transfer. With space compression, if 5 or more consecutive space characters appear in a report line, they are replaced by the space compression character and a 3-digit compression code.

Figure 1.3.11 shows an example of space compression.

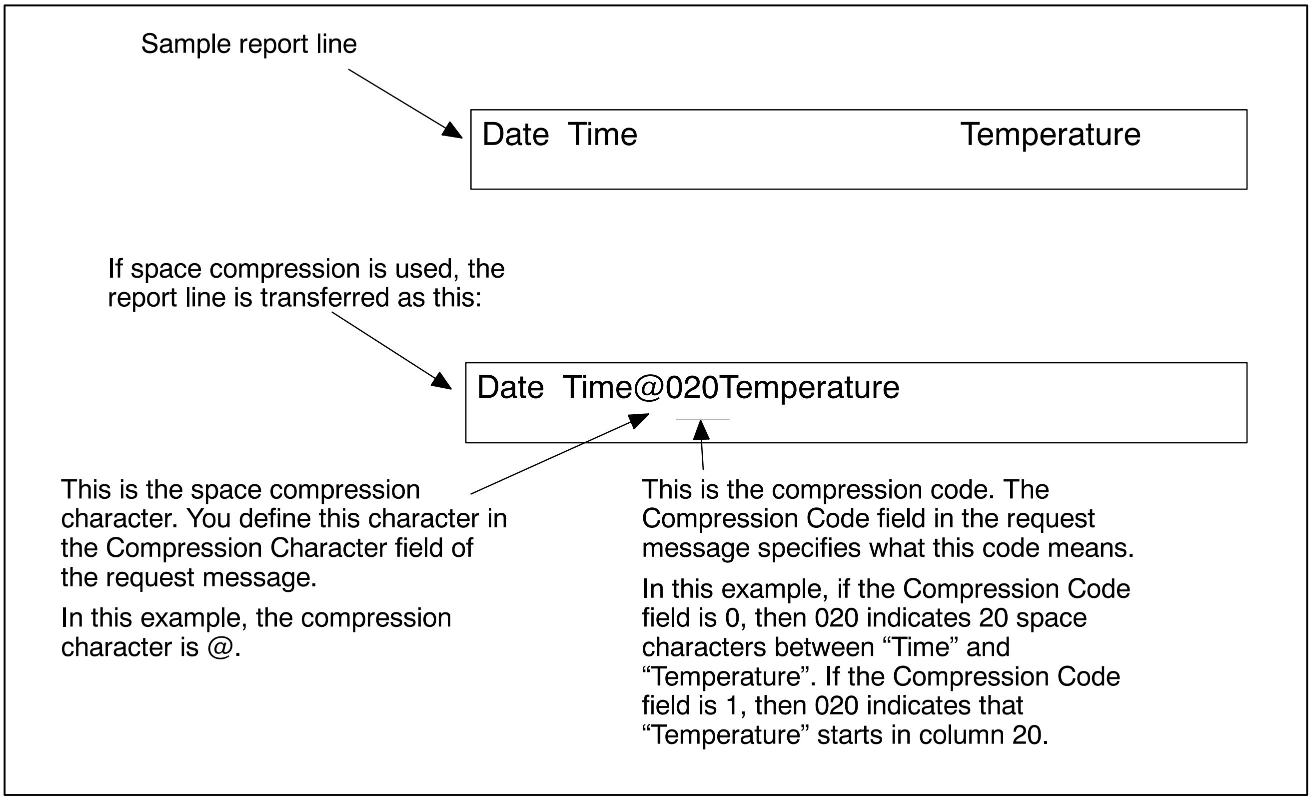


Figure 1.3.11. Sample Use of Space Compression

Raw File Transfer Read Message (Message Type 63)

Purpose

The Raw File Transfer Read message is used to transfer an ASCII file from an RS3 hard disk to the host.

Valid Files

You can transfer file data from two types of folders:

- SRU Data folder including virtual array and virtual string array files.
- ASCII Files folders including the following file types:
  - Script in ASCII format and script file parse errors
  - ASCII Batch Operations Table and ASCII Batch Operations Table parse errors
  - ASCII Batch Units Table and Batch Units Table parse errors
  - ASCII Batch Materials Table and Batch Materials Table parse errors
  - ASCII Master Recipe configuration and Master Recipe configuration parse errors
  - ASCII Control Recipe configuration and Control Recipe configuration parse errors
  - ASCII Working Recipe and Working Recipe parse errors

Message Size

Data can be transferred either in multiple or single packets:

- Multiple packet transfers return as many non-zero packets of data as will fit in the requested data size. If there are more than six consecutive 0x00's in the file, the file is broken into multiple packets to avoid transmitting null data.
- Single packet transfers are intended to aid in debugging the host program during development of the file transfer routines. Since the Raw File Transfer algorithm returns the first non-null data at or after the requested offset, you can create a test file containing no "null" (hex 0x00) data at the requested offset and then use the Memory View screen on the console to view files to aid in the debugging process.

File-dependent Data

Some file types, such as virtual array and virtual string array files in an SRU folder, have data stored in the directory entry, which is not accessible at any offset. The “Return File Data” field in the request message must contain a “1” if you want the reply message to return this data. For information about the structure of the virtual array and virtual string array files, see the *RBL Manual*.

Message Formats

Table 1.3.50 shows the Raw File Transfer Read request message format. Table 1.3.51 shows the Raw File Transfer Read reply message format.

Table 1.3.50. Raw File Transfer Read *Request* Message — Message Type 63

Format	Field	Definition
2d	Type	63—message type 63
10c	Disk Volume Name	Name of the disk containing the ASCII file to be transferred
3d	Disk Node Number	Node number of the disk containing the file to be transferred
1d	Drive Number	Number of the disk drive containing the file to be transferred **NOTE:** Either the disk name or the node/drive numbers can be blank (spaces for the disk name, zeros for the node and drive numbers) because the disk name can be derived from the numbers, or the node and drive can be derived from the name; if the name, disk, and node are all included, they are compared and an error occurs if they do not match.
2d	Folder Number	19 = SRU Data folder (.ad files) 22 = ASCII Files folder (.asc files)
2d	File Subtype	**The file type is one of the following:** SRU Data Folder: 0 = Virtual Array file; 2 = Virtual String Array file ASCII Files Folder: 0 = Script in ASCII format; 1 = Script file parse errors 2 = ASCII Batch Operations Table; 3 = ASCII Batch Operations Table parse errors 4 = ASCII Batch Units Table; 5 = ASCII Batch Units Table parse errors 6 = ASCII Batch Materials Materials Table; 7 = ASCII Batch Table parse errors 8 = ASCII Master Recipe configuration; 9 = ASCII Master Recipe configuration parse errors 10= ASCII Control Recipe configuration; 11= ASCII Control Recipe configuration parse errors 12 = ASCII Working Recipe; 13 = ASCII Working Recipe parse errors
9c	File Name	Name of the file on the RS3 disk to be transferred
1d	Return File-dependent Data	0 = Do not return file information from the directory entry 1 = Return file information from the directory entry **NOTE:** Some file types (such as virtual array and virtual string array files in an SRU folder) have data stored in the directory entry, which is not accessible at any offset. If this field is 1, the data will be returned in “File-dependent Data” portion of the reply packet.
8h	File Offset	Offset in file indicating start of the data to be read.
2h	Maximum Packet Size	Maximum allowable packet size can be between 1 and 256 bytes (expressed in hexadecimal as 00 to FF where 00 indicates 256 bytes).
1d	Number of Packets	0 = Multiple 1 = Only one **NOTE:** Option 1 (single packet of data) should be used only for debugging the source program.

Table 1.3.51. Raw File Transfer Read *Reply* Message — Message Type 63

Format	Field	Definition
2d	Type	63—message type 63
3d	Error	Error code. For a list of error codes, see Section 6.
12d	File Modification Time Stamp	Used by the host to determine whether or not a file has been changed between consecutive reads; the time stamp should not change if the file has not been modified between successive reads.
8h	Non-zero File Offset	Indicates the next data-containing file offset; "00000000" indicates no more data.
2h	Total Size of Returned Packet	Total size of returned data; can be between 0 and 256 bytes (returned as hexadecimal 00 to FF where 00 indicates 256 bytes).
	Data	Varies with the type of file and number of packets. See Table 1.3.52.
	File-dependent Data	Format depends on the folder and subtype as listed below. **NOTE:** Data is returned in this portion of the message only if a "1" appears in the "Return File Data" field of the request message which must be set if the file is a virtual array or virtual string array file in an SRU folder.
		Virtual Array (Folder = 19, Subtype = 0)
5d	Rows	Number of rows in the virtual array
5d	Columns	Number of columns in the virtual array
		Virtual String Array (Folder = 19, Subtype = 2)
3d	Version	Version of the virtual string array
3d	Maximum characters	Maximum number of characters in each character string
5d	Rows	Number of rows in the virtual array
5d	Columns	Number of columns in the virtual array

Table 1.3.52. Structure of Packet Data

Binary Format	ASCII Format	Field	Definition
			Single Packet Data
1b	2h	Data	Data; the ASCII hexadecimal format uses two digits per byte.
			Multiple Packets of Data
4b	8h	Offset	Offset of this packet of data
2b	4h	Length	Length of this packet in bytes
nb	nh	Data	Data; the ASCII hexadecimal format uses two digits per byte.

Raw File Transfer Write Message (Message Type 64)

Purpose

The Raw File Transfer Write message is used to transfer data from the host to an ASCII file on an RS3 disk.

Valid Files

You can transfer file data into two types of folders:

- SRU Data folder including virtual array and virtual string array files.
- ASCII Files folders including the following file types:
  - Script in ASCII format and script file parse errors
  - ASCII Batch Operations Table and ASCII Batch Operations Table parse errors
  - ASCII Batch Units Table and Batch Units Table parse errors
  - ASCII Batch Materials Table and Batch Materials Table parse errors
  - ASCII Master Recipe configuration and Master Recipe configuration parse errors
  - ASCII Control Recipe configuration and Control Recipe configuration parse errors
  - ASCII Working Recipe and Working Recipe parse errors

Potential Situations

When using the Raw File Transfer Write Message, the following situations might cause file corruption:

- Corrupted data in an ASCII file can have unpredictable effects on the file transfer and can cause failure of the file to parse.
- A single bit changed in a Virtual Array file can corrupt the information contained in the file.

In both of the above situations, restoring the file from a backup should correct any file corruption.

WARNING

To minimize the risk of damaging files, you should use caution when implementing Message Type 64. Incorrect use can cause irretrievable loss of data and associated damage such as ruined virtual array and ASCII files.

The following practices are recommended to assist in correcting any file corruption problems:

- Make regular backups of every disk and tape in the system, and keep those backups in a secure place where they are not subject to theft or damage.
- Make copies of files that you want to change and then modify the copy. Be certain that the changes work before you replace the original file with the modified copy.
- Take steps to prevent unauthorized access to the Supervisory Computer Interface and other nodes capable of writing directly to a file. Do not assume that the SCI password is sufficient security.

File-dependent Data

Some files types (such as the virtual array and virtual string array files in the SRU folder) have information stored in the directory entry which is not accessible at any offset. If the “File-specific Data” field in the request message is true, the “File-dependent Data” in the request message will be written into the directory entry.

Message Formats

Table 1.3.53 shows the Raw File Transfer Write request message format. Table 1.3.54 shows the Raw File Transfer Write reply message format.

NOTE: A line feed or carriage return character must be included at the end of each packet that is being written to disk; this character, however, is not included in the total packet size.

Table 1.3.53. Raw File Transfer Write *Request* Message — Message Type 64

Format	Field	Definition
2d	Type	64—message type 64
10c	Disk Volume Name	Name of the disk containing the ASCII file to be transferred
3d	Disk Node Number	Node number of the disk containing the file to be transferred
1d	Drive Number	Number of the disk drive containing the file to be transferred **NOTE:** Either the disk name or the node/drive numbers can be blank (spaces for the disk name, zeros for the node and drive numbers) because the disk name can be derived from the numbers, or the node and drive can be derived from the name; if the name, disk, and node are all included, they are compared and an error occurs if they do not match.
2d	Folder Number	19 =SRU Data folder (.ad files) 22 =ASCII Files folder (.asc files)
2d	File Subtype	**The file subtype is one of the following:** SRU Data Folder: 0 = Virtual Array file 2 = Virtual String Array file ASCII Files Folder: 0 = Script in ASCII format 2 = ASCII Batch Operations Table 4 = ASCII Batch Units Table 6 = ASCII Batch Materials Table 8 = ASCII Master Recipe configuration 10 = ASCII Control Recipe configuration 12 = ASCII Working Recipe
9c	File Name	Name of the file to be transferred
1d	File-specific Data	0 = False 1 = True **NOTE:** Some file types have data stored in the directory entry which cannot be written at any offset. If this flag is true, the “File-depended Data” included in the request message will be written into the directory entry.
	Data	Varies with the type of file as listed below:
8h	File Offset	Offset in file indicating where to start writing the data
4h	Length	Length of this packet in bytes.
nh	Data	Data; the hexadecimal format uses two digits per byte.

(continued on next page)

Table 1.3.53. Raw File Transfer Write *Request* Message — Message Type 64 (continued)

Format	Field	Definition
	File-dependent Data	Valid only for virtual array and virtual string array files in an SRU folder; data is written to the directory entry only if the File-specific Data field is specified as “true”; format depends on the folder and subtype as listed below.
		Virtual Array (Folder = 19, Subtype = 0) **WARNING** **If this data is changed on an existing file, the current array data will be scrambled. This data must be written to a newly created virtual array file before it can be used.**
5d	Rows	Number of rows in virtual array (1 to 50,000)
5d	Columns	Number of columns in virtual array (1 to 50,000)
		Virtual String Array (Folder = 19, Subtype = 2) **WARNING** **If this data is changed on an existing file, the current array data will be corrupted. This data must be written to a newly created virtual string array file before it can be used.**
3d	Version	Version of virtual string array; must be greater than 0.
3d	Maximum characters	Maximum number of characters in each character string; must be greater than 0 and less than 270.
5d	Rows	Number of rows in virtual array; must be greater than 0 and the number of rows multiplied by the number of columns must be less than 60,000.
5d	Columns	Number of columns in virtual array; must be greater than 0 and the number of rows multiplied by the number of columns must be less than 60,000.

Table 1.3.54. Raw File Transfer Write *Reply* Message — Message Type 64

Format	Field	Definition
2d	Type	64—message type 64
3d	Error	Error code. For a list of error codes, see Section 6.
12d	Previous File Modification Time Stamp*	In the form: yymmddhhmmss
12d	New File Modification Time Stamp*	In the form: yymmddhhmmss

* The host can use the time stamp as a check to see if the file has been written by another device between writes. The “Previous Time Stamp” of the current message should match the “New Time Stamp” of the previous message if the file has not been modified between consecutive write requests.

RS3 Information Messages

Purpose

RS3 information messages allow you to read and write certain RS3 information.

Messages

Message type 71—Set RS3 Time Message

- used to set the RS3 time from the host.

Message type 72—Lookup Block Address Message

- used to get a block address from a block tag.

Message type 73—ControlFile Status Message

- used to get information from a ControlFile Status screen.

Set RS3 Time Message Formats (Message Type 71)

Table 1.3.55 shows the Set RS3 Time request message format.
Table 1.3.56 shows the Set RS3 Time reply message format.

NOTE: Excessive manipulation of the system time can be disruptive to time-dependent functions such as trending, reports, and ControlBlock logic.

Table 1.3.55. Set RS3 Time *Request* Message--Message Type 71

Format	Field	Definition
2 d	Type	71—message type 71.
2 d	Year	Year that RS3 time is set to.
2 d	Month	Month that RS3 time is set to.
2 d	Day	Day that RS3 time is set to.
2 d	Hour	Hour that RS3 time is set to.
2 d	Minute	Minute that RS3 time is set to.
2 d	Second	Second that RS3 time is set to.

Table 1.3.56. Set RS3 Time *Reply* Message--Message Type 71

Format	Field	Definition
2 d	Type	71—message type 71.
3 d	Error	Error code. For a list of error codes, see Section 6.

Look Up Block Address Message Formats (Message Type 72)

Table 1.3.57 shows the Look Up Block Address request message format. Table 1.3.58 shows the Look Up Block Address reply message format.

Table 1.3.57. Look Up Tag *Request* Message--Message Type 72

Format	Field	Definition
2 d	Type	72—message type 72.
8 c *or* 16 c	Block Tag	RS3 tag for which to find an address. Upper and lower case conventions must be observed. The request message can use a block tag that is either 8 or 16 characters. **NOTES:** • If the tag is less than the designated 16 characters, fill the remainder of the field with blank spaces. • If you attempt to enter the field code without filling in unused spaces, an error code of 002 is returned. • For compatibility with RS3 Version 16 and earlier software, 8 character tags are allowed. However, for nodes configured for 16 character tags, use all 16 characters.

Table 1.3.58. Look Up Tag *Reply* Message--Message Type 72

Format	Field	Definition
2 d	Type	72—message type 72.
3 d	Error	Error code. For a list of error codes, see Section 6.
10 c	Block Address	Block address corresponding to the requested tag.

ControlFile Status Message Formats (Message Type 73)

Table 1.3.59 and Table 1.3.60 show the ControlFile Status request message format. Table 1.3.59 is the recommended format. Table 1.3.60 is an alternative format, but it can be used only on PeerWay 1.

Table 1.3.61 shows the ControlFile Status reply message format.

Table 1.3.59. ControlFile Status *Request* Message--All PeerWays--Message Type 73

Format	Field	Definition
2 d	Type	73—message type 73.
4 d	Node	Node number of desired ControlFile.

Table 1.3.60. ControlFile Status *Request* Message--PeerWay 1 Only--Message Type 73

Format	Field	Definition
2 d	Type	73—message type 73.
2 d	PeerWay	PeerWay number of the desired ControlFile.
2 d	Node	Node number of desired ControlFile.

Table 1.3.61. ControlFile Status *Reply* Message--Message Type 73

Format	Field	Definition
2 d	Type	73—message type 73.
3 d	Error	Error code. For a list of error codes, see Section 6.
3 f	CP Idle Time	CP “Idle Time” field value.
2 d	CP Avail Links	CP “Avail Link” field value.
1 d	Bubble Active	Indicates which bubble memory is active. 0=left 1=right 2=none
3 f	Bubble Free	“Bubble Memory: Free Space” field value.
5 f	Left CP Boot Rev	“Left CP: Boot” field value.
5 f	Rt CP Boot Rev	“Right CP: Boot” field value.
5 f	CP Rev	“Prgm” field value.
1 d	CP Active	Indicates which CP is currently active. 0=left 1=right
5 f	Controller Rev	“Prgm Rev” field value.
v	Entry List	List of Controller Processor status entries. The first entry is for the Controller Processor in slot A, the second entry is for the Controller Processor in slot B, and so on. The format of each entry is shown below.
		Each entry in the list has the following format:
2 d	Card Type	“Card Type” field value. 99=None 06=SMART 12=MPCAP 01=MLC 07=RBLC 13=MPCAT 02=CC 08=MPC 14=MPC2 03=SSC 09=ATMLC 15=MPC5 04=MUX 10=MPTUN 05=PLC 11=MPCAS
5 f	Cont Boot Rev	“Boot Rev” field value.
3 f	Cont Idle Time	“Idle Time” field value.
3 f	Cont Free Space	“Free Space” field value.
2 d	Cont Avail Links	“Avail Links” field value.

(continued on next page)

Table 1.3.61. ControlFile Status *Reply* Message--Message Type 73 (continued)

Format	Field	Definition
1 d	Redundancy	Redundancy / backup mode of the Controller Processor: 0= This is not a redundant Controller Processor n= This is a redundant Controller Processor. The number n indicate the slot number of the primary controller (1=A, 2=B, 3=C, 4=D, 5=E, 6=F, 7=G, 8 = H). 9= This is a primary controller constrained from switching for one of the following reasons: • The primary controller has a hardware alarm. • The primary controller has been switched too often in too short a period of time. • The settle time of the controller has not expired.
1 d	Mode	Controller Processor mode. 0=normal 1= standby
1 d	Alarm Inhibit	“Alarm Inhib” field value. 0= no 1= yes
4 f	Scan Time	“Scan Time” field value.
5 f	SCI Timeout	“SCI Time out” field value.

SCI Information Messages

Purpose

SCI messages allow the host to read and write auxiliary SCI information.

Messages

Message type 81—SCI Status Message

- used to retrieve status information about the SCI.

Message type 82—Read Message Pairs Message

- used to read the message pairs that are loaded into the SCI.

Message type 83—Message Format Message

- used for asynchronous length and X.25 protocols to specify the format of message data.

SCI Status Message Formats (Message Type 81)

Table 1.3.62 shows the SCI Status request message format.
Table 1.3.63 shows the SCI Status reply message format.

Table 1.3.62. SCI Status *Request* Message--Message Type 81

Format	Field	Definition
2 d	Type	81—message type 81.

Table 1.3.63. SCI Status *Reply* Message--Message Type 81

Format	Field	Definition
2 d	Type	81—message type 81.
3 d	Error	Error code. For a list of error codes, see Section 6.
4 d	Node	PeerWay node number.
2 d	Year	Year of current RS3 date.
2 d	Month	Month of current RS3 date.
2 d	Day	Day of current RS3 date.
2 d	Hour	Hour of current RS3 time.
2 d	Minute	Minute of current RS3 time.
2 d	Second	Second of current RS3 time.
3 d	# of Alarms	Number of alarm messages waiting to be sent to the host.
4 d	# of Points 15	Number of points in the SCI table being polled every 15 seconds.
4 d	# of Points 30	Number of points in the SCI table being polled every 30 seconds.
4 d	# of Points 60	Number of points in the SCI table being polled every 60 seconds.
4 d	# of Points 120	Number of points in the SCI table being polled every 120 seconds.
4 d	# of Points 300	Number of points in the SCI table being polled every 300 seconds.
4 d	# of Points 5	Number of points in the SCI table being polled every 5 seconds.
4 d	# of Points 10	Number of points in the SCI table being polled every 10 seconds.
4 d	# of Pts Contin	Number of points in the SCI table being continuously polled.
4 d	PeerWay A Qual	Quality of PeerWay A from the PEERWAY OVERVIEW screen.
4 d	PeerWay B Qual	Quality of PeerWay B from the PEERWAY OVERVIEW screen.
4 d	PeerWay Margin	Current PeerWay margin.
3 d	Comm Errors	Total number of all comm errors on messages received from the host.
3 d	Duplicates	Total number of duplicate messages received from the host.

Read Message Pairs Message Formats (Message Type 82)

Table 1.3.64 shows the Read Message Pairs request message format. Table 1.3.65 shows the Read Message Pairs reply message format.

Table 1.3.64. Read Message Pairs *Request* Message--Message Type 82

Format	Field	Definition
2 d	Type	82—message type 82.
3 d	Low Entry	Low entry of message pair range requested.
3 d	High Entry	High entry of message pair range requested.

Table 1.3.65. Read Message Pairs *Reply* Message--Message Type 82

Format	Field	Definition
2 d	Type	82—message type 82.
3 d	Error	Error code. For a list of error codes, see Section 6.
v c	Text List	List of text of message pairs. The format of each entry is shown below.
		Each entry in the list has the following format:
8 c	Off Text	String of characters that designate the OFF condition.
8 c	On Text	String of characters that designate the ON condition.
1 c	Comma	A comma follows all message pair items, including the last item.

Data Format Message Formats (Message Type 83)

Asynchronous length protocol

In asynchronous length protocol, this message is used to change the data format and override the “Data Format” field on the SCI Configuration Screen.

The data format specified in this message remains in effect until communication with the SCI has ended. When communication with the SCI resumes again, the data format specified in the “Data Format” field is in effect.

X.25 protocol

In X.25 protocol, this message is used to change the data format of the messages for an individual session. X.25 sessions always start with ASCII data format.

Asynchronous terminator protocol

This message cannot be used in asynchronous terminator protocol.

Message Formats

Table 1.3.66 shows the Data Format request message format.
Table 1.3.67 shows the Data Format reply message format.

Table 1.3.66. Message Format *Request* Message--Message Type 83

Format	Field	Definition
2 d	Type	83--message type 83.
1 d	Format	Message format 0=ASCII floating point and ASCII integers 1=DEC floating point and LSB integers 2=IEEE floating point and MSB integers

Table 1.3.67. Message Format *Reply* Message--Message Type 83

Format	Field	Definition
2 d	Type	83--message type 83.
3 d	Error	Error code. For a list of error codes, see Section 6.

Diagnostic Messages

Purpose

Diagnostic messages allow you to troubleshoot SCI problems.

Messages

Message type 91—Loopback message

- used to perform a loopback test on communications with the SCI. The SCI receives the data from the host, places it in its buffer, then sends the data back.

Message type 92—Debug message

- used to get a copy of a request or reply message from the SCI. The SCI sends a copy of the specified message from its buffer. The SCI keeps approximately the most recent 4K bytes of request messages and 4K bytes of reply messages.

Loopback Message Formats (Message Type 91)

Table 1.3.68 shows the Loopback request message format.
Table 1.3.69 shows the Loopback reply message format.

NOTE: The Loopback reply message does not contain an Error field.

Table 1.3.68. Loopback *Request* Message--Message Type 91

Format	Field	Definition
2 d	Type	91—message type 91.
v	Data	Anything that the host wishes to send.

Table 1.3.69. Loopback *Reply* Message--Message Type 91

Format	Field	Definition
2 d	Type	91—message type 91.
v	Data	A copy of the data received from the host, including the Message ID and Type fields.

Debug Message Formats (Message Type 92)

Table 1.3.70 shows the Debug request message format.
Table 1.3.71 shows the Debug reply message format.

NOTE: The Debug message is not supported for X.25 protocol.

Table 1.3.70. Debug *Request* Message--Message Type 92

Format	Field	Definition
2 d	Type	92—message type 92
2 d	Which Message	Which message should be returned. 0=most recent message 1=next most recent message n=nth most recent message
1 d	Request or Reply	Request or reply message. 0=request message 1=reply message

Table 1.3.71. Debug *Reply* Message--Message Type 92

Format	Field	Definition
2 d	Type	92—message type 92
3 d	Error	Error code. For a list of error codes, see Section 6.
2 d	Which Message	Echo of field value from request message.
1 d	Request or Reply	Echo of field value from request message.
v	Data	The desired message, including the Message ID and Type fields.

Section 4:
Field Codes

This section describes the field codes that are used with Field Code messages to read from and write to specific block data fields.

Note

- The host computer can write to field codes that include a “w”. For information about writing to ControlBlocks, see Section 2.
- Some numeric fields can have a configured value of “None”. A value of “None” is returned as -999999 for low limits and 999999 for high limits.
- Some SCI field codes return status values for alarm messages. These alarm messages are described in the *Alarm Messages Manual*. You can use the return status value to reference the alarm in the *Alarm Messages Manual*.

  The following status values can represent more than one text message for an alarm:

213	Floppy disk errors
447	RBLC script alarms
448	Batch script alarms
481	Batch link errors

Data Types

SCI messages provided in this section can use the following data types.

Table 1.4.1. Message Data Types

Format	Meaning
b	Binary byte.
c	ASCII printing characters.
d	Decimal number.
f	Unscaled floating point value.
h	Hexadecimal number.
m	Discrete value (0 or 1) or message pair text.
p	Scaled floating point number, unscaled floating point number, discrete value, or message pair text.
s	Scaled or unscaled floating point number.
v	Variable format data described elsewhere.

SCI Compatible Block Types

In Section 4, the Field Code messages for the RS3 block types are described in the following tables. Since table descriptions are brief, you may want to refer to the manuals indicated for additional information on block functions and fields.

Field Codes for all Block Types

Table 1.4.2	All Block Types For more information on ControlBlocks, see CB; for information on I/O Blocks, see I/O.

Field Codes for Input/Output blocks:

Table 1.4.3	AIB (Analog Input Block) For more information on AIBs, see IO: 2-1.
Table 1.4.4	AOB (Analog Output Block) For more information on AOBs, see IO: 2-2.
Table 1.4.5	CIB (Contact Input Block) For more information on CIBs, see IO: 6-1.
Table 1.4.6	COB (Contact Output Block) For more information on COBs, see IO: 6-2.
Table 1.4.7	MIB (Multiplexer Input Block) For more information on MIBs, see IO: 7.
Table 1.4.8	PLCB (PLC Block) For more information on PLCBs, see IO: 9-3.
Table 1.4.9	PIOB (Pulse I/O Block) For more information on PIOBs, see IO: 3.
Table 1.4.10	TIB (Temperature Input Block) For more information on TIBs, see IO: 4.
Table 1.4.11	SIB (Smart Input Block) For more information on SIBs, see IO: 8.
Table 1.4.12	VIB (Value Input Block) For more information on VIBs, see IO: 8.
Table 1.4.13	DIB (Discrete Input Block) For more information on DIBs, see IO: 10.
Table 1.4.14	DOB (Discrete Output Block) For more information on DOBs, see IO: 10.

Field Codes for ControlBlocks:

Field Codes for All Block Types

Table 1.4.2 lists field codes that exist for all block types.

Table 1.4.2. Field Codes for All Block Types

Code	Binary Format	ASCII Format	Field	Return Status	Description
0000	8 c	8 c	Tag		Block Tag for 8 character tags. **NOTE:** Field code 0000 will not accept 16 character tags. Use code 0017 instead.
0001	1 b	2 d	Block Type		Type of controller processor blocks:
				01	Analog Input Block (AIB)
				02	Contact Input Block (CIB)
				03	Analog Output Block (AOB)
				04	Contact Output Block (COB)
				06	Multiplexer Input Block (MIB)
				07	PLC I/O Block (PLCB)
				08	Smart Input Block (SIB)
				09	Redundant I/O Block (RIOB)
				10	ControlBlock (CB)
				11	Value Input Block (VIB)
				12	Pulse I/O Block (PIOB)
				15	Temperature Input Block (TIB)
				16	Discrete Input Block (DIB)
				17	Discrete Output Block (DOB)
0002	1 b	2 d	Alarm Priority		Alarm priority for critical and process alarms.
0003	2 b	3 d	Update Code		Update Code.
0004	2 b	3 d	Plant Unit		Plant Unit to which the block is assigned.

(continued on next page)

Table 1.4.2. Field Codes for All Block Types (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
0005	1 b	3 d	Mode		Block mode states. **NOTE:** I/O Block mode cannot be written to.
			I/O Block Types: AIB, DIB, AOB, CIB, COB, DOB, MIB, PLCB	0	Manual
				1	Auto
				2	Override *or* Fail (PLCB only)
			I/O Block types: SIB, VIB	0	Manual
				1	Auto
				2	Simulate
0005 w			ControlBlock Types: PID/ATC Non-PIDAnalog Discrete	0	Local
				1	Manual *or* Operator *(discrete only)*
				2	Auto
				3	Remote
				4	DDC
				5	Comp Sp
0006 w	4 b*	8 p	Output		Value of the ControlBlock output. **NOTE:** I/O Block output cannot be written to.
0007	2 b	4 h	System Flags		Alarm information for source and destination block registers.
0008	2 b	4 h	User Flags		Discrete output states for register Q.
0009	4 b	7 f	Eng Zero		Output zero scaling value.
0010	4 b	7 f	Eng Max		Output maximum scaling value.
0011	1 b	2 d	Resolution		Number of digits to the right of decimal point displayed for block output.
0012	8 c	8 c	Output Units		Output units text.
0013	2 b	3 d	Static Size		Size of static section in words.
0014	2 b	3 d	Dynamic Size		Size of dynamic section in words.
0015	24 c	24 c	Descriptor		User-assigned block descriptor.
0016	2 b	3 d	Message Pair		"Message Pair" value for CIB and COB. Zero-filled for other blocks.
0017	16 c	16 c	Tag		Block Tag for 16 Character tags. Unused character spaces in the tag are left blank.

* If this value is a scaled discrete message pair, then the data format is 8 m.

Input/Output Block Field Codes

The following tables provide field codes for input/output blocks:

Table 1.4.3	AIB (Analog Input Block)
Table 1.4.4	AOB (Analog Output Block)
Table 1.4.5	CIB (Contact Input Block)
Table 1.4.6	COB (Contact Output Block)
Table 1.4.7	MIB (Multiplexer Input Block)
Table 1.4.8	PLCB (PLC Block)
Table 1.4.9	PIOB (Pulse I/O Block)
Table 1.4.10	TIB (Temperature Input Block)
Table 1.4.11	SIB (Smart Input Block)
Table 1.4.12	VIB (Value Input Block)
Table 1.4.13	DIB (Discrete Input Block)
Table 1.4.14	DOB (Discrete Output Block)

Table 1.4.3. AIB (Analog Input Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
0100	2 b	3 d	Sig Char		Signal characterization curve.
				0	None
				1	SQR
				2	TC-B
				3	TC-E
				4	TC-J
				5	TC-K
				6	TC-R
				7	TC-S
				8	TC-T
				9	RTD-100
				10	RTD-1000
				11	RTD-120
				12	RTD-10
0101	4 b	7 f	Filt Time		Filter time constant applied to the field value.
0102	4 b	7 s	Field Value		Transmitter input value.
0103	2 b	3 d	Hw Alarm Code		FIC hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**
0104	4 b	7 f	Inst Low		Low hardware alarm point for field value.
0105	4 b	7 f	Inst High		High hardware alarm point for field value.
0106	4 b	7 s	Crit Low		Low critical process alarm point for block output value.
0107	4 b	7 s	Crit High		High critical process alarm point for block output value.
0108	4 b	7 s	Adv Low		Low advisory process alarm point for block output value.
0109	4 b	7 s	Adv High		High advisory process alarm point for block output value.
0110	4 b	7 f	Al Ddband		Deadband value for all block alarm points.
0111	4 b	7 f	Inst Bias		Bias signal used to modify the field value.
0112	4 b	7 s	Low Cutoff		Linearization low value.
0113	4 b	7 s	Linear Hi		**NOTE:** This field is not used.
0114	4 b	7 s	Nom Out		Not used at this time.

Table 1.4.4. AOB (Analog Output Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
0300	2 b	3 d	Hardware Alarm Code		FIC Hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**
0301	16 b	16 c	Source Address		Address of the source block of input.
0302	4 b	7 s	Actual Value		Output feedback value.
0303	1 b	1 d	Fail Safe Jumper		Output FIC jumper position.
				1	Hold
				2	Zero
0304	2 b	4 f	Output Card Rev		Output FIC revision number.
0305	1 b	1 d	Reverse Acting		Block input to output relationship.
				0	No
				1	Yes

Table 1.4.5. CIB (Contact Input Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
0200	4 b*	8 m	Filtered State		Block output value.
0201	1 b	2 d	Filt Type		Signal characterization values.
				0	None
				1	D-On
				2	D-Off
				3	Delay
				4	Glitch
				5	Settle
				6	Extend
0202	4 b	7 f	Filt Time		Filter time constant applied to the “Filt Type” field.
0203	2 b	3 d	Hardware Alarm Code	Same as 0103	FIC Hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**
0204	1 b	1 d	Rising Edge		Triggers an event message.
				0	No
				1	Yes
0205	1 b	1 d	Falling Edge		Triggers an event message.
				0	No
				1	Yes
0206	1 b	1 d	Alarm Cond		Field state that activates an alarm.
				0	None
				1	On
				2	Off

* If this value is a scaled discrete message pair, then the data format is 8 c.

(continued on next page)

Table 1.4.5. CIB (Contact Input Block) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
0207	1 b	1 d	Event Type		Event type to be generated.
				0	None
				1	Advisory
				2	Critical
				3	Hardware
0208	1 b	1 d	Field Contact		Block input to output relationship.
				0	N.O.
				1	N.C.
0209	2 b	4 f	Contact I/O Card Rev		Revision number of the installed contact FIC.

Table 1.4.6. COB (Contact Output Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
0400	4 b*	8 m	Output State		Block output value.
0401	2 b	3 d	Hardware Alarm Code	Same as 0103	FIC hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**
0402	16 c	16 c	Source address		Address of the source block of input.
0403	1 b	1 d	Fail Safe Jumper		Output FIC jumper position.
				1	Hold
				2	Zero
0404	1 b	1 d	Output Hold		Output signal characterization.
				0	None
				1	On
				2	Off
				3	Pulse
0405	4 b	7 f	Hold Time		Signal characterization time in seconds.
0406	1 b	1 d	Contact Type		Block input to output relationship.
				0	N.O.
				1	N.C.
0407	2 b	4 f	Contact I/O Card Rev		Revision number of the installed contact FIC.

* If this value is a scaled discrete message pair, then the data format is 8 c.

Table 1.4.7. MIB (Multiplexer Input Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
0500	2 b	3 d	Sig Char		MIB signal characteristics
				0	Raw Cnts
				1	10 mV
				2	20 mV
				3	80 mV
				4	400 mV
				5	2 V
				6	20 V
				7	B NBS
				8	E NBS
				9	J NBS
				10	K NBS
				11	R NBS
				12	S NBS
				13	T NBS
				14	J DIN
				15	T DIN
				16	Ref-J
				17	4–20 mA
				18	B NBS mV
				19	E NBS mV
				20	J NBS mV
				21	K NBS mV
				22	R NBS mV
				23	S NBS mV
				24	T NBS mV
				25	J DIN mV
				26	T DIN mV

(continued on next page)

Table 1.4.7. MIB (Multiplexer Input Block) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
0500	2 b	3 d	Sig Char		MIB signal characteristics.
				27	3 PT0D
				28	4 PT0D
				29	3 PT1D
				30	4 PT1D
				31	3 PT0N
				32	4 PT0N
				33	3 PT1N
				34	4 PT1N
				35	3W ohms
				36	4W ohms
				37	Ground
				38	4 Cu1N
				39	3 Cu1N
				40	4 Ni0N
				41	3 Ni0N
				42	4–20 mA SQR
0501	4 b	7 f	Filt Time		Time delay between input and output.
0502	4 b	7 s	Field Value		Current transmitter input value.
0503	2 b	3 d	Hardware Alarm Code	Same as 0103	FIC hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**
0504	4 b	7 f	Inst Low		Low hardware alarm point for field value.
0505	4 b	7 f	Inst High		High hardware alarm point for field value.
0506	4 b	7 s	Crit Low		Low critical process alarm point for block output.
0507	4 b	7 s	Crit High		High critical process alarm point for block output.
0508	4 b	7 s	Crit Rate-Chg		Critical rate of change process alarm point for block output.

(continued on next page)

Table 1.4.7. MIB (Multiplexer Input Block) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
0509	4 b	7 s	Crit Dev		Critical process alarm point for difference between block output and Nom Out.
0510	4 b	7 s	Adv Low		Low advisory process alarm point for block output.
0511	4 b	7 s	Adv High		High advisory process alarm point for block output.
0512	4 b	7 s	Adv Rate-Chg		Advisory rate of change process alarm point for block output.
0513	4 b	7 s	Adv Dev		Advisory process alarm point for block output and Nom Out.
0514	4 b	7 s	Al DdBand		Deadband value for all block alarms.
0515	4 b	7 f	Linear Lo Value		Linearization low value.
0516	4 b	4 f	Linear Hi Value		Linearization high value.
0517	4 b	7 s	Nom Out		Value used with deviation alarms.
0518	4 b	1 d	Degrees		Degrees C (1) or F (0).
0519	4 b	1 d	Not Used		Not used for any field.
0520	4 b	1 d	FEM Type		FEM type actually installed.
				0	None
				1	V/A
				2	RTD
0521	4 b	1 c	Dataport Rev		Dataport version level.

Table 1.4.8. PLCB (PLC Blocks) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
0600	2 b	3 d	Hardware Alarm Code	Same as 0103	FIC hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**
0601	16 c	16 c	Source Addr		Address of ControlBlock (data source).
0602	4 b	7 f	Value from Source		Hexadecimal value from ControlBlock to which PLC is linked.
0603	2 b	4 h	User Flags		User flags from a source link; written to a PLC from a PLCB with a data type of FLAGS.
0604	2 b	4 h	Value to PLC		Value written to the PLC.
0605	2 b	4 h	PLC Write Mask		Sets write capability of PLC to specific bits within the PLC word.
0606	1 b	1 d	Data Type		Type of word being read from or written to the PLC.
				0	Flags
				1	BCD 3
				2	BCD 4
				3	S Int
				4	U int
0607	1 b	1 d	Read		Data to be read from the PLC.
				0	No
				1	Yes
0608	1 b	1 d	Port		Communications port that is being used by the PLC.
				0	A
				1	B
0609	2 b	3 d	PLC Number		Address of the PLC with which the block communicates.
0610	4 b	5 d	PLC Word Address		Word address within the PLC with which the block communicates.
0611	16 c	16 c	Sim Tag		Tag of the ControlBlock from which data is obtained when in simulation mode.

NOTE: If the PLCB includes block flags, the fields 620 and higher are present and represent flag information. The letter i represents a number that is assigned to the flag. Flag a os 0, flag b is1, and so on. For example, field code 621 represents the Alarm Code field for flag b.

(continued on next page)

Table 1.4.8. PLCB (PLC Blocks) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
0612	2 b	3 d	Sample Time		Interval (in seconds) between the PLC read/write cycles.
0613	2 b	3 h	Alarm Mask		Which of the 16 flags are configured to generate alarms and events.
0620+i	1 b	1 d	Alarm Type		Type of alarm issued.
				0	None
				1	Event
				2	Advisory
				3	Critical
				4	Hardware
				5	Batch
0640+i	1 b	1 d	When		When to issue an alarm.
				0	Rise
				1	On
				2	Fall
				3	Off
				4	Change
0660+i	2 b	3 d	Event Type		Type of event issued. Range is from 0 to 255.
0700+i	1 b	2 d	Color of Flag		Color value: • 0 to 15 are for normal video. • 16 to 31 are for reverse video.
0720+i	1 b	1 d	Backlight State		Video state in which the flag name is displayed.
				0	Normal video when the flag value is low
				1	Reverse video when the flag value is high
0740+i	8 c	8 c	Flag Name		8 character flag tag.
0760+i	32 c	32 c	Alarm Descriptor		Description of alarm.

NOTE: If the PLCB includes block flags, the fields 620 and higher are present and represent flag information. The letter i represents a number that is assigned to the flag. Flag a os 0, flag b is1, and so on. For example, field code 621 represents the Alarm Code field for flag b.

Table 1.4.9. PIOB (Pulse I/O Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
2800	1 b	1 d	Auto Lock		Indicates whether the block is locked in Auto mode.
2801	1 b	1 d	Point Type		Type of PIOB.
				0	Pulse Input
				1	Pulse Output
				2	mA Output
2802	16 c	16 c	Source Link		Tag or address of the ControlBlock providing the target input to the PIOB block.
2803	4 b	8 f	Source Output		Value of the ControlBlock output.
2804	1 b	1 d	Type		Type of input pulse.
				0	Voltage
				1	Contact
2805	1 b	1 d	Comm Fail		PIOB output value if communication between the Pulse FIC and the Controller Processor fails.
				0	Zero Output
				1	Hold Output
2806	4 b	8 f	Block Output		Output value of block.
2807	1 b	1 d	Threshold		On/Off threshold appropriate for a particular voltage signal.
				0	Zero
				1	50 Percent
2808	1 b	1 d	Debounce		Debounce filter time.
				0	None
				1	1.00 mS
				2	2.00 mS
				3	5.00 mS
				4	10.00 mS
				5	20.00 mS
				6	50.00 mS

(continued on next page)

Table 1.4.9. PIOB (Pulse I/O Block) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
2809	1 b	1 d	Range		Expected voltage range for the input and actual 50% threshold and hysteresis.
				0	0.10 volts
				1	0.30
				2	1.00
				3	3.00
				4	10.00
				5	30.00
2810	1 b	1 d	Source HF		Hold forward flag option for source link: yes or no.
2811	1 b	1 d	Function		Type of measurement to be made.
				0	Frequency
				1	Duration
				2	Counter
				3	Loader Counter
2812	2 b	3 d	Hardware Alarm Code		FIC hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**
2813	4 b	7 f	Range Zero (Hz)		Input frequency value (in Hertz) that corresponds to a block output of 0.
2814	4 b	7 f	Range Max (Hz)		Input frequency value (in Hertz) that corresponds to a block output of 1.
2815	4 b	7 f	Low Cutoff (Hz)		Frequency input value (in Hertz) below which the frequency input is set to zero and block output flag i is set.
2816	1 b	1 d	Measure from		Input transition at which the timing is to begin and end.
				0	Fall to rise
				1	Fall to fall
				2	Rise to fall
				3	Rise to rise
2817	2 b	3 d	High Cutoff		Maximum value for input duration and maximum value of the block output.
2819	1 b	1 d	Ten Period Avg.		Specifies whether the block output is the average of the last ten measured duration values.

(continued on next page)

Table 1.4.9. PIOB (Pulse I/O Block) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
2819	1 b	1 d	Prescale		Value used to slow the maximum input frequency to less than 500 Hz or to extend the counter.
				0	1
				1	2
				2	4
				3	8
				4	16
				5	32
				6	64
				7	128
2820	4 b	8 d	Target Count		Fixed target count, maximum 12,000,000.
2821	1 b	1 b	Tied to		Point to which the loader counter is to be tied.
				0	Point 2
				1	Point 3
2822	4 b	4 f	Full Scale		The full scale output frequency.
2823	1 b	1 d	Units		Frequency output units.
				0	PPS
				1	CPH
2824	4 b	7 f	Max on Time (Sec)		Maximum number of seconds that the frequency output is on.
2825	4 b	7 f	Frame Duration		Period (1/frequency) at which the output is to be fixed.
2826	2 b	3 d	Minimum On Time (%)		Percentage of pulse on time for a source link value of 0.
2827	2 b	3 d	Maximum On Time (%)		Percentage of pulse on time for a source link value of 1.0.
2828	4 b	7 f	Pulse Rate (Hz)		Frequency of the output pulses.
2829	1 b	2 d	Duty Cycle (%)		Duty cycle of output pulses in percentage of pulse periods.
2830	1 b	1 d	Output Type		Type of output to be provided by the block.
				0	Supply
				1	Analog
				2	Loader

Table 1.4.10. TIB (Temperature Input Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
2900	1 b	1 d	Auto Lock		Indicates whether the block is locked into Auto mode
2901	4 b	7 f	Field Value		Block output value in degrees.
2902	4 b	7 f	Field Value		Block input value in ohms or mV.
2903	1 b	2 d	Sensor Type		Signal characterization curve.
				0	RTD
				1	J
				2	K
				3	T
				4	E
				5	R
				6	S
				7	B
				8	–E
				9	–T
				10	CJC
				11	OHM
				12	MV
2904	1 b	1 d	Temp		Temperature units for the RTD sensor.
				0	C (centigrade)
				1	F (Fahrenheit)
				2	K (Kelvin)
				3	R (Rankine)
2905	1 b	1 d	Range		Sensitivity range of the temperature input FIC.
				0	1
				1	2
2906	16 c	16 c	CJC Link		Source link that provides the cold junction compensation for the thermocouple.
2907	4 b	7 f	Ext. Res		Value of the external resistance for the RTD sensor.

(continued on next page)

Table 1.4.10. TIB (Temperature Input Block) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
2908	1 b	1 d	Calibrate		Mode of calibration.
			Sensor type= RTD, CJC, OHMRange =1	0	None
				1	Lo 0.0 OHM
				2	Hi 100 OHM
			Sensor type= RTD, CJC, OHMRange =2	0	None
				1	Lo 0.0 OHM
				2	Hi 100 OHM
			Range = 1 (Other sensor types)	0	None
				1	Lo −4. mv
				2	Hi 22. mv
			Range = 2	0	None
				1	Lo −16. mv
				2	Hi 88. mv
2909	1 b	1 d	Cal Mode		Select “Yes” to calibrate.
				0	No
				1	Yes
2910	4 b	7 f	CJC Value		Internally scaled temperature value.
2911	4 b	8 f	Inst High		High hardware alarm point applied to the field value.
2912	4 b	8 f	Crit High		High critical alarm point applied to the block output value.
2913	4 b	8 f	Adv High		High advisory process alarm point applied to the block output.
2914	4 b	8 f	Inst Low		Low hardware alarm point applied to the field value.
2915	4 b	8 f	Crit Low		Low critical process alarm point applied to the block output value.
2916	4 b	8 f	Adv Low		Low advisory process alarm point applied to the block output.
2917	2 b	3 d	Hardware alarm code		FIC hardware fault.
2918	4 b	7 s	Al DdBand		Deadband value applied to all of the block alarm points. **For explanations of return status values for alarms, see the Alarm Messages Manual.**

Table 1.4.11. SIB (Smart Input Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
3000	1 b	1 d	Auto Lock		Indicates whether the block is locked into Auto mode.
3001	1 b	1 d	Use		Type of transmitter signal used.
				0	Analog PV
				1	Digital PV
				2	Combined PV
3002	4 b	7 s	Field Value		Current input value and the corresponding temperature scale.
3003	1 b	1 d	Calibrate		SIB block calibration field.
				0	None
				1	Cal Low
				2	Cal High
3004	4 b	5 s	mA		Input value in mA units.
3005	4 b	7 f	Filt Time		Filter time constant applied to the field value.
3006	2 b	3 d	mfg id		Manufacturer ID.
				38	Rosemount
				46	Rosemount Analytical
				25	Kay Ray/Sensall
				31	Micro Motion
3007	2 b	3 d	Xmtr Type		Type of Transmitter
3008	2 b	3 d	Xmtr Units		Engineering units configured for the transmitter.
3009	4 b	8 f	Lo Cutoff		Block output value below which the block output is set to zero.
3010	4 b	7 s	Range 4 mA		4 mA engineering units value configured for the transmitter.
3011	4 b	7 s	Range 20 mA		20 mA engineering units value configured for the transmitter.
3012	16 c	16 c	Sim Link		Tag of the input block when the I/O block is in Simulate mode.
3013	1 b	2 h	Xmtr Status		Code number of a transmitter status message on the Transmitter Status screen.

(continued on next page)

Table 1.4.11. SIB (Smart Input Block) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
3014	1 b	2 d	Xmtr Address		Address of the transmitter.
3015	4 b	7 f	Inst Low		Low hardware alarm point applied to field value.
3016	4 b	7 f	Inst Hig		High hardware alarm point applied to field value.
3017	4 b	7 s	Deadband		Deadband value applied to all block alarm points.
3018	1 b	2 d	FIC Type		Type level of the FIC card communicating with I/O block.
3019	2 b	4 f	FIC Rev		Revision level of the FIC card communicating with I/O block.
3020	2 b	3 d	Hardware Alarm Code		FIC hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**
3021	2 b	3 d	Controller Comm Error		Controller command error messages.
3022	1 b	2 h	FIC Comm Error		FIC command error messages.
3023	1 b	2 h	Xmeter Error		Transmitter Status Error Messages.
3024	1 b	2 h	FIC alarm		FIC alarm type.

Table 1.4.12. VIB (Value Input Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
3100	1 b	1 d	Auto Lock		Indicates whether the block is locked into Auto mode.
3101	4 b	7 f	Field Value		Current input value.
3102	16 c	16 c	SIB Tag/Addr		Tag or address of the SIB block that is connected to the transmitter.
3103	16 c	16 c	Sim Link		Tag of the input block when the I/O block is in Simulate mode.
3104	1 b	1 d	SIB Input		The digital PV number to get from the transmitter.
3105	2 b	3 d	Hardware Alarm Code		FIC hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual*.*

Table 1.4.13. DIB (Discrete Input Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
3200	1 b	1 d	Auto Lock		Indicates whether the block is locked into Auto mode.
3201	1 b	1 d	Filtered State		Filtered state value.
3202	1 b	1 d	Raw State		Raw state value.
3203	1 b	1 d	Field Contact		Block input to output relationship.
				0	N.O.
				1	N.C.
3204	1 b	1 d	Filter Type		Signal characterization values.
				0	None
				1	D-On
				2	D-Off
				3	Delay
				4	Glitch
				5	Settle
				6	Extend
3205	4 b	7 f	Filt Time		Filter time constant applied to the field value.
3206	1 b	1 d	Fail safe Jumper		Input FIC jumper position.
				0	None
				1	On
				2	Off
3207	4 b	7 f	fim volt		FIM voltage value.
3208	1 b	1 d	Rising Edge		Triggers an event message.
				0	No
				1	Yes
3209	2 b	3 d	Event Type		Event type to be generated.
				0	None
				1	Advisory
				2	Critical
				3	Hardware

(continued on next page)

Table 1.4.13. DIB (Discrete Input Block) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
3210	1 b	1 d	When		Block action that generates the alarm or event message.
				0	None
				1	ON
				2	OFF
3211	1 b	1 d	Falling Edge		Triggers an event message.
				0	No
				1	Yes
3212	1 b	1 d	Alarm type		Alarm type to be generated.
				0	None
				1	Advisory
				2	Critical
				3	Hardware
3213	2 b	3 d	Hardware Alarm Code		FIC hardware fault. **For explanations of return status values for alarms, see the Alarm Messages Manual.**

Table 1.4.14. DOB (Discrete Output Block) Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
3300	1 b	1 d	Auto Lock		Indicates whether the block is locked into Auto Mode.
3301	16 c	16 c	Source Addr		Address of ControlBlock (data source).
3302	1 b	1 d	Contact Type		Block contact type.
				0	N.O.
				1	N.C.
3303	1 b	1 d	Source State		The field signal value at the input of the block.
3304	1 b	1 d	Output hold		Block input to output relationship.
				0	NONE
				1	On
				2	Off
				3	Pulse
3305	2 b	3 d	Hold Time		Signal characterization time in seconds.
3306	4 b	7 f	Fim Volt		FIM voltage.
3307	1 b	1 d	Fail Safe Jumper		Output FIC jumper position.
				0	Off
				1	Hold
3308	2 b	3 d	Hardware Alarm Code		FIC Hardware fault.

ControlBlock Field Codes

The following tables provide field codes for ControlBlocks.

Table 1.4.15	All ControlBlocks
Table 1.4.16	ControlBlock Input/Output Values
Table 1.4.17	PID (Process Integral Derivative) ControlBlocks
Table 1.4.18	LL (Lead/Lag) ControlBlock
Table 1.4.19	DT (Dead Time) ControlBlocks
Table 1.4.20	TOT (Stack Totalizer) and TOTSP (Setpoint Totalizer) ControlBlocks
Table 1.4.21	R/B (Ratio/Bias) ControlBlock
Table 1.4.22	SS (Signal Selector) ControlBlock
Table 1.4.23	VLIM (Velocity Limiter) ControlBlock
Table 1.4.24	POLY (Polynomial) ControlBlock
Table 1.4.25	PLI (Piecewise Linear Interpolator) ControlBlock
Table 1.4.26	MATH ControlBlock
Table 1.4.27	DMC (Discrete motor) ControlBlock
Table 1.4.28	DVC (Discrete valve) ControlBlock
Table 1.4.29	RBLC (RBL Controller) Block

Table 1.4.15. Field Codes for All Control Blocks

Code	Binary Format	ASCII Format	Field	Return Status	Description
1000	4 b	6 f	Sample Time		Time interval between block output updates.
1001	1 b	1 d	Rate Lim Inhibit		Rate limit inhibit flag.
1002	2 b	3 d	Function #		Discrete or continuous function values.
				000	MAN (Manual)
				001	PID (Proportional+Integral+Derivative)
				002	LL (Lead/Lag)
				003	DT (Dead Time)
				004	TOT (Stack Totalizer)
				005	RB (Ratio/Bias)
				006	SS (Signal Selector)
				007	VLIM (Velocity Limiter)
				008	POLY (7th Order Polynomial)
				009	PLI (Piecewise Linear Interpolator)
				010	Math (User-Defined Function)
				011	ATPID (Auto Tuning)
				012	RBL (Rosemount Basic Language)
				130	DISC (Discrete Block)
				131	DMC (Motor Controller)
				132	DASMC (Auto Sequence Motor Controller)
				133	DDSMC (Dual Speed Motor Controller)
				134	DDMC (Dual Direction Motor Controller)
				135	DVC (Valve Controller)
				136	DASVC (Auto Sequence Valve Controller)
				137	DMVC (Motorized Valve Controller)
				255	NONE

(continued on next page)

Table 1.4.15. Field Codes for All Control Blocks (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
1003	1 b	1 d	Auto Lock		Indicates if block is locked in Auto mode.
				0	No
				1	Yes
1004	1 b	1 d	Shed Mode		Block mode when the SCI Timer on the ControlFile Status screen expires.
				0	None
				1	Manual
				2	Auto
				3	Remote
1005	4 b	7 s	Output High Lim		High limit of output.
1006	4 b	7 s	Output Low Lim		Low limit of output.
1007	4 b	7 s	Output MN Rate Lim		Manual output rate-of-change limit. Applies only if the block mode is Local or Manual.
1008	4 b	7 s	OUT Hi Crit		High output value at which critical alarms are generated.
1009	4 b	7 s	OUT Lo Crit		Low output value at which critical alarms are generated.
1010	4 b	7 s	OUT Hi Adv		High output value at which advisory alarms are generated.
1011	4 b	7 s	OUT Lo Adv		Low output value at which advisory alarms are generated.
1012	4 b	7 s	OUT DdBand		Output alarm deadband range.
1013	1 b	1 d	SC		SCI write permission for this block.
				0	No
				1	Yes
1014	2 b 2 b	4 h 4 h	Output Flags 1 Output Flags 2		Output status flags group 1. Output status flags group 2.
1015	2 b	4 h	Status Flags		Status flags.
1016	2 b	4 h	Logic Mode Flag		Mode flags for the 16 logic steps.
1017	2 b	4 h	Logic Alm Flags		Alarm flags for the 16 logic steps.
1018	1 b	2 d	# of Analog In		Number of analog inputs.

(continued on next page)

Table 1.4.15. Field Codes for All Control Blocks (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
1019	1 b	2 d	# of Discrete In		Number of discrete inputs.
1020	1 b	1 d	DDC / CSP		DDC/CSP field value.
				0	CSP
				1	DDC

Table 1.4.16. Input/Output Field Codes for All ControlBlocks

Code	Binary Format	ASCII Format	Field	Description
1100 w	4 b	7 s	A Value	A continuous block value.
1101	2 b	4 h	A User Flags	Input user flags.
1102	2 b	4 h	A System Flags	Input system flags.
1103	16 c	16 c	A Source	Tag, address, or function of input source.
1104	4 b	7 s	A Hi Crit	High output value at which critical alarms are generated.
1105	4 b	7 s	A Lo Crit	Low output value at which critical alarms are generated.
1106	4 b	7 s	A Hi Adv	High output value at which advisory alarms are generated.
1107	4 b	7 s	A Lo Adv	Low output value at which advisory alarms are generated.
1108	4 b	7 s	A Rate	Maximum rate at which input can change before an alarm is generated.
1109	4 b	7 s	A DdBand	Input alarm range.
1110	4 b	7 f	A Eng Zero	Minimum range value for display scaling.
1111	4 b	7 f	A Eng Max	Maximum range value for display scaling.
1112	1 b	2 d	A Dec Point	Number of places to the right of the decimal point.
1113	8 c	8 c	A Units	Engineering units of measurement.
1114 w	4 b*	8 m	@a Value	Discrete input value or message pair text.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1115	16 c	16 c	@a Block Link	Discrete input link, if any, or 10 spaces.
1116	4 b*	8 m	a Flag	Discrete output value or message pair text.
1117	2 b	3 d	@a in Msg Pair	Message pair associated with the discrete input.
1118	2 b	3 d	a Flag Msg Pair	Message pair associated with the discrete output.
1120 w	4 b	7 s	B Value	A continuous block value.
1121	2 b	4 h	B User Flags	Input user flags.
1122	2 b	4 h	B System Flags	Input system flags.
1123	16 c	16 c	B Source	Tag, address, or function of input source.
1124	4 b	7 s	B Hi Crit	High output value at which critical alarms are generated.
1125	4 b	7 s	B Lo Crit	Low output value at which critical alarms are generated.
1126	4 b	7 s	B Hi Adv	High output value at which advisory alarms are generated.
1127	4 b	7 s	B Lo Adv	Low output value at which advisory alarms are generated.
1128	4 b	7 s	B Rate	Maximum rate at which input can change before an advisory alarm is generated.
1129	4 b	7 s	B DdBand	Input alarm range.
1130	4 b	7 f	B Eng Zero	Minimum range value for display scaling.
1131	4 b	7 f	B Eng Max	Maximum range value for display scaling.
1132	1 b	2 d	B Dec Point	Number of places to the right of the decimal point.
1133	8 c	8 c	B Units	Engineering units of measurement.
1134 w	4 b*	8 m	@b Value	Discrete input value or message pair text.
1135	16 c	16 c	@b Block Link	Discrete input link, if any, or 10 spaces.
1136	4 b*	8 m	b Flag	Discrete output value or message pair text.
1137	2 b	3 d	@b in Msg Pair	Message pair associated with the discrete input.
1138	2 b	3 d	b Flag Msg Pair	Message pair associated with the discrete output.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1140 w	4 b	7 s	C Value	A continuous block value.
1141	2 b	4 h	C User Flags	Input user flags.
1142	2 b	4 h	C System Flags	Input system flags.
1143	16 c	16 c	C Source	Tag, address, or function of input source.
1144	4 b	7 s	C Hi Crit	High output value at which critical alarms are generated.
1145	4 b	7 s	C Lo Crit	Low output value at which critical alarms are generated.
1146	4 b	7 s	C Hi Adv	High output value at which advisory alarms are generated.
1147	4 b	7 s	C Lo Adv	Low output value at which advisory alarms are generated.
1148	4 b	7 s	C Rate	Maximum rate at which input can change before an advisory alarm is generated.
1149	4 b	7 s	C DdBand	Input alarm range.
1150	4 b	7 f	C Eng Zero	Minimum range value for display scaling.
1151	4 b	7 f	C Eng Max	Maximum range value for display scaling.
1152	1 b	2 d	C Dec Point	Number of places to the right of the decimal point.
1153	8 c	8 c	C Units	Engineering units of measurement.
1154	4 b*	8 m	@c Value	Discrete input value or message pair text.
1155	16 c	16 c	@c Block Link	Discrete input link, if any, or 10 spaces.
1156	4 b*	8 m	c Flag	Discrete output value or message pair text.
1157	2 b	2 b	@c in Msg Pair	Message pair associated with the discrete input.
1158	2 b	3 d	c Flag Msg Pair	Message pair associated with the discrete output.
1160 w	4 b	7 s	D Value	A continuous block value.
1161	2 b	4 h	D User Flags	Input user flags.
1162	2 b	4 b	D System Flags	Input system flags.
1163	16 c	16 c	D Source	Tag, address, or function of input source.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1164	4 b	7 s	D Hi Crit	High output value at which critical alarms are generated.
1165	4 b	7 s	D Lo Crit	Low output value at which critical alarms are generated.
1166	4 b	7 s	D Hi Adv	High output value at which advisory alarms are generated.
1167	4 b	7 s	D Lo Adv	Low output value at which advisory alarms are generated.
1168	4 b	7 s	D Rate	Maximum rate at which input can change before an advisory alarm is generated.
1169	4 b	7 s	D DdBand	Input alarm range.
1170	4 b	7 f	D Eng Zero	Minimum range value for display scaling.
1171	4 b	7 f	D Eng Max	Maximum range value for display scaling.
1172	1 b	2 d	D Dec Point	Number of places to the right of the decimal point.
1173	8 c	8 c	D Units	Engineering units of measurement.
1174	4 b*	8 m	@d Value	Discrete input value or message pair text.
1175	16 c	16 c	@d Block Link	Discrete input link, if any, or 10 spaces.
1176	4 b*	8 m	d Flag	Discrete output value or message pair text.
1177	2 b	3 d	@d in Msg Pair	Message pair associated with the discrete output.
1178	2 b	3 d	d Flag Msg Pair	Message pair associated with the discrete output.
1180	4 b	7 s	E Value	A continuous block value.
1181	2 b	4 h	E User Flags	Input user flags.
1182	2 b	4 h	E System Flags	Input system flags.
1183	16 c	16 c	E Source	Tag, address, or function of input source.
1184	4 b	7 s	E Hi Crit	High output value at which critical alarms are generated.
1185	4 b	7 s	E Lo Crit	Low output value at which critical alarms are generated.
1186	4 b	7 s	E Hi Adv	High output value at which advisory alarms are generated.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1187	4 b	7 s	E Lo Adv	Low output value at which advisory alarms are generated.
1188	4 b	7 s	E Rate	Maximum rate at which input can change before an advisory alarm is generated.
1189	4 b	7 s	E DdBand	Input alarm range.
1190	4 b	7 s	E Eng Zero	Minimum range value for display scaling.
1191	4 b	7 s	E Eng Max	Maximum range value for display scaling.
1192	1 b	2 d	E Dec Point	Number of places to the right of the decimal point.
1193	8 c	8 c	E Units	Engineering units of measurement.
1194	4 b*	8 m	@e Value	Discrete input value or message pair text.
1195	16 c	16 c	@e Block Link	Discrete input link, if any, or 10 spaces.
1196	4 b*	8 m	e Flag	Discrete output value or message pair text.
1197	2 b	3 d	@e in Msg Pair	Message pair associated with the discrete input.
1198	2 b	3 d	e Flag Msg Pair	Message pair associated with the discrete output.
1200 w	4 b	7 s	F Value	A continuous block value.
1201	2 b	4 h	F User Flags	Input user flags.
1202	2 b	4 h	F System Flags	Input system flags.
1203	2 b	16 c	F Source	Tag, address, or function of input source.
1204	4 b	7 s	F Hi Crit	High output value at which critical alarms are generated.
1205	4 b	7 s	F Lo Crit	Low output value at which critical alarms are generated.
1206	4 b	7 s	F Hi Adv	High output value at which advisory alarms are generated.
1207	4 b	7 s	F Lo Adv	Low output value at which advisory alarms are generated.
1208	4 b	7 s	F Rate	Maximum rate at which input can change before an advisory alarm is generated.
1209	4 b	7 s	F DdBand	Input alarm range.
1210	4 b	7 f	F Eng Zero	Minimum range value for display scaling.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1211	4 b	7 f	F Eng Max	Maximum range value for display scaling.
1212	1 b	2 d	F Dec Point	Number of places to the right of the decimal point.
1213	8 c	8 c	F Units	Engineering units of measurement.
1214 w	4 b*	8 m	@f Value	Discrete input value or message pair text.
1215	16 c	16 c	@f Block Link	Discrete input link, if any, or 10 spaces.
1216	4 b*	8 m	f Flag	Discrete output value or message pair text.
1217	2 b	3 d	@f in Msg Pair	Message pair associated with the discrete input.
1218	2 b	3 d	f Flag Msg Pair	Message pair associated with the discrete output.
1220	4 b	7 s	G Value	A continuous block value.
1221	2 b	4 h	G User Flags	Input user flags.
1222	2 b	4 h	G System Flags	Input system flags.
1223	16 c	16 c	G Source	Tag, address, or function of input source.
1224	4 b	7 s	G Hi Crit	High output value at which critical alarms are generated.
1225	4 b	7 s	G Lo Crit	Low output value at which critical alarms are generated.
1226	4 b	7 s	G Hi Adv	High output value at which advisory alarms are generated.
1227	4 b	7 s	G Lo Adv	Low output value at which advisory alarms are generated.
1228	4 b	7 s	G Rate	Maximum rate at which input can change before an advisory alarm is generated.
1229	4 b	7 s	G DdBand	Input alarm range.
1230	4 b	7 f	G Eng Zero	Minimum range value for display scaling.
1231	4 b	7 f	G Eng Max	Maximum range value for display scaling.
1232	1 b	2 d	G Dec Point	Number of places to the right of the decimal point.
1233	8 c	8 c	G Units	Engineering units of measurement.
1234 w	4 b*	8 m	@g Value	Discrete input value or message pair text.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1235	16 c	16 c	@g Block Link	Discrete input link, if any, or 10 spaces.
1236	4 b*	8 m	g Flag	Discrete output value or message pair text.
1237	2 b	3 d	@g in Msg Pair	Message pair associated with the discrete input.
1238	2 b	3 d	g Flag Msg Pair	Message pair associated with the discrete output.
1240 w	4 b	7 s	H Value	A continuous block value.
1241	2 b	4 h	H User Flags	Input user flags.
1242	2 b	4 h	H System Flags	Input system flags.
1243	16 c	16 c	H Source	Tag, address, or function of input source.
1244	4 b	7 s	H Hi Crit	High output value at which critical alarms are generated.
1245	4 b	7 s	H Lo Crit	Low output value at which critical alarms are generated.
1246	4 b	7 s	H Hi Adv	High output value at which advisory alarms are generated.
1247	4 b	7 s	H Lo Adv	Low output value at which advisory alarms are generated.
1248	4 b	7 s	H Rate	Maximum rate at which input can change before an advisory alarm is generated.
1249	4 b	7 s	H DdBand	Input alarm range.
1250	4 b	7 f	H Eng Zero	Minimum range value for display scaling.
1251	4 b	7 f	H Eng Max	Maximum range value for display scaling.
1252	1 b	2 d	H Dec Point	Number of places to the right of the decimal point.
1253	8 c	8 c	H Units	Engineering units of measurement.
1254 w	4 b*	8 m	@h Value	Discrete input value or message pair text.
1255	16 c	16 c	@h Block Link	Discrete input link, if any, or 10 spaces.
1256	4 b*	8 m	h Flag	Discrete output value or message pair text.
1257	2 b	3 d	@h in Msg Pair	Message pair associated with the discrete input.
1258	2 b	3 d	h Flag Msg Pair	Message pair associated with the discrete output.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1260 w	4 b	7 s	I Value	A continuous block value.
1261	2 b	4 h	I User Flags	Input user flags.
1262	2 b	4 h	I System Flags	Input system flags.
1263	16 c	16 c	I Source	Tag, address, or function of input source.
1264	4 b	7 s	I Hi Crit	High output value at which critical alarms are generated.
1265	4 b	7 s	I Lo Crit	Low output value at which critical alarms are generated.
1266	4 b	7 s	I Hi Adv	High output value at which advisory alarms are generated.
1267	4 b	7 s	I Lo Adv	Low output value at which advisory alarms are generated.
1268	4 b	7 s	I Rate	Maximum rate at which input can change before an advisory alarm is generated.
1269	4 b	7 s	I DdBand	Input alarm range.
1270	4 b	7 f	I Eng Zero	Minimum range value for display scaling.
1271	4 b	7 f	I Eng Max	Maximum range value for display scaling.
1272	1 b	2 d	I Dec Point	Number of places to the right of the decimal point.
1273	8 c	8 c	I Units	Engineering units of measurement.
1274 w	4 b*	8 m	@i Value	Discrete input value or message pair text.
1275	16 c	16 c	@i Block Link	Discrete input link, if any, or 10 spaces.
1276	4 b*	8 m	i Flag	Discrete output value or message pair text.
1277	2 b	3 d	@i in Msg Pair	Message pair associated with the discrete input.
1278	2 b	3 d	i Flag Msg Pair	Message pair associated with the discrete output.
1280 w	4 b	7 s	J Value	A continuous block value.
1281	2 b	4 h	J User Flags	Input user flags.
1282	2 b	4 h	J System Flags	Input system flags.
1283	16 c	16 c	J Source	Tag, address, or function of input source.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1284	4 b	7 s	J Hi Crit	High output value at which critical alarms are generated.
1285	4 b	7 s	J Lo Crit	Low output value at which critical alarms are generated.
1286	4 b	7 s	J Hi Adv	High output value at which advisory alarms are generated.
1287	4 b	7 s	J Lo Adv	Low output value at which advisory alarms are generated.
1288	4 b	7 s	J Rate	Maximum rate at which input can change before an advisory alarm is generated.
1289	4 b	7 s	J DdBand	Input alarm range.
1290	4 b	7 f	J Eng Zero	Minimum range value for display scaling.
1291	4 b	7 f	J Eng Max	Maximum range value for display scaling.
1292	1 b	2 d	J Dec Point	Number of places to the right of the decimal point.
1293	8 c	8 c	J Units	Engineering units of measurement.
1294 w	4 b*	8 m	@j Value	Discrete input value or message pair text.
1295	16 c	16 c	@j Block Link	Discrete input link, if any, or 10 spaces.
1296	4 b*	8 m	j Flag	Discrete output value or message pair text.
1297	2 b	3 d	@j in Msg Pair	Message pair associated with the discrete input.
1298	2 b	3 d	j Flag Msg Pair	Message pair associated with the discrete output.
1300 w	4 b	7 s	K Value	A continuous block value.
1301	2 b	4 h	K User Flags	Input user flags.
1302	2 b	4 h	K System Flags	Input system flags.
1303	16 c	16 c	K Source	Tag, address, or function of input source.
1304	4 b	7 s	K Hi Crit	High output value at which critical alarms are generated.
1305	4 b	7 s	K Lo Crit	Low output value at which critical alarms are generated.
1306	4 b	7 s	K Hi Adv	High output value at which advisory alarms are generated.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1307	4 b	7 s	K Lo Adv	Low output value at which advisory alarms are generated.
1308	4 b	7 s	K Rate	Maximum rate at which input can change before an advisory alarm is generated.
1309	4 b	7 s	K DdBand	Input alarm range.
1310	4 b	7 f	K Eng Zero	Minimum range value for display scaling.
1311	4 b	7 f	K Eng Max	Maximum range value for display scaling.
1312	1 b	2 d	K Dec Point	Number of places to the right of the decimal point.
1313	8 c	8 c	K Units	Engineering units of measurement.
1314 w	4 b*	8 m	@k Value	Discrete input value or message pair text.
1315	16 c	16 c	@k Block Link	Discrete input link, if any, or 10 spaces.
1316	4 b*	8 m	k Flag	Discrete output value or message pair text.
1317	2 b	3 d	@k in Msg Pair	Message pair associated with the discrete input.
1318	2 b	3 d	k Flag Msg Pair	Message pair associated with the discrete output.
1320 w	4 b	7 s	L Value	A continuous block value.
1321	2 b	4 h	L User Flags	Input user flags.
1322	2 b	4 h	L System Flags	Input system flags.
1323	16 c	16 c	L Source	Tag, address, or function of input source.
1324	4 b	7 s	L Hi Crit	High output value at which critical alarms are generated.
1325	4 b	7 s	L Lo Crit	Low output value at which critical alarms are generated.
1326	4 b	7 s	L Hi Adv	High output value at which advisory alarms are generated.
1327	4 b	7 s	L Lo Adv	Low output value at which advisory alarms are generated.
1328	4 b	7 s	L Rate	Maximum rate at which input can change before an advisory alarm is generated.
1329	4 b	7 s	L DdBand	Input alarm range.
1330	4 b	7 f	L Eng Zero	Minimum range value for display scaling.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1331	4 b	7 f	L Eng Max	Maximum range value for display scaling.
1332	1 b	2 d	L Dec Point	Number of places to the right of the decimal point.
1333	8 c	8 c	L Units	Engineering units of measurement.
1334 w	4 b*	8 m	@l Value	Discrete input value or message pair text.
1335	16 c	16 c	@l Block Link	Discrete input link, if any, or 10 spaces.
1336	4 b*	8 m	l Flag	Discrete output value or message pair text.
1337	2 b	3 d	@l in Msg Pair	Message pair associated with the discrete input.
1338	2 b	3 d	l Flag Msg Pair	Message pair associated with the discrete output.
1340 w	4 b	7 s	M Value	A continuous block value.
1341	2 b	4 h	M User Flags	Input user flags.
1342	2 b	4 h	M System Flags	Input system flags.
1343	16 c	16 c	M Source	Tag, address, or function of input source.
1344	4 b	7 s	M Hi Crit	High output value at which critical alarms are generated.
1345	4 b	7 s	M Lo Crit	Low output value at which critical alarms are generated.
1346:	4 b	7 s	M Hi Adv	High output value at which advisory alarms are generated.
1347	4 b	7 s	M Lo Adv	Low output value at which advisory alarms are generated.
1348	4 b	7 s	M Rate	Maximum rate at which input can change before an advisory alarm is generated.
1349	4 b	7 s	M DdBand	Input alarm range.
1350	4 b	7 f	M Eng Zero	Minimum range value for display scaling.
1351	4 b	7 f	M Eng Max	Maximum range value for display scaling.
1352	1 b	2 d	M Dec Point	Number of places to the right of the decimal point.
1353	8 c	8 c	M Units	Engineering units of measurement.
1354 w	4 b*	8 m	@m Value	Discrete input value or message pair text.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1355	16 c	16 c	@m Block Link	Discrete input link, if any, or 10 spaces.
1356	4 b*	8 m	m Flag	Discrete output value or message pair text.
1357	2 b	3 d	@m in Msg Pair	Message pair associated with the discrete input.
1358	2 b	3 d	m Flag Msg Pair	Message pair associated with the discrete output.
1360 w	4 b	7 s	N Value	A continuous block value.
1361	2 b	4 h	N User Flags	Input user flags.
1362	2 b	4 h	N System Flags	Input system flags.
1363	16 c	16 c	N Source	Tag, address, or function of input source.
1364	4 b	7 s	N Hi Crit	High output value at which critical alarms are generated.
1365	4 b	7 s	N Lo Crit	Low output value at which critical alarms are generated.
1366	4 b	7 s	N Hi Adv	High output value at which advisory alarms are generated.
1367	4 b	7 s	N Lo Adv	Low output value at which advisory alarms are generated.
1368	4 b	7 s	N Rate	Maximum rate at which input can change before an advisory alarm is generated.
1369	4 b	7 s	N DdBand	Input alarm range.
1370	4 b	7 f	N Eng Zero	Minimum range value for display scaling.
1371	4 b	7 f	N Eng Max	Maximum range value for display scaling.
1372	1 b	2 d	N Dec Point	Number of places to the right of the decimal point.
1373	8 c	8 c	N Units	Engineering units of measurement.
1374 w	4 b*	8 m	@n Value	Discrete input value or message pair text.
1375	16 c	16 c	@n Block Link	Discrete input link, if any, or 10 spaces.
1376	4 b*	8 m	n Flag	Discrete output value or message pair text.
1377	2 b	3 d	@n in Msg Pair	Message pair associated with the discrete input.
1378	2 b	3 d	n Flag Msg Pair	Message pair associated with the discrete output.

* If this value is a scaled discrete message pair, then the data format is 8 m.

(continued on next page)

Table 1.4.16. Input/Output Field Codes for All ControlBlocks (continued)

Code	Binary Format	ASCII Format	Field	Description
1380 w	4 b	7 s	O Value	A continuous block value.
1381	2 b	4 h	O User Flags	Input user flags.
1382	2 b	4 h	O System Flags	Input system flags.
1383	16 c	16 c	O Source	Tag, address, or function of input source.
1384	4 b	7 s	O Hi Crit	High output value at which critical alarms are generated.
1385	4 b	7 s	O Lo Crit	Low output value at which critical alarms are generated.
1386	4 b	7 s	O Hi Adv	High output value at which advisory alarms are generated.
1387	4 b	7 s	O Lo Adv	Low output value at which advisory alarms are generated.
1388	4 b	7 s	O Rate	Maximum rate at which input can change before an advisory alarm is generated.
1389	4 b	7 s	O DdBand	Input alarm range.
1390	4 b	7 f	O Eng Zero	Minimum range value for display scaling.
1391	4 b	7 f	O Eng Max	Maximum range value for display scaling.
1392	1 b	2 d	O Dec Point	Number of places to the right of the decimal point.
1393	8 c	8 c	O Units	Engineering units of measurement.
1394 w	4 b*	8 m	@o Value	Discrete input value or message pair text.
1395	16 c	16 c	@o Block Link	Discrete input link, if any, or 10 spaces.
1396	4 b*	8 m	o Flag	Discrete output value or message pair text.
1397	2 b	3 d	@o in Msg Pair	Message pair associated with the discrete input.
1398	2 b	3 d	o Flag Msg Pair	Message pair associated with the discrete output.
1416	4 b*	8 m	p Flag	Discrete output value or message pair text.
1418	2 b	3 d	p Flag Msg Pair	Message pair associated with the discrete output.

* If this value is a scaled discrete message pair, then the data format is 8 m.

Table 1.4.17. PID ControlBlock Field Codes

Code	Binary Format	ASCII Format	Field	Return Status	Description
1500	1 b	1 d	PI Act		Parameter that P term uses in P algorithm:
				5	SP indicates that P=K(SP)
				6	PV indicates that P=K(PV)
				7	Err indicates that P=K(SP-PV)
1501	1 b	1 d	D Act		Parameter that D term uses in algorithm:
				1	SP indicates that D=KΔSP
				2	PV indicates that D=KΔPV
				3	Err indicates that D=KΔ(SP-PV)
1502	1 b	1 d	Ratio/Bias or FF		Ratio / Bias (0) or FF (1).
1503	4 b	7 s	Gap Value		Output does not change until calculation change is greater than the gap value.
1504	4 b	7 s	Err DdBand		Output does not change until value of error is greater than the deadband value.
1505	1 b	1 d	Opt		Error squared action flag.
1506	1 b	1 d	Action		Action of controller output (Q).
				0	Reverse. Decrease in PV results in increase in Q.
				1	Direct. Default condition.
1507	1 b	1 d	Tracking Value		Setpoint / measured variable tracking value.
				0	No
				1	Yes
1508	1 b	2 d	Track Input		Input source of tracking signals: 0=None 1=A 4=D 7=G 10=J 13=M 2=B 5=E 8=H 11=K 14=N 3=C 6=F 9=I 12=L 15=O
1509	1 b	1 d	Opt		Velocity Flag
1510	4 b	7 s	Setpoint Low Lim		Local setpoint low limit. Units are same as the local setpoint.
1511	4 b	7 s	Setpoint High Lim		Local setpoint high limit. Units are same as the local setpoint.

(continued on next page)

Table 1.4.17. PID ControlBlock Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Return Status	Description
1512	4 b	7 s	Setpoint Rate Lim		Local setpoint rate limit in units/seconds. Units are same as the local setpoint.
1513	1 b	1 d	Ratio/Bias Limit Type		Ratio/Bias limits apply to:
				0	ratio limits
				1	bias limits
1514	4 b	7 s	Ratio/Bias Low Lim		Ratio/Bias low limit value.
1515	4 b	7 s	Ratio/Bias High Lim		Ratio/Bias high limit value.
1516	4 b	7 s	Ratio/Bias Rate Lim		Maximum rate at which Ratio/Bias can be changed, in Units/Seconds.
1517	4 b	7 f	Prop Gain		Proportional Band or Controller Gain (Prop Band of 100)=(Controller Gain of 1).
1518	4 b	7 f	Integ Time		Integral action time constant (reset time).
1519	4 b	7 f	Deriv Time		Derivative action time constant (rate time).
1520	4 b	7 f	Derivative filter time		Derivative filter time in seconds.
1521	4 b	7 f	FF Gain		Feed forward term gain.
1522	4 b	7 s	Dev Crit		Dev critical alarm value.
1523	4 b	7 s	Dev Adv		Dev advisory alarm value.

Table 1.4.18. Lead/Lag ControlBlock (LL) Field Codes

Code	Binary Format	ASCII Format	Field	Description
1600	1 b	2 d	Track Input	Input source of tracking signal: 0=None 1=A 4=D 7=G 10=J 13=M 2=B 5=E 8=H 11=K 14=N 3=C 6=F 9=I 12=L 15=O
1601	4 b	7 f	Filter Gain KA	Gain on input A (PV).
1602	4 b	7 f	Bias Gain KB	Gain on input B (Bias).
1603	4 b	7 f	Offset Gain KC	Gain on input C (offset).
1604	4 b	7 f	Lead Time	Lead time constant.
1605	4 b	7 f	Lag Time 1	First-order lag time constant.
1606	4 b	7 f	Lag Time 2	Second-order lag time constant.

Table 1.4.19. Dead Time ControlBlock (DT) Field Codes

Code	Binary Format	ASCII Format	Field	Description
1700	4 b	7 f	Gain KA	Gain on input A.
1701	4 b	7 f	Gain KC	Gain on input C.

Table 1.4.20. Stack Totalizer ControlBlock (TOT) and Setpoint Totalizer ControlBlock (TOTSP) Field Codes

Code	Binary Format	ASCII Format	Field	Description
1800	1 b	1 d	Totalizer Type	Type of totalizer: 0=setpoint totalizer (TOTSP) 1=stack totalizer (TOT)
1801	4 b	7 f	Gain K	Gain of input A (PV).
1802	4 b	7 s	Low Cutoff	Low Cutoff value.
1803	1 b	1 d	Cutoff Type	Values that are totalized: 0=Signed 1=Band
1804	4 b	7 f	Integ Time	Integration time it takes for output to go from 0% to 100% when the input is at 100%.

Table 1.4.21. Ratio/Bias ControlBlock (R/B) Field Codes

Code	Binary Format	ASCII Format	Field	Description
1900	1 b	2 d	Track Input	Input source of tracking signal. 0=None 1=A 4=D 7=G 10=J 13=M 2=B 5=E 8=H 11=K 14=N 3=C 6=F 9=I 12=L 15=O
1901	4 b	7 s	Pre-Bias	Pre-Bias term of Ratio/Bias equation.
1902	4 b	7 f	Ratio Gain	Ratio gain term of Ratio/Bias equation.
1903	4 b	7 s	Ratio Low Lim	Ratio low limit value.
1904	4 b	7 s	Ratio High Lim	Ratio high limit value.
1905	4 b	7 s	Ratio Rate Lim	Maximum rate at which ratio value can be changed.
1906	4 b	7 s	Bias Low Lim	Bias low limit value.
1907	4 b	7 s	Bias High Lim	Bias high limit value.
1908	4 b	7 s	Bias Rate Lim	Maximum rate at which bias value can be changed.

Table 1.4.22. Signal Selector ControlBlock (SS) Field Codes

Code	Binary Format	ASCII Format	Field	Description
2000	1 b	2 d	No of Inputs	Number of inputs used by the selector.
2001	1 b	2 d	Select Number	The nth highest value to selected.
2002	4 b	7 s	Hysteresis	Hysteresis value.
2003	1 b	2 d	A Track Input	Value that the A input tracks. 0=None 1=A 4=D 7=G 10=J 13=M 2=B 5=E 8=H 11=K 14=N 3=C 6=F 9=I 12=L 15=O
2004	1 b	2 d	B Track Input	Value that the B input tracks. Refer to code 2003.
2005	1 b	2 d	C Track Input	Value that the C input tracks. Refer to code 2003.
2006	1 b	2 d	D Track Input	Value that the D input tracks. Refer to code 2003.
2007	1 b	2 d	E Track Input	Value that the E input tracks. Refer to code 2003.
2008	1 b	2 d	F Track Input	Value that the F input tracks. Refer to code 2003.
2009	1 b	2 d	G Track Input	Value that the G input tracks. Refer to code 2003.
2010	1 b	2 d	H Track Input	Value that the H input tracks. Refer to code 2003.
2011	1 b	2 d	I Track Input	Value that the I input tracks. Refer to code 2003.
2012	1 b	2 d	J Track Input	Value that the J input tracks. Refer to code 2003.
2013	1 b	2 d	K Track Input	Value that the K input tracks. Refer to code 2003.
2014	1 b	2 d	L Track Input	Value that the L input tracks. Refer to code 2003.
2015	1 b	2 d	M Track Input	Value that the M input tracks. Refer to code 2003.
2016	1 b	2 d	N Track Input	Value that the N input tracks. Refer to code 2003.
2017	1 b	2 d	O Track Input	Value that the O input tracks. Refer to code 2003.
2018	1 b	2 d	P Track Input	Value that the P input tracks. Refer to code 2003.

Table 1.4.23. Velocity Limiter ControlBlock (VLIM) Field Codes

Code	Binary Format	ASCII Format	Field	Description
2100	1 b	1 d	Track PV	Back tracking selection for the PV (A) input. 0=no 1=yes
2101	4 b	7 s	Entry Lo Lim	Entry low limit.
2102	4 b	7 s	Entry Hi Lim	Entry high limit.
2103	4 b	7 s	Rise Limit	Normal rise rate limit.
2104	4 b	7 s	Rise Hi Lim	Rise rate limit when deviation trigger is exceeded.
2105	4 b	7 s	Fall Limit	Normal fall rate limit.
2106	4 b	7 s	Fall Hi Lim	Fall rate limit when deviation trigger is exceeded.
2107	4 b	7 s	Rise Dev Trigger	Rise deviation trigger value.
2108	4 b	7 s	Fall Dev Trigger	Fall deviation trigger value.

Table 1.4.24. Polynomial ControlBlock (POLY) Field Codes

Code	Binary Format	ASCII Format	Field	Description
2200	4 b	7 f	Max Gain	Determines rate of response of tracking signal.
2201	1 b	2 d	Track Input	Input source of tracking signal: 0=None 1=A 4=D 7=G 10=J 13=M 2=B 5=E 8=H 11=K 14=N 3=C 6=F 9=I 12=L 15=O
2202	4 b	7 f	K0	K0 value.
2203	4 b	7 f	K1	K1 value.
2204	4 b	7 f	K2	K2 value.
2205	4 b	7 f	K3	K3 value.
2206	4 b	7 f	K4	K4 value.
2207	4 b	7 f	K5	K5 value.
2208	4 b	7 f	K6	K6 value.
2209	4 b	7 f	K7	K7 value.

Table 1.4.25. Piecewise Linear Interpolator ControlBlock (PLI) Field Codes

Code	Binary Format	ASCII Format	Field	Description
2300	4 b	7 f	Max gain	Determines rate of response of the tracking signal.
2301	1 b	2 d	Track Input	Input source of tracking signal: 0=None 1=A 4=D 7=G 10=J 13=M 2=B 5=E 8=H 11=K 14=N 3=C 6=F 9=I 12=L 15=O
2302	4 b	7 f	PV 1	PV Term 1
2303	4 b	7 s	OUT 1	Out term 1
2304	4 b	7 f	PV 2	PV Term 2
2305	4 b	7 s	OUT 2	Out term 2
2306	4 b	7 f	PV 3	PV Term 3
2307	4 b	7 s	OUT 3	Out term 3
2308	4 b	7 f	PV 4	PV Term 4
2309	4 b	7 s	OUT 4	Out Term 4
2310	4 b	7 f	PV 5	PV Term 5
2311	4 b	7 s	OUT 5	Out term 5
2312	4 b	7 f	PV 6	PV Term 6
2313	4 b	7 s	OUT 6	Out Term 6
2314	4 b	7 f	PV 7	PV Term 7
2315	4 b	7 s	OUT 7	Out term 7
2316	4 b	7 f	PV 8	PV Term 8
2317	4 b	7 s	OUT 8	Out Term 8
2318	4 b	7 f	PV 9	PV Term 9
2319	4 b	7 s	OUT 9	Out Term 9
2320	4 b	7 f	PV 10	PV Term 10
2321	4 b	7 s	OUT 10	Out Term 10
2322	4 b	7 f	PV 11	PV Term 11
2323	4 b	7 s	OUT 11	Out Term 11

(continued on next page)

Table 1.4.25. Piecewise Linear Interpolator ControlBlock (PLI) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Description
2324	4 b	7 f	PV 12	PV Term 12
2325	4 b	7 s	OUT 12	Out Term 12
2326	4 b	7 f	PV 13	PV Term 13
2327	4 b	7 s	OUT 13	Out Term 13
2328	4 b	7 f	PV 14	PV Term 14
2329	4 b	7 s	OUT 14	Out Term 14
2330	4 b	7 f	PV 15	PV Term 15
2331	4 b	7 s	OUT 15	Out Term 15
2332	4 b	7 f	PV 16	PV Term 16
2333	4 b	7 s	OUT 16	Out Term 16
2334	4 b	7 f	PV 17	PV Term 17
2335	4 b	7 s	OUT 17	Out Term 17
2336	4 b	7 f	PV 18	PV Term 18
2337	4 b	7 s	OUT 18	Out Term 18
2338	4 b	7 f	PV 19	PV Term 19
2339	4 b	7 s	OUT 19	Out Term 19
2340	4 b	7 f	PV 20	PV Term 20
2341	4 b	7 s	OUT 20	Out term 20
2342	4 b	7 f	PV 21	PV Term 21
2343	4 b	7 s	OUT 21	Out Term 21
2344	4 b	7 f	PV 22	PV Tern 22
2345	4 b	7 s	OUT 22	Out Term 22
2346	4 b	7 f	PV 23	PV Term 23
2347	4 b	7 s	OUT 23	Out term 23
2348	4 b	7 f	PV 24	PV Term 24
2349	4 b	7 s	OUT 24	Out term 24
2350	4 b	7 f	PV 25	PV Term 25

(continued on next page)

Table 1.4.25. Piecewise Linear Interpolator ControlBlock (PLI) Field Codes (continued)

Code	Binary Format	ASCII Format	Field	Description
2351	4 b	7 s	OUT 25	Out term 25
2352	4 b	7 f	PV 26	PV Term 26
2353	4 b	7 s	OUT 26	Out term 26
2354	4 b	7 f	PV 27	PV Term 27
2355	4 b	7 s	OUT 27	Out Term 27
2356	4 b	7 f	PV 28	PV Term 28
2357	4 b	7 s	OUT 28	Out Term 28
2358	4 b	7 f	PV 29	PV Term 29
2359	4 b	7 s	OUT 29	Out term 29
2360	4 b	7 f	PV 30	PV Term 30
2361	4 b	7 s	OUT 30	Out Term 30

Table 1.4.26. MATH ControlBlock Field Codes

Code	Binary Format	ASCII Format	Field	Description
2400	1 b	1 d	Tracking Type	Max gain back calculate (0) for inverse function (1).
2401	4 b	7 f	Max Gain	Determines rate of response of the tracking signal.
2402	1 b	2 d	Track Input	Input source of tracking signal: 0=None 1=A 4=D 7=G 10=J 13=M 2=B 5=E 8=H 11=K 14=N 3=C 6=F 9=I 12=L 15=O

Table 1.4.27. Discrete Motor ControlBlock Field Codes

Code	Binary Format	ASCII Format	Field	Description
2500	4 b	7 f	Conf On 1 Time	Confirm 1 time.
2501	4 b	7 f	Conf Off Time	Confirm off time.
2502	4 b	7 f	Conf On 2 Time	Confirm 2 time.
2503	4 b	7 f	Lockout Time	Lockout time.
2504	1 b	1 d	Dual Speed	Dual Speed motor.
2505	1 b	1 d	Auto Seq Enable	Auto sequence enabled.
2506	1 b	1 d	Interlock Enable	Interlock enabled.
2507	1 b	1 d	Security Lockup	Motor lockup on security.
2508	1 b	1 d	Stop Enab Conf	Enable confirm for stop.
2509	1 b	1 d	Motor Check	Motor control center check.
2510	1 b	1 d	Motor Retries	Motor start retries enabled.
2511	1 b	1 d	Immed Stop	Motor immediate stop enabled.
2512	1 b	1 d	Ignore Cnf on 1	Ignore confirm 1.
2513	1 b	1 d	Ignore Conf Off	Ignore confirm off.
2514	1 b	1 d	Ignore Cnf on 2	Ignore confirm 2.
2515	1 b	1 d	Ignore Interlock	Ignore interlock signal.
2516	1 b	1 d	Dropout Delay	Confirm dropout delay selected.
2517	2 b	3 d	# of Retries	Number of retries.

Table 1.4.28. Discrete Valve ControlBlock Field Codes

Code	Binary Format	ASCII Format	Field	Description
2600	4 b	7 f	Open Conf Time	Open confirm time.
2601	4 b	7 f	Close Conf Time	Close confirm time.
2602	4 b	7 f	Auto Seq Time	Auto sequence time.
2603	1 b	1 d	Auto Seq Enable	Auto sequence enable.
2604	1 b	1 d	Interlock Enable	Interlock enabled.
2605	1 b	1 d	Security Lockup	Motor lockup on security.
2606	1 b	1 d	Enab Open Conf	Enable confirm for open.
2607	1 b	1 d	Enab Close Conf	Enable confirm for close.
2608	1 b	1 d	Ign Open Conf	Ignore open confirm signal.
2609	1 b	1 d	Ign Close Conf	Ignore close confirm signal.
2610	1 b	1 d	Ignore Interlock	Ignore interlock signal.
2611	1 b	1 d	Motorized Valve	Motorized valve selected.

Table 1.4.29. RBL Controller Fields

Code	Binary Format	ASCII Format	Field	Description
2700	10 c	10 c	Volume Name	Drive volume that contains the RBL programs.
2701	9 c	9 c	File Name	File that contains the RBL script.
2702	9 c	9 c	Script Name	Script that contains the RBL program.
2703	10 c	10 c	Backup Volume Name	Backup drive volume that contains the RBL file and script.

Section 5:
Sample Instructions

This section contains samples of host computer instructions used to communicate with the SCI and the RS3 control system.

NOTE:

- These instructions are samples only and may not be valid for all applications.
- An underscore (_) denotes one blank character.

Sample Program Excerpts

This section includes the following samples of excerpts from programs:

- Opening and initializing the host communication port
- Reading SCI status
- Reading data from a controller processor
- Writing to a controller processor
- Reading from the SCI table
- Writing to the SCI table

The samples are based on the following configuration:

- 9600 baud rate
- Data Bits and Parity⇒7M
- EIA⇒None
- Checksum⇒No
- Header of linefeed + (”, a trailer of “) + carriage return

The samples are written in IBM™ BASICA™.

Opening and Initializing the Host Communication Port

The following instructions open and initialize the host serial port.

```
/*==============================================================*/
125   SCREEN 0,0: KEY OFF: CLS

130   OPEN ”COM1: 9600,M,7,1,CS0,DS0,CD0” AS #1
```

Open port through which the host communicates with the SCI (COM1) and configure as follows: 9600 BPS, mark parity, 7 data bits, 1 stop bit, ignore CTS, DSR, and DCD. The port is assigned to BASICA file #1.

```
132   END
/*==============================================================*/
```

Reading SCI Status

The following instructions use Message Type 81 to request status from the SCI, read the status reply message, and print it to the screen.

```
/*==============================================================*/
125   SCREEN 0,0: KEY OFF: CLS

130   OPEN ”COM1: 9600,M,7,1,CS0,DS0,CD0” AS #1

133   PRINT ”Requesting SCI Status”

135   PRINT ”Sending => 0081”

140   PRINT #1,”0081”
```

Send the SCI Status message (ID = 00 and type = 81) to Comp file #1 in the SCI. This request is entered on a terminal as ”0081.”

```
150   LINE INPUT #1, REPLY$
```

Read the SCI Status reply message from Comp file #1 and assign it to the string variable REPLY$. REPLY$ contains all reply data up to the carriage return at the end of the message.

```
160   PRINT ”Received => ”+REPLY$
```

Prints the reply message to the screen.

```
170   PRINT : PRINT ”Hit any key to cont”: PRINT

180   IF (INKEY$=”) GOTO 180

190   END
/*==============================================================*/
```

Reading Data from a Controller Processor

The following instructions use message type 21 to read data from a Controller Processor through the SCI.

```
/*===============================================================*/
1080  SCREEN 0,0: KEY OFF: CLS
1090  OPEN "COM1: 9600,M,7,1,CS0,DS0,CD0" AS #1
1105  CLS
1110  ITEMS$="0000,1100,1120,1140,1160,0006,1114,1134,
      1154,1174,"
```

Assign the field codes of the items to be read from the block to ITEMS$. The field codes correspond to the following items (in order): tag, A, B, C, D, Q, @a, @b, @c, @d.

```
1120  ADDRESS$="=01A-01_ _ _ "
```

The address of the first block from which the items are to be read.

```
1130  REQ$="21"+ADDRESS$+"3"+ITEMS$
```

Assign the assembled request message to REQ$. (3 means the analog values are scaled and the discrete values are message pairs.)

```
1135  PRINT "Requesting controller data from "+ADDRESS$
1140  GOSUB 1200
1145  GOSUB 1300
```

Go to the subroutines that send the request and receive the reply and print the reply items.

```
1150  ADDRESS$="=01A-02_ _ _ "
```

The address of the second block from which the items are to be read.

```
1152  REQ$="21"+ADDRESS$+"3"+ITEMS$
```

Assign the assembled request message to REQ$.

```
1155  PRINT "Requesting controller data from "+ADDRESS$
1160  GOSUB 1200
1165  GOSUB 1300
```

Go to the subroutines that send the request and receive the reply and print the reply items.

```
1170  ADDRESS$="=01B-01_ _ _ "
```

The address of the third block from which the items are to be read.

```
1172  REQ$="21"+ADDRESS$+"3"+ITEMS$
```

Assign the assembled request message to REQ$.

```
1175  PRINT "Requesting controller data from "+ADDRESS$
1180  GOSUB 1200
1185  GOSUB 1300
```

Go to the subroutines that send the request and receive the reply and print the reply items.

```
1190  GOTO 2100
/*===============================================================*/
```

Reading Data from a Controller Processor (continued)

The following subroutine adds the ID and CRC to the request message, sends the request, receives the reply, and prints the reply to the screen.

```
/*============================================================*/
1200  ID=(ID+1) MOD 256

1210  ID$=HEX$(ID)

1220  IF (ID<16) THEN ID$="0"+ID$
```

Increment the ID number and convert it into characters.

```
1230  'GOSUB CRC ROUTINE
```

Go to a subroutine that computes the CRC for the request message if one is used in your application.

```
1240  PRINT "Sending => "+ID$+REQ$+CRC$
```

Print the request message to the screen.

```
1250  PRINT #1,ID$+REQ$+CRC$
```

Send the request message to comp file #1 in the SCI.

```
1260  LINE INPUT#1,REPLY$
```

Read the reply message from comp file #1 in the SCI.

```
1270  PRINT "Received => "+REPLY$
```

Print the reply message to the screen.

```
1280  RETURN
/*============================================================*/
```

Reading Data from a Controller Processor (continued)

The following subroutine prints the items from the reply message to the screen. The starting point of the item in the string and its length are given for each item. One character between each item is allowed for a comma.

```
/*==============================================================*/
1300  PRINT ”The values of ”ADDRESS$+” are: ”

1310  PRINT ”Tag   = ”+MID$(REPLY$,11,8)

1320  PRINT ”  A   = ”+MID$(REPLY$,20,7)

1330  PRINT ”  B   = ”+MID$(REPLY$,28,7)

1340  PRINT ”  C   = ”+MID$(REPLY$,36,7)

1350  PRINT ”  D   = ”+MID$(REPLY$,44,7)

1360  PRINT ”  Q   = ”+MID$(REPLY$,52,8)

1370  PRINT ”  @a  = ”+MID$(REPLY$,61,8)

1375  PRINT ”  @b  = ”+MID$(REPLY$,70,8)

1380  PRINT ”  @c  = ”+MID$(REPLY$,79,8)

1385  PRINT ”  @d  = ”+MID$(REPLY$,88,8)

1390  PRINT :PRINT ”Hit any key to continue”: :PRINT

1392  IF (INKEY$=”) GOTO 1292

1394  CLS

1395  RETURN

1397  END
/*==============================================================*/
```

Writing to a Controller Processor

Writing to a Controller Processor is similar to reading from a Controller Processor. For sample instructions, see the heading “Reading From a Controller Processor.”

The following instructions would replace lines 1110 to 1135 of the instructions in the ”Reading From a Controller Processor” heading and could be repeated for lines 1150 to 1155 and 1170 to 1175 for different block addresses.

```
/*==============================================================*/

1110  ITEMS$=”1100=100.000,1120=000183.,1140=-10.099,
      1160=000712.,0006=0000547.,”

1112  ITEMS$=ITEMS$+”1114=_ _ _ON_ _ _,1134=_ _ _OFF_ _,
      1154=_ _OPEN_ _,”

1114  ITEMS$=ITEMS$+“1174=_ _CLOSE_,”

1120  ADDRESS$=”=01A-03_ _ _”

1130  REQ$=”22”+ADDRESS$+”31”+ITEMS$

1135  PRINT ”Writing controller data to ”+ADDRESS$
/*==============================================================*/
```

The reply for a write request contains the message type and success code. To print the reply, replace lines 1300 to 1385 of the sample program for reading data with the following line:

```
/*==============================================================*/
1300  PRINT ”Write Success Code =”+MID$(REPLY$,8,3)
/*==============================================================*/
```

Reading from the SCI Table

The following instructions initialize the SCI data table and read from it every 15 seconds. For more information about reading from the SCI table, see the ”Table Data Messages” heading in Section 3.

Initialize the SCI table

```
/*==============================================================*/
2080  SCREEN 0,0: KEY OFF: CLS

2090  OPEN ”COM1: 9600,M,7,1,CS0,DS0,CD0” AS #1

2100  ON COM(1) GOSUB 10400
```

Initialize the interrupt subroutine that reads in a reply when one is received.

```
2120  ON TIMER(1) GOSUB 10600
```

Initialize the interrupt subroutine that contains the timer and executes timeouts. Execution goes to this subroutine every second.

```
2130  TIMER ON
```

Start the timer.

```
2140  TIMEOUT=2
```

Set the timeout period at 2 seconds (i.e., after sending the request to the SCI, the host waits 2 seconds before sending the request again).

```
2150  RETRYLIMIT=3
```

Set the retry limit to 3 (i.e., the host allows 3 timeout periods to elapse before giving up on the request and stopping communication with the SCI).

```
/*==============================================================*/
```

Clear the SCI table

```
/*==============================================================*/
2210  PRINT: PRINT ”Requesting to clear out SCI data table
      entries 1 through 8”

2220  REQ$=”32001008”
```

Configure the request message (type 32) that clears entries 001 to 008 of the data table.

```
2230  GOSUB 10100
```

Go to the subroutine that sends the request to the SCI

```
/*==============================================================*/
```

Reading from the SCI Table (continued)

Configure the SCI table

```
/*=============================================================*/
2250  PRINT: PRINT "Request to configure SCI data table
      entries 1 through 8"
2260  REQ$="3308"+"001=07d-01/A_ _ _ _ _ _ _ _ _10,"

2270  REQ$=REQ$+"002=07d-01/B_ _ _ _ _ _ _ _ _10,"

2280  REQ$=REQ$+"003=07d-01/C_ _ _ _ _ _ _ _ _10,"

2290  REQ$=REQ$+"004=07d-01/D_ _ _ _ _ _ _ _ _10,"

2310  REQ$=REQ$+"005=01B-02/A_ _ _ _ _ _ _ _ _10,"

2320  REQ$=REQ$+"006=01B-02/B_ _ _ _ _ _ _ _ _10,"

2330  REQ$=REQ$+"007=01B-02/C_ _ _ _ _ _ _ _ _10,"

2340  REQ$=REQ$+"008=01B-02/D_ _ _ _ _ _ _ _ _10,"
```

Lines 2260 through 2340 configure the request message (type 33) that configures SCI table entries 1 through 8. Each entry has a scaled value (1) and an update interval of 15 seconds (0).

```
2360  CONFIG$=REQ$
```

Save the request message for printing.

```
2370  GOSUB 10100
```

Go to the subroutine that sends the request to the SCI.

```
/*=============================================================*/
```

Enable data collection

```
/*=============================================================*/
2385  PRINT: PRINT "Requesting SCI data table to begin data
      collection"

2390  REQ$="311"+LEFT$(TIME$,2)+MID$(TIME$,4,2)+RIGHT$(TIME$,2)
```

Configure the request message (type 31) that enables data collection (1) and synchronizes the clocks of the host and SCI.

```
2400  10100
```

Go to the subroutine that sends the request to the SCI.

```
2402  PRINT :PRINT "Hit any key to read data table values"

2404  IF (INKEY$=") GOTO 2404

2406  PRINT "Will read values at :05, :20, :35, :50 seconds"

2407  LOCATE 25,1,1: PRINT "Hit Ctrl-Break (Scroll Lock) to end
      demonstration"
/*=============================================================*/
```

Reading from the SCI Table (continued)

Lines 2470 to 2510 configure the request message. The request message asks for the 8 entries configured in lines 2260 to 2340 in 15 second intervals. The request message is set to allow 5 seconds to expire after the update interval before the entries are read so that the SCI has time to update the entries. Since the SCI updates the table at :00, :15, :30, and :45 seconds, the request message is sent at :05, :20, :35, and :50 seconds.

```
/*==============================================================*/
2470  IF (CLOCK<>5) GOTO 2470
```

Wait at this instruction until :05, :20, :35, or :50 seconds. At these times, CLOCK = 5 and it is time to send the request.

```
2480  SECONDS$ = RIGHT$(STR$(VAL(RIGHT$(TIME$,2))-5),2)
```

Configure the seconds field of the time to be put in the request message. This is the time for which the data will be read. Five seconds is subtracted from the actual time to arrive at this time. For sample, the data read from the table at 11:15:35 is for 11:15:30. This allows 5 seconds for the SCI to update the table.

```
2490  IF (LEFT$(SECONDS$,1)=″) THEN SECONDS$=″00″
```

Make the seconds field 00 if its left digit is a blank (i.e.,_0).

```
2500  CLS :PRINT ″Requesting the ″+MID$(TIME$,4,2)+:
      +SECONDS$+″values″

2510  REQ$=″34″+MID$(TIME$,4,2)+SECONDS$ +″01001008,″
```

Configure the request message that reads the data table entries. The time is given in MMSS format. One range (01) from 001 to 008 is read.

```
2530  GOSUB 10100
```

Go to the subroutine that sends the request to the SCI.

```
2540  'PRINT THE RETURNED VALUES TO THE SCREEN

2550  PRINT ″ address    value      mode       status″

2560  PRINT ″  ____       ____      ____       _____″

2570  FOR I=0 TO 7

2580  PRINT MID$(CONFIG$,I*22+8,16)+″==″+MID$(REPLY$,I*12+13,8);

2590  PRINT ″           ″+MID$(REPLY$,I*12+21,1) +
      ″             ″+MID$(REPLY$ I*12+22,2)

2600  NEXT
```

The FOR loop prints the values from the SCI table returned in the reply message. From the CONFIG$ variable (set = REQ$ in line 2360), the value, node, and status for each table entry are printed.

```
2610  PRINT
2620  CLOCK=6
```

Prevent the results from printing more than once per second.

```
2630  GOTO 2470
2640  END
/*==============================================================*/
```

Reading from the SCI Table (continued)

The following subroutine (lines 10100 to 10290) sends the Read Table Data request message and waits for a reply. It sends the request the number of times specified in line 2150 (RETRYLIMIT) and then returns an error status code if no reply is received. The values of DONE$ (i.e., the reply is received) and WAITING$ (i.e., waiting for a reply) are changed by the COM1 interrupt subroutine. WAITING$ is changed by the RETRYLIMIT subroutine (line 10800).

```
/*==============================================================*/
10100 RETRY=0
```

Initialize the retry count. The request can be sent 3 times before the retry limit (3) is reached.

```
10150 REPLY$=”
```

Initialize the reply string.

```
10160 DONE$=”NO”
10170 ID=(ID+1) MOD 256
10180 ID$=HEX$(ID)
10190 IF (ID<16) THEN ID$=”0”+ID$
```

Lines 10170 to 10190 increment the message ID and change it to characters.

```
10195 GOSUB CRC ROUTINE (if used in your application)
```

Go to a subroutine that computes a CRC for the request message if used in your application.

```
10200 WHILE (DONE$=”NO” AND RETRY<RETRYLIMIT)
```

The WHILE loop sends the request to the SCI and waits for the reply. Execution stays in the WHILE loop or returns to it if DONE = “NO” and RETRY < RETRYLIMIT.

```
10210 TICKS=0
```

Reset the timeout counter.

```
10220 PRINT ”Sending => ”+ID$+REQ$+CRC$
```

Print the request message to the screen.

```
10230 PRINT +1,ID$+REQ$+CRC$
```

Send the request to the SCI.

```
10240 WAITING$=”YES”
```

Start the timeout counter (line 10620).

```
10250 COM(1) ON
```

Start listening for a reply on serial port 1.

```
10260 IF (WAITING$=”YES”) GOTO 10260
```

Wait here for a reply or a timeout. The interrupt routine that reads in a reply sets WAITING$ = “NO”, causing execution to continue. The same thing happens if a timeout occurs.

```
10270 WEND
10280 IF (RETRY=RETRYLIMIT) THEN PRINT ”TIMEOUT! Nothing
      received from SCI.”
10285 IF (RETRY=RETRYLIMIT) THEN STOP
10290 RETURN
```

Return to line 2540.

```
/*==============================================================*/
```

Reading from the SCI Table (continued)

The following subroutine (lines 10400 to 10450) is activated every time a reply is received by serial port 1. Typically, the subroutine executes an interrupt while the program is waiting for a reply in line 10260. Following execution of the subroutine, execution returns to line 10260; however, now WAITING$ = “NO” and DONE$ = “YES”. Execution, therefore, leaves the WHILE loop.

```
/*==============================================================*/
10400 WAITING$=”NO”
```

Done waiting for a reply. Execution will continue from line 10260.

```
10410 COM(1) OFF
```

Stop listening for a reply.

```
10420 LINE INPUT #1,REPLY$
```

Read in the SCI reply.

```
10430 PRINT ”Received => ”+REPLY$
```

Print the reply to the screen.

```
10440 DONE$=”YES”
```

The reply is received. Execution can leave the WHILE loop.

```
10450 RETURN
```

Return to line 10260.

```
/*==============================================================*/
```

The following subroutine (lines 10600 to 10640) is activated once every second, no matter what line in the program is being executed, and returns to the line at which the interrupt occurred. If TICKS > TIMEOUT, this subroutine calls a subroutine which forces the WHILE loop to stop trying to communicate.

```
/*==============================================================*/
10600  XX=CSRLIN: YY=POS(0)
10605  LOCATE 25,70,1: PRINT TIME$

10610  LOCATE XX,YY,1

10615  CLOCK=VAL(RIGHT$(TIME$,2)) MOD 15
/*==============================================================*/
```

Reading from the SCI Table (continued)

The variable CLOCK is assigned a value that is the remainder of the division of the current seconds by 15. If the current seconds are 50, for sample, the CLOCK value is 5. The value of CLOCK determines whether execution continues from line 2470.

```
/*==============================================================*/
10620 IF (WAITING$="YES") THEN TICKS=TICKS+1
```

Increment the timeout counter if WAITING$ = "YES".

```
10630 IF (TICKS>TIMEOUT) THEN GOSUB 10800
```

If the timeout period has been exceeded, go to the retry limit subroutine.

```
10640 RETURN
```

Return to the instruction being executed when the timeout interrupt occurred.

```
/*==============================================================*/
```

The following subroutine (lines 10800 to 10830) is called from the TIMER interrupt subroutine when TICKS seconds have passed and a reply has not yet been received. It increments the retry counter. Execution returns to line 10640 in the TIMER interrupt, and then returns to line 10260.

However, WAITING$="NO", so execution continues from line 10260. DONE is still "NO", so that, if the retry limit is not reached, the WHILE loop is executed again (the request is sent again). If the retry limit is reached, execution leaves the WHILE loop, and an error message is sent.

```
/*==============================================================*/
10800 WAITING$="NO"
```

The timeout has been reached, stop waiting.

```
10810 COM (PORT) OFF
```

Disable the reply interrupt.

```
10820 RETRY=RETRY+1
```

Increment the retry count.

```
10830 RETURN
```

Return to line 10640.

```
/*==============================================================*/
```

Writing to the SCI Table

Writing to the SCI table is similar to reading from the SCI table. For sample instructions, see the heading “Reading From the SCI Table”. The following instructions would replace lines 2470 to 2510 and line 10615 of the sample program for reading from the SCI table.

```
/*============================================================*/
2500  CLS: PRINT ”Writing to the data table”
2510  ITEMS$=”111=10000.0,112=00002.9,113=00223.8,114=.987000,
      115=00675.0”

2520  REQ$=”3511105”+ITEMS$
/*============================================================*/
```

The reply contains the message type, the success code, the number of return codes in the reply, and a list of the success codes for each value written to the data table. For sample:

3500005 000,000,000,000,000,...

CRC Samples

This section discusses the “Checksum” field on the SCI Configuration screen and provides CRC calculation samples.

Checksum Field

The “Checksum” field specifies whether or not CRC (cyclical redundancy check) is performed in request and reply messages. CRC is a check performed on a block of data to determine whether or not an error occurred in transmission.

The SCI allows you to use a CRC to help insure data integrity. To use CRC, you must include subroutines in your host program that accomplish the following tasks:

- Compute the CRC of a request message and append the CRC characters to the message. The SCI computes the CRC to determine if the request message was received correctly.
- Optionally compute the CRC of a reply message and compare it to the CRC characters that the SCI included in the reply message. If the values are not equal, the message is in error and should be discarded.

This section includes a description of the CRC calculation that the SCI uses and some sample subroutines to calculate checksums.

NOTE: When you use X.25 protocol, you usually do not need to use CRC because of the data integrity of X.25 protocol.

CRC Calculation Method

The CRC is generated from all of the characters in the message, except for the header, trailer, CRC, message length, and message length negation characters. The CRC consists of a 16-bit code that is encoded as four characters. The CRC characters are calculated with the following algorithm:

- Starting with the 16-bit hex value 5A5A (CRC), the following operations are performed for each character (CHAR) in the message:

  CRC = CRC <exclusive or> CHAR CRC <right rotate> 9 bits CRC = CRC <exclusive or> CHAR

The resulting CRC value is appended to the message body.

In addition, in asynchronous terminator protocol and X.25 protocol, the 16-bit CRC is divided into four sets of four bits each. The value of each set is converted into an ASCII hexadecimal digit. For sample, the 16-bit CRC of 18AF is converted into the ASCII string 18AF.

CRC Sample for the 6809 Microprocessor

This sample uses 6809 Assembly language instructions to generate the CRC. This is an sample only and may not be valid for all applications.

Set the CRC result, store to hex 5A5A, and add each character to the result according to the following instructions:

```
Result MSB 0,Y 5A

Result LSB 1,Y 5A

Temporary store 2,Y Temp store
```

Enter with the character to be added in accumulator A.

`STA 2,Y`	Store character
`LDB 0,Y`	Load accumulator B with MSB
`EORA 1,Y`	Exclusive OR accumulator A with LSB
`ASLA`	Arithmetic shift left accumulator A
`ROLB`	Rotate accumulator B
`ADCA L0`	Add with carry 0 to accumulator A
`ASLA`	Arithmetic shift left accumulator A
`ROLB`	Rotate accumulator B
`ADCA L0`	Add with carry 0 to accumulator A
`EORA 2,Y`	Exclusive OR accumulator with original character
`ASLA`	Arithmetic shift left accumulator A
`ROLB`	Rotate accumulator
`ADCA L0`	Add with carry 0 to accumulator A
`EORB 2,Y`	Exclusive OR B with character
`STD 0,Y`	Save result 0,Y 1,Y
`RETURN`	

The computation continues until all of the characters in the message have been computed.

CRC Sample for C

This sample uses C language instructions to generate the CRC and put the result in the last four bytes of a message. This is an sample only and may not be valid for all applications.

```
/*==============================================================*/
dochk(msg, nchar)
unsigned char msg [] ;
int nchar ;
{
  int i;
  unsigned short crc;

  char hexdigit[]="0123456789ABCDEF";

  /* Form CRC */

  crc=0x5a5a;

  for (i=0;i < nchar-4;i++){
    crc∧=msg [i];
    crc=(crc<<7)|((crc>>9)&0x7f);
    crc∧=msg[i];
    }

  /*  Convert CRC to 4 hex digits  */

  msg[nchar-1]=hexdig[crc&0xf];
  crc>>=4;
  msg[nchar-2]=hexdig[crc&0xf];
  crc>>=4  ;
  msg[nchar-3]=hexdig[crc&0xf];
  crc>>=4  ;
  msg[nchar-4]=hexdig[crc&0xf];
}
/*==============================================================*/
```

CRC Sample for FORTRAN

This sample uses FORTRAN language instructions to generate the CRC and put the result in the last four bytes of a message. This is an sample only and may not be valid for all applications.

```
/*=============================================================*/
subroutine dochk (msg, nchar)
character msg (nchar)
integer nchar
integer i
integer crc
character hexdig (16)
data (hexdig (i),i=1,8)/'0','1','2','3','4','5','6','7'/
data (hexdig (i),i=9,16)/'8','9','A','B','C','D','E','F'/

c Form the checksum.
crc=23130
do 10 i=1,nchar-4
crc=ieor(crc,msg(i))
crc=ior(ishft(crc,7),iand(ishft(crc,-9),127))
10 crc=ieor(crc,msg (i))

c Express the computed checksum in ASCII as a 4 dig hex number.
msg(nchar-0)=hexdig(iand(crc,15)+1)crc=ishft(crc,-4)
msg(nchar-1)=hexdig(iand(crc,15)+1)crc=ishft(crc,-4)
msg(nchar-2)=hexdig(iand(crc,15)+1)crc=ishft(crc,-4)
msg(nchar-3)=hexdig(iand(crc,15)+1)
end
/*=============================================================*/
```

Section 6:
Error Codes

Reply messages contain a three-digit Error field that informs the host of any difficulties in processing the corresponding request message. Table 1.6.1 defines the meanings of these codes.

Table 1.6.1. Error Codes

Code	Meaning
000	Operation successful.
001	Unknown message type.
002	Message length incorrect.
003	Generic error code. An error has occurred that is not covered by another, more specific error code.
004	Host disconnected.
005	Bad password.
006	Message no longer in debug queue.
008	Request too long.
009	Reply too long.
010	No such tag.
011	No such PeerWay.
012	No such node.
013	No such volume.
014	No such node/drive.
015	No such file.
016	No such controller card.
017	No such block/link.
018	Node not a ControlFile.
020	Disc read/write error.
021	Global SCI write disabled. “Block Update Disable⇒Yes” is configured on SCI CONFIGURATION screen.

(continued on next page)

Table 1.6.1. Error Codes (continued)

Code	Meaning
022	SCI write disabled to this block. Block has “SC⇒No” configured on CB CONTINUOUS FACEPLATE screen.
023	Block mode incorrect for write.
024	Value outside limits.
025	Invalid configuration for write.
026	Too many Set RS3 Time messages (#71) sent in a period of time.
030	SCI table full.
031	SCI table address already in use.
032	SCI not synchronized with host. Send Message Type 31.
033	A write has been attempted while polling is disabled.
034	The SCI table contains another entry for this block variable. The other entry has a different update interval.
040	Raw block transfer in progress.
041	Checksum bad on raw data write.
051	The reply message may not have status information for all writes.
060	Cannot change message format in Asynchronous Terminator mode.
070	No transfers available and no transfers have timed out.
071	Transfer not active. Transfer timed out.
072	Unable to find a report that meets the request message criteria.
073	Report generation conflict. A report was generated while a report file transfer was active.
074	File transfer message not allowed for this folder and file type.
101 and up	Error in processing a specific message field. To find the field in error: 1. Look up the specific request message format. 2. Subtract 100 from the error code. The resulting number indicates which field is in error. 3. From the top of the table, count down the number of fields found in step 2. For example, if a reply message contains an error code of 112, then the error occurred in the twelfth field of the request message.

FISHER-ROSEMOUNT

RS3 ™
PeerWay Interfaces Manual

Chapter 2:
Host Mode

List of Figures

List of Tables

Section 1:
Host Mode

This section describes how to make a Host Mode connection between the RS3 and a VAX® computer.

What is Host Mode?

A Host Mode session is a connection between an RS3 68040 pixel Multitube Command Console CRT and a VAX host computer running RMT/host software.

Emulation

Host Mode provides a limited emulation of a DEC VT340 terminal in both ANSI and REGIS modes. Not all VT340 features are implemented.

Applications

Host Mode is a way for RS3 operators to use application programs on VAX/VMS computers that are running RMT/host. Host Mode applications are site specific; consult your VAX system manager to find out about programs that are available at your site.

Configuring DEC VT Keys on the RS3

Some applications might require keys that are unique to a DEC VT keyboard. You can configure RS3 callup buttons to function like VT keys. Table 2.1.1 shows the commands that are used to assign VT key functions to callup buttons.

NOTE:

- Callup buttons are configured on the Alarm Annunciation screen. For information, see CC: 1-4.
- You do not have to configure callup buttons for all of the VT keys, only for those your applications require.
- VT keys F1 to F5 are reserved for local VT functions and cannot be configured on the RS3.

❏ **To enter the character ñ :**

1. Type the character n.
2. Press [CTRL] [a]. The ~ symbol appears above the n.

Table 2.1.1. Callup Button Commands for DEC VT Keys

DEC VT Key	Callup Buttons
F6	~ñ [17~
F7	~ñ[18~
F8	~ñ[19~
F9	~ñ[20~
F10	~ñ[21~
F11	~ñ[23~
F12	~ñ[24~
F13	~ñ[25~
F14	~ñ[26~
HELP	~ñ[28~
DO	~ñ[29~
F17	~ñ[31~
F18	~ñ [32~
F19	~ñ[33~
F20	~ñ[34~
FIND	~ñ[1~
INSERT HERE	~ñ[2~
REMOVE	~ñ[3~
SELECT	~ñ[4~
PREV SCREEN	~ñ[5~
NEXT SCREEN	~ñ[6~
PF 1	~ñOP
PF 2	~ñOQ
PF 3	~ñOR
PF 4	~ñOS

NOTE: KP Stands for key pads for key pad

(continued on next page)

Table 2.1.1. Callup Button Commands for DEC VT Keys (continued)

DEC VT Key	Callup Buttons
KP 0	~ñOp
KP 1	~ñOq
KP 2	~ñOr
KP 3	~ñOs
KP 4	~ñOt
KP 5	~ñOu
KP 6	~ñOv
KP 7	~ñOw
KP 8	~ñOx
KP 9	~ñOy
KP ,	~ñOl
KP –	~ñOm
KP .	~ñOn
KP ENTER	~ñOM
RETURN	~*
VMS Commands	~Command * Examples: ~DIR * ~MAIL * ~EDIT *

NOTE: KP Stands for key pads for key pad

Beginning a Host Mode Session

❑ **To begin a Host Mode session from the RS3 console:**

- Type the following command:

  [H] [M] (*p1*,p2,p3,p4,p5,*p6*) [ENTER]

Parameters p1 through p6 are described below:

p1 The PeerWay node number (1-992) of the VAX RPQNA interface. If there is only one VAX node, this parameter can be skipped; the connection message is then sent to all of the VAX RPQNA nodes on the PeerWay.

p2 The Rosemount Plant Data Highway (PDH) Logical Unit Number (LUN). This specifies the particular LUN (1-256) on which the Host Mode program must execute. If there is only one VAX that can run the selected Host Mode program, this parameter can be skipped.

p3 Host Mode executable file name.

This is the VAX/VMS file name of the Host Mode program that is to be executed. It can be a VMS logical name. If the p1 and p2 parameters are not specified, each VAX tries to execute the specified program. No error messages are issued if the program is not found.

If parameter 3 is not specified, the program specified in RMT SYSTEM=CONFIGURE is used to select the program to execute on a LUN by LUN basis.

The default file specification for the Host Mode program is RMT_HOST_MODE:*filename*.EXE. For example, if you enter the command [H] [M] ,,TEST, then the VAX attempts to execute the file RMT_HOST_MODE:TEST.EXE.

Remember that any logical names used to specify files for Host Mode must be in the SYSTEM logical name table on the VAX that is executing the program.

PTERM Host Mode includes a terminal emulation program called PTERM. PTERM enables the RS3 to emulate some of the functions of a DEC VT220 terminal. To use PTERM, enter PTERM as parameter 3 in the host mode command.

p4
p5
p6 p4, p5, and p6 are parameters that are passed to the Host Mode program that is executed.

Examples of Commands

Two examples of commands are described below.

HM 23,,test

Executes RMT_HOST_MODE:TEST.EXE on the VAX connected to the PeerWay as node 23. If there is no VAX at node 23, then no program will run. In this example, the LUN has been eliminated.

Note that even though parameter p2 is skipped, a comma is required for the parameter.

HM ,25

Executes the program specified by RMT SYSTEM=CONFIGURE on VAX LUN 25. LUN 25 may be directly connected to the PeerWay or connected to a gateway VAX which is connected to the PeerWay. In this example, the PeerWay node and program have been eliminated. If there is no configured program on LUN 25 then the connection will not be made.

Note that even though parameter p1 is skipped, a comma is required for that parameter.

HM ,, PTERM

Executes the PTERM terminal emulation program. PTERM enables the RS3 to emulate some of the functions of a DEC VT220 terminal. After you log in with PTERM, the Host Mode system prompts you for a VAX account number and password.

Note that even though parameter p1 and p2 are skipped, commas are required for these parameters. PTERM is entered as parameter 3.

If you exit a host mode session, you can later resume execution at the point in the session where you left off by typing “**HM ,, PTERM**”. You might have to wait while the Host Mode refreshes the screen.

Host Mode Screen

The Host Mode screen is shown in Figure 2.1.1.

❑ **To move the cursor into the VAX VT 340 equivalent terminal lines:**

1. Press the down arrow key or use the trackball to move the cursor down to lines 4-27
2. Press [Select] to choose a field, if needed.

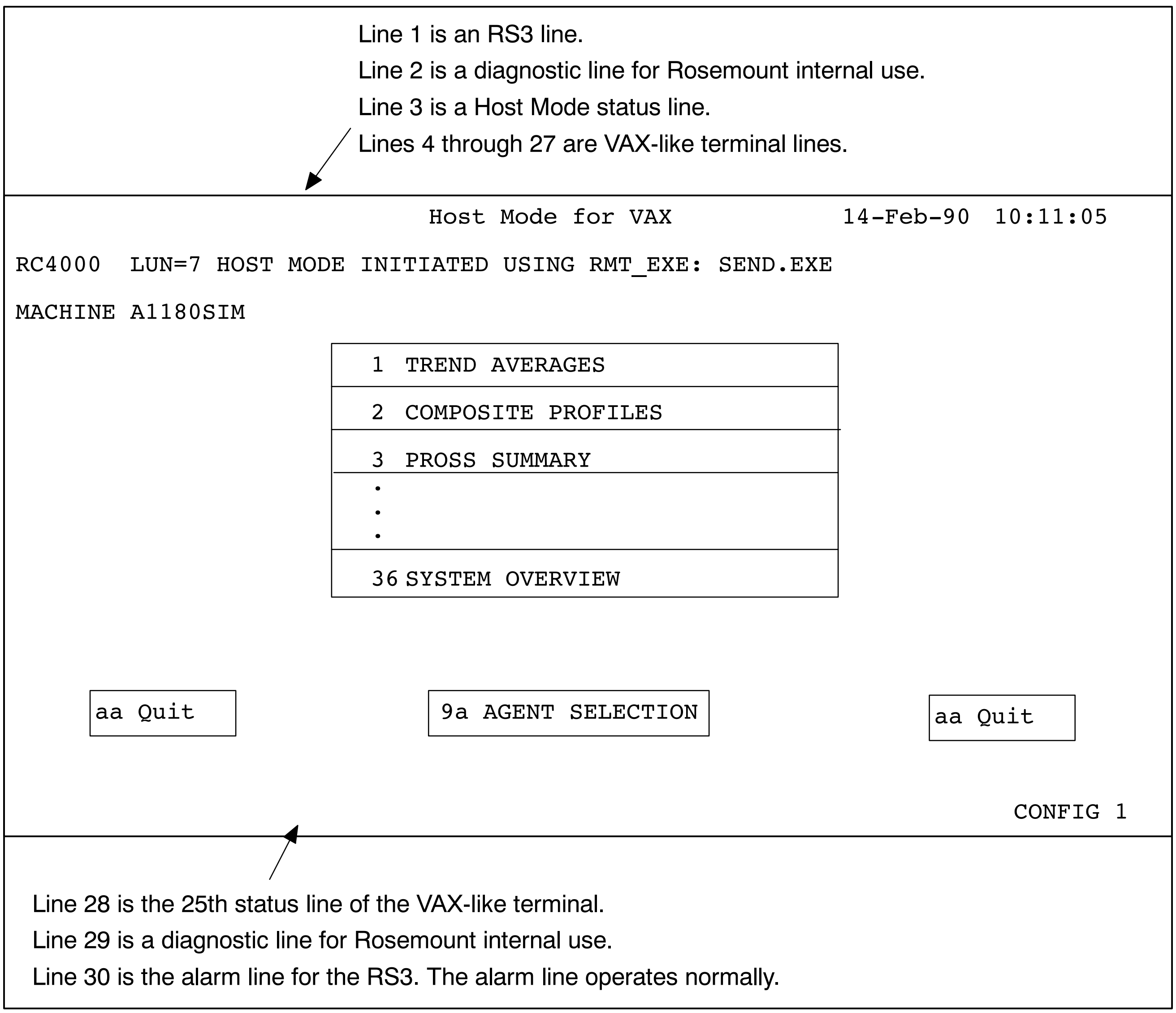


Figure 2.1.1. Host Mode Screen

Ending a Host Mode Session

Once you end a Host Mode session, the program is terminated. You cannot press [RECALL] to resume the session. If you want to communicate with the VAX again, you must start a new Host Mode session.

❑ **To end a Host Mode session:**

- Move the cursor to the command line and type a new RS3 command.

  or

- Press [RECALL].

❑ **To end a Host Mode PTERM session:**

- At the prompt, type:

  LOGOUT [RETURN]

  You can later resume execution at the point in the session where you logged out by typing:

  HM ,, PTERM [RETURN]

VAX Status Screen

The VAX Status screen provides information about the VAX/RS3 connection. Figure 2.1.2 shows an example VAX Status screen. Table 2.1.2 describes the types of information provided on the screen.

❑ **To call up the VAX Status screen from the command line:**

- Type [V] [S] [ENTER]

```
                             VAX RPQNA Status Display        19-Jun-90  13:12:55
Node  :87 (23 on PeerWay 3) QBI.PRG  V 13.20  QBI.ROM V 2.00  Prwy+QBI  140mS

 STATIC SCAN                DYNAMIC SCANS:    1       2       3          4     Spare6
ConfigScan Sec    600.00 ConfigScan Sec   30.0    30.0    30.0       30.0     h3D
LastLinkScan S    600.0  Last Scan    s   30.0    30.0    30.0       30.0     h8
Link Scan Blks    36     Blocks in scan   4       0       145        0        h9D
TOTAL ScanBlks    149    PeerwyReadMsgs   0       0       15         0        116
  VAX->QBI MSGS          FloorBetMsg mS   200     200     1875       200      0
MsgInterval  mS   3250
IntrServDly  mS   0      QBI TASKS:     ERRORS  DIAGN/Q  ACTION    T243562  INTERVAL
MsgAck Time  mS   0      StatScanInitD8   0     TO        =h0909    1885     484
Active Writes     0      StatLinkScScan   50    Tcebdaf   =89A-09   2433534  3241
InterruptCount    2      BadStatic Scan   0     Tbdcebd   =3A-99    2407523  0
RS3+QBI svc mS    0      Dynamic 1 Scan   0     Mgggg     =3AB101   623273   18000
                         Dynamic 2 Scan   0     M0 =      0                  0
UDH Dynam  Bpm    31862  Dynamic 3 Scan   0     Mvvnv     =89A-01   2435281  747
UDH Static Bpm    982    Dynamic 4 Scan   0     M0 =      0                  0
UDH Alarms Bpm    7872   DtQ&DynScReply   3     VA07C2    =89A-11   2435310  750
UDH OtherBytpm    36540  CtQ&Rp|RqToVAX   0     VC000C    M0        0        0
                         ReqstQ&MsFrVAX   0     V80010    MC0000    11270    650
Idle Activity     22346  Alarms fr RS3    48    R10101    =89?15#3  2435476  34
Hog Time    mS    70     Alarms to VAX    0     V69A      A01#312   2428034  3800
Hog Task ID       30800  T1Spare Q&askS   48    v7AC07C2  =89G?CV   1854029  241
Hog CallReturn    5E102  Last Error       50    0         0         2410855  3316
QBI Error Mask    406    Spare0  &ctrlQ   0     0         0         11270    649



                                                                      CONFIG 1
```

Figure 2.1.2. VAX Status Screen

Table 2.1.2. VAX Status Screen Fields

Field	Description
Static Scan	Displays the configured and actual static scan time, number of links scaled, and total number of blocks scanned.
Dynamic Scans	Displays the configured and actual dynamic scan rate, number of blocks per dynamic scan, and time between requests from controller.
VAX QBI Message	Time between VAX request messages, number of request interrupts, and QBI and PeerWay service time.
UDH	PeerWay message load in bytes per minute.
Idle Activity	Available idle time for QBI CPU. 1000 is very busy. 30,000 is not busy.
QBI Tasks	A diagnostic tool for Rosemount. Displays tasks, task activity and errors; includes an activity time stamp in intervals of 5 milliseconds.

Chapter 3:
Highway Interface Adapter (HIA)

List of Figures

List of Tables

Section 1:
Highway Interface Adapter

Highway Interface Adapters (HIAs) allow communications among RS3™ PeerWays. An HIA consists of a card cage and associated hardware that connects directly to the PeerWay. A pair of HIAs are connected to provide PeerWay-to-PeerWay communications.

It is recommended that PeerWays connected by HIAs be configured to provide alternate communication paths in the event of communication failure. For example, Figure 3.1.1 shows three PeerWays connected by HIAs. If communications fail between a pair of HIAs, data can still be routed to the proper destination.

NOTE: You should avoid connecting more than three PeerWays in a ring pattern. If one path should fail, the remaining paths might become overwhelmed by heavy message traffic.

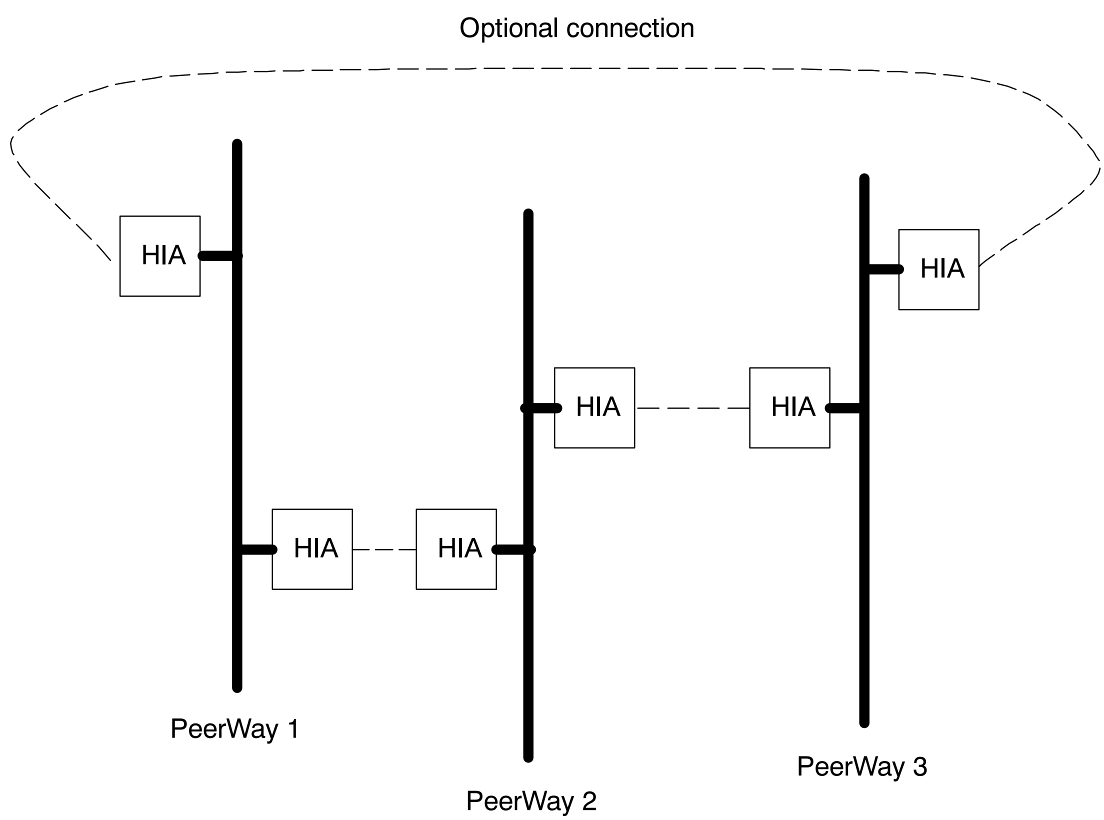


Figure 3.1.1. PeerWays Connected by HIAs

HIA Hardware

Figure 3.1.2 shows the front view of a single HIA. Figure 3.1.3 shows the rear view of two connected HIAs. The two HIAs that provide a PeerWay connection are typically contained side by side in a shelf assembly with a short cable connecting them.

For more information about HIA hardware, see the *Service Manual*.

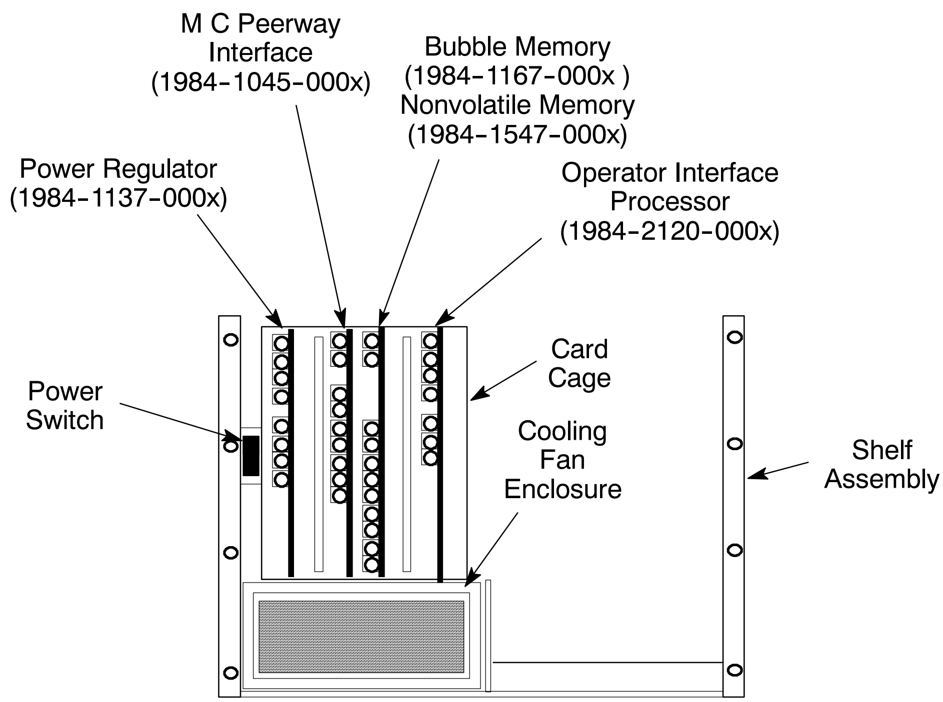


Figure 3.1.2. Single HIA (Front View)

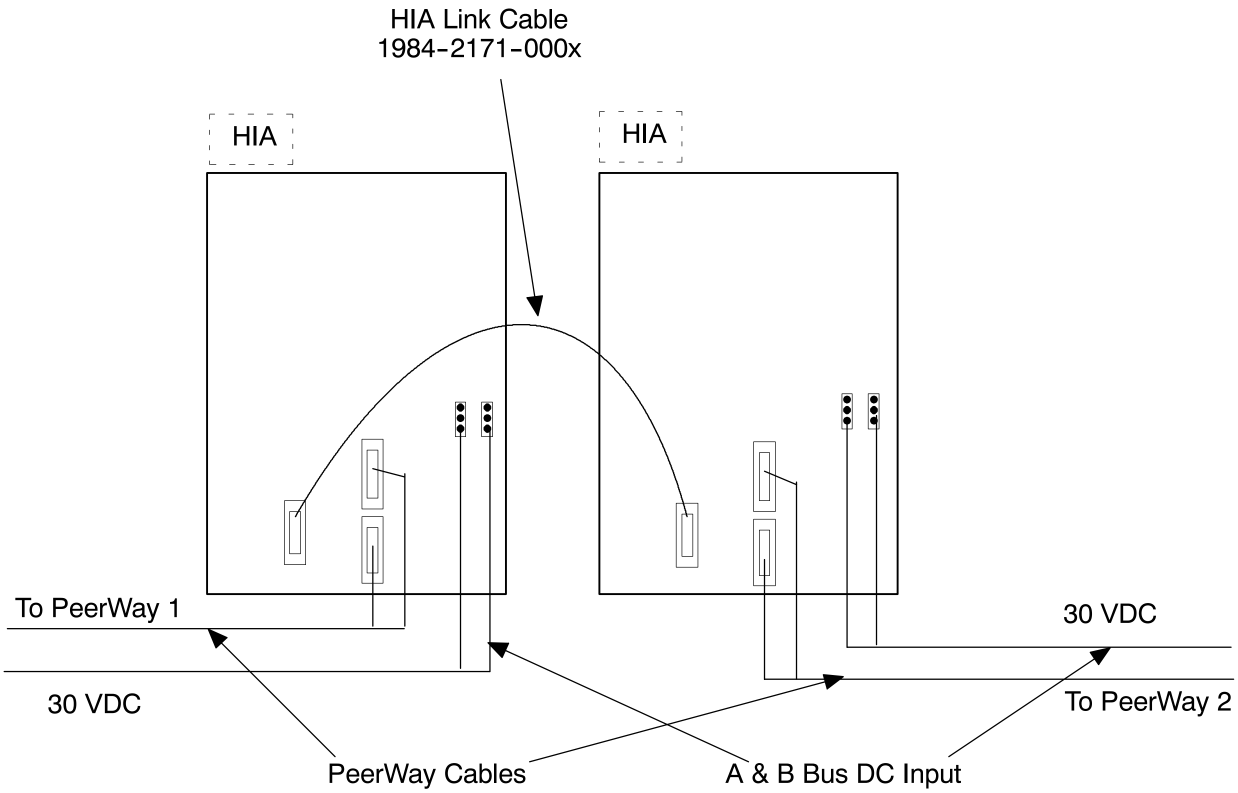


Figure 3.1.3. Two HIAs—Wire Routing—Rear View

Nonvolatile Memory Card Internal/External Clock Jumpers

The internal/external clock jumpers on the Nonvolatile Memory Card (1984-1167-000x or 1984-1547-000x) determine the source of the HIA clock synchronization. For jumper locations and positioning, see Figure 3.1.4.

- One HIA in a pair should be jumpered for internal clocking and the other HIA should be jumpered for external clocking.
- The HIA with internal clocking is the source for the clock synchronization signal.
- The “Clock src” field entry on the CONFIGURE HIA screen must match the clock jumper position.

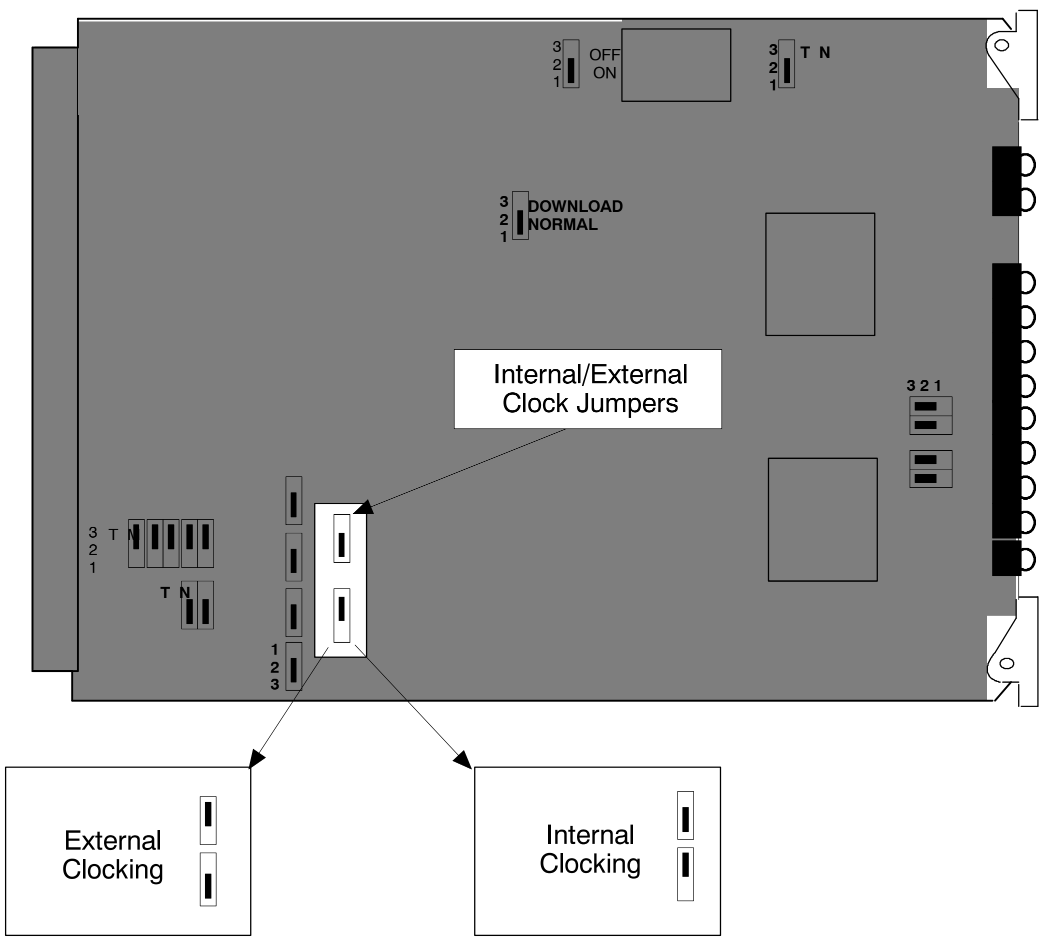


Figure 3.1.4. Clock Jumper Positioning on the Nonvolatile Memory Card

HIA Software

An HIA is configured with the CONFIGURE HIA screen. The CONFIGURE HIA screen contains the information necessary for HIA operation, as shown in Figure 3.1.5. Table 3.1.1 describes the fields on the screen.

Pressing [EXCHANGE] from the CONFIGURE HIA screen calls up the HIA STATUS screen. The HIA STATUS screen contains communications diagnostic information.

An HIA pair can support up to 40 links over the HIA connection. The number of available links can be viewed on the PEERWAY NODE screen for the HIA.

```
                              CONFIGURE HIA      23-Dec-96     11:47:12

Node number >10 (10 on PeerWay 1)       Other node   64 (32 on PeerWay 2)

PeerWay number   1                      Other PeerWay ⇒2

                 Node 10 Configuration Information
Program version  11.xx                  Boot version     x.xx

Distance penalty ⇒30                    Baud rate   ⇒614400

Clock src ⇒CLK_INT                      Link passing time  ⇒.25 s

Slot width  ⇒30                         Time correction  ⇒.00 sec/day
                                        Pass time   ⇒yes

Volume,Filename for Program ⇒xxx,xxxx

Volume,Filename for Config  ⇒xxx,xxxx

operation ⇒Save HIA configuration to config file
  ⇒Press <ENTER> to Begin
                                                                CONFIG 1
```

Figure 3.1.5. Configure HIA Screen

Table 3.1.1. Configure HIA Screen Fields

Field	Access Level	Description	Allowable Entries
Node Number	CONF	Specifies the PeerWay node address of the HIA.	1 to 992
Other node	N/A	Indicates the PeerWay node address of the other HIA during communication.	Display only
PeerWay number	CONF	Specifies the number of the PeerWay to which the HIA is connected. **NOTE:** If you make an entry in this field, all node numbers on this PeerWay are changed to reflect the new PeerWay number.	1 to 31 (Do not define a PeerWay as PeerWay 31.)
Other PeerWay	N/A	Indicates the number of the PeerWay that this HIA communicates with over the HIA link.	Display only
Program Version	N/A	Indicates the HIA software program version.	Display only
Boot Version	N/A	Indicates the boot software version.	Display only
Distance Penalty	CONF	Determines an imaginary distance for PeerWay routing. If information can be communicated to a destination over more than one route, it will take the path that has the lowest imaginary distance. Time-out (t) values for total distance penalties (dp) are: • if dp < 25, t = 2.5 seconds • if dp => 25, t = 30 seconds	1 to n **Default=3**
Baud rate	CONF	Specifies the communication speed used between HIAs. The "Baud rate" value is limited by the "Clock src" field entry, as listed below: Clock src entry — Baud Rate entry limit CLK_EXT — 614400 CLK_INT — 614400 CLK_NRZI — 40000 CLK_FM0 — 20000 CLK_FM1 — 20000	300 baud to 614400 baud

(continued on next page)

Table 3.1.1. Configure HIA Screen Fields (continued)

Field	Access Level	Description	Allowable Entries
Clock src	CONF	Specifies the type of clock signal communication used between HIAs. CLK_INT specifies that this HIA initiates the clock signal. CLK_EXT specifies that the other HIA initiates the clock signal. In order for two HIAs in an HIA pair to communicate, one HIA must be designated as "CLK_INT" and the other as "CLK_EXT". The other entries are not used. The entry must match the clock jumper position on the Nonvolatile Memory Card.	CLK_INT, CLK_EXT, CLK_NRZI, CLK_FM0, CLK_FM1
Link Passing Time	CONF	Specifies how often linking information and TIC messages are passed between PeerWays. A longer time should be used for a slow communication link.	.25, .5, 1, 2.5, 5 seconds
Slot Width	CONF	Specifies a time width used to tune performance. A value of 30 can be used for PeerWays under 2000 feet in length. Add a value of 1 for each additional 300 feet of PeerWay over 2000 feet. The higher the value, the more the PeerWay performance is reduced. This same field is also on the CONSOLE CONFIGURATION screen. A change made on one screen is automatically made on the other screen.	20 to 200
Time Correction	CONF	Specifies a time correction that allows the system clock to be adjusted in order to synchronize it with an external clock. Operates only in the console on the PeerWay with the highest number (that console is responsible for keeping time). This same field is also on the CONSOLE CONFIGURATION screen. A change made on one screen is automatically made on the other screen.	−327.00 to 327.00 seconds/day
Pass Time	CONF	Specifies whether or not the HIA connected to this PeerWay sends time synchronization TIC messages to the HIA connected to the other PeerWay. This field should have the same value for each side of the HIA. Generally, communications slower than 9600 baud should have "Pass Time⇒no".	yes, no **Default=yes**
Volume, Filename for Program	CONF	Specifies a program file for program file disk operations initiated with the "operation" field.	Entered in the form: node:volume,filename; volume,filename; or filename

(continued on next page)

Table 3.1.1. Configure HIA Screen Fields (continued)

Field	Access Level	Description	Allowable Entries
Volume, Filename for Config	CONF	Specifies program file for configuration file disk operations initiated with the “operation” field.	Entered in the form node:volume,filename; volume,filename; or filename
operation	CONF	Specifies an HIA operation. Use the [NEXT OPTION] key to scroll through the HIA operations. When the desired operation has been selected, press [ENTER]. Then cursor to the “Press <ENTER> to Begin” field and press [ENTER] to begin the operation.	Use [NEXT OPTION] key for choices.

Distance Penalties

Distance Penalty determines an imaginary distance for paths between PeerWays. It can be used to:

- Configure optimum communication paths between PeerWays
- Set timeout values

Determining Communication Paths

Figure 3.1.6 shows how message are sent when there are two paths between PeerWays. Each PeerWay transmits messages across the HIA path with lowest distance penalty.

When one path is more desirable than another, you can direct communication to that path by assigning it a lower distance penalty.

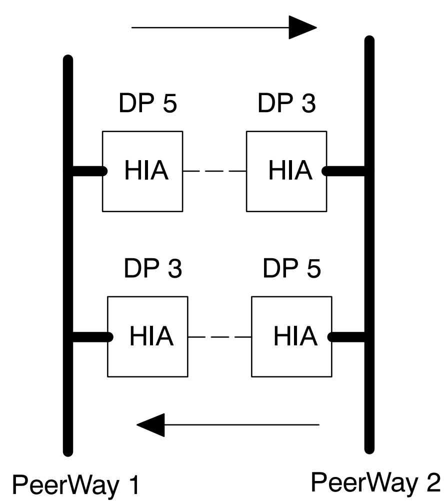


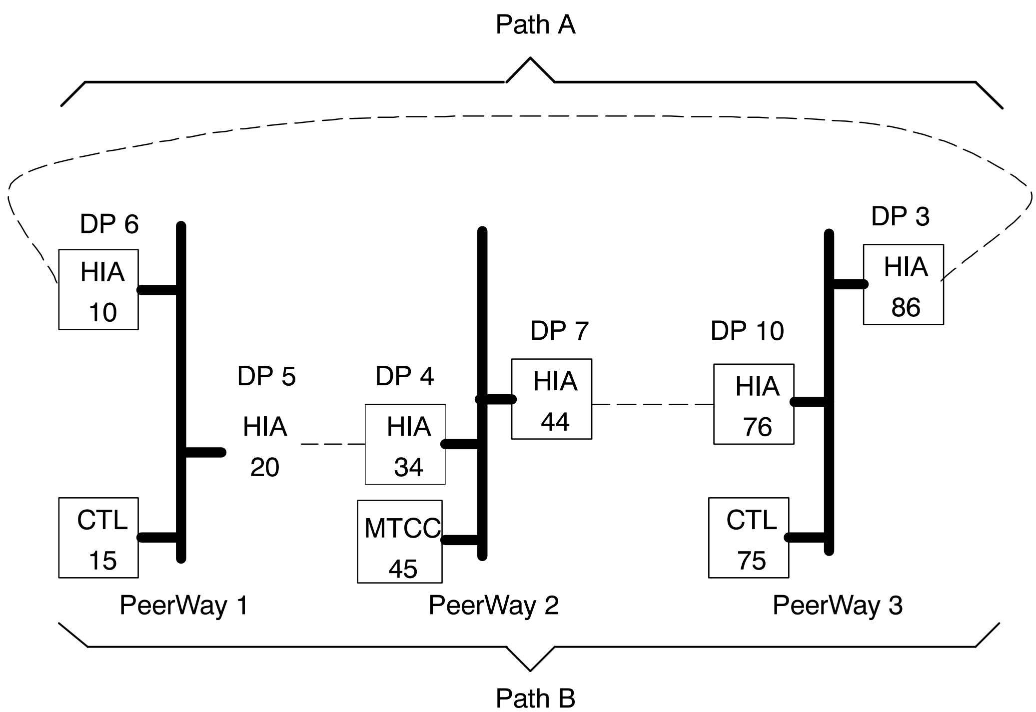
Figure 3.1.6. Selecting Paths with Distance Penalties

In Figure 3.1.7, distance penalties are used to control transmission of messages between three PeerWays. The distance penalties allow communication between the following PeerWays:

- PeerWay 1 and PeerWay 3 across Path A
- PeerWay 1 and PeerWay 2 across Path B
- PeerWay 2 and PeerWay 3 across Path B

Notice that PeerWay 1 and PeerWay 3 cannot communicate across Path B because Path B has a higher distance penalty than Path A. However, Path B could provide emergency communication between PeerWay 1 and PeerWay 3 if Path A failed.

The transmitting node adds the distance penalties for HIAs and PeerWays on each path to determine the path with the lowest distance penalty. The system automatically adds 2 distance penalties for each PeerWay on the Path.



Node	Path	Calculate DP	Total DP
15 to 75	A	2+3+2	7
15 to 75	B	2+4+2+10+2	20
75 to 15	A	2+6+2	10
75 to 15	B	2+7+2+5+2	18
15 to 45	B	2+4+2	8
45 to 15	B	2+5+2	9
45 to 77	B	2+10+2	14
75 to 45	B	2+7+2	11

NOTE: 2 distance penalties are added for each PeerWay in the path.

Figure 3.1.7. Coordinating Communications with Distance Penalties

Time-Out Retry

Distance penalties are also useful for controlling time-out delays between message retries. Each HIA node computes the total distance penalties between it and other points of communication on the network. There are two time-out delays based on total distance penalty.

- If distance penalty (dp) is less than 25, time-out (t) equals 2.5 seconds (if dp < 25, t = 2.5).
- If distance penalty (dp) is equal to or greater than 25, time-out (t) equals 30 seconds (if dp => 25, t = 30).

If the total distance penalty is equal to or greater than 25, communications between nodes take longer to recover following an upset in the PeerWay. In some cases where recovery generates many messages, it might be desirable to increase the time-out of some nodes to prevent PeerWay flooding.

In large systems with heavy message traffic, you can use a distance penalty greater than 25 to avoid overwhelming the system with alarm acknowledgements at the slowest node.

PeerWay Performance Screen

Figure 3.1.8 shows the PeerWay Performance screen. This screen is useful for monitoring the performance of the PeerWay and diagnosing PeerWay problems.

For more information on using the PeerWay Performance screen, see SV: 9-2.

❑ **To call up:**

- **PP** [ENTER] at the command line
- **PP** [PeerWay #] at the command line. Calls up the screen displaying the nodes on the specified PeerWay. Overwrites the nodes on the PeerWay Overview screen.

❑ **To access other screens:**

- [SELECT] on node number to see PeerWay Node screen

If you change the PeerWay or node numbers, wait at least 5 seconds for the screen values to update.

```
                            PEERWAY PERFORMANCE           25-Feb-96  12:07:37
 Node Msgs_Point_KBit PW_Q_Buffers Links Host+PW&S+PWto PWto+Srvr+Srvr CPU Thk
 #Type TxM,RxM TxK,RxK TxQ,RxQ,FreeOutIn toPW+Svce+Host Srvr+Svce+toPW Idl Bsy
→34
→35
→36 SRU 9,  12  7,  5   .   .   625  1 ,.  .   .    =27   3   +1    +19  0   .
→37
→38
→39
→40 HIA 47  39  29, 14  .   .   1250 . ,.  .   .    .     .   .     .    16  .
→41
→42 MTC 69, 70  23, 20  1,  .   250  1 ,.  12  +10  +0    .   .     .    16  .
→43 CTL 1,  1   5,  .   .   .   60   . ,.  .   .    .     .   .     .    20  .
→44
→45
→46
→47
→48 MTC 7,  7   7,  3   .   .   250  1 ,.  .   .    .    6   +0     +14  90  .
→49 CTL 2,  2   5,  .   .   .   125  . ,1  .   .    .    .   .      .    99  .
→50 MTC 66, 70  22, 21  .   .   250  1 ,.  .   .    .    0   +0     +17  19  ,
→51
→52
→53
→54
→55
→56
→58
 SUMpw  222,222 112, 73 =P5%         4, 1  =L2%   Data=xx%                  0
 Pway>2  IdleM 804=186% SlotW 35 Margin 508  Qual A 0   B 0  NdErr 0  Time 508
```

Figure 3.1.8. PeerWay Performance Screen

Appendix A:
Acronyms and Abbreviations

A

AIB	Analog Input Block
ANSI	American National Standards Institute
AOB	Analog Output Block
ASCII	American Standard Code for Information Interchange
ATPID	Autotuning Process Integral Derivative

B

BPS	Block Print Setup

C

CC	Command Console; Contact Controller
CCITT	Consultative Committee on International Telephone and Telegraphy (know know as the ITU-T [Telecommunications Standardization Sector of the International Telecommunications Union]
CIB	Contact Input Block
CJC	Cold Junction Compensator
COB	Contact Output Block
CPH	Counts per Hour
CRC	Cyclical Redundancy Check
CRT	Cathode-Ray Tube
CSP	CSprint
CTS	Clear to Send

D

DASMC	Discrete Auto Sequence Motor Controller
DASVC	Discrete Auto Sequence Valve Controller
DC	Direct Current
DCD	Data Carrier Detect

DCE	Data Circuit-Terminating Equipment
DDC	Direct Digital Control
DDMC	Dual Direction Motor Controller
DDSMC	Discrete Dual Speed Motor Controller
DDP	Disk Directory PeerWay
DEC	Digital Equipment Corporation
DIB	Discrete Input Block
DIN	Deutsche Industrie Normenausschuss (a German national standards organization)
DISC	Discrete Block
DMC	Discrete Motor Controller
DMVC	Discrete Motorized Valve Controller
DOB	Discrete Output Block
DP	Distance Penalty
DSR	Data Set Ready
DT	Dead Time
DTE	Data Terminal Equipment
DTR	Data Terminal Ready
DVC	Discrete Valve Controller

E

EIA	Electronic Industries Association

F

FEM	Front End Module
FIC	Field Interface Card

H

HIA	Highway Interface Adapter
Hz	Hertz

I

ID	Identifier; Identification
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output

L

LL	Lead/Lag
LSB	
LUN	Logical Unit Number

M

mA	Milliampere
MAN	Manual
MIB	Multiplexer Input Block
MLC	MultiLoop Card
MMSS	Minutes/Seconds
MPC	MultiPurpose Controller
mS	Millisecond
MSB	
MUX	Multiplexer
mV	Millivolt(s)

N

N/A	Not Applicable
NBS	National Bureau of Standards (American) – This organization is now known as the National Institute of Science and Technology.
N.C.	Normally Closed
N.O.	Normally Open

P

PDH	Plant Data Highway
PID	Process Integral Derivative
PIOB	Pulse Input/Output Block
PLC	Programmable Logic Controller
PLCB	Programmable Logic Controller Block
PLI	Piecewise Linear Interpolator
POLY	Polynomial
PPS	Pulses Per Second
PV	Process Variable

Q

QBI	QBUS to PeerWay Interface

R

R/B	Ratio/Bias
RBLC	Rosemount Basic Language Controller
RMT	REXXWARE Migration Toolkit
ROM	Read Only Memory
RTD	Resistance Temperature Detector

S

SCI	Supervisory Computer Interface
SIB	Smart Transmitter Input Block
SP	Setpoint
SRU	System Resource Unit
SS	Signal Selector
SSC	Single-Strategy Controller

T

TIB	Temperature Input Block
TOT	Stack Totalizer
TOTSP	Setpoint Totalizer
TX	Transmit

V

VA	Volts Ampere
VAX	Virtual Address Extension (a DEC family of computers)
VDC	Volt(s) Direct Current
VIB	Value Input Block
VLIM	Velocity Limiter
VMS	

FISHER·ROSEMOUNT

RS3 ™
PeerWay Interfaces Manual

Index

Numbers

A

B

C

D

E

F

H

I

L

M

N

O

P

R

S

T

U

V

W

X