ALLEN-BRADLEY

PLC-3 Family I/O Scanner Communication-Adapter Module

User Manual

Important User Information

Because of the variety of uses for this product and because of the differences between solid state products and electromechanical products, those responsible for applying and using this product must satisfy themselves as to the acceptability of each application and use of this product. For more information, refer to publication SGI-1.1 (Safety Guidelines For The Application, Installation and Maintenance of Solid State Control).

The illustrations, charts, and layout examples shown in this manual are intended solely to illustrate the text of this manual. Because of the many variables and requirements associated with any particular installation, Allen-Bradley Company cannot assume responsibility or liability for actual use based upon the illustrative uses and applications.

No patent liability is assumed by Allen-Bradley Company with respect to use of information, circuits, equipment or software described in this text.

Throughout this manual we make notes to alert you to possible injury to people or damage to equipment under specific circumstances.

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

Attention helps you:

- Identify a hazard
- Avoid the hazard
- recognize the consequences

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

Summary of Changes

Additional Information

In general, we improved the format and added greater detail to this manual. We have also corrected incorrect and confusing concepts throughout the manual.

The following table lists specific changes we made since:

We have:	To chapter/Appendix:
Added details concerning 230.4 kbps support	2
Updated LIST configuration capabilities and displays – added the new feature: DH+ Active Nodes to Status File 6	
Included the recommended numbers for assigning to slaves and masters in a Peer-to-Peer configuration link	

Added extensive text concerning DH addressing on:
• PLC-2 logical data
• PLC-3 logical binary
• Logical ASCII
  • data type
  • word range
• Additional examples of each type of addressing
	5

Added the feature of identifying remote stations in an assignment command with a symbol	
Added a new chapter: Diagnostic Methods	7
Expanded Binary Command Language	A

Using this Manual

Manual Objectives

This preface tells you how to use this manual properly and efficiently for the tasks you have to perform. Read this chapter before you program a PLC-3® family programmable controller.

What this Manual Contains

Chapter/ Appendix	Title	What's Covered
1	Using this Manual	Manual's purpose, audience, and contents
2	Scanner Hardware and Installation	Hardware components and procedures for installing a scanner
3	Configuring the Scanner Through LIST	Using the LIST function to configure the scanner
4	I/O Communication	Timing and programming considerations when using the scanner for I/O communication
5	DH and DH+ Communication	Concepts, programming, and timing considerations
6	Addressing DH and DH+ Data Transfers	Specifying data addresses in message commands that communicate over a DH or DH+
7	Programming DH and DH+ Message Procedures	Operating the message instruction to communicate over a DH or DH+
8	Diagnostic Methods	Techniques and indicators for monitoring scanner status
A	Binary Command Language	Commands used by an external control device to communicate with a PLC-3 family controller through channel 0 of the front panel
B	DH/DH+ Error Codes	DH and DH+ error codes and meanings
C	DH and DH+ Command Set	DH and DH+ commands supported by the scanner
D	Specifications	Scanner operating and environmental specifications

Audience

In this manual, we assume that you know how to program and operate an Allen-Bradley PLC-3 or PLC-3/10® programmable controller system. If you are not familiar with these controllers, refer to the following publications:

Publication	Title
1770-6.2.2	Data Highway Cable and Data Highway Plus Installation Manual
1770-6.5.15	PLC-3 Industrial Terminal (cat. no. 1770-T4) User's Manual
1770-6.5.16	Data Highway/Data Highway Plus Protocol Command Set User's Manual
1771-6.5.83	Remote I/O Adapter Module (cat. no. 1771-ASB) User's Manual
1775-6.3.1	PLC-3 Backup Concepts Manual
1775-6.4.1	PLC-3 Family Programming Manual
1775-6.7.1	PLC-3 Family Installation and Operation Manual
1784-6.5.1	Industrial Terminal (cat. no. 1784-T50) User's Manual
6200-6.5.3	PLC-3 Programming Software User's Manual

Vocabulary

We refer to certain types of equipment and terms throughout this manual. To make the manual easier for you to read and understand, we avoid repeating full product names where possible.

We refer to the:	As the:
I/O Scanner-Communication Adapter Module (cat. no. 1775-S5, -SR5)	scanner
PLC-3 or PLC-3/10 programmable controller system	controller or the system
ladder-diagram or user program that controls PLC-3 processor operation	ladder program
Data Highway link	DH link
Data Highway Plus™ link	DH+™ link
Industrial terminal system, e.g., Allen-Bradley 1784-T4, -T47, -T53, or 6160-T60 terminals, or an computer with 6200 software	programming terminal

In addition, you may encounter words in different typefaces. We use these conventions to help differentiate descriptive information from information that you enter while programming your scanner.

- The Return key looks like this (boldface and in brackets):

  [Return]

- Words or commands that you enter appear in boldface. For example:

  $047   **$FILE_A**

- Variables that you enter appear in italics. For example:

  Type *link*:*node* and press **[Return]**

- Messages or prompts on the screen look like this:

  ENTER STATION AND INPUT FILE #>

- "Type" means — type in the information

- "Enter" means — type in the information and then press the **[Return]** key.

Chapter 1

Scanner Hardware and Installation

Chapter Objectives

This chapter describes:

- the features and functions of the 1775-S5, -SR5 scanners
- the hardware components on the scanner
- how to install the scanner
- how to connect and configure the scanner to a(n):
  - backup system
  - Data Highway (DH) or Data Highway Plus (DH+) communication network
  - I/O adapter for I/O scanning
  - programming terminal
  - scanner in another PLC-3 or PLC-3/10 chassis

Scanner Features and Functions

The 1775-S5, -SR5 scanner has four communication channels to communicate with any of the following:

- I/O adapters (cat. no. 1771-AS, -ASB)
- node adapter devices
- a scanner in a backup PLC-3 or PLC-3/10 processor
- PLC-5™ products in adapter mode
- up to six PLC-3 or PLC-3/10 processors

On channel 4, you can communicate with these additional devices:

- up to 64 stations on a DH or DH+ link
- a programming terminal

In the following chapters, we describe how to configure and operate these channels. Table 1.A summarizes the operating features and functions.

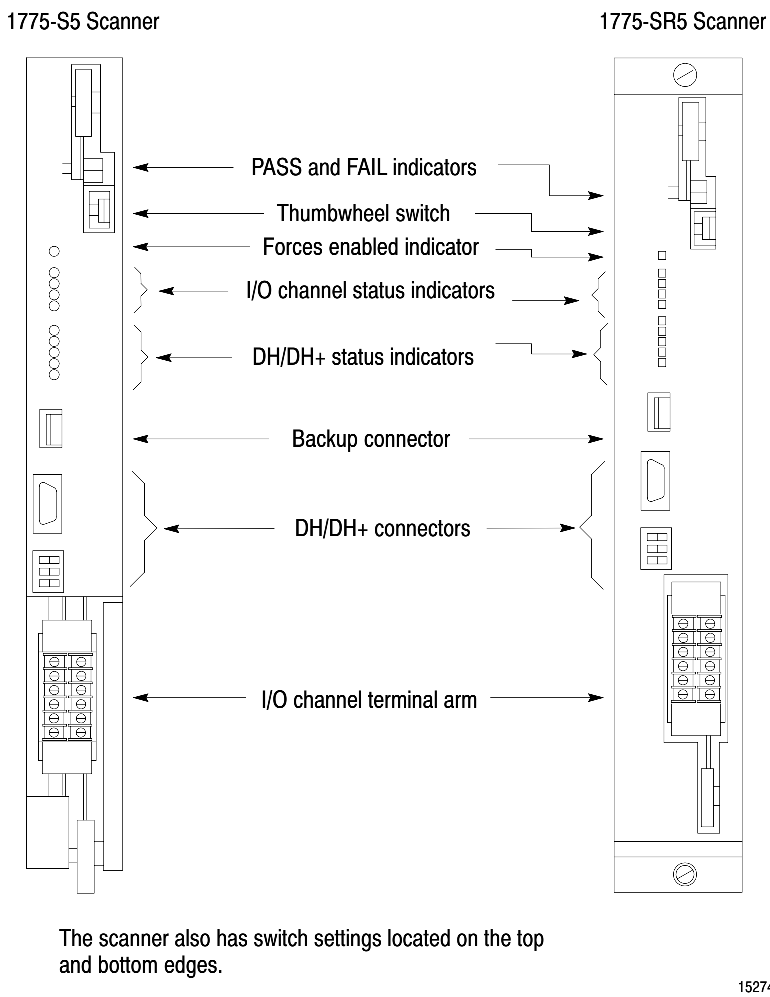
Table 1.A
1775-S5, -SR5 Scanner Features and Functions

Features	Functions
Four I/O communication channels	Communicate with I/O adapter modules in an I/O chassis. You can connect up to 32 I/O chassis to one I/O channel. The scanner can communicate with up to 4,096 I/O.
I/O scan priority	Scan the I/O chassis according to a sequence that you select.
DH/DH+ channel	Communicate with other Allen-Bradley controllers and/or computers on a DH or a DH+ channel. Also provides direct connection for a programming device.
Status indicators	Keeps you informed on the scanner's status including: • forces in the system • DH/DH+ channel • I/O communication channels
Thumbwheel switch	Distinguishes one scanner from another. You can have up to: • 15 scanners in a PLC-3 system (S5 scanner) • 2 scanners in a PLC-3/10 system (SR5 scanner)
Backup connector	Transfers control to a backup PLC-3 or PLC-3/10 system if a fault shuts down the primary system.
Terminal arm	Makes connections to: • 1771 I/O chassis up to 10,000 cable feet away from the scanner via twinaxial cable (cat. no. 1770-CD) for I/O communication • Communication channels on scanners in up to six separate PLC-3 or PLC-3/10 systems via twinaxial cable for peer-to-peer communication • Communication channel on a scanner in a backup PLC-3 or PLC-3/10 system via twinaxial cable for backup communication
Message procedure commands	Easy to use commands that you can use to transfer data over a DH or DH+ link and combine in procedures for: • complex logic decisions, looping, and nesting • symbolic representation of data and addresses • embedded arithmetic expressions and logic operations • error checking and reporting • decimal, octal, or binary-coded-decimal data entry • functions for converting data values to and from BCD

Hardware Features

Figure 1.1 shows the hardware components on your scanner.

Figure 1.1
Hardware Features on the Scanner



Pass and Fail Indicators

The indicators labeled PASS and FAIL (Figure 1.2) keep you informed about the general condition of the scanner:

PASS (green)	FAIL (red)	Meaning
on	off	normal operation
off	on	module fault
on	on	power-up or system reset
off	off	system is not on or module has lost power

Forces Enabled Indicator

The yellow indicator labeled FORCE (Figure 1.2) illuminates when I/O forcing is enabled in the system.

I/O Channel Status Indicators

The four green indicators labeled CH1, CH2, CH3, and CH4 (Figure 1.2) correspond to one of the four I/O channels.

Indicator	Status	Description
CHx Configured for I/O scanning	ON	Communication between scanner module and the I/O chassis on the corresponding I/O channel is properly established.
	FLASHING	There is a fault on one or more of the I/O chassis on the corresponding I/O channel.
	OFF	No I/O chassis are configured on the corresponding I/O channel or the channel is inactive.
CHx Configured for peer-to-peer communication	ON	The channel is functioning properly.
	FLASHING	The input file is too small at the processor receiving data. The slave or master does not exist. Communication retry.
	OFF	The channel is inactive.
CHx Configured for backup communication	ON	The channel is functioning properly.
	FLASHING	The input file is too small at the processor receiving data. The partner is not responding.
	OFF	The channel is inactive.

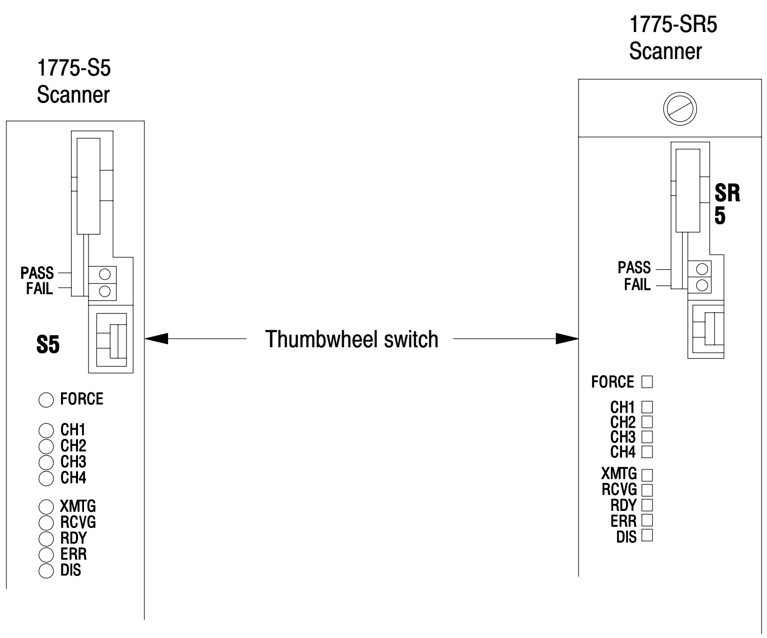
DH/DH+ Status Indicators

The five indicators labeled XMTG, RCVG, RDY, ERR, and DIS (Figure 1.2) show you the status of the DH or DH+ channel.

Indicator	Color	When this led is on, (the scanner is)
XMTG	green	transmitting a message.
RCVG	green	receiving a message.
RDY	green	ready to transmit a message.
ERR	red	programming or communication error detected.
DIS	yellow	DH/DH+ connectors are disabled or duplicate DH+ station address if blinking.

When the scanner is polling, both the XMTG and RCVG LEDs turn on.

Figure 1.2
Indicator Locations for 1775-S5 and 1775-SR5 Scanner Modules



Configuring the Scanner Hardware

Before you install the scanner, you need to configure the module. You use the connectors, terminal arm, and switches when:

- setting the thumbwheel and switches
- installing the scanner

We describe these connections and configurations in the following sections.

Setting the Thumbwheel and Switches

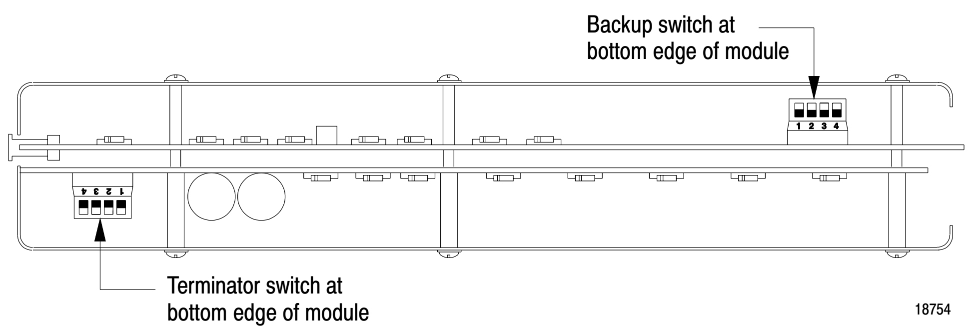
Before inserting the scanner into a chassis:

1. Set the thumbwheel switch to a unique number (1 to 15) to enable the processor to distinguish one scanner from another. You must have a scanner with the thumbwheel set for 1.

2. Make the necessary switch settings.

Figure 1.3 shows the location of the backup switches for the 1775-S5 and 1775-SR5 scanner moduels.

Figure 1.3
Location of Backup Switches for 1775-S5 and -SR5 Scanner Modules

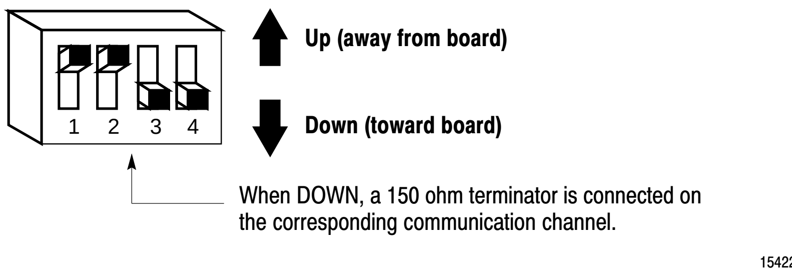


Setting aTerminator on a Communication Channel
On the bottom edge near the front of the scanner is a set of four switches. With these switches, you can connect a terminator across the line when the scanner is an end device on a DH, DH+, I/O, peer-to-peer, or backup communication link (Figure 1.4) on the corresponding channel.

Important: When using internal terminators, removing the terminal swing arm from the scanner causes the communication channels to be unterminated. You may want to leave the switch up and install the optional 1770-XT external 150 ohm terminator.

When a channel is configured for a communication rate of 230.4 kbps, the corresponding terminator switch must be up away from the board. An external 82 ohm terminator must be used in this configuration.

Figure 1.4
Setting a Terminator on a Communication Channel



Setting the Protocol, Communication Rate, and Station Address for Channel Four

On the top edge near the rear of the scanner are two sets of switches. You can use these two sets of switches to set the protocol, communication rate, and station address (Figure 1.5 and Figure 1.6).

You can set the protocol for:

- DH link
- DH+ link
- I/O link
- or LIST configurable (selections in the LIST set the protocol)

If you set the protocol switches for:	Then you must specify the station address by:
DH or DH+	setting the set of eight switches next to the protocol and communication rate switches.
list configurable	using the station number selection in LIST. For more information, see chapter 2.

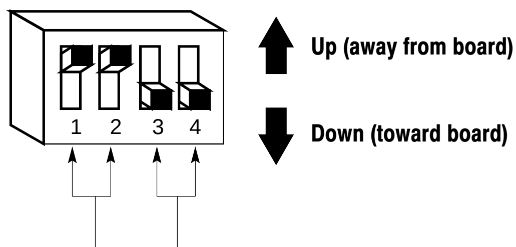
You can set the communication rate to the following:

Select:	For:
57.6 kbps	I/O, peer-to-peer, backup, DH, or DH+ communication
115.2 kbps	I/O, peer-to-peer, backup, or DH+ communication only
230.4 kbps	I/O, peer-to-peer, or backup communications

Not all DH+ stations are capable of operating above 57.6 kbps. You must reference the appropriate users manual for the maximum communication rate for each device.

ATTENTION: Channel 4 may not be configured for DH or DH+ when operating any of the other communication channels at 230.4 kbps. Otherwise, I/O rack retries and missing inputs in a listen only mode backup system may result.

Figure 1.5
Setting the Protocol, Communication Rate, and Station Address for Channel Four



Communication Rate (bps)	Switch	
	1	2
57.6k	down	down
115.2k	up	down
230.4k	down	up

Protocol	Switch	
	3	4
DH	down	down
DH+	up	down
I/O scan	down	up
Configurable through LIST	up	up

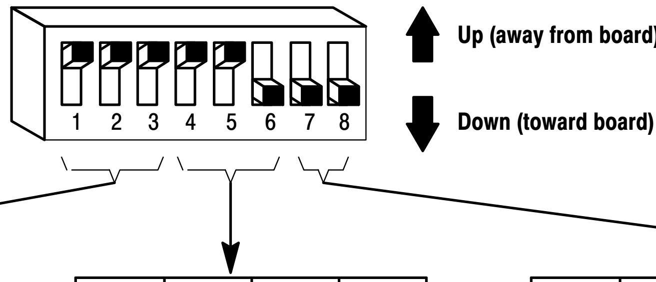
If you set the protocol switches for DH or DH+ communication, you must set the station number (Figure 1.6). Valid station numbers range from 000 to 376 octal for DH and 00 to 77 octal for DH+.

Switch numbers:	correspond to the:
7 and 8	first digit (MSD)[1]
4, 5 and 6	second digit
1, 2 and 3	third digit (LSD)[2]

[1] MSD = Most Significant Digit
[2] LSD = Least Significant Digit

Figure 1.6
Setting the Station Number for the DH or DH+ Channel

If you are using channel 4 for DH or DH+ communication, then set the station number with these switches.



Digit	1	2	3
0	down	down	down
1	up	down	down
2	down	up	down
3	up	up	down
4	down	down	up
5	up	down	up
6	down	up	up
7	up	up	up

(Least Significant)

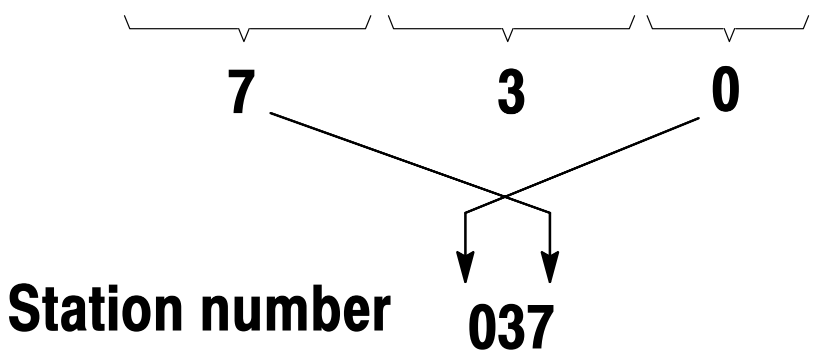
Digit	4	5	6
0	down	down	down
1	up	down	down
2	down	up	down
3	up	up	down
4	down	down	up
5	up	down	up
6	down	up	up
7	up	up	up

Digit	7	8
0	down	down
1	up	down
2	down	up
3	up	up

(Most Significant)

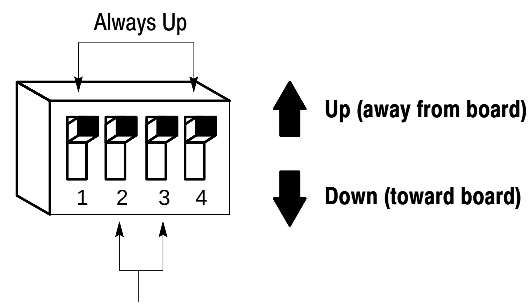
Example:
The switch settings for station number 037 are:

1	2	3	4	5	6	7	8
up	up	up	up	up	down	down	down



Setting Backup System Functions
On the bottom edge (Figure 1.7) near the rear of the scanner is a set of 4 switches. If you are using a backup configuration, then you can set switches 2 and 3 to define the switchover type. For detailed information on backup, refer to the PLC-3 Backup Concepts Manual (publication 1775-6.3.1).

Figure 1.7
Setting Backup System Functions



Switchover Type	Switch 2	Switch 3	System (primary or backup)
Auto with controllable switchback	down	up	both
Auto with no switchback	down up	up up	primary backup
Manual	up	down	both

[1] If you are not operating a backup system, then set switches 2 and 3 to the UP position.

The backup switches for scanners other than thumbwheel number 1 should be set away from the board.

Installing the Scanner

After you configure the scanner, insert the scanner into any slot of a:

- PLC-3 processor chassis (cat. no. 1775-A1, -A2)
- PLC-3/10 processor chassis (cat. no. 1775-A3)

PLC-3 and PLC-3/10 CPUs, scanners, and memories are not interchangeable.

The chassis electromechanically interlocks helping to guard against inserting or removing modules while power is on.

ATTENTION: Do not change the thumbwheel setting on a scanner while power is on. This could result in equipment damage.

PLC-3 and PLC-3/10 systems require a scanner with the thumbwheel set to 1. If you are using a multi-chassis system, the scanner set to 1 must be in the chassis with the front panel.

You can operate a 1775-S5 scanner with a 1775-S4A, -S4B scanner in a PLC-3 system and a 1775-SR5 with a 1775-SR scanner in a PLC-3/10 system. However, note the following cautions.

ATTENTION: You can replace a 1775-S4A, -SR scanner with a 1775-S5, -SR5 scanner, however, you must:

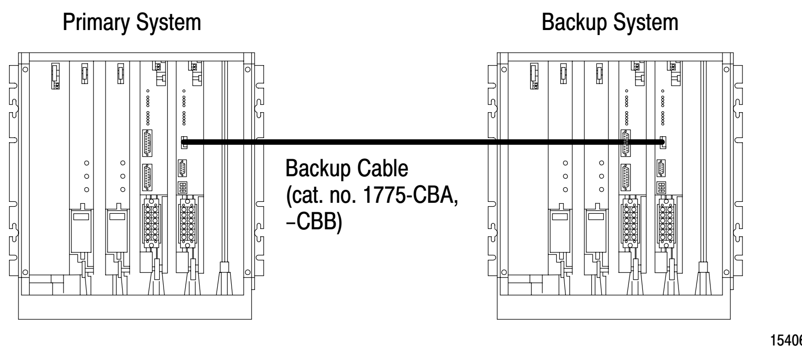
- perform a logical save of memory without saving the module status area (E2)
- clear memory after replacing one scanner with another.
- restore the previously stored memory
- configure each 1775-S5, -SR5 scanner communication channel in LIST.

Failure to observe this caution could result in equipment damage and/or undesired machine operation.

Connecting to a Backup System

With a backup system, if a major fault occurs, the primary system shuts down and the backup system takes over the outputs to enable your process to continue. To set up a backup system, connect a backup cable (cat. no. 1775-CBA, -CBB) from the 6-pin connector labeled BACK UP on scanner number 1 in a primary system to the backup connector on scanner number 1 in a backup system (see Figure 1.8).

Figure 1.8
Connecting the Scanner to a Backup System



ATTENTION: When using a backup system:

- Do not backup a 1775-S5, -SR5 scanner with a 1775-S4A, -S4B, or -SR scanner.
- If you are using manual switchover and switchover occurs, you must wait at least 60 seconds before switching back to the primary system.

Failure to observe these cautions could result in equipment damage and/or unpredictable machine operation.

For detailed information on installing and operating a backup system, refer to the PLC-3 Backup Concepts Manual (publication 1775-6.3.1).

Connecting to a DH or DH+ Network

The scanner enables the connectors labeled DH/DH+ when you set communication channel 4 for a DH or DH+ link. You can use the:

- 9-pin connector to connect the programming terminal for programming the processor on a DH+ network
- 3-pin connector to connect the controller to a DH or DH+ netwrok

Programming Terminal Connections
The 9-pin connector provides direct connection via the Industrial Terminal Processor Cable.

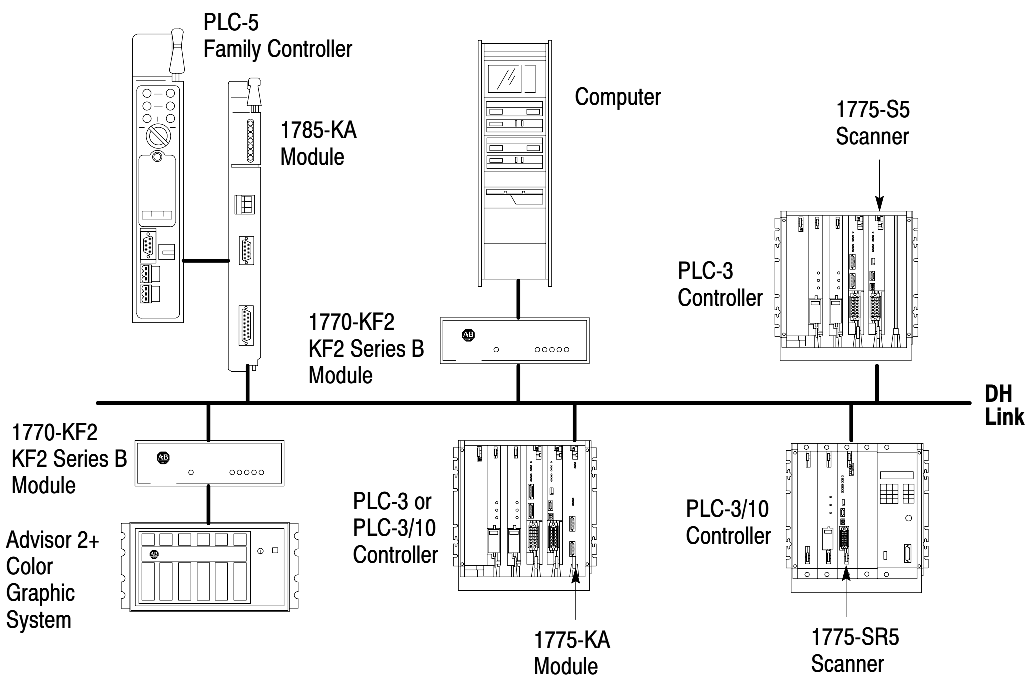
Cable	Communication Board
1784-CP5	1784-KTK1
1784-CP	1784-KT

For detailed information on installing and operating the Industrial Terminal (cat. no. 1784-T50) with the controller, refer to the Industrial Terminal T53 User's Manual (publication 1784-6.5.1).

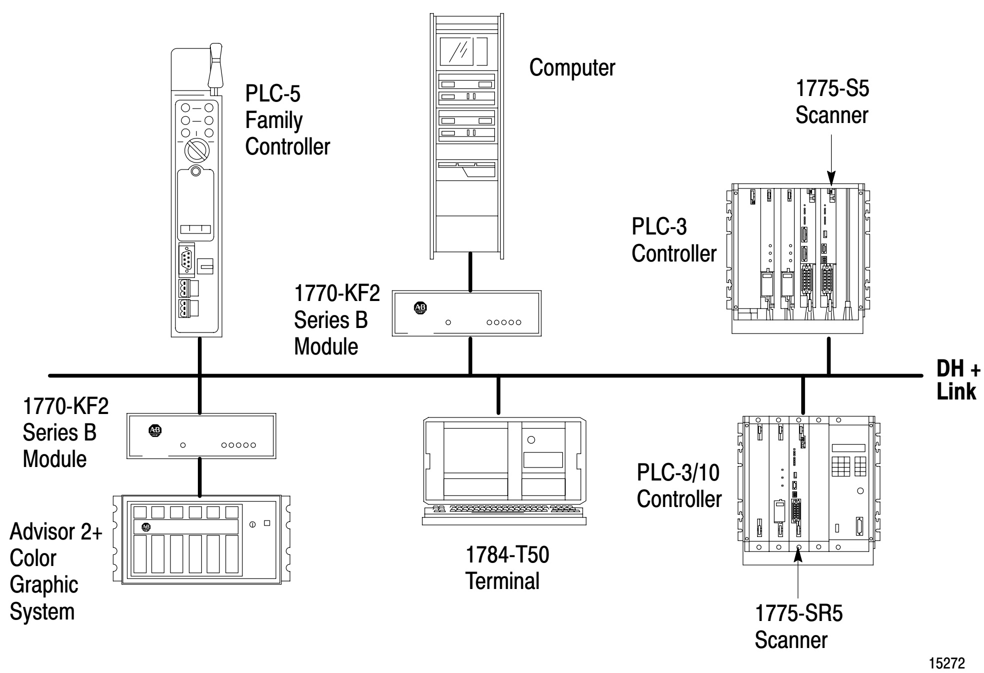
DH/DH+ Connections
The 3-pin connector provides direct connection via Twinaxial Cable (cat. no. 1770-CD) to a DH or DH+ link (Figure 1.9). For detailed information on installing a DH or DH+, refer to DH Cable Assembly and Installation Manual (publication 1770-6.2.2).

Figure 1.9
Example of DH and DH+ Configurations

Data Highway (DH) Link

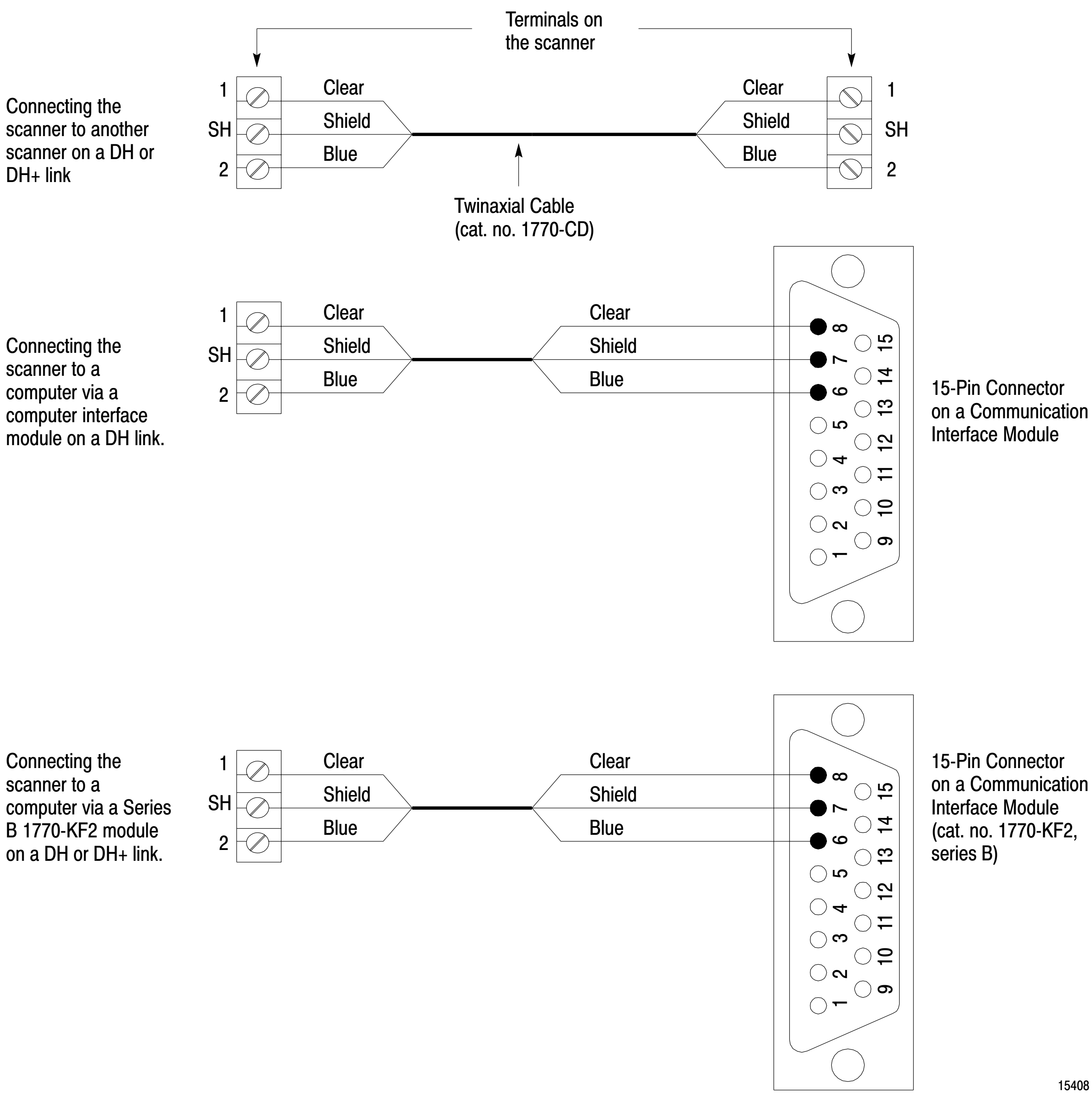


Data Highway Plus (DH+) Link



To make connections to the scanner, connect the 1770-CD cable to the screw terminals on the 3-pin connector (Figure 1.10).

Figure 1.10
Connecting the Scanner to a DH or DH+



The scanner has an on-board switch-selectable terminator. If the scanner is an end device in the DH or DH+ link, set the terminator switch corresponding to I/O channel 4 (see "Setting the Thumbwheel and Switches," page 1-6).

Connecting to the I/O Channel Terminal Arm

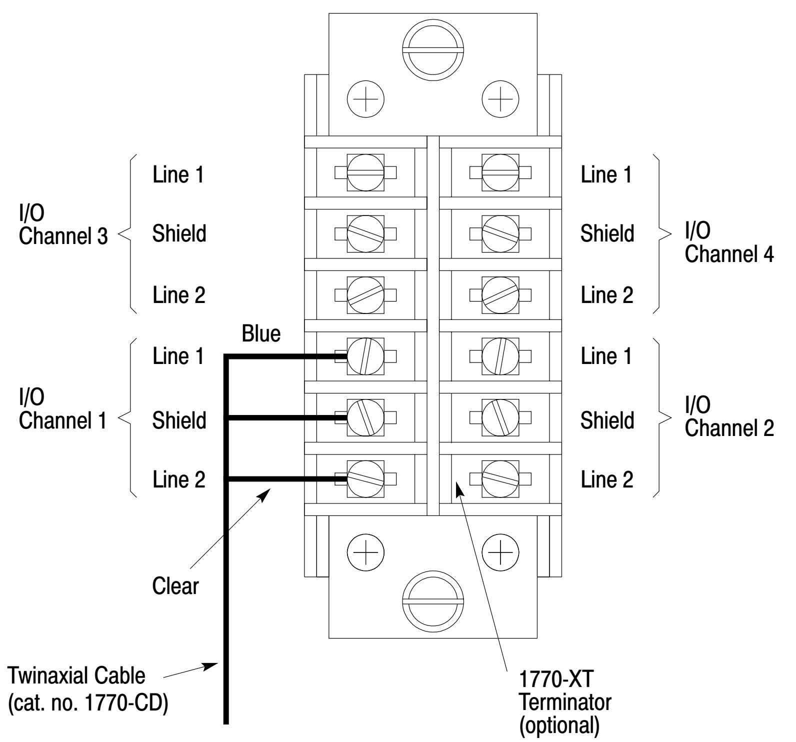
The terminal arm provides connections for four I/O communication channels (Figure 1.11). Functions of these channels include:

- I/O communication
- backup communication
- peer-to-peer communication

If you select:	Then the maximum I/O channel cable length can be:
57.6 kbps	10,000 feet
115.2 kbps	5,000 feet
230.4 kbps	2,000 feet

Important: If you set the protocol for channel four to a DH or DH+ link, the scanner automatically disables the channel four terminals, and enables the DH/DH+ connectors (see “Connecting to a DH or DH+ Network,” page 1-13).

Figure 1.11
I/O Channel Terminal Arm Connections



Connects to a Remote I/O Device for I/O communication or a scanner in another PLC-3 or PLC-3/10 chassis for peer-to-peer or backup communication.

If the scanner is at the end of the channel, set the terminator switch corresponding to the channel number. If the channel is configured for I/O at 230.4 kbps, disable the terminator for the channel and install an external 82 ohm terminator

15409

Electrostatic Discharge

ATTENTION: Electrostatic discharge can degrade performance or damage the module.

Electrostatic discharge can damage integrated circuits or semiconductors in the scanner if you touch backplane connector pins. It can also damage the scanner when you set configuration plugs and/or switches inside the module. Avoid electrostatic damage by observing the following precautions:

- Touch a grounded object to rid yourself of charge before handling the module.
- Do not touch the backplane connector or connector pins.
- When not in use, keep the module in its static shield bag.

Chapter 2

Configuring the Scanner through LIST

Chapter Objectives

You use LIST function to configure the scanner. This chapter describes the LIST selections. Read this chapter to learn how to:

- access the LIST function
- configure the front panel and channel 0 through scanner number 1
- configure communication channels for I/O, backup, or peer-to-peer communication
- configure communication channel 4 for I/O, DH, or DH+ communication

Accessing the LIST Function

You can access the LIST function for the scanner through a programming terminal or the data access panel on the PLC-3 or PLC-3/10 main chassis (cat. nos. 1775-A1, -A3). Refer to the PLC-3 Family Programmable Controller Installation and Operation Manual (publication 1775-6.7.1) for detailed information on operating the LIST function.

Important: The scanner limits the number of executing LIST functions to four for any PLC-3 or PLC-3/10 system.

Figure 2.1 shows you the parameters that you select for the scanner. You access these selections by doing the following:

1. Enter LIST.

The system displays the initial LIST menu.

2. Select option 6 for MODULE STATUS.

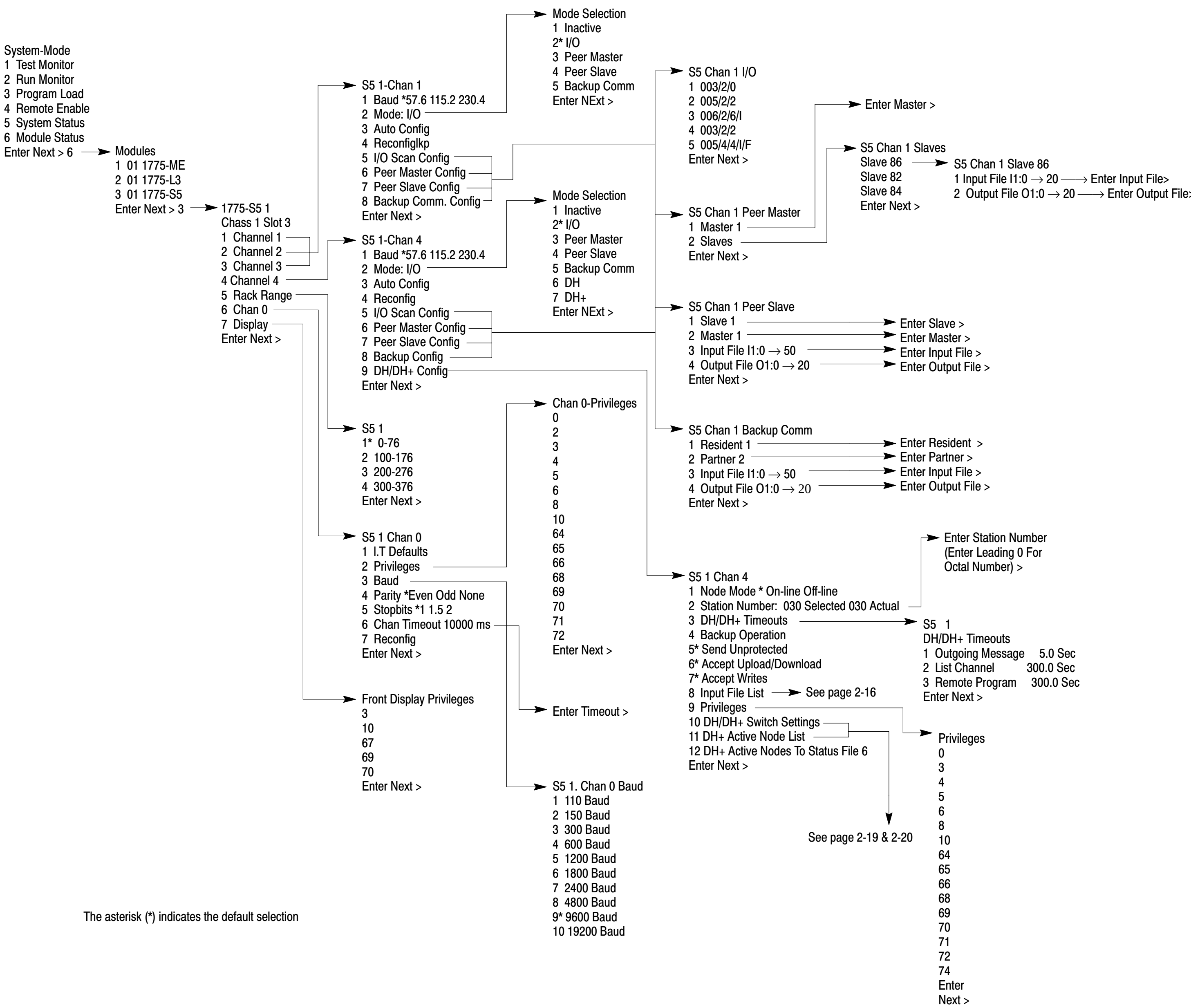
3. Select the number corresponding to the scanner that you want to configure.

The system displays the following menu:

```
1775-S5 1
CHASSIS 0 SLOT 3
1 CHANNEL 1
2 CHANNEL 2
3 CHANNEL 3
4 CHANNEL 4
5 RACK RANGE
6 CHAN 0
7 DISPLAY
ENTER NEXT >
```

The heading for this menu shows the scanner's thumbwheel setting and slot location in the processor chassis. Selections 6 and 7 only display for scanner number 1. We describe these selections in the rest of this chapter.

Figure 2.1
LIST Selections for the Scanner



Configuring the Communication Channels

To configure channels 1 through 4, you select the following parameters:

- communication rate (bps) (page 2-3)
- operating mode (page 2-4)
- auto configuration (page 2-4)
- reconfiguration (page 2-5)
- I/O scan configuration (page 2-6)
- peer-to-peer master configuration (page 2-8)
- peer-to-peer slave configuration (page 2-8)
- backup communication configuration (page 2-9)
- DH or DH+ configuration (channel 4 only) (page 2-10)

Communication Rate

You can select one of the following communication rates for the corresponding communication channel:

If you select:	Then the maximum I/O channel cable length can be:
57.6 kbps	10,000 feet
115.2 kbps	5,000 feet
230.4 kbps	2,000 feet

Important: If you are configuring channel 4, note the following:

- You must set the protocol switches for LIST configurable for the scanner to modify this selection.
- For DH communication, the scanner communicates at 57.6 kbps only.
- For DH+ communication, the scanner communicates at 57.6 kbps or 115.2 kbps. Not all DH+ stations are capable of operating above 57.6 kbps. You must reference the appropriate users manual for the maximum communication rate for each device.

ATTENTION: Channel 4 cannot be configured for DH or DH+ when operating any of the other communication channels at 230.4 kbps. Otherwise, I/O rack retries and missing inputs in a listen only mode backup system can result.

Operating Mode

You can select one of the following operating modes for the corresponding communication channel:

- inactive
- I/O scan
- peer-to-peer master
- peer-to-peer slave
- backup communications
- DH communications (channel 4 only)
- DH+ communications (channel 4 only)

These selections identify what the channel is being used for. The default is I/O scan.

An asterisk (*) displays next to the current mode of the channel. To change the mode, you select the number that corresponds to the desired mode and reconfigure the channel.

For channel 4, the protocol switch must be selected for LIST configurable for you to change the operating mode through LIST (refer to "Setting the Protocol, Communication Rate, and Station Address for Channel Four," page 1-7).

Important: If you are not using a communication channel, set the operating mode for the channel to inactive. This causes the scanner to stop communicating through the channel improving the communication times on the other active channels.

Auto Configuration

Auto configuration first creates a new I/O chassis list in which each I/O chassis has equal priority with no attributes assigned. Then the scanner reconfigures.

The scanner performs an auto configure at power-up for an I/O-scan-configured channel that has no entries in its I/O chassis scanning sequence list.

When forming the I/O chassis list during an auto configuration, the scanner polls all valid addresses. If the scanner receives response to an address, it adds that address to the list. To assign attributes or priorities to the I/O chassis, add them manually through LIST and reconfigure the channel.

You can only perform an auto configuration when:

- the controller is in program load mode
- power is applied to the I/O chassis
- the processor restart lockout switch is set to allow the I/O chassis to be restarted from the processor

If power is not applied to the I/O chassis, the processor attempts to perform an auto configure, and since the I/O chassis does not respond, the scanner does not enter it in the I/O chassis list. For an entry to get into the I/O chassis list in auto configure, a valid path must exist between the scanner and the I/O adapter module for the I/O chassis.

ATTENTION: You can replace a 1775-S4A, -SR scanner with a 1775-S5, -SR5 scanner, however, you must:

- perform a logical save of memory without saving the module status area (E2)
- clear memory after replacing one scanner with another.
- restore the previously stored memory
- configure each 1775-S5, -SR5 scanner communication channel in LIST.

Failure to observe this caution could result in equipment damage and/or undesired machine operation.

Reconfiguration

The reconfiguration selection sets configuration parameters for the scanner. The reconfigure selection implements changes that you have made in LIST for the communication channel. For example, suppose you list an I/O chassis 3 times in the I/O chassis scanning sequence list, and you change the list to include that chassis 6 times. The scanner does not change its polling sequence until you select reconfigure for the I/O channel.

Important: When selecting reconfigure, the scanner executes a reconfiguration of the selected channel only. Also, when you power-up the PLC-3 controller, a reconfiguration automatically executes.

If an asterisk is next to the reconfiguration selection, changes have been requested but not implemented. Upon selecting reconfigure, the scanner implements the changes and removes the asterisk.

ATTENTION: Selecting reconfiguration while the controller is in the run mode and executing block transfers could cause a bad address fault.

If this situation occurs, you should:

- Clear the bad address fault in a fault routine.
- Clear the block transfer error bit.
- Restart the block transfer in the ladder program

I/O Scan Configuration

When you configure an I/O channel for I/O scan, you can change the order and priority that the scanner scans the I/O chassis. You do this by listing the I/O chassis in the order that you want the scanner to communicate with them. This allows you to assign a higher priority to some I/O chassis than to others by listing the higher priority chassis more than once.

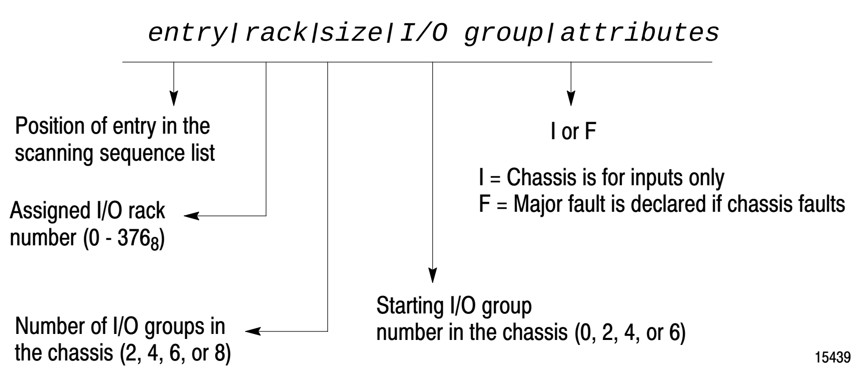
For example, suppose your I/O chassis scanning sequence list has six entries, and entries 1 and 4 are the same. Then the scanner updates the chassis listed as entries 1 and 4 twice as often as the other chassis.

Each I/O chassis is listed once in the default configuration at initial power-up or after auto configure. You can alter the list to repeat a chassis as often as desired, provided that the list contains no more than 32 entries.

An asterisk (*) appearing before an entry in the list indicates that the corresponding I/O chassis or adapter is faulted.

To enter a chassis into the I/O chassis scanning sequence, use the format shown in Figure 2.2.

Figure 2.2
Entering a Chassis into the I/O Chassis Scanning Sequence List



To delete entries from the list, press:

`entry/d [Return]`

When forming this list, remember:

- Selection of I/O rack numbers greater than 37_8 increases the program scan time.
- I/O rack numbers that are not assigned consecutively cause greater memory requirements because memory has to be allocated for the unused racks.
- You can connect up to 32 I/O adapters to one I/O communication channel on a scanner.
- You can assign up to 16 different I/O rack numbers to one scanner.
- For duplicate I/O addressing (complementary I/O or parallel output configurations), you must have two chassis with the same assigned I/O rack number and I/O group number on **different** I/O communication channels of the same scanner.

  Refer to the PLC-3 Family Controller Installation and Operation Manual (publication 1775-6.7.1) for detailed information on complementary and duplicate I/O configurations.

ATTENTION: Do not assign the same I/O rack number and I/O group number to two chassis on the same I/O communication channel. Failure to observe this warning could result in equipment damage and/or unpredictable machine operation with injury to personnel.

- The controller uses assigned I/O rack numbers 77_8, 177_8, 277_8, 377_8 for internal communication. Do not assign these numbers to an I/O rack (although you can use the associated addresses for internal storage).

For detailed information on 1/2-slot, 1-slot, and 2-slot I/O addressing considerations, refer to the Remote I/O Adapter (Cat. No. 1771-ASB) User’s Manual (publication 1771-6.5.83).

Peer-to-Peer Master Configuration

When you configure a channel as a master on a peer-to-peer communication channel, you must enter the following parameters:

Parameter	Description
Master number	identifies the master for communication with the slaves. Each peer-to-peer communication channel can have one master only. The master number and all slave numbers must be unique and selected from numbers 81 through 89.
Slave numbers	identify the slaves that communicate with the master. Each master can communicate with up to 6 slaves. The slave number must be unique and selected from numbers 81 through 89.
Input files	identify the files that receive data from each slave. You must specify an input file for each slave in the input image section of the data table. If you do not make a selection, the file defaults to input file 1. The files also must be large enough to store the data from each slave's output file. If the file is not large enough, the processor declares a minor fault.
Output files	identify the files that send data to each slave. You must specify an output file for each slave in the output image section of the data table. If you do not make a selection, the file defaults to output file 1.

LIST displays the size of the input and output files. For example, I1:0 $\rightarrow$ 19 shows that input file 1 contains 20 octal words. You can create a larger file by entering the last word desired in the file when prompted for the file number in LIST. By entering I1:50 for the input file, the system will automatically increase input file 1 to 50 octal words.

To delete slave entries from the list, press:

```
slave entry /d  [Return]
```

Peer-to-Peer Slave Configuration

When you configure a channel as a slave on a peer-to-peer communication channel, you must enter the following parameters:

Parameter	Description
Slave number	identifies the slave that communicates with the master. The slave number must be unique and selected from numbers 81 through 89.
Master number	identifies the master on the channel. The master number and all slave numbers must be unique and selected from numbers 81 through 89.
Input file	identifies the file that receives data from the master. This file must be in the input image section of the data table. If you do not make a selection, the file defaults to input file 1. This file must be large enough to store the data from the master's output file. Otherwise the processor sets the peer-to-peer minor fault bit.
Output file	identifies the file that sends data to the master. This file must be in the output image section of the data table. If you do not make a selection, the file defaults to output file 1.

LIST displays the size of the input and output files. For example, I1:0 → 19 shows that input file 1 contains 20 octal words. You can create a larger file by entering the last word desired in the file when prompted for the file number in LIST. By entering I1:50 for the input file, the system will automatically increase input file 1 to 50 octal words.

To delete slave entries from the list, press:

slave entry **/d [Return]**

Backup Communications Configuration

When you configure a channel for backup communication, you must enter the following parameters:

Parameter	Description
Resident number	identifies the system being configured on the channel. The resident number must be different from the partner number and selected from numbers 81 through 89.
Partner number	identifies the other system on the channel. The partner number must differ from the resident number and selected from number 81 through 89.
Input file	identifies the file that receives data from the partner. The input file must be in the input image section of the data table. If you do not make a selection, the file defaults to input file 1. This file must be large enough to store the data from the partner system's output file. Otherwise, the processor sets the backup communication minor fault bit.
Output file	identifies the file that sends data to the partner. The output file must be in the output image section of the data table. If you do not make a selection, the file defaults to output file 1.

LIST displays the size of the input and output files. For example, I1:0 → 19 shows that input file 1 contains 20 octal words. You can create a larger file by entering the last word desired in the file when prompted for the file number in LIST. By entering I1:50 for the input file, the system will automatically increase input file 1 to 50 octal words.

To delete slave entries from the list, press:

slave entry **/d [Return]**

DH or DH+ Configuration

When you configure channel 4 for DH or DH+ communication protocol, you can set the following parameters:

- node mode (page 2-11)
- station number (page 2-11)
- DH/DH+ timeouts (page 2-12)
- backup operation (page 2-13)
- send unprotected (page 2-14)
- accept upload/download (page 2-15)
- accept writes (page 2-15)
- input file list (page 2-16)
- privileges (page 2-17)
- DH/DH+ switch settings (page 2-18)
- DH+ active node list (page 2-19)
- DH+ active nodes to status file 6 (page 2-20)

When configuring channel four for a communication protocol, the menu contains these selections:

```
S5           1-CHAN 4
 1  NODE MODE     *ON–LINE     OFF–LINE
 2  STATION NUMBER:     23 SELECTED     255 ACTUAL
 3  DH/DH+ TIMEOUTS
 4  BACKUP OPERATION
 5  *SEND UNPROTECTED
 6  *ACCEPT UPLOAD/DOWNLOAD
 7  *ACCEPT WRITES
 8  INPUT FILE LIST
 9  PRIVILEGES
10  DH/DH+ SWITCH SETTINGS
11  DH+ ACTIVE NODE LIST
12  DH+ ACTIVE NODES TO STATUS FILE 6
ENTER NEXT >
```

The following subsections describe these selections.

1 – Node Mode

Select the NODE MODE option from the configuration screen to allow or disallow the station communicating on the link. By selecting NODE MODE, you can configure the channel to be online or offline.

If you select:	Then the channel is:
on-line	an active station on the link.
off-line	inactive and the DH/DH+ disable indicator is turned on.

Important: A reconfigure is not necessary to change this selection. The NODE MODE changes to the opposite selection and takes affect immediately.

An asterisk (*) displays in front of the selected modes.

2 – Station Number

Select the STATION NUMBER option from the configuration screen to identify the PLC-3 station on the DH or DH+ link. You can specify an octal station number by starting it with a leading 0. Otherwise, the scanner treats it as a decimal number.

Allowable station numbers are:	for:
0 to 376 octal	DH
0 to 77 octal	DH+

The number 377 octal is illegal. Entering 377 as the station number automatically disables the channel, and you cannot enable it again until you select a different station number in LIST. If you make no selection, the scanner sets the station number for 377 octal (255 decimal) by default.

Important: You must set the protocol select switches for LIST configurable before the scanner can modify this selection.

If you set the station number by using the switches (see chapter 1), the scanner displays that number as decimal in LIST.

This line may show two values for the station number: selected and actual. Both numbers display if they are different, and if these two situations occur:

- you changed the station number in line 2 but have not executed a reconfigure to change the actual station number to the new number
- the scanner in a backed up system assumes an address different from the primary scanner until switchover occurs

3 – DH/DH+ Timeouts

Select the `DH/DH+ Timeouts` option to set these timeout values in seconds:

- outgoing message
- LIST channel
- remote program

Outgoing Message
This timeout is the maximum amount of time that the scanner waits for another station to reply to one of the messages. The valid entries are 0 to 999.9. In LIST, you can enter timeout values as whole numbers or in tenths of seconds.

For example:

If you enter a timeout value of:	The scanner sets the timeout period for:
100	100.0 seconds
50.2	50.2 seconds

Important: If you enter 0, the processor disables the timeout parameter.

The timeout period applies to each individual transmission. Some messages consist of several packets of data because of their size. Each message packet requires a separate transmission. Therefore, the timeout restarts for each packet.

LIST Channel
This timeout detects a lack of activity by a station on the link still having an allocated LIST resource. Set this value high enough to prevent a timeout while running LIST but not so high that a station on the link keeps the LIST resource allocated unnecessarily.

Once this time expires, the next list selection you enter displays the initial list menu. You can then continue through your selections.

Important: This timeout takes affect anywhere from one to two times the value you enter through LIST.

Remote Program
This timeout detects a lack of activity by a previously connected programming station. Should a connected programming station be abruptly disconnected from the link, this timeout would make sure that the edit resource and histogram resources are returned to the scanner.

Important: This timeout takes affect anywhere from one to two times the value you enter through LIST.

4 – Backup Operation

Select the `Backup Operation` option from the configuration screen to configure the DH and DH+ station numbers when backed up. See Table 2.A.

Table 2.A
Backup Operation Options

If backup is:	The DH and DH+ station number of the backup scanner module:
Selected	is automatically modified before and after switchover. This allows you to have identical configurations in both the primary and the backup processor.
Not selected	remains the same both before and after switchover. The primary and backup processors must have unique station numbers.

Regardless of which mode you select, both the primary and the backup DH and DH+ channels are active. There is no selection for the backup DH or DH+ channel to remain silent on the link until switchover.

When you select backup operation, you assign the same DH or DH+ station number to both the scanner module in the primary processor and the one in the backup processor.

As long as the primary processor is controlling the outputs, the station address of the DH or DH+ channel in the backup processor is modified. Table 2.B and Table 2.C show the station number modifications for DH and DH+, respectively. At switchover, the DH or DH+ station number in the backup processor assumes the station address assigned to it with the LIST function.

Table 2.B
DH Station Numbers for 1775-S5, -SR5 Modules

If the station address between:	The backup scanner module assumes a station address that is:
000 and 276_8	100_8 higher than the assigned station number
300_8 and 376_8	100_8 lower than the assigned station number

Important: Do not give the 1775-S5, -SR5 module the reserved station address of 377_8; and, when you select the backup operation, you cannot give the module a station address of 277_8.

Table 2.C
DH+ Station Numbers for 1775-S5, -SR5 Modules

If the station address between:	The backup scanner module assumes a station address that is:
00 and 37_8	40_8 higher than the assigned station number
40_8 and 77_8	40_8 lower than the assigned station number

When you do not select backup, you assign a unique DH (000_8 to 376_8) or DH+ (00_8 to 77_8) station number to the scanner module in the primary processor and the one in the backup processor. The backup continues to operate with the same station number even after the switchover.

ATTENTION: To guard against personal injury and damage to equipment, do not assign the same station number to both scanner modules, if you have not selected backup mode.

If both scanners are assigned the same number and backup is not selected, two stations with the same station number attempt to communicate on the link. This can cause unpredictable machine motion.

5 – Send Unprotected

Select the `Send Unprotected` option from the configuration screen to determine if the scanner can send unprotected command messages to other stations. You use command messages to communicate with other controllers on a DH or DH+ network. You can use:

- an unprotected command to read or write to any area of another station's data table
- a protected command to write to those areas of another station's data table specified by the station that receives the command

If you:	Then the scanner can send:
select send unprotected	both protected and unprotected commands
do not select send unprotected	protected commands only

An asterisk displays in front of this selection to indicate that it is enabled.

6 – Accept Upload/Download

Select the `Accept Upload/Download` option from the configuration screen to determine if the scanner can execute upload and download commands sent by another station. You use a sequence of upload and download commands to transfer memory information from the controller to another station or from another station to the controller.

If you:	Then the scanner:
select accept upload/download	can execute both upload and download commands
do not select accept upload/download	cannot execute upload and download commands

An asterisk (*) displays in front of this selection to indicate that it is selected.

7 – Accept Writes

Select the `Accept Writes` option from the configuration screen to determine if the scanner accepts write commands sent by a remote station.

If you:	Then the scanner:
select accept writes	accepts write commands if the channel has the privilege to write to the specified memory area, and the assigned privilege is independent of the memory protect keyswitch (see "Privileges", page 2-17)
do not select accept writes	does not accept write commands under any condition

If the assigned privilege is not independent of the keyswitch, the scanner checks the keyswitch position before writing:

If the keyswitch is set to:	Then the scanner:
memory protect on	does not accept write commands and generates error code 86 (see appendix B)
data change or memory protect off	accepts write commands

An asterisk (*) displays in front of this selection to indicate that it is selected.

8 – Input File List (PLC-2® Compatibility Mode Only)

Select the `Input File List` option from the configuration screen for a list of files that the scanner accesses when receiving data using PLC-2 logical data addressing (see chapter 5). When a PLC-2 command comes in from a station on a DH or DH+ link, the scanner accesses the input file associated with that remote station number. You must make sure the input file is created in memory.

If you do not list an input file for a remote station, the scanner assigns a default file to it. The default file number matches the remote station number (except for remote station zero which is assigned input file 8):

PLC-2 compatible remote station number in octal	Assigned input file for read/write access
000	I008
001	I001
002	I002
003	I003
004	I004
005	I005
006	I006
007	I007
010	I010
011	I011
012	I012
.	.
.	.
.	.
077	I077
100	I100

Station address 000 is assigned to input file I008. Otherwise PLC-3 input files with an 8 or 9 in their addresses are not used for read/write access by a PLC-2 station.

If you do not want to use the default input file, you can use the input file list selection. Upon selecting input file list, the list of remote station addresses and associated input files displays. Initially, the list is empty and default assignments do not display unless you enter them.

Important: You can assign a remote station to only one input file, but you can assign different remote stations to the same input file.

To add a station address and input file to the list, enter a colon (:) followed by the station address, a space, and the input file number. For example:

```
INPUT FILE LIST
STATION                    FILE
:0                         2
:5                         2
:64                        6
:65                        7
1:37                       4
ENTER STATION AND INPUT FILE #> :6 4
```

You can specify an octal station number by starting it with a leading 0. Otherwise, the scanner treats it as a decimal number. The file number is always treated as a decimal number.

To specify station addresses for stations on remote links, you can enter:

link:*node*

To remove a station from the list, enter the station number followed by **/D**. For example:

ENTER STATION AND INPUT FILE #> **1:37/D**

This example removes link 1, station number 37 from the input file list.

Another example:

ENTER STATION AND INPUT FILE #> :**5/D**

This example removes station number 5 from the input file list.

9 – Privileges

Select the `Privileges` option from the configuration screen to select operating parameters for the DH/DH+ communication channel. These operating parameters are called privileges.

For example, you must have privilege 4 selected in order to enter and edit a ladder program. If you want to protect your ladder program from editing, you could deselect privilege 4. Another user could still monitor or read your ladder program, but could not alter it.

Available privileges are:

Privilege Number	Allows the device on the channel to
0	write to the system status area of memory
1	write to the system pointers area of memory
2	write to the module status area of memory
3	write to the data table area of memory
4	write to the ladder program area of memory
5	write to the message area of memory
6	write to the system symbols area of memory
7	write to the system scratchpad area of memory
8	write to the converted procedures area of memory
10	write to the force tables area of memory
64	On-line edit or edit the ladder program while the processor is in the run mode
65	Create and delete sections of memory
66	Change privileges list
67	Change the operating mode without having the remote enable section in LIST active
68	Change the operating mode only when the remote enable selection in LIST is active
69	Change system parameters in LIST such as system clock, watchdog timer, and current context
70	Change module dependent parameters in LIST
71	Test the ladder program by putting the processor in test mode
72	Perform a physical write
73	Accept keyboard input for GA Basic, report generation, or message procedure commands. By removing privilege 73, you can prevent an operator or other device, such as a bar code reader, from allocating a device on the channel.
74	Abort a GA Basic task in LIST

To modify a privilege, do the following:

If you want to:	Then:
add a privilege	enter the privilege number
make a privilege independent of the memory protect keyswitch	enter the privilege number followed by /I
remove a privilege	enter the privilege number followed by /D

10 – DH/DH+ Switch Settings

You can determine the state of the DH/DH+ switch settings without removing the scanner from the chassis. Select the `DH/DH+ Switch Settings` from the configuration screen to determine the state of the DH/DH+ switch settings.

When you select this option, the programming terminal displays:

```
  DH/DH+ SET-UP SWITCH SETTINGS             (VIEW FROM TOP OF MODULE)

                        S1                              S2
                                                                        MODULE
DOWN/ON      X   X           X   X                X       X   X         FRONT

UP/OFF               X   X           X   X            X

SWITCH #     8   7   6   5   4   3   2   1        4   3   2   1

               STATION NUMBER IN OCTAL             MODE   BAUD


CURRENT SELECTION       63                      DH+   57.6KB

1 EXIT THIS MENU      NOTE:  SWITCH CHANGES ARE ONLY READ AFTER POWER-UP
ENTER NEXT >
```

11 – DH+ Active Node List

Select the `DH+ Active Node List` option from the configuration screen to display a list of active nodes on the local link for channel four:

- an `X` next to a station number indicates a remote station on the link.
- a `US` next to a station number indicates the station number of the scanner whose list configuration you are currently viewing.

```
LIST OF ACTIVE STATIONS ON THE LOCAL DH+ LINK

TOTAL NUMBER OF STATIONS ON LOCAL LINK     7

   0        X       10       20       30       40       50       60       70
   1        X       11       21       31       41       51       61       71
   2        X       12       22       32       42       52       62       72
   3        X       13       23       33       43       53       63       73
   4        X       14       24       34       44       54       64       74
   5        X       15       25       35       45       55       65       75
   6        US      16       26       36       46       56       66       76
   7                17       27       37       47       57       67       77

1  EXIT THIS MENU
2 REDISPLAY MENU
```

12 – DH+ Active Nodes to Status File 6

Select the `DH+ Active Nodes to Status File 6` option from the configuration screen to maintain the DH+ active node table in status file 6. The list of active nodes is stored in status file 6. If you enable this option, you must also create status file 6.

Status file 6 is partitioned into 4 words for each of the 15 possible scanner thumbwheels. To provide for 15 scanners, you must create S6:59. Each of the 64 bits contained in the 4 word groups represent the possible DH+ station numbers of 0 through 77 octal.

If the bit is a:	Then the station represented by that bit is:
1	active on the link.
0	not on the link.

Important: This table is updated even when the system is in program mode.

Setting the Rack Range

The rack range selection in LIST sets the range of assigned I/O rack numbers. The default setting for the rack number range is 0-76_8.

If you plan to use rack numbers greater than 37_8, consider the following:

- Each instruction in the ladder program that addresses an input or output in a rack greater than 37_8 uses one additional word of memory and increases the program scan time. Typically, the program scan increases about 0.01 milliseconds for each relay logic instruction (XIC, XIO, OTE, OTL, OTU) that address a bit in a rack greater than 37_8.
- The amount of memory required for the input and output sections depends on the highest rack number containing input or outputs respectively. Therefore, skipping rack numbers wastes memory.

Configuring the Front Panel

The list configurable items for scanner number 1 also include selections for front panel channel 0, and the front panel data access display.

Select:	To enable you to set:
channel 0	parameters for the RS-232-C port labeled CHANNEL 0 on the front panel. The parameter selections are described in the following sections.
display	the privileges for the front panel display. These privileges determine what you are allowed to accomplish through the front panel keypad. The default for these privileges is 3, 10, 67, 69, and 70.

I.T. Defaults / Configuration

The **I.T. defaults** selection in LIST configures channel 0 for Binary Command Language (BCL) protocol communication with an industrial terminal (see appendix A). You can use the industrial terminal as a programming terminal for such functions as ladder programming. When you select I.T. defaults, the default values are used as shown in the following table.

The remaining **configuration** selections allow you to modify the front panel parameters shown in the following table to something other than their default settings.

Characteristic	Description	Default
privileges	This selection allows you to select operating parameters for channel 0. Refer to the section entitled "Privileges" under "DH or DH+ Configuration" on page 2-17 for a full description of the privileges and how to assign them.	0, 2, 3, 4, 5, 6, 8, 10, 64, 65, 66, 68, 69, 70, 71, 72
communication rate	The rate at which the RS-232-C device communicates to the processor through channel 0 is the communication rate. You can select one of the following communication rates: • 110 bps • 150 bps • 300 bps • 600 bps • 1200 bps • 1800 bps • 2400 bps • 4800 bps • 9600 bps • 19200 bps Set the communication rate by typing the number corresponding to the desired rate. An asterisk displays next to the current communication rate.	9600 bps
parity	You can configure channel 0 to communicate using the following parity selections: • even — the channel transmits an even parity bit with each character and checks for an even parity bit in each character received • odd — the channel transmits an odd parity bit with each character and checks for an odd parity bit in each character received • none — the channel does not transmit a parity bit and does not check for parity bits An asterisk (*) displays next to the current parity state.	even
stop bits	You can specify the number of stop bits that channel 0 uses to communicate. The stop bit selections are 1, 1.5, and 2. An asterisk displays next to the current stop bit selection. To change the number of stop bits, type the number corresponding to the desired selection.	1
channel timeout	You can specify the amount of time that the processor allows between operations on a channel before terminating communication. This amount can be set up to a maximum of 32,767ms (32.767s). If you want to disable the timeout function, set the value to 0.	10,000 ms

Reconfiguration

This selection implements changes that you have made in LIST for channel 0. If channel 0 is being used when you reconfigure it, the changes are not implemented until communication on the channel is terminated and then re-established.

An asterisk appears next to Reconfigure when a change has been made to any of the channel 0 parameters. The asterisk is removed after a successful reconfigure.

Chapter 3

I/O Communication

Chapter Objectives

This chapter describes timing and programming considerations when using the scanner for I/O communication and helps you learn how:

- I/O scan affects ladder program execution
- to calculate block-transfer times
- to program and calculate times for a peer-to-peer or backup communication channel

Effect of I/O Scan on Program Execution

You must remember that in scanning the ladder program and the I/O:

- input changes do not instantly appear in the input image table. The length of time between an input change and the update of the input table depends on the input module delay, the signal propagation time, backplane access time, and the I/O scan time.
- the I/O scan is not synchronized with the ladder program scan. Therefore, the same inputs can be in different states at different times during the program scan.

Calculating the I/O Scan Time

The amount of time that the scanner takes to scan all the I/O on a given channel depends on the communication rate and the number of:

- entries in the I/O chassis scanning sequence list
- active channels on the scanner
- block transfers
- retries required due to noise

Table 3.A shows the scan times for one I/O adapter on each active channel with no block transfers.

Table 3.A
I/O Scan per I/O Chassis in Milliseconds

bps	Chan 4 Config	Number of Active I/O Channels 1	2	3	4
57.6k	Not DH/DH+ DH/DH+	7 8	7 8	7.5 9	8 N/A
115.2k	Not DH/DH+ DH/DH+	4.5 5	5 6	5 7	6.5 N/A
230.4k	Not DH/DH+ DH/DH+	3 N/A	3.5 N/A	5 N/A	6 N/A

Each I/O chassis that you add to the scanning sequence list (including repeated I/O chassis in the list) increases the I/O scan time.

When you add block-transfer modules to the channel, the I/O scan time increases by approximately 1ms per adapter on all channels while the scanner executes the block-transfer instruction.

Calculating Block-Transfer Times

The time required to complete a block transfer depends on the number of:

- I/O channels on the scanner that contain block-transfer I/O modules
- active I/O channels on the scanner
- entries in the I/O chassis scanning sequence list for the channel
- block-transfer I/O modules on the channel

To calculate the time required for the scanner to execute all block transfers on the channel and be ready to execute the block-transfer again:

1. Determine the number of active I/O channels on the scanner.

2. Determine the number of I/O channels with block-transfer I/O modules.

3. Determine the nominal block-transfer time (see Table 3.B).

Table 3.B
Nominal-Block-Transfer Times in Milliseconds for a Block-transfer Instruction

Channels with block-transfer I/O modules	Number of active channels			
	1	2	3	4
1	27	27	28	35
2	x	29	30	36
3	x	x	31	38
4	x	x	x	40

4. Count the number of block-transfer I/O modules on the channel. If the chassis containing a block-transfer I/O module appears more than once in the I/O chassis scanning list, count the module once each time the chassis appears in the list.

5. Count the number of entries in the I/O chassis scanning sequence list for the channel.

6. Calculate the time between block transfers as follows:

```
T = (NBT x NM)ms + (NE – 1) x 7ms
```

Where:	Is the:
T	time between block transfers
NBT	nominal block-transfer time (Table 3.B)
NM	number of block-transfer modules (step 4)
NE	number of entries in the I/O scan list

Peer-to-Peer and Backup Communication

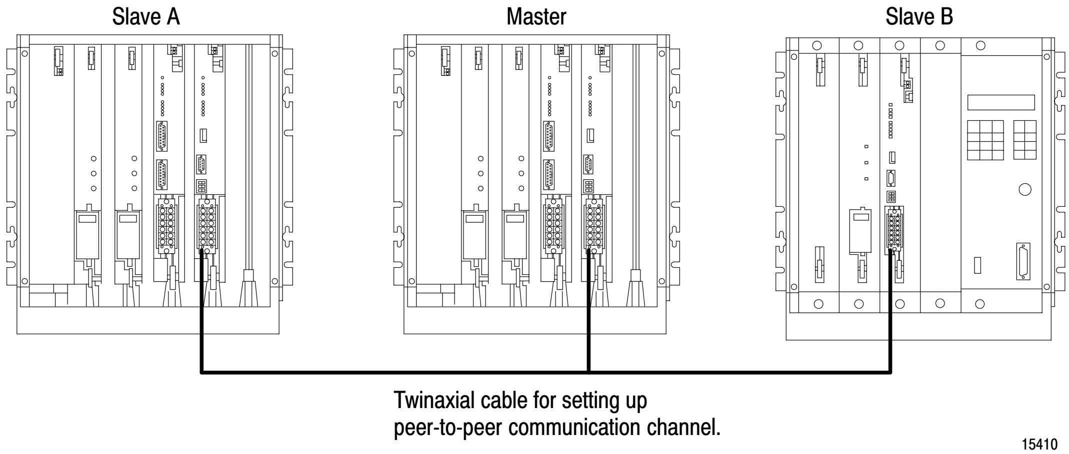
PLC-3 family controllers can communicate with other PLC-3 family controllers using peer-to-peer or backup communication channels.

Peer-to-Peer Communication

A peer-to-peer communication channel includes one PLC-3 or PLC-3/10 controller configured as the master and up to six PLC-3 or PLC-3/10 controllers configured as slaves. The master communicates with all slaves on the channel, while each slave communicates only with the master (Figure 3.1).

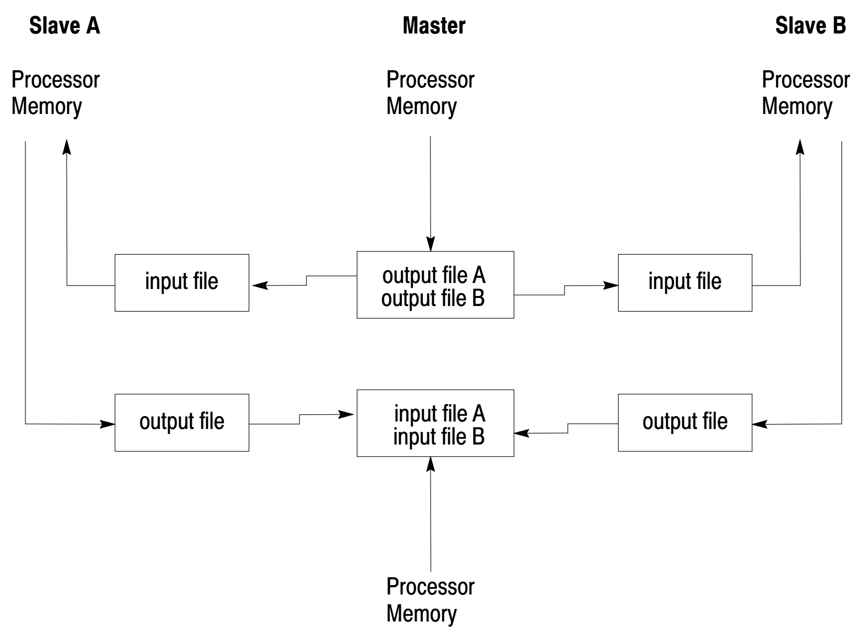
Also, when configured in a backup system, the backup slave receives all input data that are sent to the primary slave if the slave numbers are the same.

Figure 3.1
Using a Peer-to-Peer Communication Channel



You configure the master and slave controllers for the peer-to-peer channel through LIST. Communication occurs via files that you specify in LIST. Figure 3.2 shows the communication flow between master and slave controllers on a peer-to-peer channel.

Figure 3.2
Communication Flow Between Master and Slave Controllers on a Peer-to-Peer Communication Channel



Each input file must be large enough to store the data from the corresponding output file. Otherwise, a peer-to-peer communication minor fault occurs.

Programming Peer-to-Peer Communication

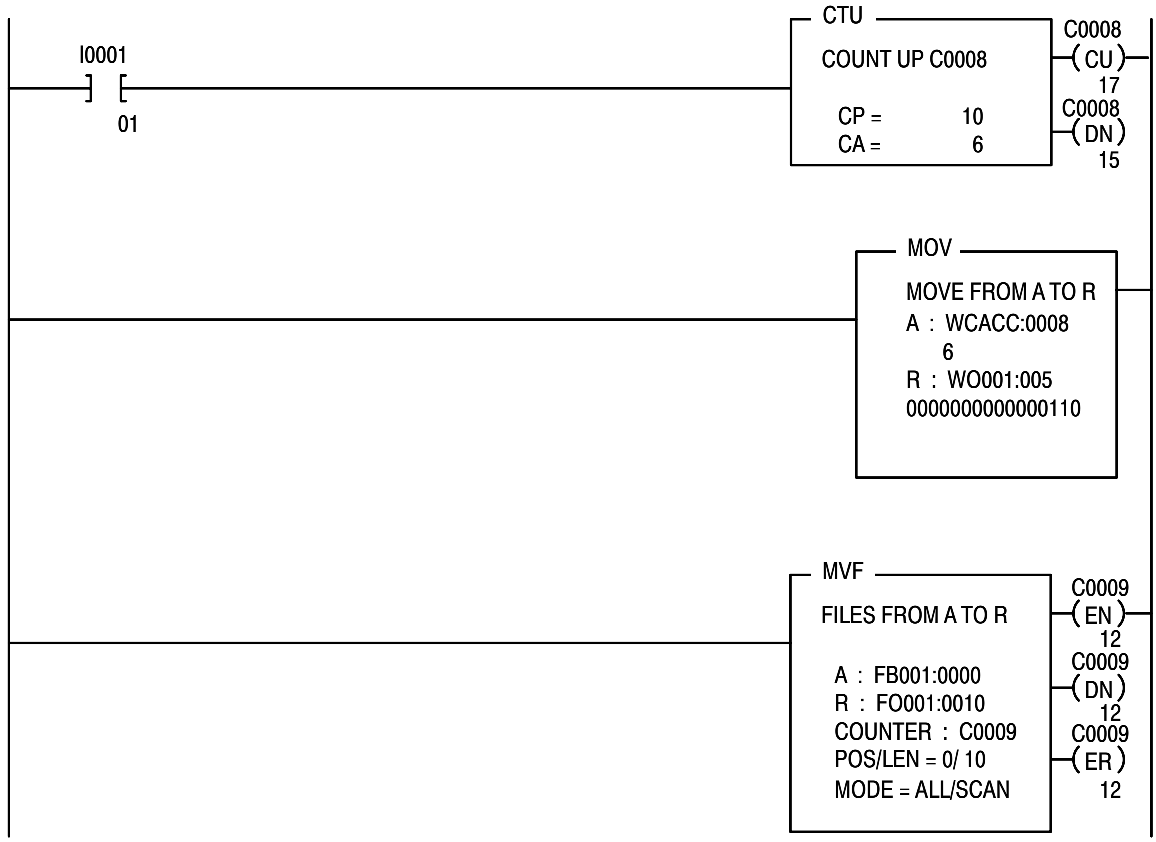
Programming PLC-3 family controllers for peer-to-peer communication requires that the ladder program in the:

- master transfers the appropriate data to its output file
- receiving slave knows the proper use for each word in its input file

The controller automatically handles the actual data transfer. Each slave's input file must be equal to or larger than the master's output file. Otherwise, a peer-to-peer communication minor fault occurs.

Figure 3.3 shows an example of the ladder rungs that transfer data to output file FO001 in the master controller. The rungs that move data into the output file are unconditional. If they are conditioned, "old" data transfers whenever the rung conditions are false, because the transfer occurs regardless of whether the output file is updated.

Figure 3.3
Example Rungs in the Transmitting Controller on a Peer-to-Peer Communication Channel



Important: You can control the information that is transferred by using multiple rungs to transfer different information to the output file, depending on the rung conditions. However, remember that the output file contains only the last data that moves into each location.

The peer-to-peer transfer is independent of program scan and a file can be transferred before being fully updated by the ladder program.

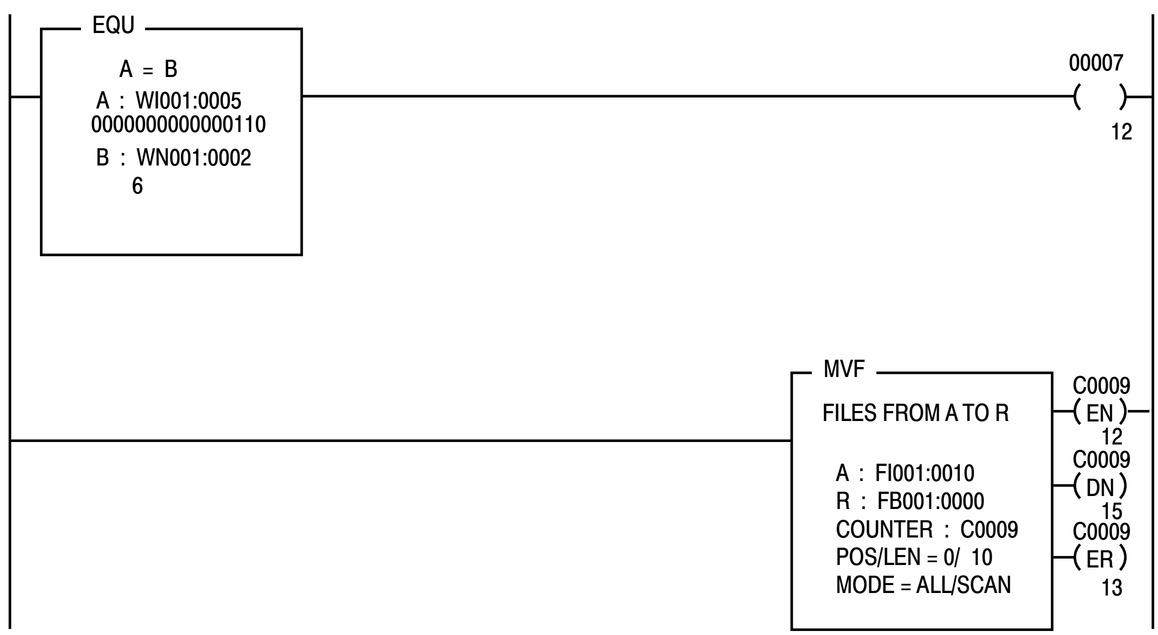
Figure 3.4 shows an example of the ladder rungs that receive data in input file FI001. Data in the input file of the receiving controller corresponds to the data in the output file of the transmitting controller.

In the example rungs shown in Figure 3.3 and Figure 3.4:

- input word 5 of the receiving controller reflects accumulated value for counter 8 in the transmitting controller
- the first 10 words of binary file 1 in the receiving controller reflect the status of the corresponding words in the transmitting controller

In both examples, remember that either the transmitting or receiving controller could be the master, provided that the other controller is a slave. The master communicates with each slave on the channel, but the slaves communicate only with the master.

Figure 3.4
Example Rungs in the Receiving Controller on a Peer-to-peer Communication Channel



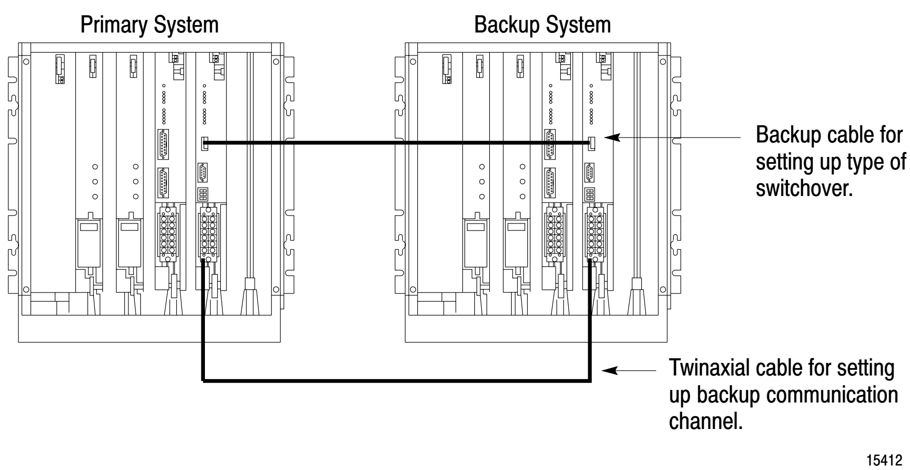
Important: Remember to reset the counter for the MVF, or the move operation does not function properly.

Backup Communication

You can use the backup communication with the PLC-3 or PLC-3/10 backup feature. The backup feature has an I/O listening mode that enables the backup system to monitor input data. To transfer other information to the backup system, you can install a backup communication channel (Figure 3.5).

With a backup communication channel, you can program data transfers between the primary and backup controller (Figure 3.5). Then, when the primary controller shuts down, the backup controller has up-to-date data for running your application. Such data could be storage, block transfer, timer, or counter information.

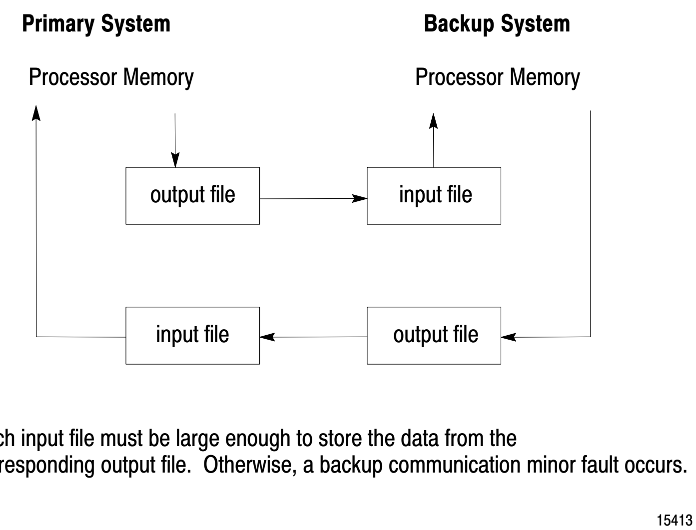
Figure 3.5
Using a Backup Communication Channel



Important: You cannot use peer-to-peer communication to communication between the primary and backup system because the backup processor listens but does not respond to communication received on a peer-to-peer communication channel.

You configure the primary and backup controllers for the backup communication channel through LIST. Communication occurs via files that you specify in LIST (Figure 3.6).

Figure 3.6
Communication Flow Between Primary and Backup Controllers on a Backup Communication Channel



A backup communication channel also prevents contention over the I/O channel should the backup cable break or become disconnected. If such a situation occurs, the system with the highest resident number takes control over the I/O. You assign the resident and partner numbers (see chapter 2).

Installation of peer-to-peer and backup communication channels is described in the PLC-3 Family Controller Installation and Operations Manual (publication 1775-6.7.1) and PLC-3 Programmable Controller Backup Systems Manual (Publication 1775-6.3.1).

Programming Backup Communication

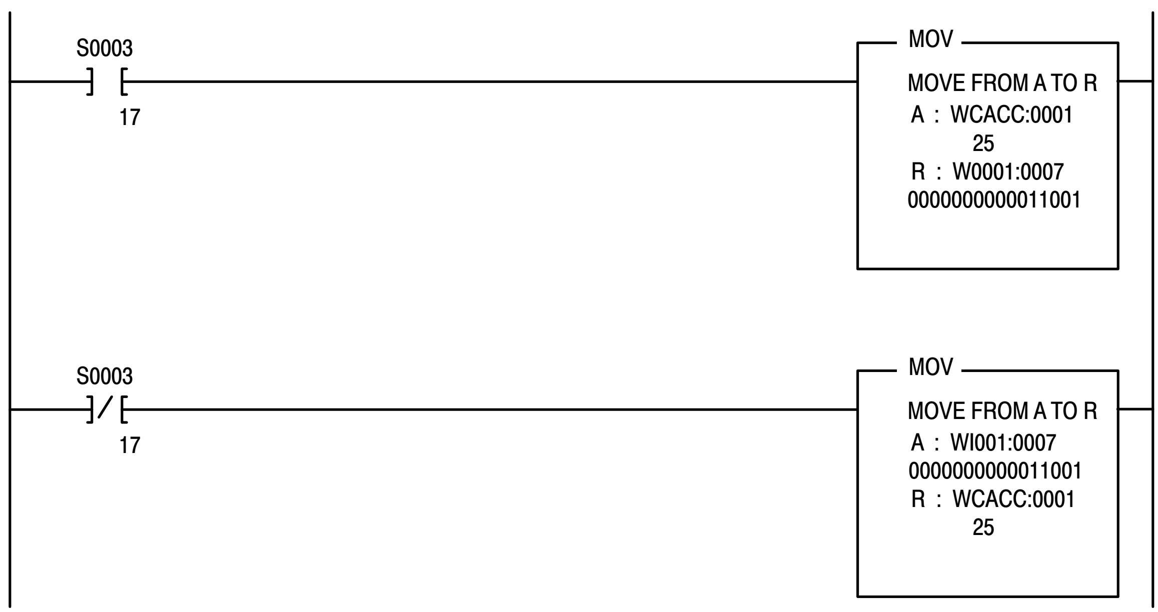
Backup communication channels operate in the same manner as peer-to-peer communication channels with these considerations:

- In using a backup communication channel, you can send information back and forth between the primary and backup systems. Since information moves both ways, you can monitor the run/backup bit (data table status section, file 0, word 3, bit 17) to determine whether to use the received data.
- The input file in the backup system must be equal to or larger than the output file in the primary system. Otherwise a backup communication minor fault occurs.

Figure 3.7 shows example rungs using the run/backup bit (S000:3/17). To use these rungs for backup communication, you must enter them into the primary and backup systems. The run/backup bit is set in the primary system and reset in the backup system. The counter is updated only in the backup system.

In this example, both the primary and backup systems use input file 1 as the input file and output file 1 as the output file for the backup communication channel. For detailed information on the backup feature, refer to the PLC-3 Family Controller Backup Concepts Manual (publication 1775-6.3.1).

Figure 3.7
Example Rungs for a Backup Communication Channel



Calculating Peer-to-Peer and Backup Communication Times

The time required to transfer a file across a peer-to-peer or backup communication link depends on the number of:

- active channels on the scanner
- communication rate
- words in the output files

To calculate the time needed to update every input file on the link, use the following procedure:

1. Determine the number of words in all output files of the master and each slave on the peer-to-peer communication link. Add them, divide by 16, and round up to the nearest whole number.

Important: Remember that the master has an output file for each slave.

2. Determine the greatest number of active channels on any scanner on the peer-to-peer communication channel.

3. Find the nominal peer-to-peer time by using Table 3.C and the result of step 2.

Table 3.C
Peer-to-Peer and Backup Communication Times per Slave in Milliseconds

bps	Chan 4 Config	Number of Active I/O Channels			
		1	2	3	4
57.6k	Not DH/DH+ DH/DH+	12 12.5	12 14	12.5 14	13 N/A
115.2k	Not DH/DH+ DH/DH+	8 9	8.5 10	9 11	10.5 N/A
230.4k	Not DH/DH+ DH/DH+	6.5 N/A	7 N/A	8 N/A	9 N/A

4. Multiply the result of step 3 by the result of step 1.

```
Update Time = (Nominal time) x (total words/16)
```

To calculate the time needed to update a specific input file on the peer-to-peer or backup communication link, use the following procedure:

1. Determine the number of words in the output file being sent to this specific input file. Divide by 8 and round up.

2. Determine the total number of slaves on the link.

3. Determine the greatest number of active channels on any scanner on the peer-to-peer communication link.

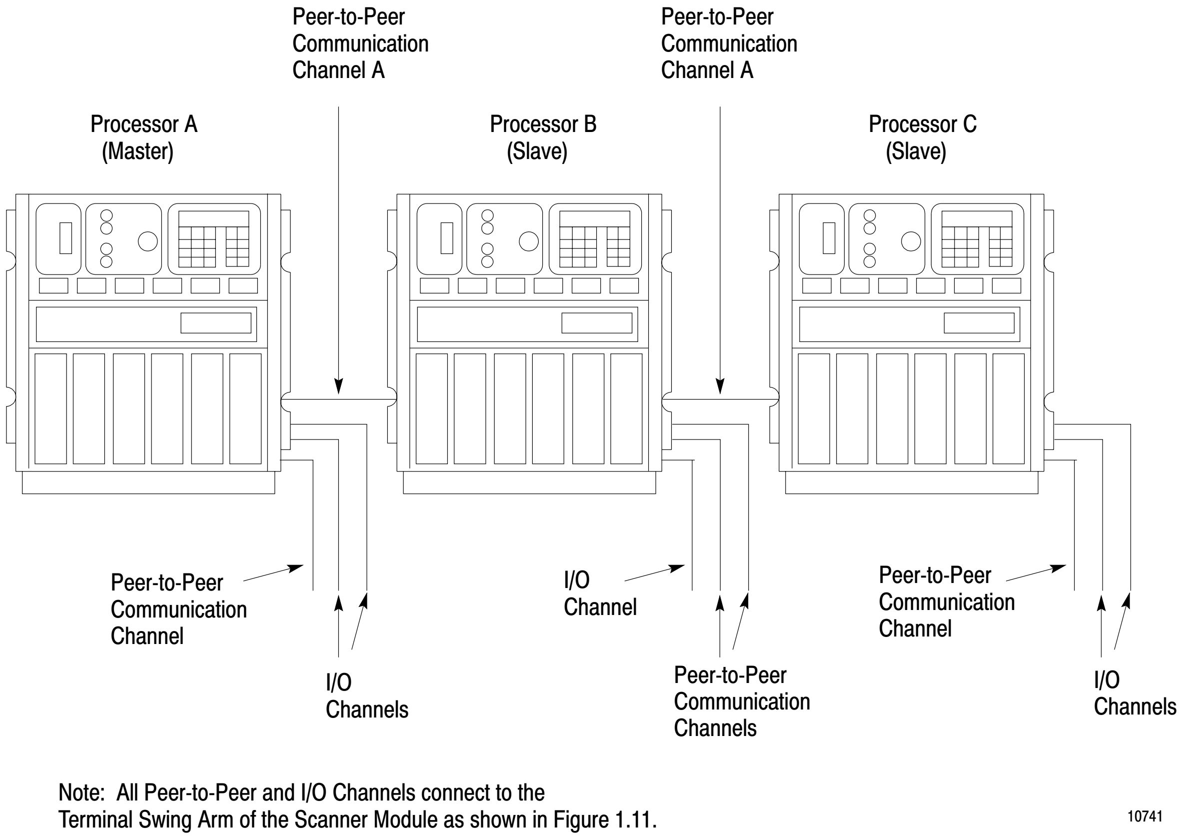
4. Find the nominal peer-to-peer time by using Table 3.C and the result of step 2.

5. Multiply the results of steps 1, 2 and 4.

```
Update time = (Nominal time) x (output words/8) x
              (number of slaves)
```

Figure 3.8 shows an example of 3 PLC-3 controllers using peer-to-peer communication to communicate with each other on one channel and with other PLC-3 controllers on other channels. For this example, assume that each output file has 200 words.

Figure 3.8
Example Configuration for a Peer-to-Peer Communication Channel



To calculate the time required to send 200 words from the master to both slaves and have 200 words returned from each slave, use the following procedure:

1. 400 words (Two 200-word output files in master)
   400 words (One 200-word output file in each slave)
   800 words/16 and rounding up = 50.
2. All the scanners used for peer-to-peer communication channel A have 4 active channels.
3. From Table 3.C, the nominal peer-to-peer time for 4 active channels and a communication rate of 115.2k is 10.5ms.
4. `Update time = 10.5ms x 50 = 525ms.`

Chapter 4

DH and DH+ Communication

Chapter Objectives

This chapter introduces some of the concepts and terminology involved with operating the scanner on a DH or DH+ communication.

Introduction

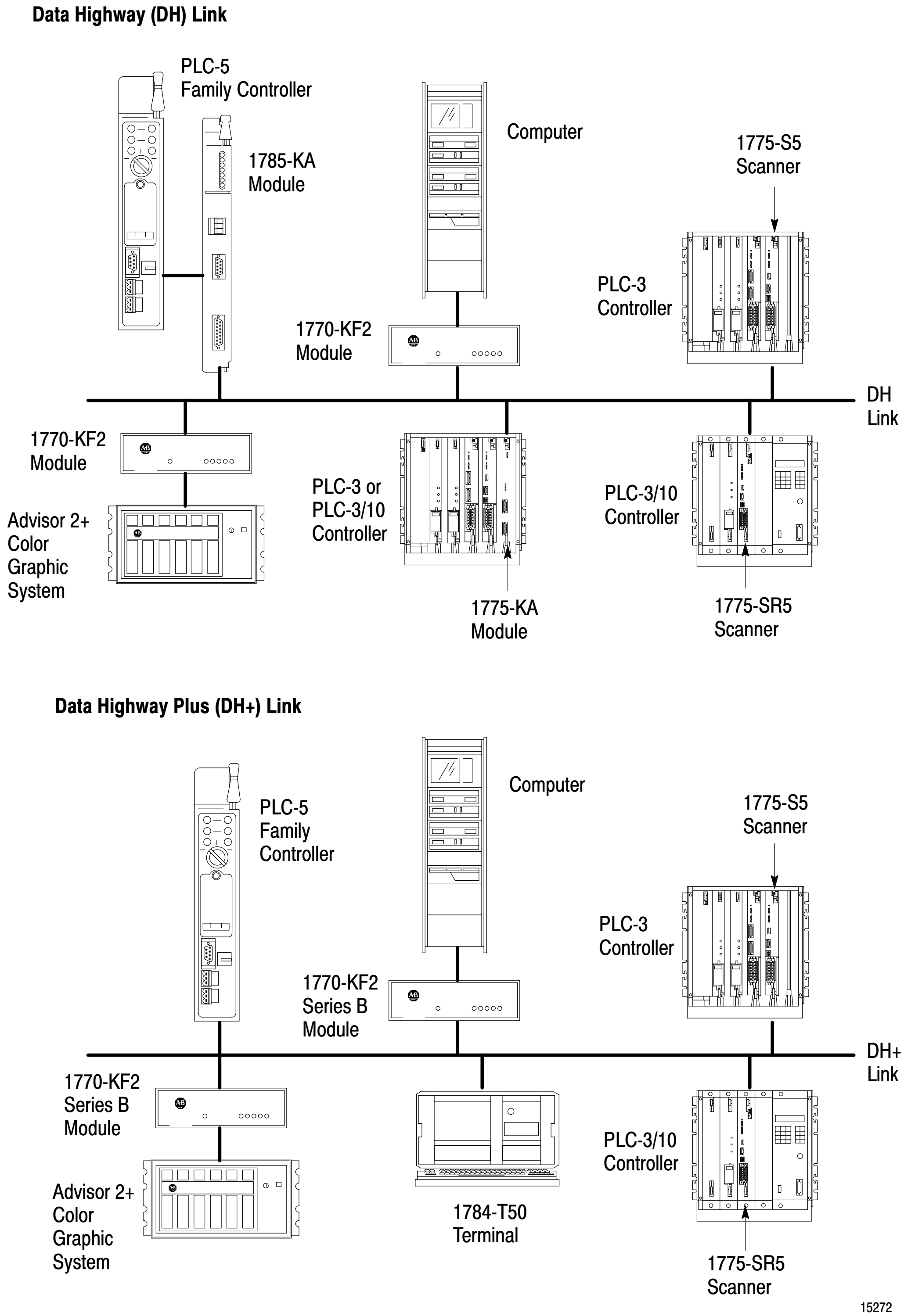
The Allen-Bradley DH and DH+ links are communication networks for industrial control applications. Both links consist of a central trunkline cable that can be up to 10,000 feet long. This cable can link together as many as 64 distinct communication points (or nodes) called stations (Figure 4.1).

Each station consists of some type of processor and a station interface module. The station interface module enables the processor to communicate with other stations on the DH or DH+ link. The scanner is the station interface module for PLC-3 family controllers. Table 4.A lists combinations of station interface modules and controllers.

Table 4.A
Devices the Scanner Can Communicate with Across the DH and DH+ Network

With this link type:	You can communicate with a:	Using this device:
DH	PLC-2 processor	1771-KA interface module 1171-KA2 communication adapter module
	PLC-3 processor	1775-S5 scanner 1775-KA interface module
	PLC-3/10 processor	1775-SR5 scanner 1775-KA interface module
	PLC-5/250 processor	5130-RM module 5130-KA module
	Computer interface using DF1 protocol	1770-KF2 interface module 1771-KE interface module 1771-KF interface module
	DH+ bridge interface	1785-KA interface module
	DF1, Master, or Slave bridge interface	1770-KF2 interface module 1771-KE interface module 1771-KF interface module
DH+	PLC-2 processor	1785-KA3 communication adapter module
	PLC-3 processor	1775-S5 scanner
	PLC-3/10 processor	1775-SR5 scanner
	PLC-5 processor	PLC-5
	PLC-5/250 processor	5130-RM module 5130-KA module
	Computer interface using DF1 protocol	1785-KE interface module 1770-KF2/B interface module
	DH bridge interface	1785-KA interface module
	DH II bridge interface	1779-KP5 foreign device interface module
	DF1, Master. or Slave bridge interface	1785-KE interface module 1770-KF2/B interface module

Figure 4.1
Example DH and DH+ Configurations



Communication Terms

Stations communicate with each other by sending two types of messages over a link:

Type of messages:	Function:
command messages	either gives (writes) data to, or requests (reads) data from one station to another.
reply messages	is a station's response to a command message.

You program command messages into the scanner. Execution of a message command is controlled by the message (MSG) instruction in the ladder program. When a scanner receives a command message from another station, the module automatically generates the appropriate reply message.

The local station is the one currently initiating some actions, or the one we are currently doing something with. All other stations are then remote.

We can also describe stations in terms of their relationship to a message.

The station that:	Is the:
sends the message	transmitting station
gets the message	receiving station
transmits a command message	command station
transmits a reply message	reply station

Solicited and Unsolicited Messages

The scanner can receive solicited and unsolicited messages depending on whether the message is a command from a remote station or a reply to a command from the local station.

- If the local station issues the command message, the corresponding reply message is said to be **solicited** because the local station has solicited, or requested the data contained in the reply message.
- If a remote station issues the command message, that message is said to be **unsolicited**.

Either station initiates the data transfer by issuing a command message.

For	a local station receives data from a remote station during a	and a local station sends data to a remote station during a
solicited messages	read operation	write operation
unsolicited messages	write operation	read operation

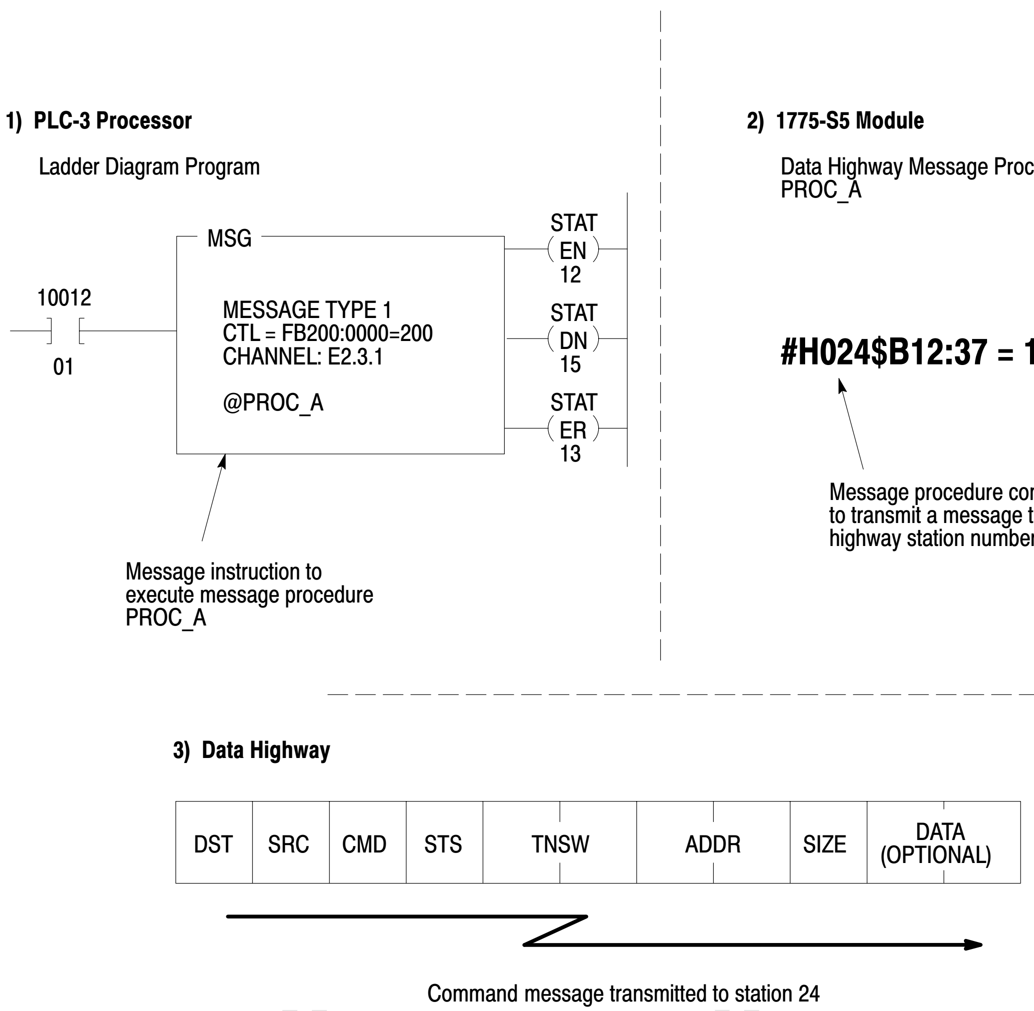
In read operations, the command message requests the data transfer, but the corresponding reply message actually contains the data being transferred.

In write operations, the command message contains the data being transferred, and the reply message merely reports the status (receipt or non-receipt) of the transfer.

Levels of Programming

The PLC-3 processor must be free to control its own processes at the same time that the scanner is communicating over a DH or DH+. For this reason, both the processor and the scanner have their own programs and programming languages. Figure 4.2 illustrates how these two programming levels (processor and scanner) interrelate.

Figure 4.2
Levels of Programming in DH or DH+ Communication



Ladder Program

The first link in the communication process is your ladder program. You send a DH/DH+ command message by means of the message (MSG) instruction (Figure 4.2).

When the rung becomes true, the processor informs the scanner to begin sending command(s) across the link. At the same time, bits in a control file word change their state (Table 4.B) to reflect the status of the message instruction.

Important: Once enabled, even if the rung becomes false, the scanner continues to send commands across the DH/DH+ network until the message is done or errors.

Table 4.B
The Status of Bits in a Control File Word

When the:	The processor:
message instruction is true	sets the enable bit (16) and the latch-enable bit (12)
scanner receives the message instruction	sets the request bit (17)
scanner begins operation	sets the busy bit (14)
scanner completes operation	resets the busy bit (14) and sets either the done bit (15) or the error bit (13)
rung becomes false	resets the request bit (17), busy bit (140), enable bit (16), and the latch-enable bit (12)
rung becomes true a second time	resets the done bit (15) or the error bit (13)
message errors appear	sets the error bit (13) and records the error code in bits 00 thru 07

You can program the message instruction with either:

- a single message command (see chapter 6) that can be up to 76 characters long.
- the name of a DH/DH+ message procedure that contains a group of commands and is stored in the message area of PLC-3 or PLC-3/10 memory.

You specify the scanner that receives the command with an extended address.

This address always takes the form:

E2.3.*nn*

Where:	Function:
E2	specifies that this command addresses the module-status area of memory
3	specifies that you are sending the message instruction through the scanner
nn	is replaced with the thumbwheel setting on your particular scanner (1-15)

To enter a message instruction, do the following:

1. Enter a condition that, when true, activates the message instruction. In Figure 4.2, we use an examine-on for input word 12, bit 01.

2. Enter MSG for the message instruction.

3. Choose a control file where status information about the message command can be stored. The file should also be binary. In Figure 4.2, we use binary file 200, word 0.

4. Enter an extended address for the channel. In Figure 4.2, we address the module status area of memory, specify the scanner, and a thumbwheel setting of 1.

5. Specify message type 1.

6. Enter either a command or a command procedure. Figure 4.2 uses the command procedure PROC_A.

DH/DH+ Message Procedure

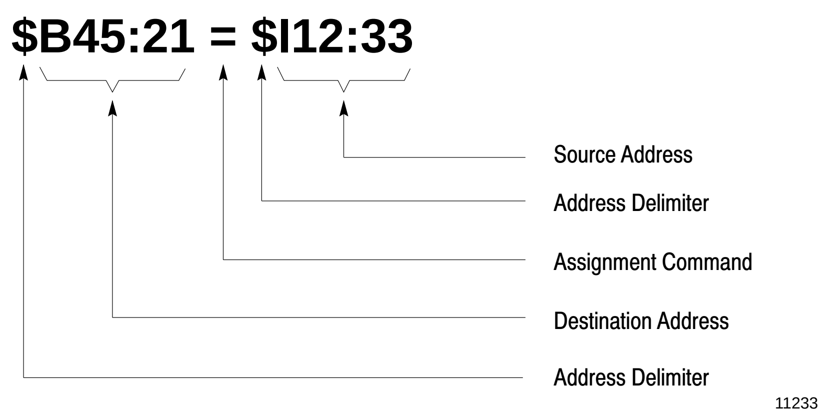
The scanner has its own programming language that consists of commands (see chapter 6). A group of related commands make up a DH/DH+ message procedure. These commands and message procedures determine what messages are transmitted over the DH or DH+ link.

Data Transfers

The purpose of DH/DH+ communication is to transfer data from one station processor memory location to another. To accomplish these data transfers, you can program the assignment command into the scanner.

Figure 4.3 is an example of an assignment command line. For details, see chapter 5. Since the following example does not specify a remote station, this transfer occurs within the local systems memory.

Figure 4.3
Example Assignment Command Line



In the above example (Figure 4.3), the assignment command copies a word (16 bits) of data from the source to the destination location. The source of the data is always specified on the right of the equal sign (=), and the destination is always on the left.

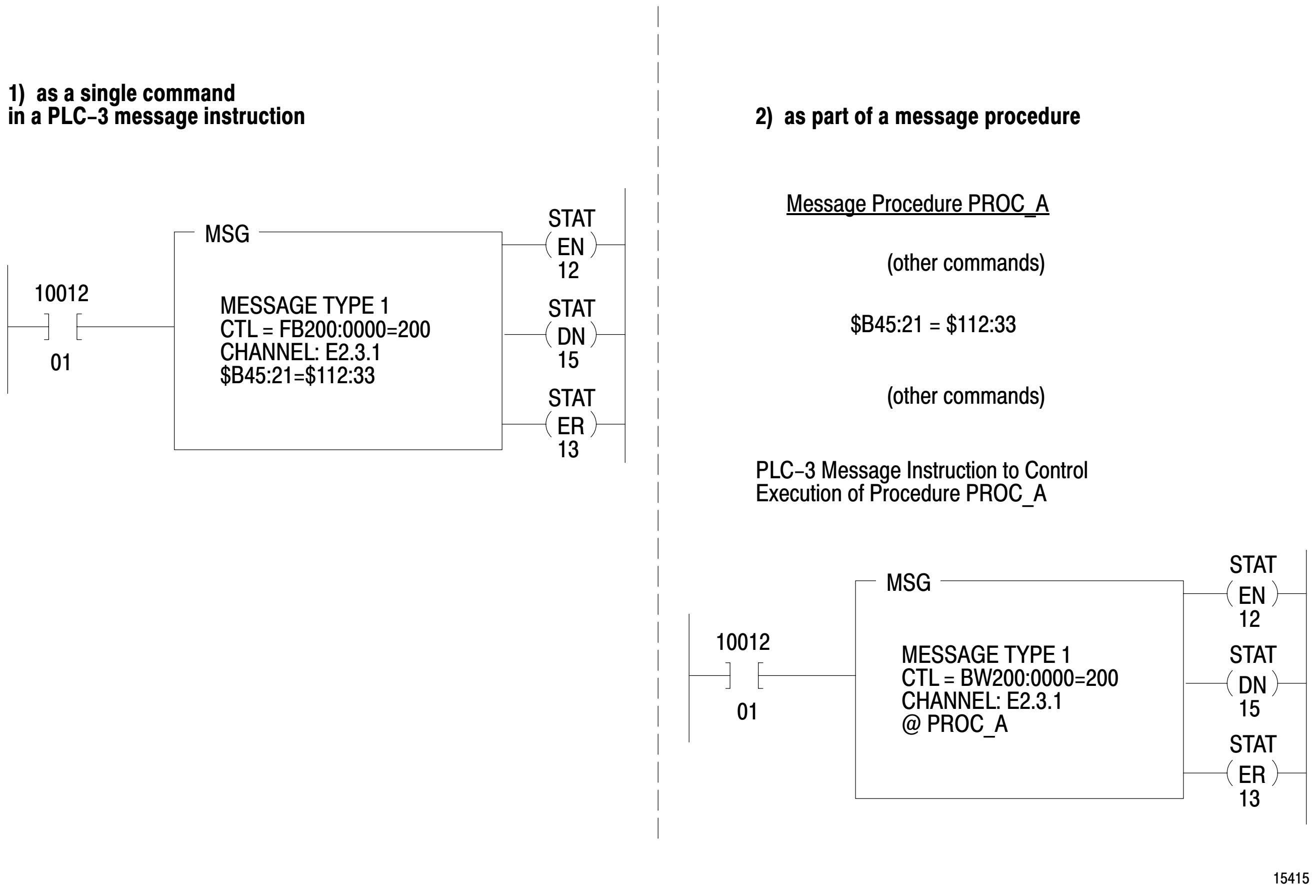
An assignment command does not destroy the data at the source location; it copies the source data at the destination location. When the assignment executes, both the source and destination contain the same data.

You can use a data transfer command with the scanner:

- as a single command within a message instruction
- as one of multiple commands within a message procedure

Figure 4.4 illustrates both of these methods for the same assignment command. A message instruction in the ladder program controls execution of the command in either case.

Figure 4.4
Two Ways to Use Message Commands



Access Privileges

Not every station can read or write to every other station. In general, read and write access privileges depend on two factors:

- type of processor at the transmitting and receiving stations
- protections set at the receiving station

The rest of this section explains how these access privileges vary according to the above factors.

PLC-3 Stations
PLC-3 stations can read and write to other PLC-3 station data tables. A local PLC-3 station can prevent remote PLC-3 stations from writing to the local station's data table by setting the memory protect keyswitch.

To allow writes to local memory depending on the memory protect keyswitch setting, select `Accept Writes` in the DH/DH+ options menu in LIST at the local station.

PLC-2 Stations
For communication with a PLC-2 station, write access privilege depends on switch settings at that remote station. For an explanation on how to set the switches for write access, refer to the Communication Adapter Module User's Manual (publication 1771-6.5.1).

Accessing a PLC-2 Station
Access to a PLC-2 station also depends on the type of command transmitted to that station. There are two types of commands:

- **protected write commands** — can only write to specified sections of the data table in a PLC-2 processor. Memory-access rungs in the PLC-2 ladder program specify where in the data table the PLC-3 can write data.
- **unprotected read and write commands**— can read or write to any section of the data table at a PLC-2 station. Refer to publication 1771-6.5.1 for an explanation of protected and unprotected commands and memory access rungs.

A PLC-3 station can read from any part of a PLC-2 data table. However, a PLC-3 station cannot write to a PLC-2 station if the switch settings at the PLC-2 station do not allow access.

If the switches at the PLC-2 station are set to accept only:	then a PLC-3 station can write to the PLC-2 data table in:	by transmitting:
protected write commands	memory areas defined by the PLC-2 ladder program	protected write commands.
unprotected write commands	any area of the data table	unprotected write commands.

Accessing a PLC-3 Processor from a PLC-2 Processor

While a PLC-3 controller can address any area of a PLC-2 data table, a PLC-2 reads an input file that is a part of the PLC-3 data table. That file is the PLC-3 input file with a number that corresponds to the station number of the PLC-2 station. For example, the read/write files assigned to PLC-2 stations 1 to 100 (octal) is:

PLC-2 compatible station number in octal	Assigned input file for read/write access
000	I008
001	I001
002	I002
003	I003
004	I004
005	I005
006	I006
007	I007
010	I010
011	I011
012	I012
.	.
.	.
.	.
077	I077
100	I100

Station address 000 is assigned to input file I008. Otherwise PLC-3 input files with an 8 or 9 in their addresses are not used for read/write access by a PLC-2 station.

The PLC-2 station can use either protected or unprotected commands to access its assigned PLC-3 file. However, the PLC-2 station cannot access its assigned file until that file is created and allocated at the PLC-3 station. To create a file, refer to the user’s manual for your programming terminal.

You can have two PLC-3 stations communicate with each other as if they were PLC-2 stations. To do this, allocate the appropriate PLC-2 addressing format (see chapter 5) in the assignment commands. Similarly, a computer can send PLC-2 commands to a PLC-3 station by using the appropriate message packet formats.

PLC-4 Stations
PLC-4 stations can only communicate on a DH. To read or write to a PLC-4 station, you can send either protected or unprotected commands.

Switches:	at the 1773-KA module specify whether the PLC-4 station accepts unprotected or protected commands, respectively through the:
2 and 3 (on the second row of switches)	DH port of the 1773-KA module.
1 and 3 (on the third row of switches)	RS-232-C port of the 1773-KA module.

In all cases, if the switch is set to the closed position, the module accepts that type of command.

PLC-5 Stations
The PLC-3 processor can read/write to a PLC-5 family processor in two ways:

- as if the PLC-5 processor was a PLC-2 family processor. See "PLC-2 Stations," page 4-10 for detailed information.
- using logical ASCII data type addressing (See chapter 5).

Choosing Between DH or DH+ Communication

The difference between the communication links is the means of communication. DH link uses a floating-master function to handle communication between stations. This function allows a station to access the DH based on 3 factors:

- station's readiness to transmit a message
- message priority
- station number

In this way, the floating-master function prevents any single station from "hogging" the DH link, and it enables the stations to become disabled.

The DH+ protocol uses a token-passing function to handle communication between stations. The token is constantly passed from station to station even if no messages are sent. Each station obtains time to send a message. A station becomes master when it obtains the token. Then it can send a message to another station. When a station is done communicating, the token automatically passes to the next higher station number on the link.

DH+ communication is faster than DH communication for configurations of 16 stations or less. The delay time for link access increases as you increase the number of stations on the DH+ link.

Calculating DH and DH+ Communication Times

The time required for the scanner in a local station to send or receive a message with a scanner in another station on a DH or DH+ link generally depends on the number of:

- stations on the DH+ link
- messages transmitted from active stations
- bytes of data of all transmitted messages
- message requests that are queued ahead of the subject message

Table 4.C lists typical I/O scan times and throughput times for a DH or a DH+ configuration. The **throughput time** is the execution time for the DH or DH+ message. This time starts with setting the enable bit and ends with setting the done bit in the ladder program of the station sending or receiving the message.

When calculating these times, the DH/DH+ link had one other station on the link, and the message instruction was sending 10 words to the second station. The second station was not transferring any words.

Table 4.C
Throughput Times for a DH or DH+ Configuration in Milliseconds

Number of Active I/O Channels	Throughput Times	
	DH	DH+
0	109	66
1	110	67
2	113	67
3	119	67

Important: I/O communication channels can affect the time required to complete DH+ functions initiated on the same scanner. If any of the devices in the I/O rack lists are not responding to the scanner (retrys), then the time to execute some programming terminal DH+ functions, such as ladder logic searches, will take longer.

Operating Backup Configurations on a DH or DH+ Link

When using a scanner for DH or DH+ communication, you can set up two types of backup configurations:

- two scanners in the same processor chassis for backing up the link

  Both scanners are active, and each one is an independent station on the link. You assign unique station numbers to each scanner. If you want to send the same message through both scanners, you must program separate message instructions.

- scanners in a backup system

  One scanner is in the primary system and one scanner is in the backup system.

For detailed information on backup systems, refer to the PLC-3 Family Controller Backup Concepts Manual (publication 1775-6.3.1).

Operating Multiple Links in One System

By inserting multiple scanners in the same PLC-3 or PLC-3/10 system, you can operate a single system on more than one link. In this configuration, each scanner connects to a different link, and each has a unique station on its associated link. However, all the scanners in the same system can have either the same or different station numbers.

ATTENTION: If such a PLC-3 station is communicating through a PLC-2 buffer file on a DH link, and all of the station's scanners have the same station number, then all of the scanners transfer data through the same buffer file. This can cause unpredictable results if several scanners try to read or write to the buffer file at the same time (see "Accessing a PLC-3 Processor from a PLC-2 Processor," page 4-11).

When such a PLC-3 station transmits a command message to a remote DH or DH+ station, the thumbwheel number specified in the message instruction determines which scanner actually transmits the command.

Chapter 5

Addressing DH and DH+ Data Transfers

Chapter Objectives

In chapter 4, we described how you can send a single message command or a message procedure containing a group of commands. This chapter explains some general rules for specifying data addresses in message procedures for communicating over a DH or DH+ link.

Addressing Field Parameters

To transfer data from one DH or DH+ station to another, you program an assignment command into the scanner. The assignment command enables you to copy information from a source to a destination. In this chapter, we use the following abbreviations to denote the parameters that you can specify as a source or destination for a data transfer:

Parameter:	Meaning:
`bit`	number of a particular bit within the addressed word
`fileaddr`	logical address of a PLC-3 file
`filesym`	symbolic address of a PLC-3 file
`offset`	number of words between the beginning of the file and the desired word (offset is zero for the first word of a file)
`size`	number of words that transfer
`wordaddr`	logical address of a PLC-3 word
`wordsym`	symbolic address of a PLC-3 word

You can use a value or an expression for any of the above parameters in an assignment command.

Interpreting Addresses

When using the above parameters, the scanner interprets values as decimal (base 10) unless you indicate that they are octal (base 8). Specify an octal number by starting it with a leading zero. For example, the scanner interprets 17 as decimal 17, but 017 as octal 17.

Important: An exception to the above rule occurs when addressing a word in the input or output sections of PLC-3 memory. In these cases, the scanner normally interprets the word address `wordaddr` as an octal number, regardless of leading zeros. To express an input or output word address as a decimal value, enclose the word address within parentheses and eliminate leading zeros.

In addressing individual bits, parentheses have no effect on the address interpretation. The scanner interprets the bit address `bit` as an octal number if it starts with a leading zero and as a decimal number if it does not start with a zero.

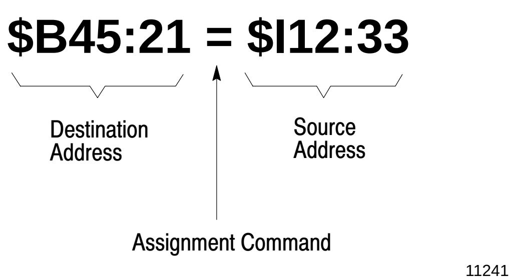
Some examples are given below:

Address:	Interpretation (expressed in decimal):
I12:15	input file 12, word 13
I12:15/15	input file 12, word 13, bit 15
I12:015/015	input file 12, word 13, bit 13
I12:015	input file 12, word 13
N43:15	integer file 43, word 15
N43:015	integer file 43, word 13
N043:15	integer file 35, word 15

Addressing Data

You reference data by its address in memory. In a message command, you must precede an address with a dollar sign ($). The dollar sign acts as a delimiter to tell the scanner that it has encountered a data address (Figure 5.1).

Figure 5.1
Example Assignment Command Showing Addressing Format



Symbols can also be used to represent data in memory. You must precede a symbolic address with an at sign (@).

Specifying Addresses

There are numerous commands that have been defined for operation on the DH/DH+ data link. You can program the scanner module to transmit a subset of these commands as listed in Appendix C, Table C.A. The command sent by the scanner is dependent on the addressing method you choose in an assignment command when addressing the data in the remote station.

The S5 and SR5 scanners offer four different addressing methods for use in DH and DH+ assignment commands:

- PLC-2 Logical Data Addressing
- PLC-3 Logical Binary Addressing
- Logical ASCII Data Type Addressing
- Logical ASCII Word Range Addressing

Although the scanner can transmit and receive any of these addressing methods, not all remote stations are able to interpret these methods. Table 5.A indicates which addressing methods are accepted by other controllers.

Table 5.A
Acceptable Addressing Methods

	Accepted by Receiving Station					
		PLC-3				
Addressing	PLC-2	KA	S5	PLC-4	PLC-5	PLC-5/250
PLC-2 Logical Data	X	X	X	X	X	X
PLC-3 Logical Binary	--	X	X	--	--	--
Logical ASCII Data Type	--	--	X	--	X	X
Logical ASCII Word Range	--	--	X	--	X	X

Using PLC-2 Logical Data Addressing

A PLC-2 logical data address references a dedicated file in the remote station. This addressing method simply provides an offset into this PLC-2 data table or PLC-2 compatibility file. The offset is interpreted as octal if a leading zero is included, otherwise the scanner interprets the number as decimal.

The receiving station must be able to interpret this form of addressing (see Table 5.A).

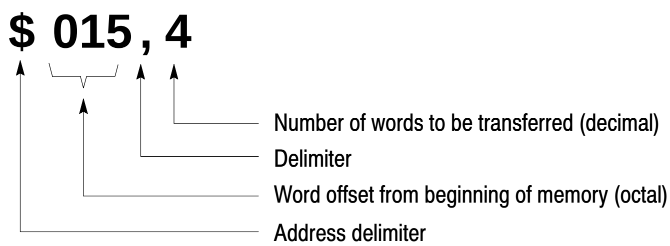
Addressing Words

To address a group of consecutive words in the remote stations memory, use the following format:

Format:	Example:
offset, size	**$047,20**

You can only address a group of words as the source field in an assignment command. The destination must be a file that is as large as, or larger than, the source size plus the offset.

Figure 5.2
Example for Addressing Consecutive PLC-2 Words



Addressing a single word is similar to addressing a word range only without specifying a size:

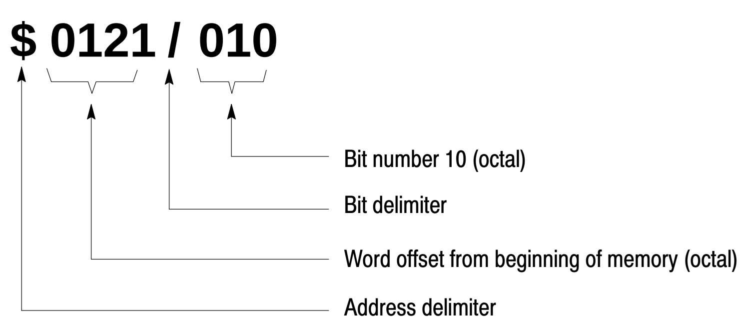
Format:	Example:
offset	**$047**

Addressing a Bit

To address a specific bit within a word, use the following format:

Format:	Example:
offset/bit	**$047/015**

Figure 5.3
Example for Addressing Specific Bits in PLC-2 Memory



Using PLC-3 Logical Binary Addressing

When you use a PLC-3 logical binary address, the scanner transmits the address to the remote station as a six level extended address. This addressing method is normally reserved for communication between PLC-3 scanners since the receiving station must be able to interpret this form of addressing (see Table 5.A). The following rules apply when specifying a PLC-3 logical binary address:

1. You must define symbolic addresses to either the word level or the file level of specification.

2. A word address can be either a:
- logical word address
- symbolic address of a word
- symbolic file address followed by a colon (:) and an offset
- extended address specified to the word level

3. You must precede a size specification with a word address and a comma (,).

4. You must precede an offset specification with a file address and a colon (:).

5. You must precede a bit number with a word address and a slash (/).

6. You can use extended addressing to access the pointer section of memory.

Addressing a File

To address a file, use either a file address or a file symbol:

Format:	Example:
fileaddr	**$N0**
filesym	**@FILE_A**

For assignment commands that copy data from one file to another, both the source and the destination file must be exactly the same size.

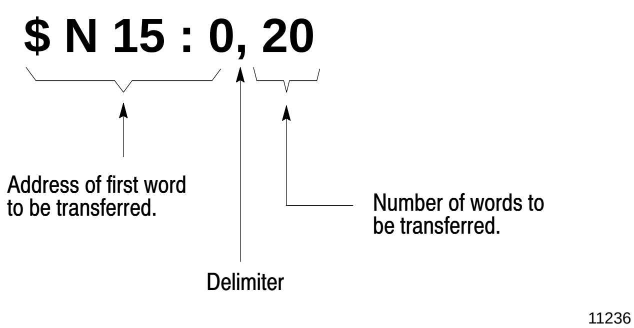
Addressing Words

To address a group of consecutive words in memory, use one of the following formats:

Format:	Example:
wordaddr,*size*	**$N0:47,20**
filesym:*offset*,*size*	**@FILE_A:15,25**
wordsym,*size*	**@WORD_1,20**

You can only address a group of words as the source field in an assignment command. The destination must be a file that is as large as, or larger than, the source range.

Figure 5.4
Example for Addressing Consecutive PLC-3 Words



Addressing a single word is similar only without specifying a size.

Format:	Example:
wordaddr	**$N0:47**
filesym:*offset*	**@FILE_A:15**
wordsym	**@WORD_1**

The scanner interprets *wordaddr* as an octal value if the addressed word is in an input or output file. Otherwise, it interprets *wordaddr* as a decimal value.

You cannot use the pointer data table specifier in an assignment command. To access words in the pointer section of the PLC-3 data table, you must use the PLC-3 extended addressing format. For example:

$E3.1.12.0.5.0 (1st word of pointer 5)
$E3.1.12.0.5.1 (2nd word of pointer 5)

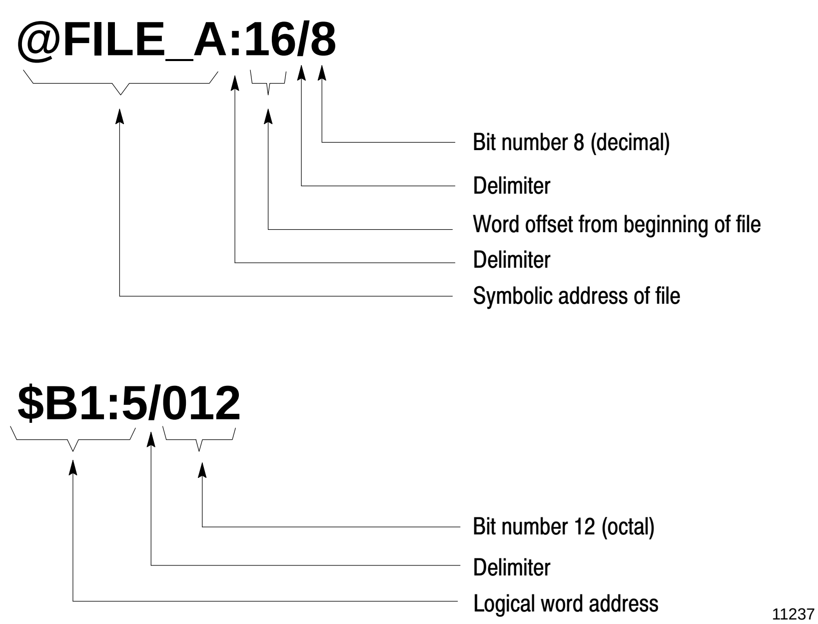
Refer to the PLC-3 Family Controller Programming Manual (publication 1775-6.4.1) for detailed information on extended addressing.

Addressing a Bit

To address a specific bit within a word, use one of the following formats:

Format:	Example:
wordaddr/bit	**$N0:47/015**
filesym:*offset/bit*	**@FILE_A:15/8**
wordsym/bit	**@WORD_1/012**

Figure 5.5
Examples for Addressing Specific Bits in PLC-3 Memory



Using Logical ASCII Addressing

The logical ASCII addressing method allows you to communicate with another device by specifying an address based on the internal memory structure of the target device. The scanner sends the address as a string of ASCII characters to the remote station. Upon receiving the string of ASCII characters, the remote station converts the data into a usable logical address.

The scanner offers two types of logical ASCII addressing; data type and word range. In both methods, the scanner transmits the address as an ASCII string, but the packet of data transmitted by the scanner on the DH/DH+ link differs.

Data Type

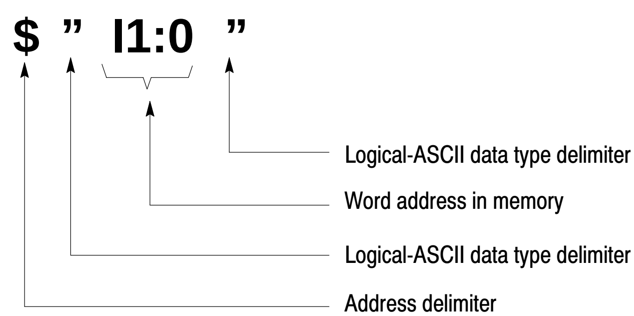
You can use Logical ASCII Data Type addressing to transfer data table sections without counting the actual words per data table structure. With a data type packet, the receiving station is notified of the type of data being sent.

Each station has a common understanding of how the different data types will be received. It is the responsibility of the sending station to translate the way it stores a particular data type into this common structure before transmitting. The receiving station then translates the structure into it’s method of storage for that particular data type.

To format a logical ASCII data type address, you enclose the address parameters in double quotes (”):

Format:	**Example:**
`$”`*`address`*`”,`*`size`*	**`$”N55:0”,20`**

Figure 5.6
Example of Logical ASCII Address that Accesses the First Word in File 1



Symbols are allowed inside the double quotes if they are defined in the remote station.

Logical ASCII data type addressing treats data being received and sent from the floating-point data table section as a special case. The PLC-3 stores floating-point data using DEC-F format and the PLC-5 family store the same data using IEEE format.

To handle this, whenever you send floating-point data using logical ASCII data type addressing, the scanner translates the data into IEEE format before transmitting on the link. Likewise, if floating-point data is received using logical ASCII data type addressing, the scanner translates the data from IEEE format into DEC-F format before storing into memory.

Word Range

A logical ASCII word range DH/DH+ packet does not indicate to the receiving station anything about the data structure. The user assumes responsibility for possibly sending unlike data types to the remote station. For example, integer data to a BCD file.

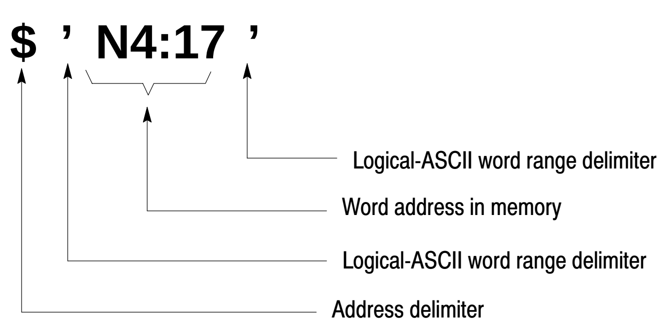
The sending station simply sends the data from the data table address given and the receiving station stores that same data at the indicated address. No error checking is done for compatible data table areas.

To format a logical ASCII word range address, you enclose the address parameters in single quotes ('):

Format:	Example:
`$'`*address*`',`*size*	`$'N55:0',15`

Symbols are allowed inside the single quotes if they are defined in the remote station.

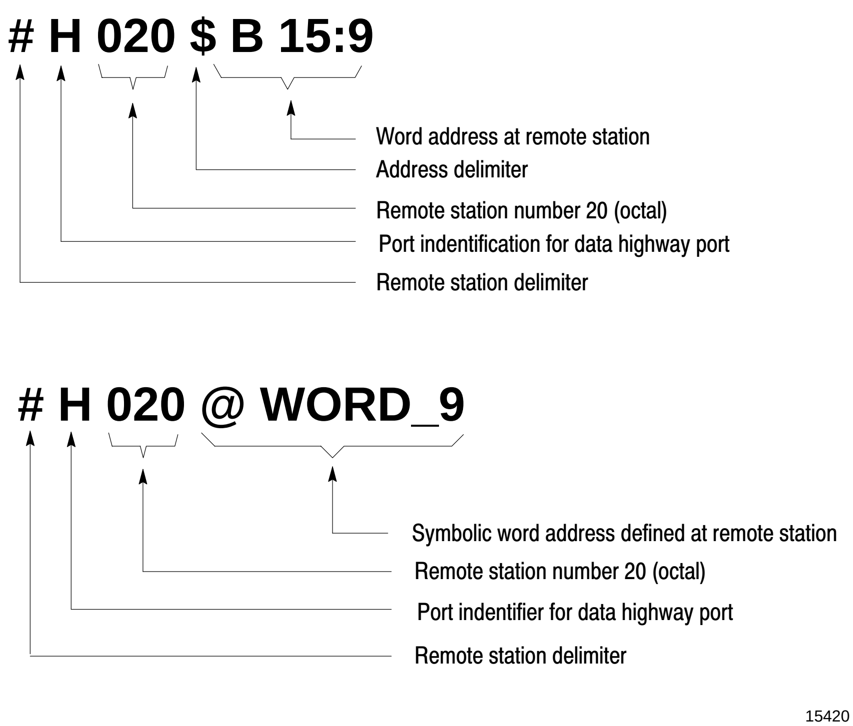
Figure 5.7
Example of Logical ASCII Address that Accesses the 17th Word in File 4



Addressing Stations on a Local Link

Figure 5.8 shows the format for addressing data at remote stations on a local link. The pound sign (#) specifies a remote station on a DH or DH+ link.

Figure 5.8
Example for Addressing a Word in a Remote Station on a Local DH/DH+ Link



Remote station addresses have the following restrictions:

- You can only use a remote address with the single equals sign (=) type of assignment command.
- In the assignment command, either the source or the destination, but not both, can be a remote address.
- A remote address can contain an embedded expression, but you cannot embed a remote address in an expression.
- You can substitute symbols or logical addresses for the remote station number.

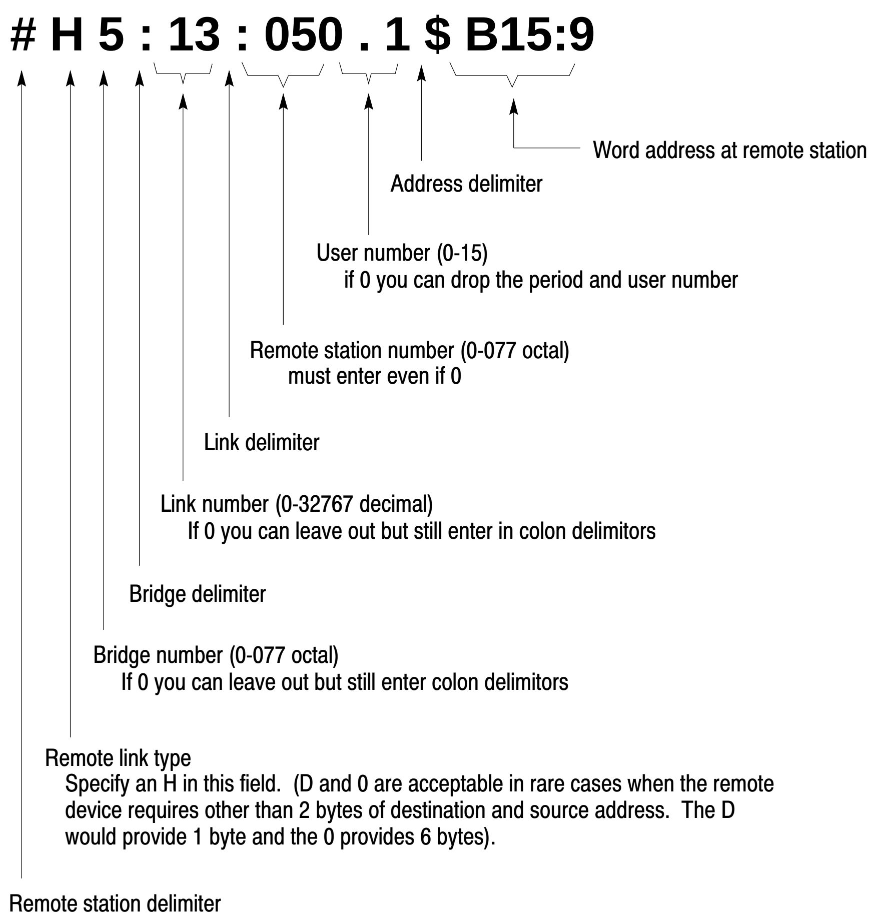
  For example, the remote station being addressed by #H($N1:0)$B15:9 will be the integer value contained in N1:0.

Addressing Stations on a Remote Link

In DH+ protocol, you can use the format shown in Figure 5.9 to address a remote station that is located across a bridge on another link.

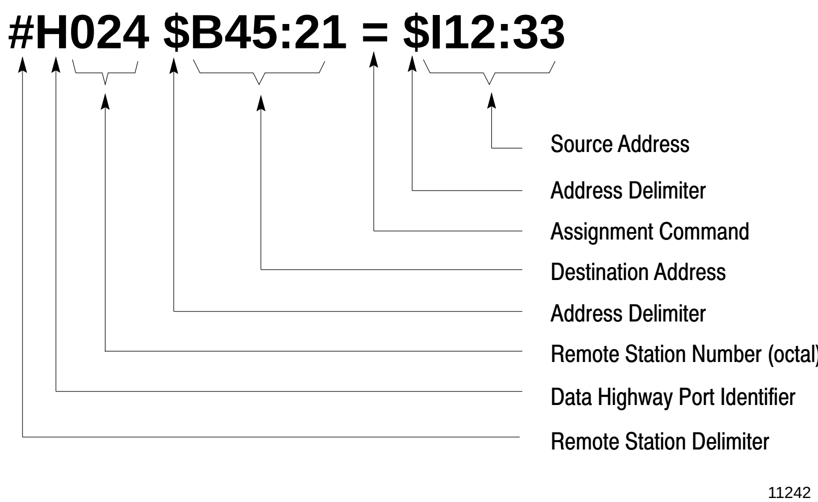
If you have the scanner configured for DH protocol, using this format causes a communication error to occur.

Figure 5.9
Example for Addressing a Remote Station Located on a Remote DH+ Link



For data locations at remote stations, a pound sign (#), followed by the remote link type (H for DH or DH+ communication) and the remote station number, precede the data address (Figure 5.10).

Figure 5.10
Example Assignment Command that Addresses a Remote Station



Assignment Command

You can use the assignment command to copy data from a source location to a destination location (Table 6.D). Source types include:

- direct values
- procedural user symbols
- interprocedural user symbols
- logical addresses
- local symbolic addresses
- global symbolic addresses
- expressions

Destination types include:

- procedural user symbols (except when the source is remote)
- interprocedural user symbols (except when the source is remote)
- logical addresses
- local symbolic addresses
- global symbolic addresses

You can use any source type listed above with any destination. If the destination is a user symbol, the scanner checks to see if it has previously been defined. If the user symbol was not previously defined, a new symbol is generated.

Important: You cannot transfer data from another station and place it into a user symbol defined at the local controller.

Formatting an Assignment Command

The equal sign (=) is the assignment command. The destination is on the left of the equal sign, and the source is on the right. When the scanner executes an assignment command, it assigns or copies the source value to the destination. For example:

```
$I12:024 = US_5
```

The scanner copies the value of user symbol US_5 into word 24 (octal) of input file 12 (see "Using Symbols," page 6-4).

Available Modifiers

You can add several modifiers to the assignment command to change the:

- scope of the assignment
- priority level of a DH message
- type of command message transmitted

Scope of the Assignment
If the destination for an assignment command is a user symbol, then you can use a double-equal sign (==) in place of the single equal sign to specify that the user symbol be interprocedural in scope. The single equal sign defines a procedural user symbol. For example:

```
US_2 == 6
```

The scanner defines US_2 to an interprocedural user symbol and assigns it the value 6 (see "Using Symbols," page 6-4).

Message Priority (DH only)
If you want to give priority to a DH message, you can use the less-than sign with the equal sign (<=). Without the less-than sign, the assignment command generates a normal message. For example:

```
#H027$I15:4 <= $I12:24
```

The scanner transmits a priority message to station 27 (octal).

You can use the priority modifier with either type of assignment (= or ==).

Important: For DH communication, stations with high priority messages are given priority over stations with no-priority messages throughout the command/reply cycle. For this reason, you should only give a high-priority designation to a command when special handling of specific data is required. Using an excessive number of high priority commands defeats the purpose of this feature and could delay or inhibit the transmission of no-priority messages.

Important: The scanner ignores the message priority modifier for DH+ messages.

Command Message Type
Command messages can be protected or unprotected:

- **Protected commands** can access only specified areas of data table memory at a PLC-2 station. You need to send a protected write command only if a switch at the remote PLC-2 station prohibits other stations from sending unprotected write commands.
- **Unprotected commands** can access any area of the data table.

For more information, see chapter 4.

By default, the assignment command generates protected command messages. To generate an unprotected command message, enter a space followed by a U after the assignment command statement.

For example:

```
#H027$0121 = 17407
```

The scanner generates a protected write command to write the value 17407 to word 121 of DH station 27.

```
#H027$0121 = 17407 U
```

The scanner generates an unprotected write command to write the value 17407 into word 121 of DH station 27.

You can disable the transmission of unprotected commands through LIST options (see chapter 2).

Identifying Remote Stations

Remote stations in an assignment command can be identified with an integer value (Figure 5.10), a symbol, a PLC-3 data table address, or system symbol.

For example:

```
#H(STANUM)$N0:0 = $N0:0    ; STANUM = 19

#H($N1:0)$N0:0 = $N0:0     ; $N1:0 = 19

#H(@SMITH)$N0:0 = $N0:0    ; @SMITH is equated to $N1:0
```

Programming Examples of Assignment Commands

This section gives examples of assignment commands using the four available addressing methods. Often, the choice of addressing method and the structure of the data table section which is being addressed, determines what data is actually sent.

Timers, counters, high-order integers, floating point, and pointers are all stored in the PLC-3 memory as multiple words. Each timer and counter contained in the PLC-3 uses 3 consecutive words; one for the control bits, one for the preset value, and one for the accumulated value. Timers and counters are stored in the following order:

CTL PRE ACC

The PLC-3 stores high-order integers and floating point numbers as two words. Pointers consume three words each in PLC-3 memory.

Table 5.B
PLC-2 Logical Data Addressing

Data Addressing	Description
`#H021$040 = $B3`	write data from entire binary file 3 to remote station 21 beginning at word 40 octal.
`#H021$040 = $B3:5,20`	write the twenty words beginning at binary file 3, word 5 to remote station 21, beginning at word 40 octal.
`#H021$40 = $B3:5`	write binary file 3, word 5, to remote station 21 word 40 decimal.
`#H021$040/5 = $B3:5/13`	write binary file 3, word 5, bit 13, to remote station 21 word 40 octal, bit 5.
`#H021$040 <= $B3:5`	as a priority message, write binary file 3, word 5 to remote station 21 word 40 octal. If the scanner is configured for DH+, the priority modifier is ignored.
`#H021$040 = $B3:5 U`	send an unprotected write of binary file 3, word 5, to remote station 21 word 40 octal.

Table 5.C
PLC-3 Logical Binary Addressing

Binary Addressing	Description
`#H02$F0:0 = $F0:0,6`	write 6 contiguous words beginning with floating point file 0, word 0 to remote station 2 floating point section file 0 word 0. A total of three complete floating point words (2 words each) are being transferred.
`$D0:0 = #H054$E0.0.0.14`	read the year from the system clock of remote station 45 and store the data in decimal file 0, word 0. Writes to non-data table addresses are not allowed.
`#H014$T0:5 = $T4,2`	write two words beginning with timer four to remote station 14 timer 5. In this example, the data in $TCTL:4 and $TPRE:4 is written to $TCTL:5 and $TPRE:5 respectively in the remote station.
`#H046$N0:0 = $N4:1`	write integer file 4, word 1 into remote station 46's integer file 0, word 0.
`#H055$N1 = $N0`	the entire integer file 0 is written into integer file 1 of remote station 55. The file sizes must be identical.
`#H02$H0:0 = $H0:0,6`	high order integers are treated as 32 bits or 2 words. This example writes 6 complete high order integers to remote station 2.

Table 5.D
Logical ASCII Data Type Addressing

Data Type Addressing	Description
`$N0:10 = #H003$"N55:0",100`	read 100 words from remote station 3 beginning with integer file 55, word 0 and store the data in integer file 0, beginning with word 10.
`$D0:0 = #H055$"N0:0"`	read integer file 0, word 0 from remote station 55 and store data in decimal file 0, word 0.
`#H020$"B10:10" = 15`	write a decimal value of 15 into binary file 10, word 10 of remote station 20.
`#H014$"T5" = $T4,4`	write the contents of four complete timers (three words each) beginning with timer 4 into remote station 14 beginning with timer 5.
`#H02$"F0:0" = $F0:0,6`	write the contents of 6 complete floating point numbers (two words each) into remote station 2 beginning with floating point number 0.

Table 5.E
Logical ASCII Word Range Addressing

Word Range Addressing	Description
#H024$'D1:0' = $D1:0,2	write two words beginning with decimal file 1, word 0 into remote station 24's decimal file 1, word 0.
#H040$'@PUNCH' = $A1:0	write the data stored at ascii file 1, word 0 into remote station 40 which will store the data at the address defined by symbolic address @PUNCH. This symbol must be created and defined in the remote station.
#H014$'T0:5' = $T4,6	write 6 contiguous words beginning with $TCTL:4, into remote station 14 beginning with timer five. In this example, timers 4 and 5 (3 words each) would be transferred complete.
#H014$'C5' = $C4,5	write 5 contiguous words beginning with $CCTL:4 into remote station 14 beginning with counter 5. In this example, $CCTL:4, $CPRE:4, $CACC:4, $CCTL:5, and $CPRE:5 would be transferred into $CCTL:5, $CPRE:5, $CACC:5, $CCTL:6, and $CPRE:6 respectively.

Chapter 6

Programming DH and DH+ Message Procedures

Chapter Objectives

This chapter explains how to create, edit, and use the message instruction to operate the DH and DH+ communication channel on the scanner. This chapter helps you learn how to:

- enter a message instruction into the ladder program
- program message command, functions, and procedures
- add comments to a message procedure
- program message procedures to monitor and recover from errors that occur during message command execution

Message Instruction Considerations

You do not always have to create a message procedure. If you want to execute just a single assignment command that is not longer than 120 characters, you can enter that command in the message-instruction block (Table 6.A). If you want to execute more than one assignment command, you must create a message procedure to contain those commands.

The execution of a single assignment command takes place once the scanner receives the message instruction from the CPU. However, when a message instruction contains a message procedure, the scanner must first store the symbols within your procedure into system memory and begin retrieving the procedure lines from system memory. Allow for this initialization time when initiating procedures. After this initial time, the commands within a procedure execute as fast as single commands within a message instruction.

Editing the Message Instruction

The general steps for editing a DH or DH+ message procedure are:

1. Create and edit the ladder program containing message instructions that control execution of the message procedure.

2. Allocate memory to the necessary data files.

3. Create and edit the message procedure.

You can perform these steps through your programming terminal. Refer to its user's manual for detailed information on creating and editing the ladder program.

You can also perform the third step through an RS-232-C data terminal connected to an RS-232-C channel on a Peripheral Communication Module (cat. no. 1775-GA).

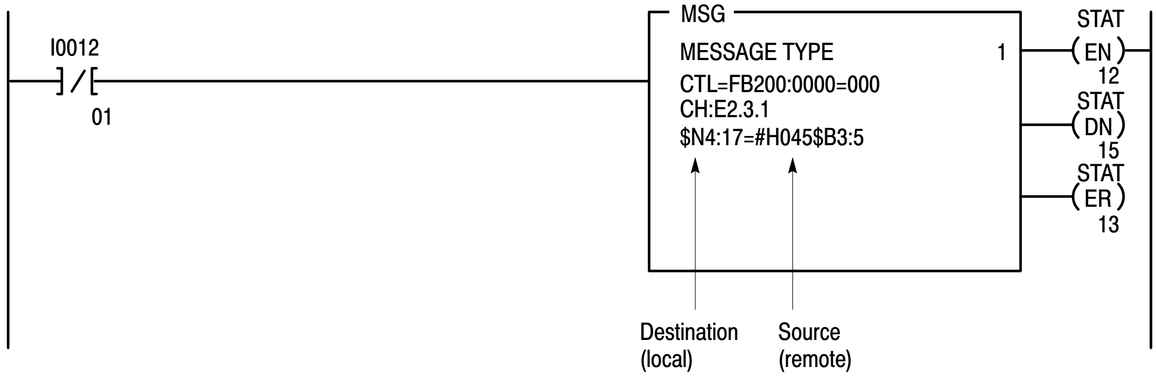
Table 6.A gives an example of how to edit the message instruction in the ladder program. For more details on ladder-program editing, refer to the user's manual for your programming device.

Table 6.A
Entering the Message Instruction into the Ladder Program

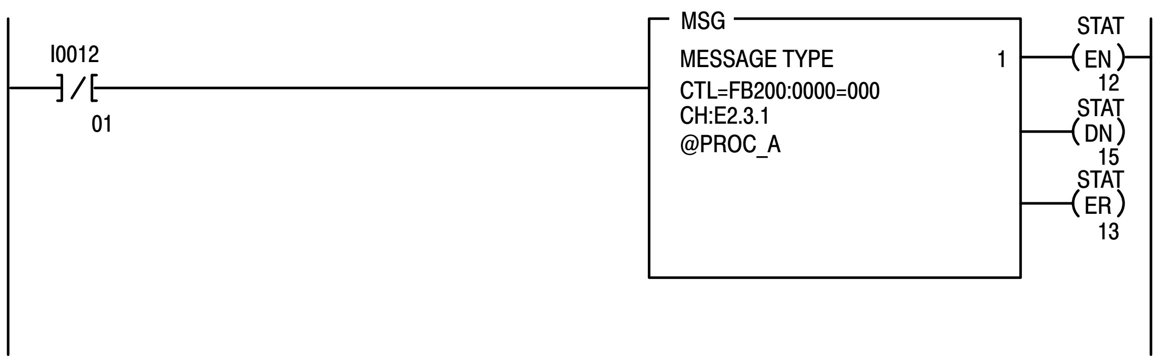
Prompt (if any)	Action
	Start editing session. Insert a new rung. Enter the bit that conditions the message rung. Enter the message instruction.
ENTER FILE ADDRESS	Enter the address of the file where the message resides in memory. This file should be binary.
ENTER SYSTEM ADDRESS OR SYMBOL	Enter the channel designation for the scanner: For example, E2.3.4: • E2 is the module status • 3 is the scanner module type • 4 is the thumbwheel number
ENTER MESSAGE TYPE	Enter the message type. This is always 1. Enter a single assignment command or the name of a message procedure. In this case, the message procedure name is PROC1. End the editing session.

Figure 6.1
Examples of Message Instructions

Reading word 5 of binary file 3 at station 045 into word 17 of integer file 4.



Execute procedure @PROC_A



Allocating Memory

Before the scanner can transfer data to or from any file in memory, the file must exist and be large enough to accommodate the data transfer. You can create and allocate a file using memory management through your programming terminal. Refer to the user's manual for your programming terminal for detailed information.

Editing Message Procedures

Table 6.B shows how to edit a message procedure through an RS-232-C data terminal connected to a 1775-GA module.

Table 6.B
Editing a Message Procedure Through an RS-232-C Data Terminal

Prompt (if any)	Action	Keystrokes
GA1>	Enter the edit mode and create the message procedure name. The GA module creates the symbol definition for the message procedure name.	**EDIT/H @PROC1 [RET]**
<EOB> *	Enter the insert mode for editing. Enter the message procedure commands. You must use an EXIT or STOP command to end each procedure. Exit the insert mode. Exit from the editing mode.	**I [RET]** **(other commands)** **#H022$B0:5=$N3:1 [RET]** **$B0:6=$N3:1 [RET]** ***2 [RET]** **EXIT [RET]** **[RET]** **E[RET]**
GA1>		

Using Symbols

You can also use symbols to represent data and data addresses in message commands.

- A symbol can consist of numeric digits, alphabetic characters, and the underscore (_). The scanner does not allow any other special characters.

  Important: The first character in a symbol must be a letter (A - Z).

- The scanner recognizes both upper-case and lower-case letters in a symbol as different characters

  For example, ASYMBOL and Asymbol are two different symbols.

- A symbol can be any length, but it must be unique in its first eight characters

  For example, the scanner can distinguish SYMBOL_A and SYMBOL_B in a message procedure, but NEW_SYMBOL_A and NEW_SYMBOL_B are not.

Important: The scanner does not flag indistinguishable symbols as programming errors. It treats similar symbols as equivalents.

You cannot use certain words and character combinations as symbols because they are reserved for special uses in message procedures. The reserved words are:

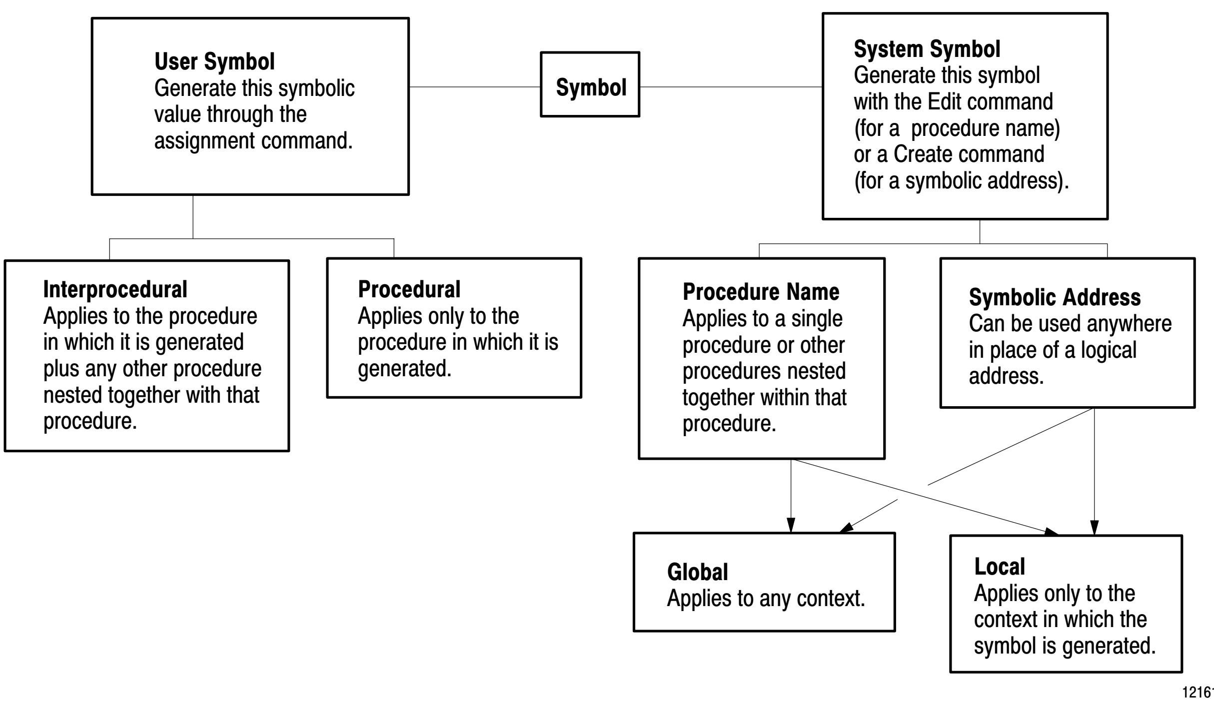
- CREATE
- DELETE
- ERROR
- EXIT
- GOTO
- IF
- ON_ERROR
- STOP
- PROT
- UNPRO

You also cannot use any abbreviated form of the above words. For example, you cannot use the single letter C as a symbol because it is an abbreviation of the word CREATE. Similarly, PRO is an invalid symbol.

The two types of symbols are (Figure 6.2):

- user symbols
- system symbols

Figure 6.2
Types of Symbols



Using User Symbols

A user symbol represents a numeric value. You can generate a user symbol and assign a value to it by means of the assignment command (see chapter 5).

Type	Description
procedural	The scanner recognizes procedural user symbols only in the procedure that you generate it in.
interprocedural	The scanner recognizes interprocedural user symbols in the procedure that you generate it in and any other procedure nested within that procedure.

User symbols can contain data that is up to 32 bits long. If the high-order bits are insignificant (that is, if they can be truncated without changing the value of the data), then the scanner can store the contents of the user symbol in a data field that is less than 32 bits long. Attempting to put a data value into a field that is too small, causes the scanner to generate error code 189 (see appendix B).

Using System Symbols

Use a system symbol as either a procedure name or a symbolic address. You must name a system symbol following the general rules given in "Using Symbols," page 6-4. The character @ delimits a system symbol and distinguishes it from a user symbols.

Symbol Type	Description
procedure names	A procedure name is a way of referring to a message procedure. You assign a procedure name at the time you generate or edit the message procedure. One procedure can execute a second procedure simply by stating the name of that second procedure. This allows for nesting of procedures up to 3 levels deep.
symbolic addresses	A symbolic address is another way of representing the logical address of data. You can generate a symbolic address by using the CREATE command. You can use a symbolic address for a logical address anywhere in a message procedure. The system symbols area of memory stores all symbolic addresses.
scope of system symbols	System symbols can be either local or global in scope. The scanner recognizes a global system symbol in any context. It recognizes a local system symbol only in the operating context that the system symbol is created in. For detailed information on context, refer to the PLC-3 Family Controller Programming Manual (publication 1775-6.4.1). At the time you generate the system symbol, you specify it to be local or global. If you do not specify the scope of the system symbol, it defaults to local. **Important:** Do not confuse the terms local and global symbols with local and remote stations. Both local and global symbols have meaning only at the station where you generate them.

Using Expressions

Expressions use operators to combine two or more numeric values into a single value. Table 6.C lists the operators that you can use in an expression. We list these operators from highest priority (1) to lowest priority (10). The result of an expression depends on the order the scanner executes the operations. The order of execution depends on the type of operator and on left-to-right placement within the expression.

The result of an expression is a 32-bit value. If the high-order bits are not significant (that is, if they can be truncated without changing the value of the expression), then a data field that is less than 32 bits long can store the value. Attempting to put a value into a field that is too small results in error code 189 (see appendix B).

Table 6.C
Expression Operators

Operator	Order of Execution	Description
/	1	bit operator
.NOT.	1	logical complement
˜ or .BNOT.	1	bitwise 32-bit complement
*	2	multiplication of 32 bits
%	2	division of 32 bits
+	3	addition of 32 bits
–	3	subtraction of 32 bits
<<	4	left arithmetic shift
>>	4	right arithmetic shift
& or .BAND.	5	bitwise 32-bit AND
^ or .BXOR.	6	bitwise 32-bit EXCLUSIVE OR
| or .BOR.	7	bitwise 32-bit OR
.EQ.	8	compare equals
.GT.	8	compare greater than
.GE.	8	compare greater or equal
.LT.	8	compare less than
.LE.	8	compare less or equal
.NE.	8	compare not equal
.AND.	9	logical AND
.OR.	10	logical OR

You can nest expressions within other expressions by enclosing the inner expression within parentheses.

For example, the following command stores the value 12 in word 45 of binary file 67.

```
$B67:45 = 6+3*2
```

This occurs because the scanner multiplies before it adds.

If an expression contains several operators within the same order of execution, the scanner executes those operators in the left-to-right order in which they appear within the expression.

You can enter extra sets of parentheses to change the order of execution. In such cases, the scanner evaluates the expression within the inner-most set of parentheses first.

For example, the following command stores the value 2 in word 45 of binary file 67 (% is the operator for division):

```
$B67:45 = 36%((6+3)*2)
```

You can use expressions anywhere for direct numeric values in a message procedure, including within an address field.

For example, in the following statement, the expression (WORD+3) specifies the address of a word within binary file 67:

```
$B67:(WORD+3) = 5
```

The parentheses are necessary to indicate that +3 is part of the word address in this case.

Interpreting Expressions

Within an expression, the scanner always interprets direct numeric values as decimal (base 10) numbers unless you indicate that they are octal (base 8). Specify an octal value by starting the number with a leading zero.

For example, the scanner interprets 17 in an expression as decimal 17, but it interprets 017 as an octal 17 (or decimal 15).

Bit Operator

The bit operator allows you to address a specific bit of a value stored under a user symbol. For example:

```
$l12:24/7 = US_13/4
```

This statement puts the value (0 or 1) of bit number 4 of user symbol US_13 into input file 12, word 24, bit 7.

The bit address itself can also be a user symbol or an expression. For example:

```
$l12:24/7 = US_3/(4+US_1)
```

The expression (4+US_1) specifies a particular bit within user symbol US_3.

The value appearing after the bit operator must be within the range of values allowed for bit addresses. Since user symbols are 32-bit values, a bit address for a user symbol must be in the range of 0 to 31 (decimal). Bit addresses for data table words must fall in the range of 0 to 15 (decimal).

Logical Operators

The logical operations are NOT, AND, and OR. These operations construct logically true or false conditions. You generally use them in decision statements such as the IF command.

The result of a logical complement is 1 (true) if the expression following the .NOT. is a value of 0. Otherwise, the result is 0 (false).

For example:

```
$l12:24 = .NOT.SYMBOL_A
```

If SYMBOL_A is:	Then the scanner stores this value in input file 12 word 24:
0	1
not 0	0

The result of a logical AND is 1 (true) if the expression preceding the .AND. and the expression following the .AND. are both non-zero. Otherwise, the result is 0 (false).

The result of logical OR is 1 (true) if either the expression preceding the .OR. is non-zero, the expression following the .OR. is non-zero, or both expressions are non-zero. Otherwise, the result is 0 (false).

Bitwise Operators

Bitwise operators manipulate the individual bits in a 32-bit operand.

The bitwise 32-bit complement (.BNOT.) inverts the state of each bit in the 32-bit expression. That is, bits set to 1 invert to 0, and bits set to 0 are invert to 1.

The bitwise 32-bit AND (.BAND.) forms a bit-by-bit logical AND of two 32-bit operands. No carry occurs from one bit position to the next within the operand. For example:

A contains the bit pattern:
`10101010010011110010101010101011`

B contains the bit pattern:
`01110101011100100010101110001010`

Then the assignment C=A.BAND.B yields

C contains the bit pattern:
`00100000010000100010101010001010`

The bitwise 32-bit EXCLUSIVE OR (.BXOR.) forms a bit-by-bit logical XOR of two 32-bit operands. No carry occurs from one bit position to the next within the operand.

The bitwise 32-bit OR (.BOR.) forms the bit-by-bit logical OR of two 32-bit operands. No carry occurs from one bit position to the next within the operand.

Arithmetic Operators

The arithmetic operations are addition, subtraction, multiplication, and division. These are binary (not BCD) operations that produce 32-bit signed integer results.

You should assign a result from these arithmetic operations to a 32-bit destination. However, you can assign the result to a 16-bit destination if the result is small enough in absolute value (less than 65,535) to fit into 16 bits. If you assign the result to a 16-bit word and the result is too large to fit into 16 bits, then an error code of 215 results.

The scanner does not indicate overflow or underflow conditions with arithmetic operations.

Shift Operators

The scanner supports the following shift operators:

If you see:	It means the bit shifts:
<<	left
>>	right

The shift operators shift binary values a specified number of bit positions to the left or right.

When a left arithmetic shift (<<) executes, zeros shift into the rightmost bits of the expression. The leftmost bits shift out of the expression and are lost, thus possibly changing the sign bit. The left bit shift operates in this manner whether operating on 16 or 32 bit values.

```
PUMP = (PUMP << 1)
```

PUMP Value:	Bit Pattern (32-bits):	Decimal Value:
Before operation	100...00000101	–2147483643
After operation	000...00001010	10

In a similar manner, when a right bit shift (>>) is executed on a 16 bit value in the data table, zeros shift into the leftmost bits of the expression. The rightmost bits shift out of the expression and are lost. As shown in this example, the sign can change when using the shift operators on 16-bit values.

```
$N0:1 = ($N0:1 >> 1)
```

$N0:1 Value:	Bit Pattern (16-bits):	Decimal Value:
Before operation	10000011 00000000	–32000
After operation	01000001 10000000	16768

However, the right bit shift works differently when the operation is performed on 32 bit values such as are created with user symbols, and also values in the high order integer data table section. In this case, the leftmost bit of the expression does not change.

If the leftmost bit is a:	Then:
1	a 1 is shifted in from the left.
0	a 0 is shifted in from the left.

Since the leftmost bit of an expression is the sign bit, this means that the right bit shift on a 32-bit value does not change the sign of the numeric value. The rightmost bits still shift out of the value and are lost as shown in the following examples:

```
GRINDER = (GRINDER >> 1)
```

GRINDER Value:	Bit Pattern (32-bits):	Decimal Value:
Before operation	100...00000101	–2147483643
After operation	110...00000010	–1073741822

```
$H0:1 = ($H0:1 >> 1)
```

$H0:1 Value:	Bit Pattern (32-bits):	Decimal Value:
Before operation	010...00000101	1073741829
After operation	001...00000010	536870914

Comparison Operators

Comparison operators result in a value of 1 if the comparison is true and 0 (zero) if the comparison is false. For example:

```
$I12:24 = ($CACC:1.GE.$CACC:2)
```

If the accumulated value of counter 1 is:	Then the scanner stores this value in input file 12 word 24:
greater than or equal to the accumulated value of counter 2	1
less than the accumulated value of counter 2	0

Message Procedure Commands

The scanner has its own command language that you can use in programming message procedures. Table 6.D summarizes these commands.

Table 6.D
Message Procedure Commands

Command	Format and Explanation	See page:
(assignment)	*destination = source* Assign a numeric value to a user symbol or copy data from the source to the destination (see chapter 5).	
CREATE	C *@system symbol$logical address* Create a symbolic address and equate it to a logical address.	6-14
DELETE	D *@system symbol* Delete a symbolic address or an entire message procedure from memory.	6-14
(execute)	*@system symbol* Execute the named message procedure.	6-15
EXIT	E Terminate execution of the current message procedure.	6-15
GOTO	G *label* Continue executing the current procedure from the point specified by the label.	6-15
IF	I *expression embedded command* Execute the embedded command only if the specified expression is true.	6-16
ON_ERROR	O *embedded command* Execute the embedded command only if an error occurs after this statement in the procedure.	6-16
STOP	S Terminate execution of the message (MSG) instruction in the ladder program.	6-17

You can abbreviate each command to the letters shown in the format and Explanation column of Table 6.D. We recommend that you use the command abbreviations because they:

- make the commands easier to program
- save memory space
- reduce execution time

You can insert spaces in command lines to make the message procedure easier to read. However, you should keep spaces to a minimum, because they use memory space and slow execution of the message procedure.

CREATE Command

The CREATE command generates a symbolic address and assigns it to a logical address. To create a local symbolic address, use the CREATE command by itself (the default modifier /LOCAL is optional). To create a global symbolic address, use the modifier /GLOBAL after the CREATE command. A procedure executing in any context can use any global symbolic address, whereas local symbolic addresses are recognized only by procedures executing in the same context. In either case, the symbol has meaning only at the station where it is created.

You can abbreviate the modifier /GLOBAL to /G and /LOCAL to /L.
For example:

```
C/G @TOTAL $E0.0.0.7
```

The scanner creates the global system symbol TOTAL to represent the logical address E0.0.0.7.

Do not confuse this CREATE command for generating symbolic addresses with the CREATE command for allocating file space in PLC-3 programming.

DELETE Command

Use the DELETE command to:

- delete message procedures from memory
- delete symbolic addresses
- delete interprocedural user symbols

Using the DELETE command on a:	Deletes the:
procedure name	name and erases the named procedure from memory.
symbolic address or user symbol	symbol, but the data stored under that symbol remains intact.

To delete a local symbol, a local procedure or a procedural user symbol, use the DELETE command by itself (the modifier /LOCAL is optional). To delete a global symbol, a global procedure, or an interprocedural user symbol, use the modifier /GLOBAL after the DELETE command.

You can abbreviate the modifier /GLOBAL to /G and /LOCAL to /L.
For example:

```
D/G @PARTS_PG
```

The scanner deletes the global procedure PARTS_PG from PLC-3 memory.

Execute Command

To execute a message procedure, enter the delimiter @ followed by the procedure's name. For example:

```
@FIRST_PROC
```

The scanner executes the procedure FIRST_PROC.

You can use procedure names anywhere that commands are used. This way one procedure can execute (call) another procedure. This allows for nesting of procedures. However, you cannot nest more than 3 layers deep.

EXIT Command

The EXIT command terminates execution of the current message procedure. If another procedure calls (executes) the current procedure, the EXIT command returns control to the calling procedure at the line following the execute statement.

The format for the EXIT command is the single letter E without any modifiers or parameters.

Each main procedure and nested procedure must end with either an EXIT command or a STOP command. The EXIT command is preferred because the STOP command results in error 179 (see appendix B).

GOTO Command

The commands in a message procedure normally execute sequentially. The GOTO command can change the order of execution.

The parameter for a GOTO command is a label. Labels are signposts, or tags, that mark a location within the message procedure.

To generate a label, enter it on any one of the lines in a message procedure. For example:

```
LABEL_A:
```

Nothing else can appear on the same line with the label. The label itself must conform to the same rules of construction as user symbols do.

Important: The trailing colon (:) is required when you first generate the label, but do not use the colon any other time you refer to the label.

When a GOTO command executes, execution of the message procedure resumes with the first command after the label specified in the GOTO command. You cannot use the GOTO command to jump from one procedure to another, even if the procedures are nested.

IF Command

The IF command makes logical decisions in the message procedure. The first parameter of the IF command is an expression (see "Using Expressions," page 6-7). You must enclose the entire expression in parentheses. Use multiple operators for complex or nested expressions.

The second element in the IF command is an embedded command. If the value of the expression is true (1), the embedded command executes. If the value of the expression is false (0), the embedded command does not execute. The embedded command can be any available command except another IF or an ON_ERROR command.

Figure 6.3 shows the combination of a label, a GOTO command, and an IF command to construct a simple loop that assigns the integers 0 through 7 to successive words in binary file 50.

Figure 6.3
Example of Looping

```
(other commands)
NUM = 0
LOOP:
$B50:(NUM) = NUM
NUM = (NUM + 1)
IF (NUM .LE. 7) GOTO LOOP
(other commands)
```

ON_ERROR Command

The ON_ERROR command specifies what action should be taken if an error occurs during execution of the message procedure. The ON_ERROR command does not execute sequentially in the procedure; it executes only when an error occurs.

The ON_ERROR command contains an embedded command and applies to all other commands between itself and the next ON_ERROR command. For example:

```
command line 1
command line 2
ON_ERROR GOTO RECOVER
command line 3
command line 4
ON_ERROR ERR_CODE = $B2:16
command line 5
```

In these command lines, the first ON_ERROR command applies to command lines 3 and 4, while the second ON_ERROR command applies to command line 5.

If an error occurs in a command line that is not associate with an ON_ERROR command, the procedure stops executing. See “Recovery from Erros,” page 7-1.

Refer to Appendix B for a listing of the error conditions.

STOP Command

The STOP command terminates execution of the message instruction in the ladder program. This means that the STOP command stops execution of the current procedure and all procedures nested together with the current one.

The format of the STOP command is the single letter S without any modifiers or parameters.

Each main procedure and nested procedure must end with either an EXIT or a STOP command. The STOP command is a extreme means of terminating a message procedure, so you should only use it when no other action is possible. The normal means of terminating a procedure is the EXIT command. The STOP command generates error code 179 (see Appendix B).

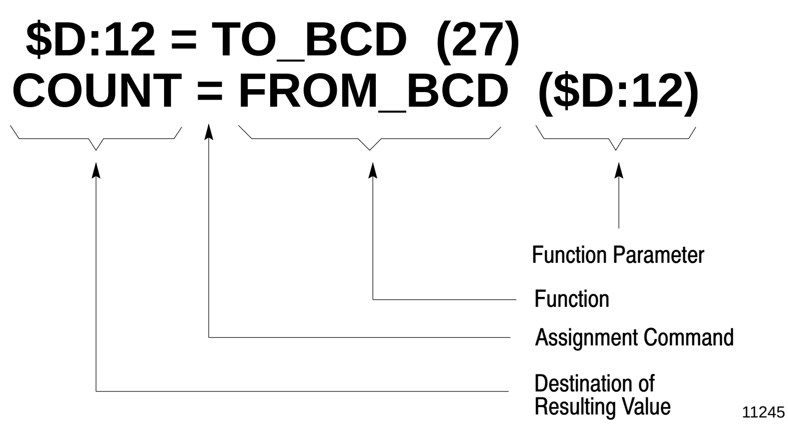
Using Functions

In addition to containing commands and nested procedures, a message procedure can also contain functions. You can use the following functions anywhere expressions are used (Figure 6.4):

- TO_BCD
- FROM_BCD

The TO_BCD and FROM_BCD perform opposite conversion functions.

Figure 6.4
Example Command Lines Using the TO_BCD and FROM_BCD Functions



You must enclose the parameter of the function in parentheses. The parameter can be a/an:

- direct numeric value (either decimal or octal)
- expression
- user symbol
- logical address
- symbolic address

TO_BCD Function

The TO_BCD function converts its parameter into a binary coded decimal value that is 32 bits long. For example, the TO_BCD function in Figure 6.4 stores the number 27 in binary-coded-decimal format in word 12 of the decimal section of memory. After this function executes, word 12 contains the following bit pattern:

```
0000 0000 0010 0111
```

FROM_BCD Function

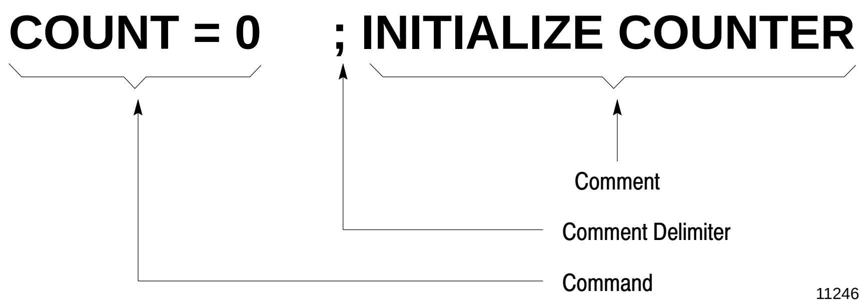
The FROM_BCD function converts its parameter from binary-coded-decimal format to binary format. The resulting value is 32 bits long. For example, the FROM_BCD function in Figure 6.4 converts the contents of decimal word 12 from binary coded decimal to a regular decimal value of 27. After this function executes, the FROM_BCD function stores the following bit pattern in user symbol COUNT:

```
0000 0000 0000 0000 0000 0000 0001 1011
```

Adding Comments to Your Message Procedure

You can add comments to any command line in a message procedure. To do this, enter a semicolon (;) after the command. Then enter your comment after the semicolon (Figure 6.5).

Figure 6.5
Format for Adding Comments to Command Lines



Anything that appears between a semicolon and the end of the command line is a comment. Comments can be any length. The end of the command line, and therefore the end of your comment, is delimited by the carriage-return and line-feed pair of characters.

Programming Examples for a Message Procedure

Figure 6.6 and Figure 6.7 provide examples of DH or DH+ message procedures that monitor the state of a status bit in a remote station.

Figure 6.6
Example Message Procedure that Executes Read and Write Commands over a DH or DH+

Procedure – @DATA

```
A = 10
B = $N0:0                                 ;procedural user symbol
IF ($10:012/017) $D2:0 = #H011$D6:0       ;read word D6:0 from station 11 into word D2:0
IF ($10:012/016) #H011$D6:1 = $D2:1       ;write word D2:1 into word D6:1 of station 11
IF (A .EQ. B) GOTO LABEL
EXIT
LABEL:
$D7 = #H011$D3                            ;note that the destination file must be ≥ the source
                                          file
#H011$N3:5 = $N2:4,10                     ;write 10 words starting at word N2:4 into
                                          ;station 11 starting at word N3:5
```

Figure 6.7
Example Message Procedure that Monitors the State of a Bit in a Remote Station and Records Any Error Condition

Procedure –– @REM_TURNON

```
;This procedure monitors the state of a bit in a remote station.
;When that bit goes true, the scanner turns on a bit locally for
;either 300 seconds or until the remote bit goes false.

ON_ERROR @LOG_ERROR                     ;log errors and time of day
A == 0                                  ;initialize error pointer (interprocedural user symbol)
CREATE @TIM_START $B0:0                 ;timer start word (local system symbol)
CREATE @TIM_CTL $TCTL:1                 ;timer control word
CREATE @TIM_PRE $TPRE:1                 ;timer preset word
T_ON_BIT = 0                            ;timer on bit (procedural user symbol)
T_DONE_BIT = 017                        ;timer done bit
CREATE @PROCESS $N3:7                   ;process word
P_ON_BIT = 5                            ;process on bit
ON = 1
OFF = 0
LOOP1:                                  ;check remote bit in loop
$B0:0/1 = #H023$B5:3/2                  ;fetch and save remote bit
IF ($B0:0/1 .EQ. OFF) GOTO LOOP1
@TIM_PRE = 300                          ;set timer for 300 sec
@TIM_START/T_ON_BIT = ON                ;turn timer rung condition on
@PROCESS/P_ON_BIT = ON                  ;turn process on
LOOP2:                                  ;check timer and remote bit in loop
$B0:0/1 = #H023$B5:3/2                  ;fetch and save remote bit
IF (($B0:0/1 .EQ. ON) .AND. (@TIM_CTL/T_DONE_BIT .EQ. OFF)) GOTO LOOP2
@PROCESS/P_ON_BIT = OFF                 ;turn process off
EXIT
```

Procedure –– @LOG_ERROR

```
;This procedure reads the error block out of the module status area and
;records it along with the time of day in status file 9.

CREATE @STATUS $S9
CREATE @ERR_BLK $E2.3.TH.3.5.4.0
CREATE @TOD $S1:3
@STATUS:(A) = @ERR_BLK,26               ;copy error block (26 words)
@STATUS:(A) = @TOD,2                    ;copy time of day (hrs, mins)
IF ((ERROR .GE. 81) .AND. (ERROR .LE. 92)) GOTO NO_STN
                                        ;no station-fatal error
IF (A .GE. 252) GOTO TIMEOUT            ;after ten errors, tell operator
A == A + 28                             ;update error pointer
EXIT
NO_STN:
$S4:3/5 = 1                             ;energize 1775-GA report generation message rung
STOP                                    ;exit procedure with an error
TIMEOUT:
$S4:3/4 = 1                             ;energize 1775-GA report generation message rung
A==0                                    ;clear error pointer
EXIT                                    ;return to @REM_TURNON
```

The 300-second timer in the example procedure given in Figure 6.7 is not an accurate, real-time clock. This is because the time between successive executions of the bit/timer check depends on activity on the DH or DH+ and on the activity of the local PLC-3 controller.

For example, if the 300-second timer times out immediately after its done bit is checked, the scanner does not detect this condition until its next pass through LOOP2. If the link is busy with other activity, it takes a while for LOOP2 to check the remote bit.

The example procedure also assumes that the referenced memory areas have been created. Specifically:

1. Status file S9 must be big enough to hold ten errors of 28 words each.

2. Timer T1 is a one-second timebase timer. Bit B0:0/0 controls the ladder rung that activates the timer. Figure 6.7 refers to this bit as TIM_START/T_ON_BIT.

3. Bit S4:3/4 and S4:3/5 activate message instructions that execute report-generation procedures. In this way, the scanner can indirectly cause execution of a report-generation procedure that displays a message on the operator’s terminal.

Chapter 7

Diagnostics Methods

Chapter Objectives

This chapter describes how the scanner detects and reports various types of errors. Appendix B lists all the errors reported by the scanner.

DH/DH+ Message Procedure Diagnostics

Error Reporting

When an error occurs, the scanner automatically creates and stores the error code in the interprocedural user symbol ERROR. You should reserve the symbol ERROR exclusively for error reporting.

ERROR contains only the last error encountered during execution of a command or message procedure. The scanner clears the ERROR symbol whenever a procedure is finished, even a called procedure. If you want to save the error code or manipulate it in any way, you can use an assignment command to copy the code into a more permanent storage word.

Recovery from Errors

Unless you specify differently, the scanner stops executing the current message procedure as soon as the module detects an error. To specify a different action, use the ON_ERROR command in the message procedure. Then, when the scanner encounters an error, it performs the action specified in the nearest preceding ON_ERROR command. After the module completes executing the ON_ERROR action, it resumes executing the message procedure at the command line following the one in which the error occurred.

For example, a message procedure can contain the command:

```
ON_ERROR @RECOVER
```

When an error occurs in the procedure, the ON_ERROR command causes the scanner to execute the procedure named RECOVER. The procedure RECOVER might be a routine for monitoring error codes. After executing RECOVER, the scanner resumes executing the original procedure at the command line following the one where the error occurred.

Error Monitoring

To aid in error monitoring, the scanner maintains a 26-word-error block in the module status area of memory. The extended address for the beginning of the error block file is:

$E2.3.*thumbwheel_number*.**3.5.4.0**

You can access this error block by:

- displaying it on the controller front panel (displayed in hex)
- using the data monitor mode of your programming terminal (displayed in hex)
- using the move status (MVS) command in the ladder program
- using procedure commands in the following modules:
  - 1775-S5 or 1775-SR5 module
  - 1775-GA module
  - 1775-KA module

Error Block

This error block contains the following information:

Word	Contains the
0	last generated error code (see appendix B)
1	total number of errors that have occurred
2-7	error information for message task 1
8-13	error information for message task 2
14-19	error information for message task 3
20-25	error information for message task 4

Four groups of words are provided for the message tasks since the scanner can execute four message instructions at once. Within each message task, the words contain the following information:

Task Word	Contains the
0	error code for the last error that occurred in the message task
1	total number of errors that occurred in the message task
2	value 1
3	line number where the error occurred in the highest level (nest level 1) message procedure
4	line number where the error occurred in the next highest level (nest level 2) message procedure
5	line number where the error occurred in the lowest level (nest level 3) message procedure

The line number is the relative location of a command line from the beginning of the message procedure containing the line. The first line of each procedure is line number 1, and any following lines are numbered in ascending sequence. Nested procedures begin with line 1 again, thus the need for words 3, 4 and 5 in the error block.

Important: You do not enter the line numbers for a procedure; the scanner automatically keeps track of the line numbers for you.

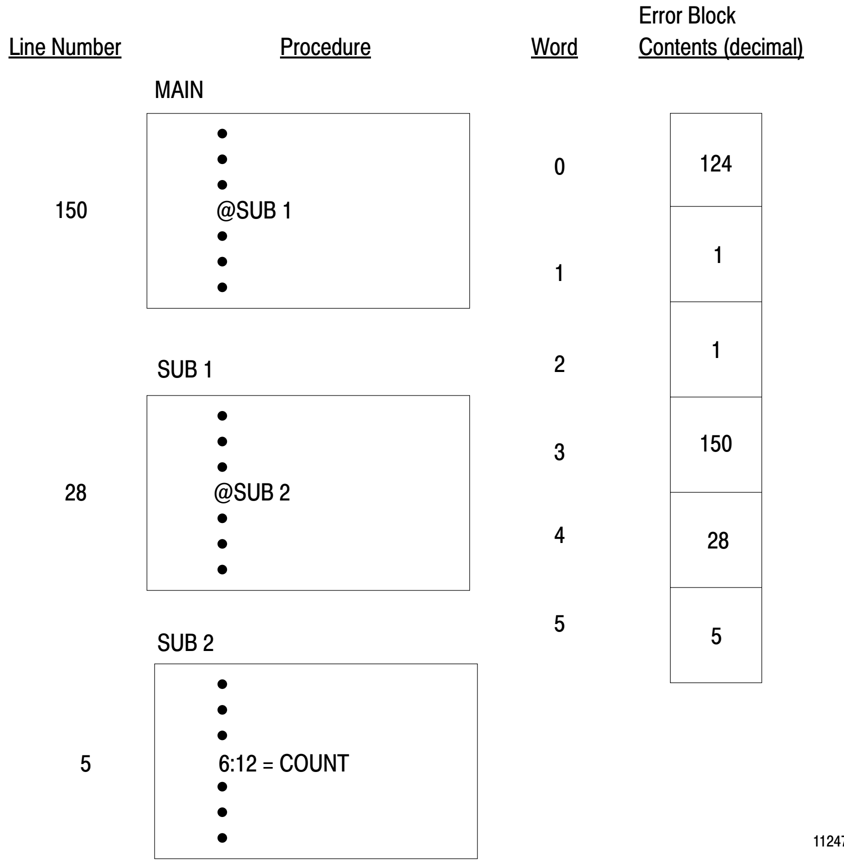
The error block retains its data even after the message procedures complete execution. The block does not reset automatically.

Error Block Operation

Figure 7.1 illustrates how the error block works. In this figure, an addressing error (invalid destination address) occurs in procedure SUB2, which is nested 3 levels deep.

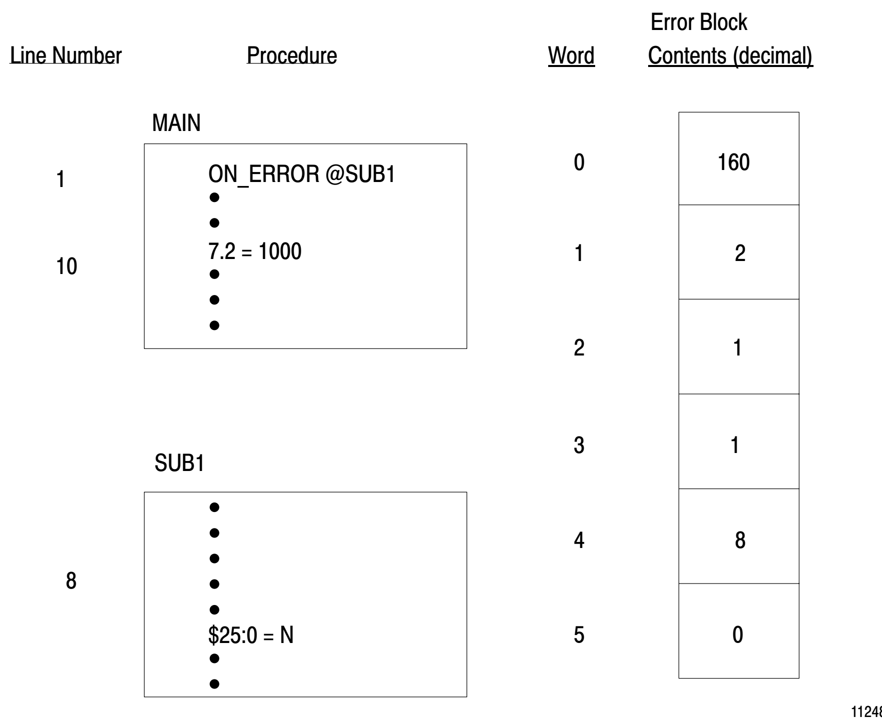
Word:	of the error code gives the line number where error occured in procedure:
5	SUB2.
4	SUB1 that executed procedures SUB2.
3	MAIN that executed procedure SUB1.

Figure 7.1
Error Block Operation



ON_ERROR or IF commands can contain an embedded command to execute another procedure. In these cases, the scanner treats the embedded execute command just like a nesting level (Figure 7.2).

Figure 7.2
Example Procedures Showing ON_ERROR Nesting



In this example (Figure 7.2), an addressing error in line 10 of procedure MAIN causes the ON_ERROR command to execute. The ON_ERROR command calls for execution of procedure SUB1. But SUB1 also contains an error. The error in SUB1 is the last one detected, so it is the one finally reported in the error block. Since procedure SUB1 is called by the ON_ERROR command in procedure MAIN, the nesting for SUB1 is 2 levels deep.

Diagnosing Faults with Module Status Indicators

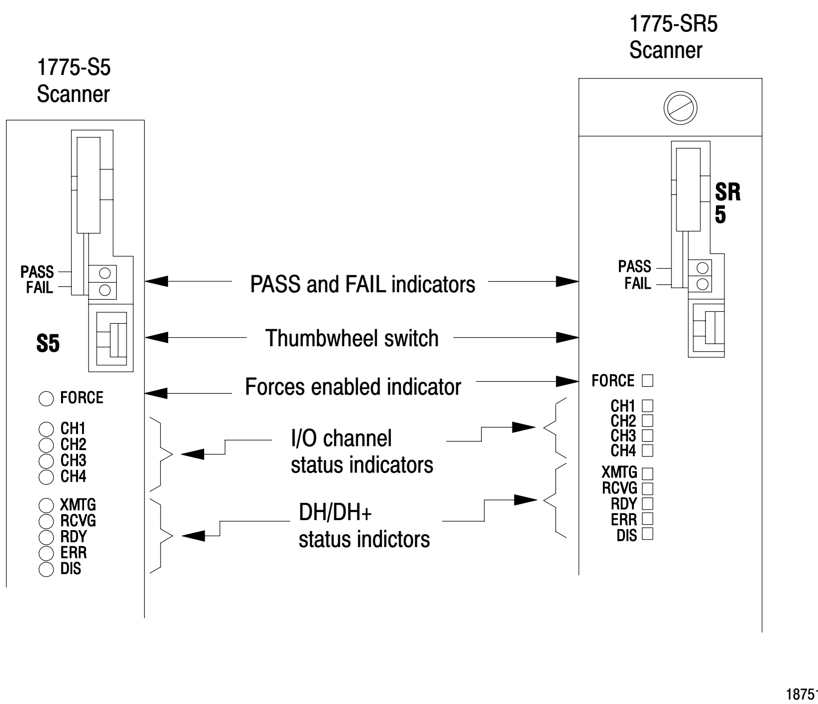
All PLC-3 modules have two self-test indicators. Table 7.A describes how to interpret these indicators.

Table 7.A
Module Self-test Indicators

Indicator and Status		Description
PASS	FAIL	
ON	OFF	Normal operation
OFF	ON	Module fault
ON	ON	Power-up or system reset
OFF	OFF	Processor shut off

The scanner modules have several indicators to monitor its status (see Figure 7.3).

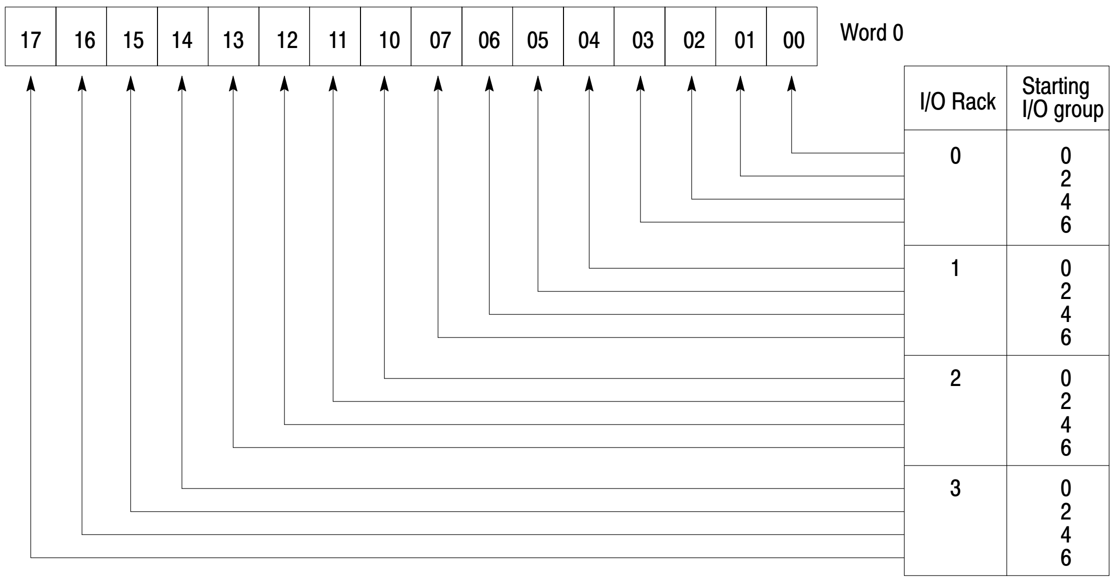
Figure 7.3
Indicator Locations for 1775-S5 and 1775-SR5 Scanner Modules



Remote I/O Adapter Faults (Status File 2)

Figure 7.4 shows the word organization for status file 2.

Figure 7.4
Remote I/O Device Fault Status Bits (Status File 2)



To find the word that stores the fault bits for an I/O adapter:

Rack # in decimal / 4 = word in file 2 containing fault bits for the adapter.
The remainder tells you the bit numbers within the word:

0 – bits 00 to 03
1 – bits 04 to 07
2 – bits 10 to 13
3 – bits 14 to 17

For example, the fault bits for I/O rack 10 are in word 2 (S2:2), bits 10 to 13.

15180

Important: The processor does not create status file 2 at power-up. You must create it in memory by using the create command.

I/O Communication Retry Counts (Status File 3)

A communication retry is a re-transmission of data that occurs when the original transmission is unsuccessful. If the I/O adapter does not respond or sends invalid data when the scanner communicates to the I/O adapter, the scanner executes a retry.

You can track the retry count by monitoring status file 3 (see Figure 7.5).

Figure 7.5
I/O Communication Retry Counts (Status File 3)

17	16	15	14	13	12	11	10	07	06	05	04	03	02	01	00
word 0															
word 1															
word 2															
word 3															
word 4															
word 5															
word 6															
word 7															
. . .															
word n															

Adapter #	Starting I/O group
0	0 2 4 6
1	0 2 4 6

To find the word that stores the retry count for an I/O adapter:

Adapter # in decimal x 4 = the first of the words that stores the retry counts for the adapter. The particular word depends on the starting I/O group.

For example, the retry counts for adapter 10 are in words 40 to 43 (S3:40 to S3:43)

15181

Important: The processor does not create status file 3 at power-up. You must create it in memory by using the create command.

If the retry count is high for an I/O channel, check to see if it is caused by one of the following:

- loose connection with the twinaxial cable (cat. no. 1770-CD) that connects from the scanner's terminal arm to the I/O adapter
- noise problem with the twinaxial cable (cat. no. 1770-CD) that connects from the scanner's terminal arm to the I/O adapter
- improper installation of I/O terminator resistors along the I/O channel

Refer to the PLC-3 Family Controller Installation and Operation Manual (publication 1775-6.7.1) for detailed information.

Upon executing a retry, if the I/O adapter responds properly, normal operation continues. If the I/O adapter does not respond properly, the scanner continues to execute retries until:

- the I/O adapter returns a valid response, or
- the processor declares a major or minor I/O fault based on how you configure the rack list in LIST.

Adapters on channel:	Number of consecutive retries that execute before the processor declares an I/O fault:
1	10
2	8
3 to 7	6
8 to 16	4

If the processor declares an I/O fault, it sets the following bits:

- I/O adapter fault status bit file 2 that corresponds to the I/O adapter
- major or minor I/O fault bit in system status

The scanner continues scanning the I/O chassis. When it returns to the faulted I/O adapter, it attempts to reset the input or output and moves on to the next I/O adapter.

Important: Upon declaring a major or minor I/O fault, if normal communication returns to the I/O adapter, the processor does not reset the I/O adapter fault bit in status file 2 or the major I/O or minor I/O fault bit in system status.

If you do not want normal communication to return to a faulted I/O adapter, set the processor-reset-lockout switch on the I/O chassis. With this switch set, the I/O adapter does not reply to the scanner once a fault is declared until you cycle power at the I/O chassis or press the reset button on the I/O adapter.

Retries for a Peer-to-Peer or Backup Communication Channel

If you configure an I/O channel for peer-to-peer or backup communication, and communication problems occur between two scanners (master, slave, primary, or backup) on the channel, a retry executes. If the communication occurs properly, normal operation continues.

If communication problem continues to occur, the scanners:

- set the peer-to-peer or backup communication minor fault status bit in system status
- flash the LED corresponding to the channel on their front edges

Also, the scanner configured as the master on a peer-to-peer communication channel declares a peer-to-peer communication minor fault when it cannot successfully communicate with a slave after two consecutive retries.

Important: Once set, the peer-to-peer and backup communication minor status bit remain set until you reset it.

Table 7.B
I/O Channel Indicators for All Scanner Modules

Indicator	Status	Description
CHx Configured for I/O Scanning	ON	Communication between scanner module and the I/O chassis on the corresponding I/O channel is properly established.
	FLASHING	There is a fault on one or more of the I/O chassis on the corresponding I/O channel.
	OFF	No I/O chassis are configured on the corresponding I/O channel or the channel is inactive.
CHx Configured for peer-to-peer communication	ON	The channel is functioning properly.
	FLASHING	The input file is too small at the processor receiving data. The slave or master does not exist. Communication retry.
	OFF	The channel is inactive.
CHx Configured for backup communication	ON	The channel is functioning properly.
	FLASHING	The input file is too small at the processor receiving data. The partner is not responding.
	OFF	The channel is inactive.

DH/DH+ Indicators

Table 7.C shows the DH/DH+ status indicators for 1775-S5, -SR5 scanner modules.

Table 7.C
DH/DH+ Status Indicators for 1775-S5, -SR5 Scanner Modules

Indicator	Status	Description
XMTG	ON	transmitting a message
RCVG	ON	receiving a message
RDY	ON	ready to transmit a message
ERR	ON	programming or communication error detected
DIS	ON	DH/DH+ channel is disabled

Important: When the scanner is polling, both the XMTG and RCVG LEDs trun on. Also, if there is a duplicate node on DH+, the yellow disable indicator blinks until the condition is removed.

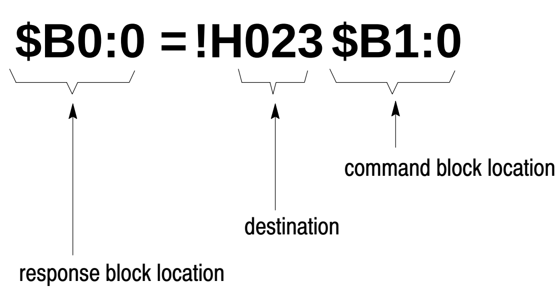
DH/DH+ Diagnostic Assignment Command

The scanner can send out DH and DH+ diagnostic commands using a special format of the assignment command. As with other assignment commands, you can program this command directly into the message instruction or use it within a DH/DH+ procedure.

ATTENTION: If the diagnostic assignment command is not properly programmed, unanticipated data can be sent to remote stations on the DH or DH+ link causing unpredictable operation.

This assignment command requires that data table be reserved for both the command to be sent and the response that will be returned from the remote station. You must reserve a minimum of 126 words of data table for the response.

The command is formatted as follows (note the special use of the exclamation point):



Where:	Is the:
`response block location`	starting address of the data table address which will be used to save the response data from the remote station.
`destination`	remote station address to where this command is being sent. This address can be any remote station address and can include a bridge address when using DH+ (for example !H1:2:3).
`command block location`	Starting address of the data table address which contains the command being sent to the remote station.

Revision A/F or later scanners execute the diagnostic assignement command, however the command block configuration was modified with the release of the A/H scanner. Table 7.D and Table 7.E show the command and response that are stored as a series of bytes with the following meaning:

Table 7.D
Command Block

Byte	Meaning
1 & 2	number of bytes minus one to follow this word
3 & 4	no care byte and command byte
5 & 6	status byte (always zero) and tns byte (always zero)
7 & 8	tns byte (always zero) and function byte
9 thru n	command data dependent on command & function

Table 7.E
Response Block

Byte	Meaning
1 & 2	no care byte and status byte (00 = success)
3 & 4	number of bytes to follow this word
5 & 6	command byte and status byte
7 & 8	tns word
9 thru n	response data (stored low byte/ high byte)

You can use the following diagnostic commands with the diagnostic assignment command:

Command	Command Byte	Function Byte
Diagnostic Loop	06	00
Diagnostic Status	06	03
Diagnostic Read	06	01
Diagnostic Counters Reset	06	07

Diagnostic Loop Command
You can use this command to check the integrity of transmissions over the communication link. The command message transmits up to 243 bytes of data to a remote station. The receiving station should reply to this command by transmitting the same data back to the scanner.

For example:

Command block in data table:
`0012 0006 0000 0000 0102 0304 0506 0708 090A 0B0C 0D00`

Response block in data table:
`0000 0011 4600 0200 0102 0304 0506 0708 090A 0B0C 0D00`

Diagnostic Status Command
You can use this command to read a block of status information from the remote station. The reply to this command contains the diagnostic status information in the response data. Each remote station defines the data sent back in response. The scanner sends back 18 bytes as defined in Table 7.H.

For example:

Command block in data table:
`0005 0006 0000 0003`

Response block in data table:
`0000 0016 4600 0300 0246 1003 5088 5E00 4002 0000 111B 0400 0000`

Diagnostic Read Command
You can use this command to read the remote stations diagnostic counters. Use the diagnostic status command to obtain the starting address of the diagnostic counters.

The first two bytes of the command data is the address of the counters (low byte/high byte) as returned from a diagnostic status command. The third byte is the number of bytes to be returned; 18 bytes for the scanner.

The response data contains the diagnostic counters. The scanner DH counters are defined in Table 7.F and DH+ counters in Table 7.G.

Command block in data table:
`0008 0006 0000 0001 5E00 1200`

Response block in data table:
`0000 0016 4600 1700 7E4B 7F4B 0000 0000 0000 0000 0000 0000 0000`

Diagnostic Counters Reset Command

Use this command to reset to zero all the diagnostic counters in the remote station.

Command block in data table:
`0005 0006 0000 0007`

Response block in data table:
`0000 0004 4600 1900`

Table 7.F
DH Diagnostic Counters

Byte:	This counter increments when the scanner:
1	receives an ACK with a bad CRC
2	does not receive an ACK before the timeout value
3	while master, detects a message transmission by another station (contention)
4	receives an error in the received ACK
5	adds errors in bytes 1, 2, and 4
6	receives a WAK when no receive buffer space is available at the remote station
7	does not detect a master on the link before its timeout period expires and so assumes mastership
8	as master, does not receive a reply from its polling sequence that has been narrowed done to a single station
9	receives an ACK even though it does not have mastership
10	receives an illegal packet of less than 6 bytes
11	receives a packet with an incorrect destination or a packet in which the source is equal to the destination
12	(not used)
13	receives a packet with a bad CRC
14	received a frame that was beyond the legal limit of 250 bytes
15	does not have the buffer space for a received message
16	received a message that was previously received
17	received aborted frame (line noise)
18 & 19	successfully sent a message
20 & 21	successfully received a message
22 & 23	successfully sent a command
24 & 25	successfully receives a reply
26 & 27	successfully receives command
28 & 29	successfully sends a reply
30	can not send a reply

**Table 7.G
DH+ Diagnostic Counters**

Byte:	This counter increments when the scanner:
1 & 2	successfully sent a message
3 & 4	successfully received a message
5	can not deliver a message due to a NAK or after using up maximum retries
6	did not receive a response before the response timeout value was reached
7	received a NAK
8	must send a message retry as a result of CRC error, illegal length, not its destination, the source was not the expected station, or a timeout
9	sent a NAK because there was no memory available
10	sent a NAK because of the received message contained an undefined LSAP
11	receives a message which was received previously
12	has to send a token pass retry
13	receives an aborted packet
14	receives a packet that had a CRC error
15	receives an illegal size packet which was either less than 3 bytes or larger than 271 bytes
16	detects a duplicate token on the link
17	recovers from a duplicate node condition
18	detects a link dead timeout

Table 7.H
Data Sent in Response to a Diagnostic Status Command

Byte:	Bit:	Description:
1		Operating status of PLC–3 processor
	0–1	00 = Program mode 01 = Test mode 10 = Run mode
	2	Not used
	3	0 = Normal 1 = Major processor fault
	4	0 = Normal 1 = Shutdown request
	5	0 = Normal 1 = Shutdown in effect
	6-7	Not used
2		Type of station interface and processor
	0-3	6 = scanner
	4-7	4 = PLC–3 processor
3	all	Current context stored in bits 4 thru 7
4	all	Thumbwheel
5-6	all	Mode control word The extended address of the mode control word is E0.0.0.8
7-8	all	Starting byte address of the diagnostic counters.
9		Series and revision number of the module. Even though the scanners are series A, they appear as series E modules
	0-3	0 = Revision A scanner 1 = Revision B scanner etc.
	4-7	0 = Series A 1 = Series B etc.
10		Type of channel communications
	0-1	0 = Inactive 1 = DH 2 = DH+ 3 = other
	2-3	0 = A/G or earlier S5 or SR5 1 = A/H or later S5 2 = A/H or later SR5 3 = unused
	4-7	unused
11-14	all	The physical address of the unused word of PLC–3 system memory. This is the physical address corresponding to the extended address E60.0.0.0
15-18	all	The total number of words in the PLC–3 system memory (both used and unused)

Appendix A

Binary Command Language

Introduction

The scanner communicates through the peripheral channel 0 on front panel with a programming device or another external control device such as a computer. The scanner uses the binary command language (BCL) for communication. This chapter describes the protocol and commands of BCL, and briefly reviews PLC-3 addressing.

Protocol

BCL is a master/slave communication protocol in which the scanner is the slave and the external device must act as the master. As slave, the scanner cannot initiate any communications. The BCL protocol defines the rules of communication between the scanner and an external control device. It includes the:

- method of initiating communication
- format in which information must be sent to the scanner
- format of scanner responses
- amount of time which may elapse between commands without causing the scanner to declare a channel timeout and stop communicating

Timeouts

BCL protocol uses four timeouts to determine if a communication problem occurs. These timeouts guard against stopping or unnecessarily slowing down the scanner or the master device when communication is terminated unexpectedly (such as when a cable is detached).

If you see:	It means:
Channel Timeout	scanner waiting for command
Character Timeout	master or scanner waiting for next character in block
Response Timeout	master waiting for response block after ACK
Acknowledge Timeout	master or scanner waiting for ACK

Channel Timeout (Scanner waiting for command)
A channel timeout occurs when the scanner is waiting for a command and does not receive one within the specified time.

You can select the channel timeout through the LIST function. The front panel timeout value defaults to 10 seconds. Selecting a value of zero disables the channel timeout.

When a channel timeout occurs, the scanner stops communicating through channel 0. The master must then re-initialize the communication.

Character Timeout (Master or scanner waiting for next character in block)
A character timeout occurs when the time between characters (bytes) in a command block, response block, CCR, or CCR response is greater than the character timeout value.

The master can set the scanner's character timeout value by sending the desired value within the CCR. If the master sends a zero value, the scanner defaults to the character timeout value appropriate for the current communication rate configuration as shown in Table A.A.

Table A.A
Communication Rate Configuration

Communication Rate (bps)	Character Timeout Default Value
110	1 second
150	600 ms
300	300 ms
600	150 ms
1200 to 19.2	80 ms

If the master or a scanner records a character timeout, it must be treated as if a NAK were received. They stop transmitting for one character timeout value and then reissue the command or response.

The scanner sends a NAK when it detects this timeout.

Response Timeout (Master waiting for response block after ACK)
A response timeout occurs when the scanner does not send a response block back to the master within a given time after acknowledging receipt of a command block.

The master device determines the time allowed (the scanner is not aware of this timeout). Typically, a one minute response timeout provides enough time for the scanner to respond without unnecessarily slowing the master device.

After a response timeout, the master device:

- sends a CCR to the scanner
- checks the protocol fault code in the CCR response (refer to Table A.A)

If the fault code is:	The master device:
00 hex (no fault) or 05 hex (no input buffer available)	clears its error count and retransmits the last block.
another fault code (besides 00 or 05 hex) or if the scanner does not respond to the CCR within one character timeout period	terminates communication or tries to re-initialize communication.

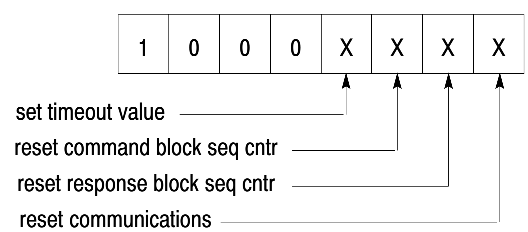
Acknowledge Timeout (Master or scanenr waiting for ACK)
The maximum amount of time the scanner waits for an ACK, NAK, or SO from the other device after sending a response or command is 2 seconds (S4A used character timeout value). Master must also select a value for this timeout.

This timeout should be treated as if a NAK were received. The driver stops transmitting for one character timeout value and then reissues the command or response.

Circuit Control Request (CCR)

The master uses the circuit control request (CCR) to initiate, reset, or verify communications with the scanner. The CCR consists of five bytes of data defined as follows:

STX	Control Code	Character Timeout Value	ETX	Checksum

Command:	Description:
STX (start of text)	the first byte of the control block (02 hex).
Control code	In the control code byte, bits 7, 6, 5, and 4 contain the value 1, 0, 0, and 0 respectively. This is necessary for the scanner to recognize the byte as a control code. The meaning of the other bits in the control code is as follows: 
Character Timeout Value	The third byte is the character timeout value. If bit 3 in the above control code is set, this byte sets the character timeout value. The timeout is 10 ms times the value in this byte, which allows timeout values between 10 and 2550ms. If bit 3 in the control code is not set, this byte is zero. A zero in this byte causes a default character timeout as shown in Table A.A.
ETX (End of Text)	03 hex.
Checksum	The fifth and last byte in the CCR is a checksum. The checksum is the least significant eight bits of the sum of the previous four bytes in the CCR.

Circuit Control Request Response

The scanner response to a master's CCR is a fourteen byte response.

STX	Control Code	Number of Bytes	Mod Type and thumbwheel	Series	Rev	Max command block size	Max response block size

Current command block seq	Current response block seq	Character timeout value	Protocol fault code	ETX	Checksum

Command:	**Description:**
STX (start of text)	the first byte of the control block (02 hex).
Control Code	Value echoes the control code (byte two) of the CCR.
Length of response block	Count of the remaining bytes including this byte and the ETX.
Module type and thumbwheel number	Module type in the upper four bits and the thumbwheel number in the lower four bits.
Scanner Series	Scanner module's series number.
Scanner Revision	Scanner module's revision number.
Maximum command block size	Scanner module's maximum command block size in bytes.
Maximum response block size	Scanner module's maximum response block size in bytes.
Current command block sequence	The current command block sequence count as maintained by the scanner.
Current response block sequence	The current response block sequence count as maintained by the scanner.
Character timeout value	The character timeout value in 10's of ms. If zero then see Table A.A for default.
Protocol fault code	The protocol fault code. The possible fault codes are listed in Table A.B.
ETX (End of text)	03 hex.
Checksum	The last byte in the CCR response is a checksum. The checksum is the least significant eight bits of the sum of all the above bytes in the response.

Table A.B
Fault Code

HEX fault code value	Description
00	No fault
01	Checksum error
02	Format fault – missing ETX
03	Format fault – other
04	Input buffer overflow
05	No input buffer available
06	Previous buffer still in progress
07	Command block sequence count error
08	Receiver error
FF	Unknown protocol failure

Block Sequence Counts

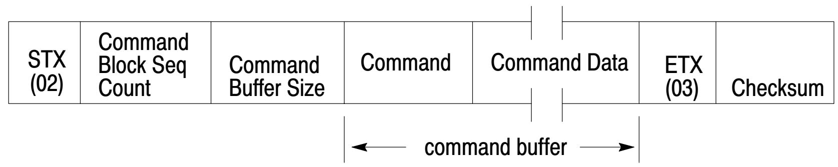
Both the master and scanner maintain a command block sequence count and a response block sequence count. These counters are separate entities and do not necessarily agree. The scanner:

- checks the command block sequence when received
- compares it to its own version of the command block sequence count
- increments the command block sequence count when a valid command is received.

The response block sequence count is incremented and sent out by the scanner. The master checks the response block sequence count when received and compares it to its own version of the response block sequence count.

Command Block

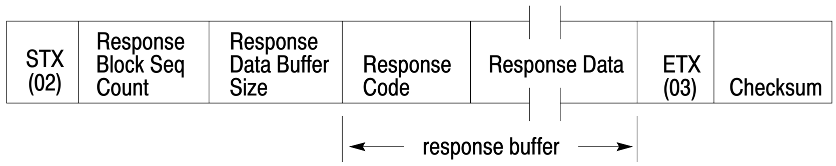
With the exception of the CCR, all commands sent from the master to the scanner are in the form of a command block. The command is always structured as follows:



Command:	**Description:**
STX (start of text)	the first byte of the control block (02 hex).
Command block sequence count	Command block sequence count (0 to 7F hex)
Command buffer size	Number of bytes to follow in the command buffer. The buffer includes all bytes between this buffer size byte and the ETX.
Command	The fourth byte in the command block contains the hex code corresponding to the desired command.
Command data	The command data contains all parameters associated with the indicated command. When the command data contains data in word format, the lower byte must be sent first. **Important:** The master can send multiple commands in one command buffer by including another command and command data after the previous one. Each command must have all necessary parameters and the entire command block cannot be larger than the scanner is capable of receiving. This maximum size should be one less than is reported in a CCR response. One word is reserved for scanner use.
ETX (End of text)	Following the command buffer is the ETX code (03 hex).
Checksum	The last byte is a checksum which is the sum of all previous bytes from the STX (02) through ETX (03) inclusively.

Response Block

With the exception of the CCR response, all responses from the scanner to the master are in the form of a response block. The response is always structured as follows:



Command:	**Description:**
STX (start of text)	the first byte of the control block (02 hex).
Response block sequence count	Response block sequence count (0 to 7F hex).
Response buffer size	Number of bytes to follow in the response buffer. The buffer includes all bytes between this buffer size byte and the ETX.
Response code	This byte contains the scanner's reaction to the previous master's command in the form of a response code. Table A.C outlines the response code possibilities.
Response data	The response data includes response parameters associated with the previous command as received from the master. When the response data contains data in word format, the lower byte is sent first. **Important:** The scanner can send multiple responses in one response buffer provided that each response has all necessary parameters.
ETX (End of text)	Following the response data buffer is the ETX code (03 hex).
Checksum	The last byte is a checksum which adds from the STX (02) through ETX (03) inclusively.

Table A.C
Response Code

HEX code	Response	Description
00	Success	Operation was completed as requested.
01	Size Too Big	Operation stopped because the response was too big to fit in the response block. Success or failure depends on the operation.
02	Available	Interpretation of this response depends on the operation. See the appropriate command descriptions for details.
03	Unavailable	Interpretation of this response depends on the operation. See the appropriate command descriptions for details.
04	Address Does Not Exist	The specified address has not been allocated.
05	Address Invalid	This response is returned when the specified start address is valid, but another address used in the operation has not been allocated.
06	Access Not Allowed	The channel does not have access to the area specified for the purpose requested.
07	Allocated	The function or resource requested is allocated and therefore unavailable (unless it is already allocated to your channel).
08	Address Not Complete	The address given is not specified to a low enough level to perform the requested function.
09	Unknown Command	The scanner does not recognize the command code.
0A	Invalid Parameter	A parameter in the command block is invalid.
0D	Processor Error	
0E	Processor Error	
0F	No Privilege	One or more privileges required for the attempted operation are not assigned to the channel.
FF	Unknown Failure	The function failed for a reason which cannot be described by any other response code.

Initializing Communications

To initiate communications with the scanner the master sends a break character followed by a CCR. The CCR should have the reset communications bit (bit 0) of the control code set. This instructs the scanner to return to its initial communications state, including resetting the command block sequence and the response block sequence count.

The master must see break go away before sending the CCR. The scanner sends break for a character timeout period. The scanner identifies a break character as any high state (voltage) which remains on the line longer than the time required to transmit one character (byte). The approximate character times are listed in Table A.D.

Table A.D
Approximate Character Times

Communication Rate in bits/s	Approximate character transmit time	
	10 bit character	8 bit character
110	91.0 ms	72.7 ms
150	66.7 ms	53.3 ms
300	33.3 ms	26.7 ms
600	16.7 ms	13.3 ms
1200	8.3 ms	6.7 ms
1800	5.6 ms	4.4 ms
2400	4.2 ms	3.3 ms
4800	2.1 ms	1.7 ms
9600	1.0 ms	0.83 ms
19200	0.52 ms	0.41 ms

The break character tells the scanner to terminate any previous communication. Therefore, send a break character whenever you initiate or terminate communication. Both the scanner and the master device respond to a break at any time by stopping transmission (even in the middle of a block).

Although not configurable on the scanner, some PLC-3 family modules which use BCL, can also be configured for BREAK DISABLED. This option is selected when the master is not capable of detecting a break sent from the PLC-3. When the channel is configured with BREAK DISABLED, the master must send break for a minimum of one second.

If the master device does not support a break character, you can initiate communications by sending a valid command to the scanner.

Important: If you use this method, remember that the PLC-3 scanner may not be in its initial communication state when you begin communicating with it. This can cause the PLC-3 to respond with invalid results until a timeout or other error causes it to terminate communication, return to its initial state, and start over.

Communication

The scanner must supply an ACK or NAK within one character timeout period of receiving a command block. As the slave on the link, the scanner has until a response timeout occurs before issuing a response block.

If the scanner receives a command block correctly, it responds with an ACK code (06 hex) and increments the command block sequence count. The master also increments the command block sequence count and resets the error count at this time.

If the scanner detects an error in the command block which it receives, it responds with a NAK code (15 hex). Some typical reasons for sending a NAK are as follows:

- received invalid command block sequence count
- unexpected command block
- unable to handle command block at this time
- character timeout value exceeded

The master stops transmitting within one-half of the character timeout period after receiving the NAK code. The master increments its error count and:

If the error count is:	The master:
less than three	retransamits the block after one character timeout period.
three	stops trying to transmit the block.

If the master does not receive an ACK or NAK code within one character timeout period after transmitting the command block, it continues as though it received a NAK.

Handshaking

After receiving a response block from the scanner the master responds within a character timeout period with one of these codes:

If you see:	It means that:
ACK (06 hex)	the command block was received correctly and that normal communication is to continue.
NAK (15 hex)	an error was detected in the response block and instructs the scanner to retransmit the block. The scanner tries three times before terminating communication.
SO (0E hex) shift out	the master needs time to process the response, and instructs the scanner to wait. The scanner must wait for the master to send a shift in character (0F hex) before continuing communication. A shift in character indicates that the master is ready to continue communication.

When the master is ready to restart communication with the scanner:

- It sends a shift in character.
- The scanner echoes the shift in character and resumes communication.
- If the scanner does not echo the shift in character within one character timeout period, the master device assumes that the character is not received, and retransmits up to a maximum of three times.
- After 3 unsuccessful attempts, the master sends a CCR to the scanner and checks the protocol fault code in the CCR response. If the fault code is:
  - 00 hex (no fault) or 05 hex (no input buffer available), the master should clear its error count and retransmit the last block.
  - Any other fault code returns, or if the scanner does not respond to the CCR within one character timeout period, the device should terminate communication.

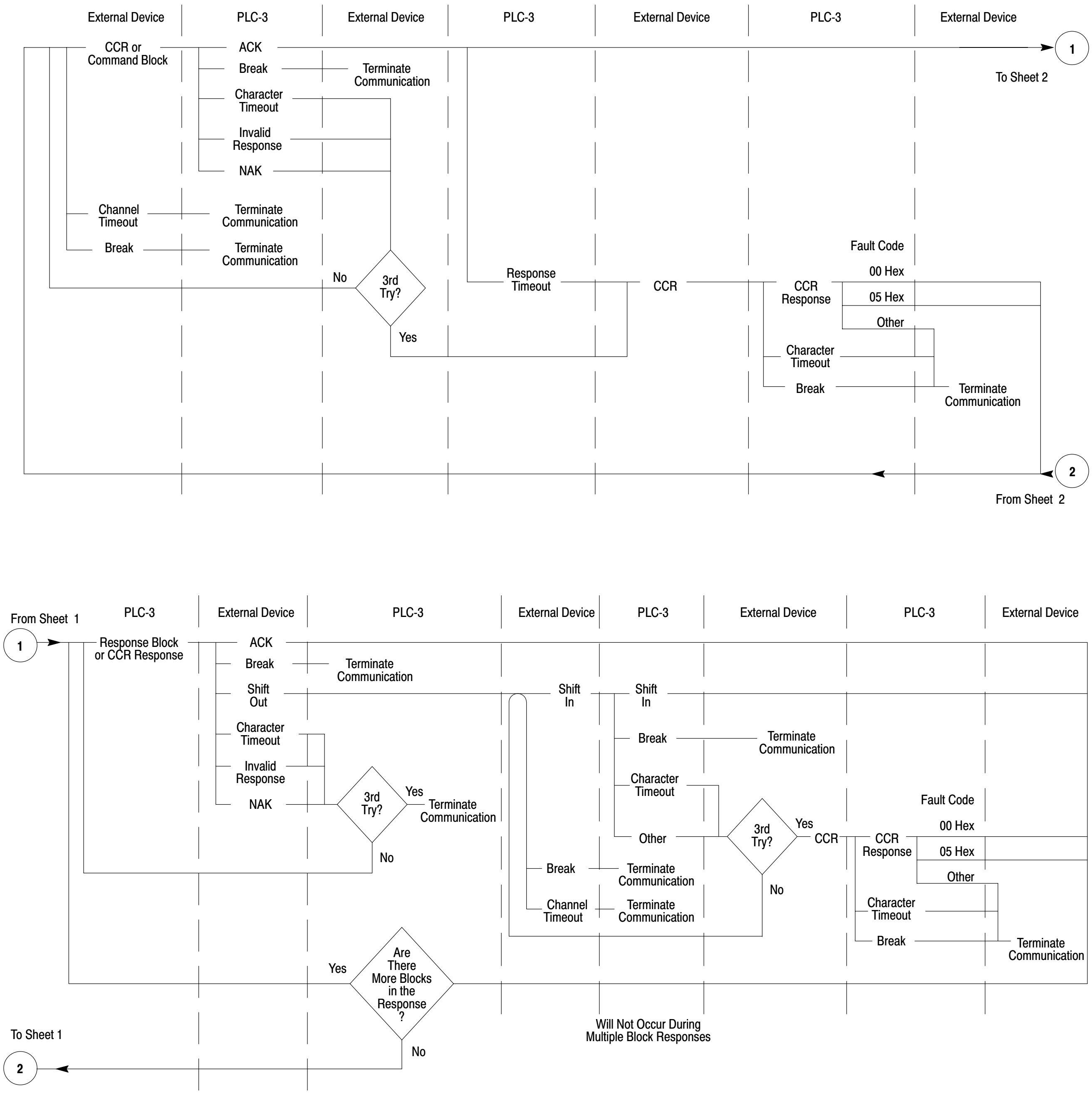
Protocol Summary

Figure A.1 summarizes the protocol used with BCL. To read this diagram:

Use the:	To see
left column	what the processor expects to receive from your device
second column	possible processor responses to your transmission
each remaining column	either • a processor response or • the action which the processor expects your device to take in response to its last transmission

Important: Both the processor and the external device must respond to a break at any time. Figure A.1 shows break conditions after each transmission, but be aware that either the processor or the external device stops transmitting (even in the middle of a block) when it sees a break.

Figure A.1
BCL Protocol Diagram



Extended Addressing

Most BCL commands and responses that require an address as one of the parameters use the extended address format. The exceptions are:

- read I/O word command which requires an I/O address
- read block physical and write block physical commands (used for uploading and downloading respectively) which use physical addresses

A detailed description of processor extended addressing appears in the PLC-3 Family Controller Programming Manual (publication 1775-6.4.1). Processor memory organization is summarized in Table A.E.

When you send an extended address to the processor, it must follow a mask byte. The mask byte tells the processor which address levels the following address defines and which levels default.

A bit in the mask which is set to:	Indicates that the:
1	corresponding level of memory is specified in the address to follow
0	default value is used for the corresponding level of memory

Bit 0 in the mask corresponds to the major area. Bits 1 to 7 in the mask each correspond to the next level in the specified section.

For example, in the data table (area 3), bit 1 corresponds to the context and bit 2 corresponds to the data table section (e.g., output, input).

You can extend the mask byte into a mask word to specify more than 7 levels of an address by sending a byte with the value FF hex followed by two bytes of the mask word. This is for future expansion of processor extended addressing, as only 6 levels are presently used (Table A.E).

The default value for:	Is :
the major memory area (bit 0)	3 (data table)
context (bit 1)	initially the context that the processor is operating when communication is established
all other memory levels	0

You can change the default context by using the set operating context command. Unless otherwise specified, examples in this chapter assume that the default context is 1. You can use the default for any level except the lowest level of the address.

Table A.E
Extended Addressing

Area	E#	.x	.x	.x	.x	.x
system status	E0	context = 0	section = 0	word = 0 - 20	not used	not used
module status	E2	module type = 1 – memory 2 – main processor 3 – scanner 5 – communication adapter 6 – expansion 7 – S4B I/O scanner 8 – peripheral communication 9 – DH II interface 14 – memory communication	thumbwheel switch = 1 - 15	module data	module data	module data
data table	E3	context = 1 - 15	section = 1 – output image 2 – input image 3 – timers 4 – counters 5 – integers 6 – floating point 7 – decimal 8 – binary 9 – ASCII 10 – high order int. 12 – pointers 13 – status	file = 0 - 999 0 - 999 0 0 0 - 999 0 - 999 0 - 999 0 - 999 0 - 999 0 - 999 0 0 - 999	structure = 0 0 0 - 9999 0 - 9999 0 0 0 0 0 0 0 - 9999 0	word = 0 - 07777 octal 0 - 07777 octal 0 - CTL, 1 - PRE, 2 - ACC 0 - CTL, 1 - PRE, 2 - ACC 0 - 9999 0 = lower 16 bits, 1 = upper 16 bits 0 - 9999 0 - 9999 0 - 9999 0 = lower 16 bits, 1 = upper 16 bits 0 = section/file, 1 = word 0 - 9999
ladder program	E4	context = 1 - 15	section = 0 - program status 1 - main 2 - subroutine 3 - fault routine	rung = 0 - 32, 767	instruction = 0 - 32, 767	word = 0 - 32, 767
message	E5	context = 1 - 15	section = 1 - report generation 2 - rung comments 3 - terminal (MACROS) 4 - DH 5 - assistance (HELP)	message = 0 - 32, 767	word = 0 - 32, 767	not used
system symbols	E6	context = 1 - 15	type = 1	symbol = 0 - 32, 767	word = 0 - 32, 767	not used
converted procedures	E8	context = 1 - 15	section = 1 - report generation	message = 0 - 32, 767	word = 0 - 32, 767	not used
force table	E10	context = 1 - 15	force type = 0 - status 1 - forced output 2 - forced input	rack = not used 0 - 64 0 - 64	word = 0 0 - 15 0 - 15	bit = 0 - input forces enabled/disabled 1 - output forces enabled/disabled not used not used

For example, two ways to send the address E3.1.2.0.0.0 are:

- send the mask $3F_{16}$ (00111111) followed by the bytes 03_{16}, 01_{16}, 02_{16}, 00_{16}, 00_{16}, 00_{16}
- send the mask 24_{16} (00100100) followed by the bytes 02_{16}, 00_{16}

You can extend address bytes into address words just as you extend mask bytes into mask words. For example, the mask 24_{16} followed by the bytes 02_{16}, FF_{16}, 00_{16}, 08_{16} corresponds to the address E3.1.2.0.0.2048.

Commands

The scanner supports commands for the following functions:

- reading, writing, and verifying data
- reading the size of memory sections
- adding, expanding, deleting, or reducing sections
- setting processor operating mode
- accessing the LIST function
- setting the default context for communication

The sections that follow describe each of these functions.

Reading Data

You can read data from the processor either in individual words using the read word command, or in blocks using the read block command. You can also read from the data table I/O section using the read I/O word command. This command obtains an I/O word and the associated force information from the processor. The byte increment/decrement command adds or subtracts one to a specified byte. We describe these four read-type commands in the following subsections.

Read Word Command

Hex: 12

Parameters: Address of required word

Description: The following example command block reads file 0, word 3 of the data table status section:

STX	ISC	SIZ	CMD	DAT			ETX	CHK
02	05	04	12	24	0D	03	03	54

The data bytes store:

- address mask (24 hex)
- address byte specifying status section 13 (0D hex)
- address byte specifying word 3 (03 hex)

The processor responds with one of the following codes:

Response Codes

Hex	Description
00	Success
04	Address unknown
06	Access not allowed
08	Address not complete

If a success code is returned, the next two bytes in the response block contain the data stored at the requested address, with the lower byte sent first:

STX	OSC	SIZ	RSP	DAT		ETX	CHK
02	01	03	00	05	08	03	16

The data bytes store the data word:

- lower byte (05 hex)
- upper byte (08 hex)

If any other code returns, no data follows.

Read Block Command

Hex: 1F

Parameters: Starting address, size (in words) of the block

Description: The read block command reads a block of information starting at a specific address. If you specify 0 for the size, the processor returns the data from the address specified to the end of the section if this does not include more than 63 words, or 63 words starting with the address specified if the section is larger. The following example command block reads the first two words of the input section:

STX	ISC	SIZ	CMD	DAT							ETX	CHK
02	01	08	1F	3C	02	00	00	00	02	00	03	6D

The data bytes store:

- address mask (3C hex)
- address byte specifying section 2 (input) (02 hex)
- address byte specifying file 0 (00 hex)
- address byte specifying structure 0 (00 hex)
- address byte specifying word 0 (00 hex)
- lower byte of size word (02 hex)
- upper byte of size word (00 hex)

Notice that the size is specified in a word (two bytes). This is true of all size specifiers within the data buffer of a command block.

The processor responds to a read block command with one of the following response codes:

Response Codes

Hex	Description
00	Success
01	Size too big
04	Address unknown
05	Address invalid – end of section
06	Access not allowed
08	Address not complete

If a **success, size too big, or address invalid – end of section** code returns, the next byte contains the number of data words returned. The data follows this byte.

A success code returns when the number of data words returned equals the number requested. For example:

STX	OSC	SIZ	RSP	DAT					ETX	CHK
02	01	06	00	02	04	03	0F	02	03	26

The data bytes store:

- size byte (02 hex)
- lower byte of the first data word (04 hex)
- upper byte of the first data word (03 hex)
- lower byte of the second data word (0F hex)
- upper byte of the second data word (02 hex)

A **size too big** code returns if the number of words requested requires a response block larger than 63 words. In this case, 63 words return.

An **address invalid – end of section** code returns when the number of words requested extends beyond the end of this section. The data from the specified address to the end of the section returns.

If the processor returns any other response code, no data follows.

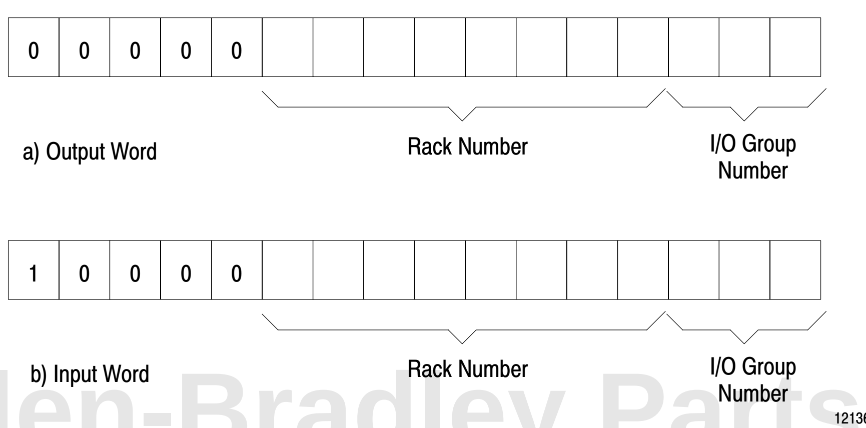
Read I/O Word

Hex: 23

Parameters: Address of required word

Description: You must send the I/O address using the following format (Figure A.2):

Figure A.2
I/O Address Format



The following example command block uses the read I/O word command to request the data and force information for the output word representing rack 6, module group 4:

STX	ISC	SIZ	CMD	DAT		ETX	CHK
02	01	03	23	34	00	03	60

The data bytes store:

- lower byte of the I/O address (34 hex)
- upper byte of the I/O address (00 hex)

The processor responds to a read I/O word command with one of the following response codes:

Response Codes

Hex	Description
00	Success
03	Forces unavailable
04	Address unknown
06	Access not allowed
08	Address not complete

When the processor returns a success response, the data buffer also contains the data stored at the requested address, and force-on and force-off masks associated with that data. The following example shows a response block for a read I/O word command when forces exist:

STX	OSC	SIZ	RSP	DAT						ETX	CHK
02	01	07	00	61	1C	02	86	80	00	03	92

The data bytes store:

- lower byte of the data word (61 hex)
- upper byte of the data word (1C hex)
- lower byte of the force-on mask (02 hex)
- upper byte of the force-on mask (86 hex)
- lower byte of the force-off mask (80 hex)
- upper byte of the force-off mask (00 hex)

If no force table has been created in processor memory, the processor responds to a read I/O word command with a forces unavailable response code. The data stored at the specified address follows this response code in the data buffer. The following example response block shows a read I/O word command when no forces exist:

STX	OSC	SIZ	RSP	DAT		ETX	CHK
02	01	03	03	61	1C	03	89

The data bytes store:

- lower byte of the data word (61 hex)
- upper byte of the data word (1C hex)

If any other response code is returned, no data follows.

Writing Data

You can write data into processor memory either in individual words with the write word command, or in blocks with the write block command. You can also modify words in processor memory with the read-modify-write command. The following two subsections explain the writing data commands.

Write Word Command

Hex: 14

Parameters: Address that needs to be updated and the new data

Description: The following example command block writes new data into address E3.1.5.0.0.0:

STX	ISC	SIZ	CMD	DAT							ETX	CHK
02	01	08	14	3C	05	00	00	00	08	91	03	FC

The data bytes store:

- address mask (3C hex)
- first address byte (05 hex)
- second address byte (00 hex)
- third address byte (00 hex)
- fourth address byte (00 hex)
- lower data byte (08 hex)
- upper data byte (91 hex)

The processor responds to a write word command with one of the following response codes:

Response Codes

Hex	Description
00	Success
04	Address unknown
06	Access not allowed
08	Address not complete
0F	No privilege

If any code other than success is returned, no data is written into memory. The following response block shows a successful write word command:

STX	OSC	SIZ	RSP	ETX	CHK
02	01	01	00	03	07

Write Block Command

Hex: 20

Parameters: Starting address, number of words, and the new data

Description: The following example command block writes new data into the first 3 words of integer file 0:

STX	ISC	SIZ	CMD	DAT													ETX	CHK
02	01	0E	20	3C	05	00	00	00	03	00	10	02	08	00	7D	8E	03	9D

The data bytes store:

- address mask (3C hex)
- first to fourth address bytes (05, 00, 00, 00 hex)
- lower and upper bytes of:
  - size word (03, 00 hex)
  - first data word (10, 02 hex)
  - second data word (08, 00 hex)
  - third data word (7D, 8E hex)

The processor responds to a write block command with one of the following response codes:

Response Codes

Hex	Description
00	Success
04	Address unknown
05	Size too big
06	Access not allowed
08	Address not complete
0F	No privilege

If the **size too big** response returns, the processor stores data which fits into the existing memory locations. The rest is lost. If any code besides **success** or **size too big** is returned, no data is written into memory. The following response block shows a successful write block command:

STX	OSC	SIZ	RSP	ETX	CHK
02	01	01	00	03	07

The size is specified in one byte. This is true of all size specifiers within the data buffer of a response block.

Read-Modify-Write Command

Hex: 16

Parameters: Address of the word to modify, one word AND mask, one word OR mask

Description: The following example command block executes the read-modify-write command:

STX	ISC	SIZ	CMD	DAT							ETX	CHK
02	0D	08	16	24	05	03	55	55	0F	0F	03	07

The data bytes store:

- address mask (24 hex)
- first and second address bytes (05, 03 hex)
- lower and upper bytes of AND mask (55 hex)
- lower and upper bytes of OR mask (0F hex)

When the processor receives a read-modify-write command,
it (Figure A.3):

1. Reads the data at the specified address.
2. Performs a logical AND operation between the data read in step 1 and the AND mask.
3. Performs a logical OR operation between the result of step 2 and the OR mask.
4. Writes the result of step 3 into the word at the specified address.

Figure A.3
Read-Modify-Write Operation

Bit Pattern	Description
1111000011110000	Initial state of specified word
0101010101010101	AND mask
0101000001010000	Result of logical-AND operation
1010000001011111	Final result

The processor responds to a read-modify-write command with one of the following response codes:

Response Codes

Hex	Description
00	Success
04	Address unknown
06	Access not allowed
08	Address not complete
0F	No privilege

The following response block shows a successful read-modify-write command:

STX	OSC	SIZ	RSP	ETX	CHK
02	01	01	00	03	07

Verify Block Command

Hex: 15

Parameters: Starting address, number of words, data to compare with memory

Description: The processor can compare a block of data in memory to reference data from an external device. The following command block verifies the first three words in the binary section of the data table (section eight):

STX	ISC	SIZ	CMD	DAT											ETX	CHK
02	14	0C	15	24	08	00	03	00	04	41	E1	02	E3	15	03	89

The data bytes store:

- address mask (24 hex)
- first address byte (08 hex)
- second address byte (00 hex)
- lower byte of size word (03 hex)
- upper byte of size word (00 hex)
- lower byte of the first data word (04 hex)
- upper byte of the first data word (41 hex)
- lower byte of the second data word (E1 hex)
- upper byte of the second data word (02 hex)
- lower byte of the third data word (E3 hex)
- upper byte of the third data word (15 hex)

The processor responds to a verify block command with one of the following response codes:

Response Codes

Hex	Description
00	Success
04	Address unknown
05	Address invalid – end of section
06	Access not allowed
08	Address not complete

If the success response code returns, the next byte contains the number of words in the reference block which do not match the block in memory. The following response block shows a successful verify block command:

STX	OSC	SIZ	RSP	DAT	ETX	CHK
02	01	01	00	00	03	07

The data byte stores the miscomparison count. If any other response code returns, no data follows.

Uploading/Downloading

You can read from and write to physical locations in processor memory using the read block physical and write block physical commands. These commands are executed faster than read block and write block commands. However, physical addresses change when you edit the program, which makes it difficult to keep track of where a given word or block of data is stored. Therefore, use the read block physical and write block physical commands only for uploading and downloading the entire memory contents to/from a computer.

The read block physical and write block physical commands use physical addresses which are different than the addressing used for other BCL commands. The address requires two words (four bytes) with the least significant byte sent first. The most significant byte must contain the value 01 hex. The value contained in the other bytes ranges from 0 to the highest address in the processor programmable processor (depending on the amount of memory in the system). We explain these two commands in the following 4 subsections.

Read Block Physical

Hex: 07

Parameters: Physical address of the first word in the block, number of words in the block

Description: The following command block uses a read block physical command to read 16 words starting with word 66120 (10248 hex):

STX	ISC	SIZ	CMD	DAT						ETX	CHK
02	05	07	07	48	02	01	01	10	00	03	70

The data bytes store:

- least significant byte of the address (48 hex)
- second byte of the address (02 hex)
- third byte of the address (01 hex)
- most significant byte of the address (01 hex)
- least significant byte of the size word (10 hex)
- most significant byte of the size word (00 hex)

The processor response to a read block physical command is the same as the response to a read block command (see "Read Block Command, page A-18).

When uploading to a computer, you need to know how much memory is actually used. You can find the address of the last word used in memory with the following algorithm:

1. Read words 35 and 36.

2. Use the least significant byte of word 35 plus word 36 as the three least significant bytes in a physical address.

3. Subtract 9 from the physical address in step 2.

4. Read 2 words starting from the result of step 3.

5. The least significant byte of the first word plus the second word read in step 4 contain the three least significant bytes of the physical address of the last word used in memory (Figure A.4):

Figure A.4
Finding the Address of the Last Used Memory Word

Result of step	Word 35	Word 36	Description
1	1101100000000000	0000000001101001	---------
2	0000000100000000	0000000001101001	Points to word 105
3	0000000100000000	0000000001100000	Points to word 96
4	1000001100000001	0000000001000000	Contents of words 96 and 97
5	0000000100000001	0000000001000000	Points to word 65600 (the last unused memory word)

Although uploading does not require a complete shutdown of the processor, memory must not be created or deleted during the upload operation. Therefore, be sure that no scanner or other module is creating or deleting memory, and making changes in LIST during an upload operation.

Example Upload

The following sequence of command and response blocks performs an upload. The upload includes 4DD (hex) words, and is performed in 10 (hex) word blocks. All values are shown in hexadecimal.

	STX	ISC/OSC	SIZ	CMD/RSP	DAT						ETX	CHK
Device:	STX 02	ISC	SIZ 07	CMD 07	00	00	00	01	10	00	ETX 03	CHK
PLC-3:	STX 02	OSC	SIZ 22	RSP 00	10	...data...					ETX 03	CHK
Device:	STX 02	ISC	SIZ 07	CMD 07	10	00	00	01	10	00	ETX 03	CHK
PLC-3:	STX 02	OSC	SIZ 22	RSP 00	10	...data...					ETX 03	CHK
Device:	STX 02	ISC	SIZ 07	CMD 07	20	00	00	01	10	00	ETX 03	CHK
PLC-3:	STX 02	OSC	SIZ 22	RSP 00	10	...data...					ETX 03	CHK
• • •												
Device:	STX 02	ISC	SIZ 07	CMD 07	C0	04	00	01	10	00	ETX 03	CHK
PLC-3:	STX 02	OSC	SIZ 22	RSP 00	0D 48	...data...					ETX 03	CHK
Device:	STX 02	ISC	SIZ 07	CMD 07	D0	04	00	01	0D	00	ETX 03	CHK
PLC-3:	STX 02	OSC	SIZ 24	RSP 00	10	...data...					ETX 03	CHK

Write Block Physical

Hex: 08

Parameters: Physical address of the first location in processor memory to receive data, number of words in the block of data, data to write into processor memory

Description: The processor only recognizes the write block physical command when in the shutdown mode. To put the processor into the shutdown mode, use the shutdown command (06 hex). The following example command block uses the shutdown command:

STX	ISC	SIZ	CMD	ETX	CHK
02	04	01	06	03	10

The response to a shutdown command is a success response code (00 hex). Any other response indicates that the processor did not correctly receive the transmission.

The following example command block uses the write block physical command:

STX	ISC	SIZ	CMD	DAT										ETX	CHK
02	05	0B	08	01	20	00	01	02	00	19	05	20	01	03	80

The data bytes store:

- least significant byte of the address (01 hex)
- second byte of the address (20 hex)
- third byte of the address (00 hex)
- most significant byte of the address (01 hex)
- least significant byte of the address (02 hex)
- most significant byte of the address (00 hex)
- least significant byte of first data word (20 hex)
- most significant byte of second data word (01 hex)

The processor response to a write block physical command is the same as the response to a write block command (see the section entitled write block command).

After the download operation is complete, send a reset command (09 hex) to exit the shutdown mode. The following command block uses the reset command:

STX	ISC	SIZ	CMD	ETX	CHK
02	06	01	09	03	15

The processor does not issue any response to a reset command, so just continue communication by sending the next command. However, the reset command resets the sequence counts.

Example Download

The following sequence of command and response blocks performs a download. The download includes 4DD (hex) words, and is performed in 10 (hex) word blocks. All values are shown in hexadecimal.

Device:

STX	ISC	SIZ	CMD	ETX	CHK
02		01	06	03	

PLC-3:

STX	OSC	SIZ	RSP	ETX	CHK
02		01	00	03	

Device:

STX	ISC	SIZ	CMD	DAT		ETX	CHK
02		27	08	01	...data...	03	

PLC-3:

STX	OSC	SIZ	RSP	ETX	CHK
02		01	00	03	

Device:

STX	ISC	SIZ	CMD	DAT		ETX	CHK
02		27	08	10	...data...	03	

PLC-3:

STX	OSC	SIZ	RSP	ETX	CHK
02		01	00	03	

Device:

STX	ISC	SIZ	CMD	DAT		ETX	CHK
02		27	08	20	...data...	03	

PLC-3:

STX	OSC	SIZ	RSP	ETX	CHK
02		01	00	03	

•
•
•

Device:

STX	ISC	SIZ	CMD	DAT		ETX	CHK
02		27	08	CO	...data...	03	

PLC-3:

STX	OSC	SIZ	RSP	ETX	CHK
02		01	00	03	

Device:

STX	ISC	SIZ	CMD	DAT		ETX	CHK
02		27	08	D0	...data...	03	

PLC-3:

STX	OSC	SIZ	RSP	ETX	CHK
02		01	00	03	

Device:

STX	ISC	SIZ	CMD	ETX	CHK
02		01	09	03	

Read Section Size Command

Hex: 19

Parameters: Address of section to read

Description: The following example command block requests the size of the data table in context 1:

STX	ISC	SIZ	CMD	DAT			ETX	CHK
02	08	04	19	03	03	01	03	31

The data bytes store:

- address mask (03 hex)
- first and second address bytes 903, 01 hex)

The processor responds to a read section size command with one of the following response codes:

Response Codes

Hex	Description
00	Success
04	Address unknown
06	Access not allowed

If the processor returns a success response code, the next four bytes contain the section size, with the least significant byte transmitted first. The two bytes following the section size contain the number of next levels of addressing.

For example, in context 1 of the data table, the number of next levels corresponds to the number of data table sections such as output, input, or integer. And in the integer section, the number of next levels corresponds to the number of integer files.

The following example response block shows successful read section size command:

STX	OSC	SIZ	RSP	DAT						ETX	CHK
02	14	07	00	08	28	00	00	19	00	03	69

The data bytes store the data word:

- first size byte (08 hex)
- second size byte (28 hex)
- third size byte (00 hex)
- fourth size byte (00 hex)
- first byte of next level's word (19 hex)
- second byte of next level's word (00 hex)

If any other response code is returned, no data follows.

Create Command

Hex: 17

Parameters: Address of the last word, structure, file or section.

Description: The following example command block creates 10 words in integer file 5 (if integer file 6 does not already exist):

STX	ISC	SIZ	CMD	DAT					ETX	CHK
02	0C	06	17	3C	05	05	00	0A	03	7E

The data bytes store:

- address mask (3C hex)
- first address byte (section) (05 hex)
- second address byte (file) (05 hex)
- third address byte (structure) (00 hex)
- fourth address byte (word) (0A hex)

If the words have already been created, the processor sends an available response code (02 hex). The command also tells the processor to create the pointers for integer files 0 thru 4 if they did not already exist, although no words are allocated to these files.

The processor responds to a create command with one of the following response codes:

Response Codes

Hex	Description
00	Success
01	Size too big
02	Available – address specified already exists
04	Address could not exist – invalid values
08	Address insufficient – exists, but no specific to word level
0F	No privilege

The following example response block shows a successful create command:

STX	OSC	SIZ	RSP	ETX	CHK
02	11	01	00	03	17

Delete Command

Hex: 18

Parameters: Address of the first word, structure, file, or section to delete

Description: The delete command deletes or reduces sections in PLC-3 memory. If you specify a word, structure, or file, the processor deletes all higher addresses at the same level.

For example, if integer file 5 contains words 0 thru 100, deleting word 10 also deletes word 11 thru 100. On the other hand, deleting the integer section does not delete any other section.

The following example command block deletes from integer file 5 to the end of the integer section:

STX	ISC	SIZ	CMD	DAT			ETX	CHK
02	0D	04	18	0C	05	05	03	44

The data bytes store:

- address mask (0C hex)
- first address byte 905 hex)
- second address byte (05 hex)

The processor responds to a delete command with one of the following response codes:

Response Codes

Hex	Description
00	Success
03	Unavailable – section does not exist
04	Address could not exist – invalid values
06	Access not allowed
08	Insufficient address – not specified to at least three levels
0F	No privilege

The following example response block shows a successful delete command:

STX	OSC	SIZ	RSP	ETX	CHK
02	11	01	00	03	17

LIST Command

Hex: 29

Parameters: none required

Description: The LIST command accesses the LIST function. This command allows you to send ASCII characters to the processor. These characters are interpreted in the same manner as if they were entered from the processor front panel or the keyboard of a programming device (see chapter 3).

The processor remains in the LIST processing mode until it receives either a command block which does not contain a LIST command or a break character. The following example command block uses the LIST command (with no data in the command buffer):

STX	ISC	SIZ	CMD	ETX	CHK
02	31	01	29	03	60

The processor responds to the LIST command with one of the following response codes:

Response Codes

Hex	Description
00	Success
01	Size too big

Both responses are followed by ASCII data in the data buffer. The size too big response indicates that the response contains more than 64 words. In this case, the processor continues to send response block until the entire response has been transmitted. All response blocks except one containing the last block of data use the size too big response. The last block uses the success response.

The following example response block shows a successful LIST command:

STX	OSC	SIZ	RSP	DAT	ETX	CHK
02	32	10	00	...ASCII data...	03	

The first LIST command which you send should not have any data in the command buffer. The processor response to this first command contains the initial data which appears on the terminal CRT after you enter the LIST command. Other LIST commands generally contain data in the command block.

Set CPU Mode Command

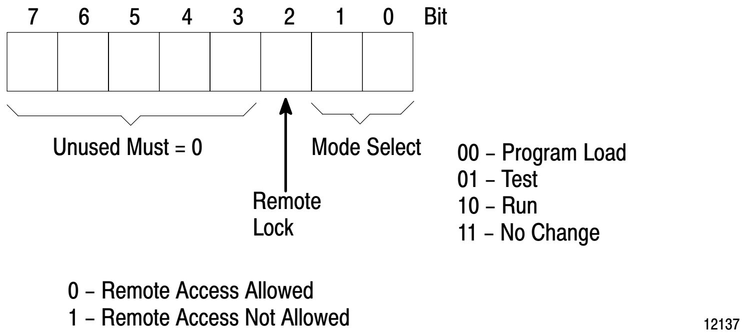
Hex: 2A

Parameters: Flag byte

Description: You can change the processor operating mode with the set CPU mode command. The flag byte allows you to change the operating mode to program load, test, or run (Figure A.5). It also allows you to keep other remote devices from communicating with the processor by setting the remote lock bit.

Important: The remote lock bit is not the same as remote enable (which is set in LIST).

Figure A.5
CPU Mode Flag Byte



The following example command block resets the remote lock bit (allowing remote access to other devices) and selects run mode:

The data byte stores the flag byte (02 hex).

The processor responds to a set CPU mode command with one of the following response codes:

STX	ISC	SIZ	CMD	DAT	ETX	CHK
02	03	02	2A	02	03	36

Response Codes

Hex	Description
00	Success
06	Access not allowed
0F	No privilege

The following example response block shows a successful set CPU mode command:

STX	OSC	SIZ	RSP	ETX	CHK
02	11	01	00	03	17

Set Operating Context Command

Hex: 2B

Parameters: New default context (any context between 1 and 15)

Description: You can change the default communication context by using the set operating context command. If you send a value greater than 15, the processor uses the lower 4 bits only. The processor does not accept a default context of 0. The following example command block changes the default context values to 2:

STX	ISC	SIZ	CMD	DAT	ETX	CHK
02	03	02	2B	02	03	37

The data byte stores the new default context.

The processor responds to a set operating context command with one of the following response codes:

Response Codes

Hex	Description
00	Success
03	Unavailable – context = 0
06	Access not allowed
0F	No privilege

The following example response block shows a successful set operating context command:

STX	OSC	SIZ	RSP	ETX	CHK
02	11	01	00	03	17

Appendix **B**

DH/DH+ Error Codes

Introduction

This appendix describes the error codes that the scanner uses to report DH/DH+ communication errors. Errors are of three types:

- local
- reply
- remote

Local Error Codes

The scanner generates local errors while trying to execute one of its own DH/DH+ message procedures. The message instruction sets the erro bit and records the error in the message control word (see “Ladder Program,” page 4-6). The module also stores the local error codes in the user symbol ERROR (see “DH/DH+ Message Procedure Diagnostics,” page 7-1). Possible local errors and their meanings are listed in Table B.A.

Reply Error Codes

The scanner generates reply errors while trying to respond to a command message received from a remote DH or DH+ station. The scanner inserts the reply error code in the STS or EXT STS bytes of any reply message packet it transmits to a remote station. For reply errors, there is a direct correlation between the error codes in the STS and EXT STS bytes of reply messages and the error codes reported at the remote station:

If the scanner generates these codes:		Then the command station stores the error code in decimal
STS byte (hex)	EXT STS byte (in hex)	
00	not used	no error
10	not used	81
30	not used	83
40	not used	84
50	not used	85
60	not used	86
70	not used	87
F0	1	231
F0	2	232
F0	3	233
F0	4	234
F0	5	235
F0	6	236
F0	7	237
F0	8	238
F0	9	239
F0	A	240
F0	B	241

A value of F0 (hex) in the STS byte indicates that the EXT STS byte actually contains the error code for the reply message.

The meaning of each error code depends on the command message that the local station receives from a remote station. Table B.A describes the error conditions that the various commands can generate. The error codes are listed according to the decimal value that the command at the initiating station stores.

When a remote station transmits a command, the local scanner might issue a reply message that contains one of the error codes listed in Table B.A. Error codes 81–87 appear in the STS byte of the reply message, and codes 231–241 appear in the EXT STS byte.

Remote Error Codes

The scanner receives remote error codes in a reply to a command it has sent to a remote station. The message instruction sets the error bit and records the error in the message control word (see “Ladder Program,” page 4-6). The module also stores the local error code in the user symbol ERROR (see “DH/DH+ Message Procedure Diagnostics,” page 7-1).

The meaning of a particular remote error code varies depending on the type of communication interface module at the remote station. If the remote station is a PLC-3 controller with a scanner as the interface module, the remote error codes have the meanings listed in Table B.A. For the meanings of other remote error codes, refer to the appropriate user’s manual for the communication interface module at the remote station.

Table B.A
Local and Reply Error Codes

Error Code	Error Type	Associated Commands	Meaning
32	local	all	The size of the local file involved in a file assignment command is greater than 65,535 bytes.
34	local	all	A station number greater than 376 (octal) is specified for the remote address in an assignment command.
35	local	all	Attempt to send unprotected command is invalid.
37	local	all	The per-packet timeout, set through LIST, runs out before receiving a reply. This means that the remote station acknowledges (ACK) the command message but does not send the reply in the allotted time (cf. error 92).
81	reply	diagnostic read	• A 2-byte ADDR and 1-byte SIZE field is missing after the FNC byte in the command message. • The number of bytes of data requested in the SIZE field is greater than the maximum number allowed per reply packet (244), or SIZE is 0 (zero).
	reply	PLC/PLC-2 read	• The required 2-byte ADDR and 1-byte SIZE field are missing in the command message. • The ADDR value is odd (it does not specify a word address). • The value SIZE is 0. • The value SIZE is greater than 244. • The SIZE value specifies an odd number of bytes.
	reply	PLC/PLC-2 bit write	Incomplete bit description because the number of bytes after the TNSW is not a multiple of four.
	reply	PLC/PLC-2 word write	• A 2-byte ADDR field is expected after the TNSW word, but only one byte is present. • An odd number of data bytes in the command packet. • The ADDR value is odd (it does not specify a word address).
	reply	PLC-3 read	• More than one byte of data is after the byte address. • Number of bytes to read is odd. • Number of bytes to read is zero. • Number of bytes to read is greater than the maximum allowed in a reply packet (244). • Sum of packet offset and size of data in words is greater than 65,535. • Sum of packet offset and size of data in words is greater than the total transaction size.
	reply	PLC-3 bit write	More than four bytes of data exist after the address in the command message.
	reply	PLC-3 write	• Less than two bytes of data are after the end of the block address. • An odd number of data bytes is after the end of the block address. • Sum of packet offset and size values specifies more than 65,535. • Sum of packet offset and size is greater than total transaction size.
83	reply	PLC/PLC-2/PLC-3 read and write	The local scanner has executed a shutdown request to the local PLC-3 controller.

Error Code	Error Type	Associated Commands	Meaning
84	reply	diagnostic status	Backplane error occurred during determination of the physical address of the end of the ladder program or of the end of user memory. In polled mode, the RS-232-C port has received a NAK, which causes a system reset.
	reply	PLC/PLC-2 read, write	Local PLC-3 backplane error (either memory parity or timeout/disconnect). In polled mode, the RS-232-C port has received a NAK, which causes a system shutdown.
	reply	PLC-3 read and write	Backplane error (memory parity or timeout/disconnect). In polled mode, the RS-232-C port has received a NAK, which causes a system reset.
85	reply	diagnostic read	The command is an illegal request to read from the scanner's backplane window.
	reply	PLC/PLC-2 read	• PLC-3 file does not exist. • PLC-3 file is too small. • PLC-3 file is more than 65,535 words long.
	reply	PLC/PLC-2 bit write	• PLC-3 file does not exist. • Destination bits do not exist in PLC-3 file. • Length of PLC-3 file is greater than 65,535 words.
	reply	PLC/PLC-2 word write	• Destination file does not exist in PLC-3 memory. • Destination word does not exist in the destination PLC-3 file. • Length of the destination file is greater than 65,535 words.
86	reply	PLC/PLC-2 bit write	Keyswitch setting at local PLC-3 controller prohibits access.
	reply	PLC/PLC-2 word write	Local keyswitch settings prohibit writing into desired destination file.
	reply	PLC-3 write	Keyswitch setting disallows access to file.
87	reply	PLC/PLC-2/PLC-3 read and write	The local PLC-3 controller is in program mode. There may or may not be a major system fault.
91	local	all	• Handshaking lines on the RS-232-C link are not connected properly. • Multiple wacks on link.
92	local	all	The remote station specified does not acknowledge (ACK) the message.
94	local	all	Remote station is disabled.
112	local	all	Undefined operator in an assignment statement or expression.
114	local	all	Illegal expression syntax.
115	local	all	Illegal unary (prefix) operator in an expression.
117	local	all	Undefined data following a valid address in a CREATE command, or undefined data following a valid symbol in a delete command.
121	local	all	Symbol undefined because it appears as the source in an assignment command before it is defined. For example, a statement of the form A = A + 6 generates this error if user symbol A has not appeared previously.
123	local	all	System symbol must be a symbolic address. This error occurs if a procedure name is used in place of a symbolic address in an assignment statement or if the system symbol referenced in an assignment does not exist.

Error Code	Error Type	Associated Commands	Meaning
124	local	all	Illegal destination in an assignment command or invalid data following a number on a command line. For example, the lines 5 = 4 + 1 or 6ASDFGHJ generate this error code. The line $rbWERTYUI generates error code 140 (unrecognized command).
125	local	all	Illegal modifier for the CREATE command. Accepted modifiers for the create command are LOCAL and GLOBAL.
126	local	all	The CREATE command was specified, but the symbol did not begin with an @.
127	local	all	$ missing in a CREATE system symbol address command line.
129	local	all	Attempt to delete nonexistent symbol.
140	local	all	• Unrecognized or ambiguous command. • Channel 4 not configured for DH/DH+.
142	local	all	Illegal data following GOTO command.
143	local	all	Illegal use of label (e.g., not in a procedure).
144	local	all	Label not found.
145	local	all	Duplicate label. User symbols must be distinct from labels.
146	local	all	Too many nested procedures.
147	local	all	Insufficient privilege for the specified operation. This error can occur when an attempt is made, via the assignment command, to write into a major section of memory in which the scanner does not have access privileges (namely, major section 0, 1, or 2).
148	local	all	Unbalanced parenthesis in expression.
149	local	all	A procedure name is used in a field that requires a symbolic address or a user symbol variable.
150	local	all	A label is used in a field that requires a symbolic address or a user symbol variable.
154	local	all	Error in reading address for symbol entry.
156	local	all	Illegal symbol in expression.
159	local	all	Bad level specified in extended address: • More than nine levels specified in an extended address. • Something other than a (or a number followed a in an extended address.
160	local	all	Unrecognized section specifier. An illegal character followed the $ in an address.
161	local	all	Bad timer or counter specification: • The first letter of the data table address is a T, C, or P, but four characters are not in specification. Addresses that are incorrect causing this error include $TAC:15, $C5:3, $TACCUM:23, etc. • The key data table word specifier is four characters long and began with a T, C, or P, but it does match the legal word specifiers (e.g., $TACM:3). • No colon following a legal word specifier.

Error Code	Error Type	Associated Commands	Meaning
163	local	all	Missing colon between file and word.
164	local	all	Illegal word specifier in a data table address.
165	local	all	Illegal context specifier. When an expression determines the context in a data table address, or when a data table address specifies the global context (context 0) and a colon follows the context.
166	local	all	Attempt to execute a symbol not defined as a process. The system symbol exists but refers to a symbolic address rather than to a process. Not a DH/DH+ procedure.
169	local	all	Either the number or the expression following the \ in an address has a value outside the range 0 to 15 (decimal).
171	local	all	Value specified in a bit assignment statement was other than a zero or a one.
177	local	all	Illegal use of EXIT command.
178	local	all	Illegal use of STOP command.
179	local	all	STOP command encountered in procedure.
188	local	all	Attempt to read/write at bad address.
189	local	all	Unable to evaluate the expression in the given base. This occurs if the argument of a FROMBCD function is not a valid BCD bit pattern, or if invalid characters occur in numeric values (e.g., "57 + 12X").
192	local	all	Function being used is not defined.
194	local	all	Expression is too complex.
199	local	all	Attempt to divide by zero.
200	local	all	Bad port specifier. The character following the # is other than H, h, M, m, D, d, O, or o.
201	local	all	User symbol used as part of remote address specification.
202	local	all	Undefined data following assignment command. This error occurs if the modifier UNRPOT is entered instead of UNPROT.
203	local	all	Error in remote specification: • A character other than @ or $ following the station number specification (.. . . . . = #H045*T . . .). • Something other than EOL, PROT, or UNPROT following a remote source address (... = #H012$S5:8 + 9).
204	local	all	Third-party transfer. In an assignment command, both the source and the destination are remote addresses.
205	local	all	Error in evaluating a PLC-2 address, or PLC-2 address greater than 65,535.
206	local	all	Zero range specified in an assignment command.
207	local	all	Word range specified in destination address.
208	local	all	Destination and source addresses disagree in type.

Error Code	Error Type	Associated Commands	Meaning
209	local	all	Not of DH message type.
210	local	all	Use of a non-PLC-3 type address in a local address operand.
211	local	all	In an assignment command, one of the local files does not exist, or the word specified is beyond the end of the file.
212	local	create	During CREATE command execution, the scanner was not able to create any symbols.
213	local	all	A local file exists, but the action specified refers to addresses beyond the end of the file. Possible causes include: • In a word assignment statement, the offset is greater than the file size. • In a word range assignment statement, the sum of the base address and the offset is greater than the total file size. • In a file assignment statement, the destination file is smaller than the source file. If the source file is remote, a single packet is fetched from the remote station's file.
214	local	all	Local source and destination files differ in size.
215	local	all	The value resulting from operations specified on the left side of an assignment statement does not fit into the destination specified on the right side: • The source is in the H section and the destination is in the N section, but the number is too large (i.e., outside the range −32768 to +32767). • A word is transferred from a binary section (I, O, or B section) to the N or C section and the high-order bit is a 1. • The destination is in the decimal section, but the number is not a valid BCD bit pattern.
217	local	all	More than eight levels specified in file address.
218	local	all	• File size changed between packets of a multi-packet transaction. • Destination not created.
230	local	all	Reply packet too small.
231	reply	PLC-3 read and write	Error in converting the block address (major section 63, context 15, or section 15).
232	reply	PLC-3 read and write	Three or fewer addressing levels specified for a PLC-3 word address.
233	reply	PLC-3 read and write	Conversion of a file address to a block address resulted in more than nine addressing levels.
234	reply	PLC-3 read and write	Symbolic address not found.
235	reply	PLC-3 read and write	Symbolic address is of length zero or is longer than eight bytes.

Error Code	Error Type	Associated Commands	Meaning
236	reply	PLC-3 read	• File not found. • Destination address does not have enough levels to specify a PLC-3 word (for word-range reads) or a file (for file reads). • The PLC-3 address specifies more levels than required. • Word specified by the PLC-3 address does not exist.
	reply	PLC-3 bit write	• File not found. • Destination address does not specify a PLC-3 word. • The PLC-3 address specifies more levels than required. • Word specified by the PLC-3 address does not exist.
	reply	PLC-3 write	• Destination file not found. • Destination address does not point to a word (for word-range writes) or a file (for file writes). • Destination address specifies more levels than required. • First word of destination location does not exist.
237	reply	PLC-3 read and write	• Any word in the total transaction does not exist in the destination file. • The source and destination files are not the same size.
238	reply	PLC-3 read and write	The file size decreased between packets of a multi-packet transaction and became too small for the total transaction.
239	reply	PLC-3 read and write	File is larger than 65,535 words.
240	reply	PLC-3 read and write	Sum of total transaction size and the word level of PLC-3 addressing is greater than 65,535.
241	reply	PLC-3 write	Remote station does not have access to the destination file.
243	reply	all	• No privilege for upload/download • Words already created or already deleted.
245	reply	restart	No previous shut down command received.
255	local	all	Module being sent message does not exist. Should also get BAD ADDRESS major system fault.

Appendix C

DH and DH+ Command Set

Introduction

Table C.A lists the DH and DH+ commands supported by the scanner. For detailed information on using these commands, refer to the DH/DH+ Command Set User's Manual (publication 1770-6.5.16).

The scanner can transmit 12 different commands on the link. The addressing method determines which command is to be sent.

Table C.A
Basic Commands

Addressing Method	Command	CMD (Hex)	FNC (Hex)
PLC-2 logical data	Protected write-block	00	N/A
	Unprotected read-block	01	N/A
	Protected write-bit	02	N/A
	Unprotected write-bit	05	N/A
	Unprotected write-block	08	N/A
PLC-3 logical binary or Logical ASCII word range	Write block	0F	00
	Read block	0F	01
	Write bit	0F	02
	Write file	0F	03
	Read file	0F	04
Logical ASCII data type	Write block	0F	67
	Read block	0F	68

Appendix **D**

Specifications

Specification	
Function	• I/O communication • DH communication • DH+ communication • backup communication • peer-to-peer communication • front-panel support
Location	• 1775-S5 – single slot in a PLC-3 chassis • 1775-SR5 – single slot in a PLC-3/10 chassis
I/O Capacity per Scanner	• 2,048 I/O (any mix) • 4,096 I/O (complementary)
Communication Rates (in kbps)	• 57.6 at 10,000 cable ft. (DH, DH+, I/O) • 115.2 at 5,000 cable ft. (I/O only) • 230.4 at 2,000 cable ft. (I/O only)
Cabling	• Twinaxial Cable (cat. no. 1770-CD) for I/O, DH or DH+ communication channels • Industrial Terminal T50 Cable (cat. no. 1784-CP5) for DH+ communication to T50 industrial terminal
Nominal I/O Scan Times Per I/O Adapter at 115.2Kbaud	• 4.5ms for 1 channel • 5ms for 2 channels • 5ms for 3 channels • 6.5ms for 4 channels • 8ms for 3 channels + DH • 9ms for 3 channels + DH+
Backplane Current	• 6.0A maximum from +5V • 20mA maximum from ±15V
Environmental Conditions	• Operating Temperature: 0 to 60° C (32 to 140°F) • Storage Temperature: -40 to 85° C (-40 to 185° F) • Relative Humidity: 5 to 95% (without condensation)

Numbers

A

B

C

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

As a subsidiary of Rockwell International, one of the world's largest technology companies — Allen-Bradley meets today's challenges of industrial automation with over 85 years of practical plant-floor experience. More than 11,000 employees throughout the world design, manufacture and apply a wide range of control and automation products and supporting services to help our customers continuously improve quality, productivity and time to market. These products and services not only control individual machines but integrate the manufacturing process, while providing access to vital plant floor data that can be used to support decision-making throughout the enterprise.

With offices in major cities worldwide

WORLD HEADQUARTERS
Allen-Bradley
1201 South Second Street
Milwaukee, WI 53204 USA
Tel: (1) 414 382-2000
Telex: 43 11 016
FAX: (1) 414 382-4444

EUROPE/MIDDLE EAST/AFRICA HEADQUARTERS
Allen-Bradley Europe B.V.
Amsterdamseweg 15
1422 AC Uithoorn
The Netherlands
Tel: (31) 2975/43500
Telex: (844) 18042
FAX: (31) 2975/60222

ASIA/PACIFIC HEADQUARTERS
Allen-Bradley (Hong Kong) Limited
Room 1006, Block B, Sea View Estate
28 Watson Road
Hong Kong
Tel: (852) 887-4788
Telex: (780) 64347
FAX: (852) 510-9436

CANADA HEADQUARTERS
Allen-Bradley Canada Limited
135 Dundas Street
Cambridge, Ontario N1R 5X1
Canada
Tel: (1) 519 623-1810
FAX: (1) 519 623-8930

LATIN AMERICA HEADQUARTERS
Allen-Bradley
1201 South Second Street
Milwaukee, WI 53204 USA
Tel: (1) 414 382-2000
Telex: 43 11 016
FAX: (1) 414 382-2400

Publication 1775-6.5.5 — November 1992
Supersedes 1775-6.5.5 — November 1987
PN 955113-06