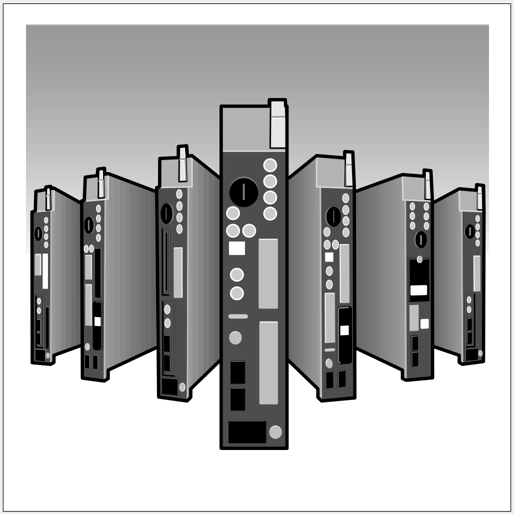
Allen-Bradley

Enhanced and Ethernet PLC-5 Programmable Controllers

(Cat. Nos. 1785-L11B, -L20B, -L30B, -L40B, -L40L, -L60B, -L60L, -L80B, -L20E, -L40E, -L80E, -L26B, -L46B, -L86B)

User Manual



Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. "Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls" (Publication SGI-1.1) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will the Allen-Bradley Company be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, the Allen-Bradley Company cannot assume responsibility or liability for actual use based on the examples and diagrams.

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

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

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

Attentions help you:

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

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

Summary of Changes

Introduction

This release contains new and updated information.

To help you find new and updated information, look for the change bars as shown on this paragraph.

Updated Information

For this new/updated information:	See chapter:
2000 elements per data table file	4
recommendations for using 230.4K bit/s	6, 10
performing block-transfers on the extended-local I/O channel	8
enhancements when using the serial port in master mode	11
communicating with ControlLogix devices over Ethernet	12
extended force tables	14
EEPROM information	20

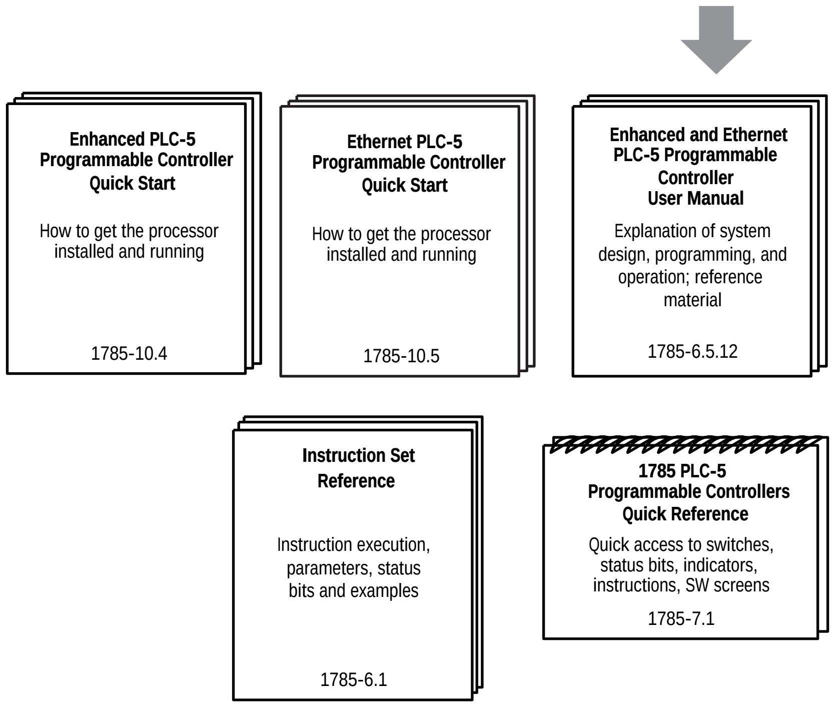
Notes:

Using This Manual

How to Use Your Documentation

Your PLC-5® programmable controller documentation is organized according to the tasks you perform. This organization lets you find the information that you want without reading through information that is not related to your current task. The arrow in Figure P.1 points to the book that you are currently using.

Figure P.1
Enhanced and Ethernet® PLC-5 Programmable Controller Documentation


Enhanced PLC-5 Programmable Controller Quick Start
How to get the processor installed and running
1785-10.4
Ethernet PLC-5 Programmable Controller Quick Start
How to get the processor installed and running
1785-10.5
Enhanced and Ethernet PLC-5 Programmable Controller User Manual
Explanation of system design, programming, and operation; reference material
1785-6.5.12
Instruction Set Reference
Instruction execution, parameters, status bits and examples
1785-6.1
1785 PLC-5 Programmable Controllers Quick Reference
Quick access to switches, status bits, indicators, instructions, SW screens
1785-7.1

MORE

For more information about PLC-5 programmable controllers or the above publications or other related publications, contact your local sales office, distributor, or system integrator.

For Ethernet, ControlNet, and DeviceNet information, see this web site:

- http://www.ab.com/networks

For additional Ethernet information, see these web sites:

- http://standards.ieee.org/catalog/sol/lan_man.html
- http://www.ietf.cnri.reston.va.us/

For additional information on TCP/IP protocol and networking in general, see these publications:

- Comer, Douglas E. *Internetworking with TCP-IP, Volume 1: Protocols and Architecture.* Englewood Cliffs, N.J.: Prentice-Hall, 1990. ISBN 0-13-468505-9.
- Tanenbaum, Andrew S. *Computer Networks*, 2nd ed. Englewood Cliffs, N.J.: Prentice-Hall, 1989. ISBN 0-13-162959-*X*.

Purpose of This Manual

This manual is intended to help you design and operate an Enhanced and/or Ethernet PLC-5 programmable controller system. Use this manual to help:

- determine the features of the processors and how you can use them
- select the proper hardware elements for your system
- plan your PLC-5 system
- operate your PLC-5 system

Conventions

This manual uses the following conventions:

This icon:	Indicates:
Design Tip	the information pertains to system design These tips are also referenced in the index.
MORE	the topic is discussed further in the publication referenced

- Words in square brackets represent actual keys that you press. For example: **`[Enter]`** or **`[F1] - Online Programming/Documentation`**
- Words that describe information that you have to provide are shown in italics. For example, if you have to type a file name, this is shown as: *`filename`*
- Messages and prompts that the terminal displays are shown as: `Press a function key`

The programming examples in this manual use screens from RSLogix 5 programming software.

Terms Used in This Manual

Become familiar with the following terms and definitions.

Term	Definition
Block-transfer data	data transferred, in blocks of data up to 64 words, to/from a block-transfer I/O module (for example, an analog module)
Discrete-transfer data	data (words) transferred to/from a discrete I/O module
Enhanced PLC-5 processors	references PLC-5/11™, -5/20™, -5/26™, -5/30™, -5/40™, -5/46™, -5/40L™, -5/46L™, -5/60™, -5/60L™, -5/80™, and -5/86™ processors PLC-5/26™, -5/46™, and -5/86™ processors are protected processors. See the PLC-5 Protected Processors Supplement, publication 1785-6.5.13 This term also refers to the PLC-5/V30B™, -5/V40B™, -5/V40L™, and -5/V80B™ processors when applicable. See the PLC-5/VME VMEbus Programmable Controllers User Manual, publication 1785-6.5.9, for more information
Ethernet	a local area network with a baseband communication rate of 10M bps designed for the high-speed exchange of information between computers and related devices
Ethernet PLC-5 processors	references PLC-5/20E™, -5/40E™, and -5/80E™ processors
Extended-local I/O	I/O connected to a processor across a parallel link to achieve higher throughput, thus limiting its distance from the processor
Extended local I/O link	a parallel link for carrying I/O data between a PLC-5/40L or -5/60L processor and extended-local I/O adapters
PLC-5 processor	used to generically reference Enhanced PLC-5 and Ethernet PLC-5 processors **in this manual only**
Processor-resident local I/O chassis	the I/O chassis in which the PLC-5 processor is installed
Remote I/O link	a serial communication link between a PLC-5 processor port in scanner mode and an adapter as well as I/O modules that are located remotely from the PLC-5 processor
Remote I/O chassis	the hardware enclosure that contains an adapter and I/O modules that are located remotely on a serial communication link to a PLC-5 processor in scanner mode

Manual Overview

This manual has three main sections:

- Design
- Operation
- Reference

Section:	For information about:	See Chapter:	Title:
Design	An overview of the PLC-5 processors' capabilities and keyswitch	1	Understanding Your Processor
	Guidelines for selecting and placing I/O modules	2	Selecting and Placing I/O
	The proper environment for your PLC-5 system	3	Placing System Hardware
	Choosing addressing mode, assigning rack numbers, and understanding PLC-5 memory	4	Addressing I/O and Processor Memory
Operation	Configuring the processor for processor-resident I/O, transferring data, and monitoring status	5	Communicating with Processor-Resident I/O
	Configuring a system for remote I/O communication, designing a remote I/O link, transferring data, and monitoring status	6	Communicating with Remote I/O
	Configuring a PLC-5 adapter channel, transferring data, and monitoring status	7	Communicating with a PLC-5 Adapter Channel
	For PLC-5/40L, -5/46L, and -5/60L processors only: Configuring an extended-local I/O system, transferring data, and monitoring status	8	Communicating with Extended-Local I/O
	General and specific performance considerations	9	Maximizing System Performance
	Configuring a system for Data Highway Plus™ and monitoring channel status	10	Communicating with Devices on a DH+ Link
	Configuring a system for serial communications and monitoring channel status	11	Communicating with Devices on a Serial Link
	For PLC-5/20E, -5/40E, and -5/80E processors only: Configuring a system for Ethernet communications and monitoring channel status	12	Communicating with Devices on an Ethernet Network
	Assigning passwords and privileges	13	Protecting Your Programs
	PLC-5 programming feature overview	14	Programming Considerations
	Defining power-up procedure	15	Preparing Power-Up Routines
	Defining, programming, and monitoring fault routines	16	Preparing Fault Routines
	Configuring and monitoring main control programs	17	Using Main Control Programs
	Using, defining, and monitoring selectable timed interrupts	18	Using Selectable Timed Interrupts
	Using, defining, and monitoring processor input interrupts	19	Using Processor Input Interrupts
Reference	System specifications	20	System Specifications
	Listing of the processor status file words and meaning	21	Processor Status File
	Guide to ladder instructions and execution times	22	Instruction Set Quick Reference
	How to set system switches	23	Switch Setting Reference
	Potential problems and recommended solutions	24	Troubleshooting
	Guidelines for choosing and making cables	25	Cable Reference

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 sales/support offices, 512 authorized distributors and 260 authorized systems integrators located throughout the United States alone, plus Rockwell Automation representatives in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

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

Technical Product Assistance

If you need to contact Rockwell Automation for technical assistance, please review the information in the *Troubleshooting* chapter first. Then call your local Rockwell Automation representative.

Your Questions or Comments on this Manual

If you find a problem with this manual, please notify us of it on the enclosed Publication Problem Report.

Notes:

Table of Contents

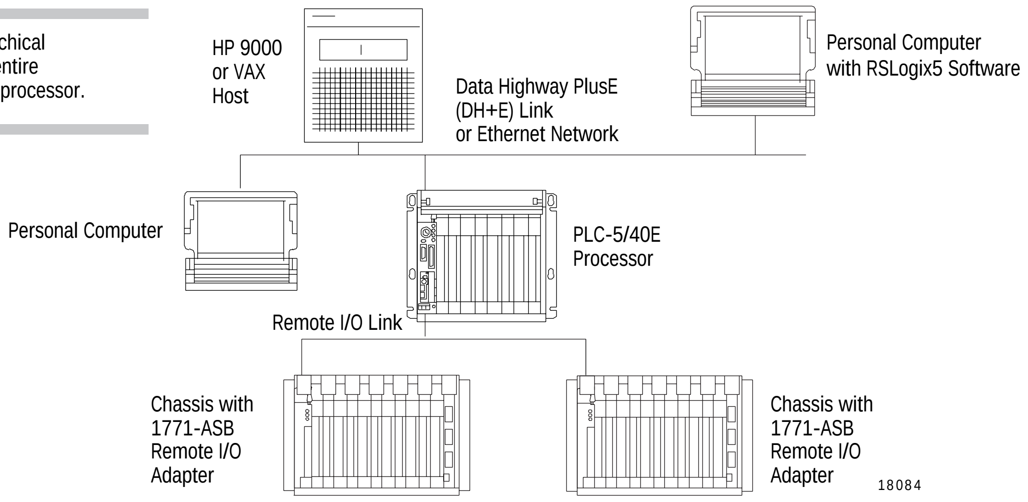
Understanding Your Processor

Using This Chapter

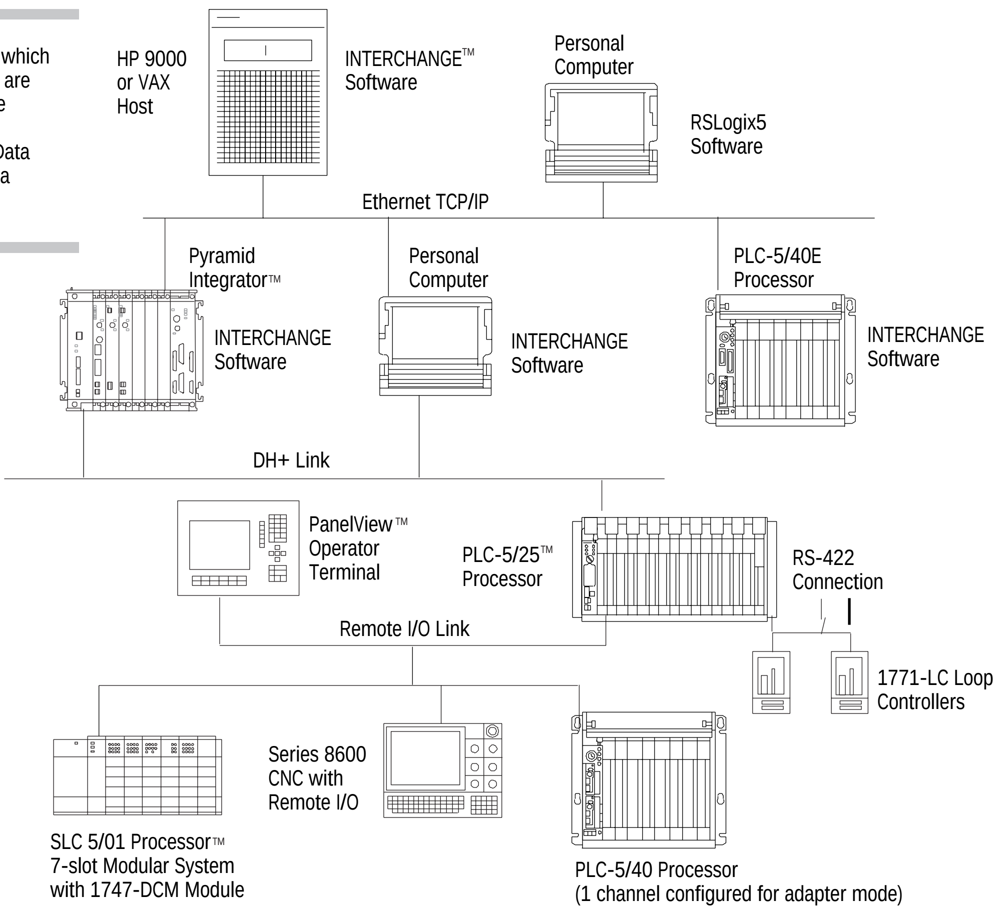
Designing Systems

You can use PLC-5 processors in a system that is designed for centralized control or in a system that is designed for distributed control.

Centralized control is a hierarchical system where control over an entire process is concentrated in one processor.



Distributed control is a system in which control and management functions are spread throughout a plant. Multiple processors handle the control and management functions and use a Data Highway™ link, an Ethernet link, or a bus system for communication.

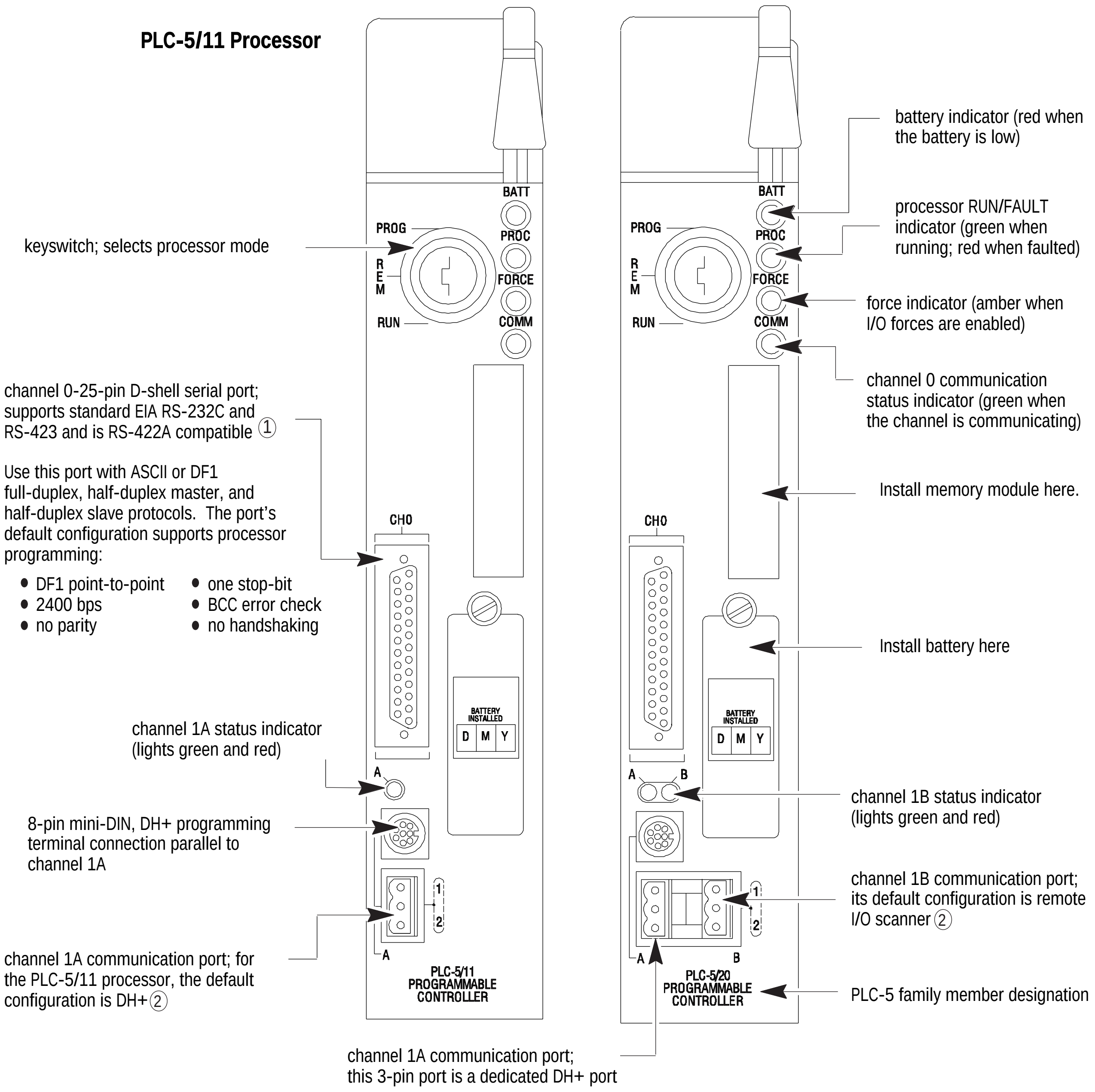


Identifying PLC-5 Processor Components

To become familiar with the processor's front panels, use these figures:

For the front panels of:	See:	Page:
PLC-5/11, -5/20 and -5/26 processors	Figure 1.1	1-3
PLC-5/30 processors	Figure 1.2	1-4
PLC-5/40, -5/46, -5/60, -5/80 and -5/86 processors	Figure 1.3	1-5
PLC-5/20E processors	Figure 1.4	1-6
PLC-5/40E and -5/80E processors	Figure 1.5	1-7
PLC-5/40L and -5/60L processors	Figure 1.6	1-8

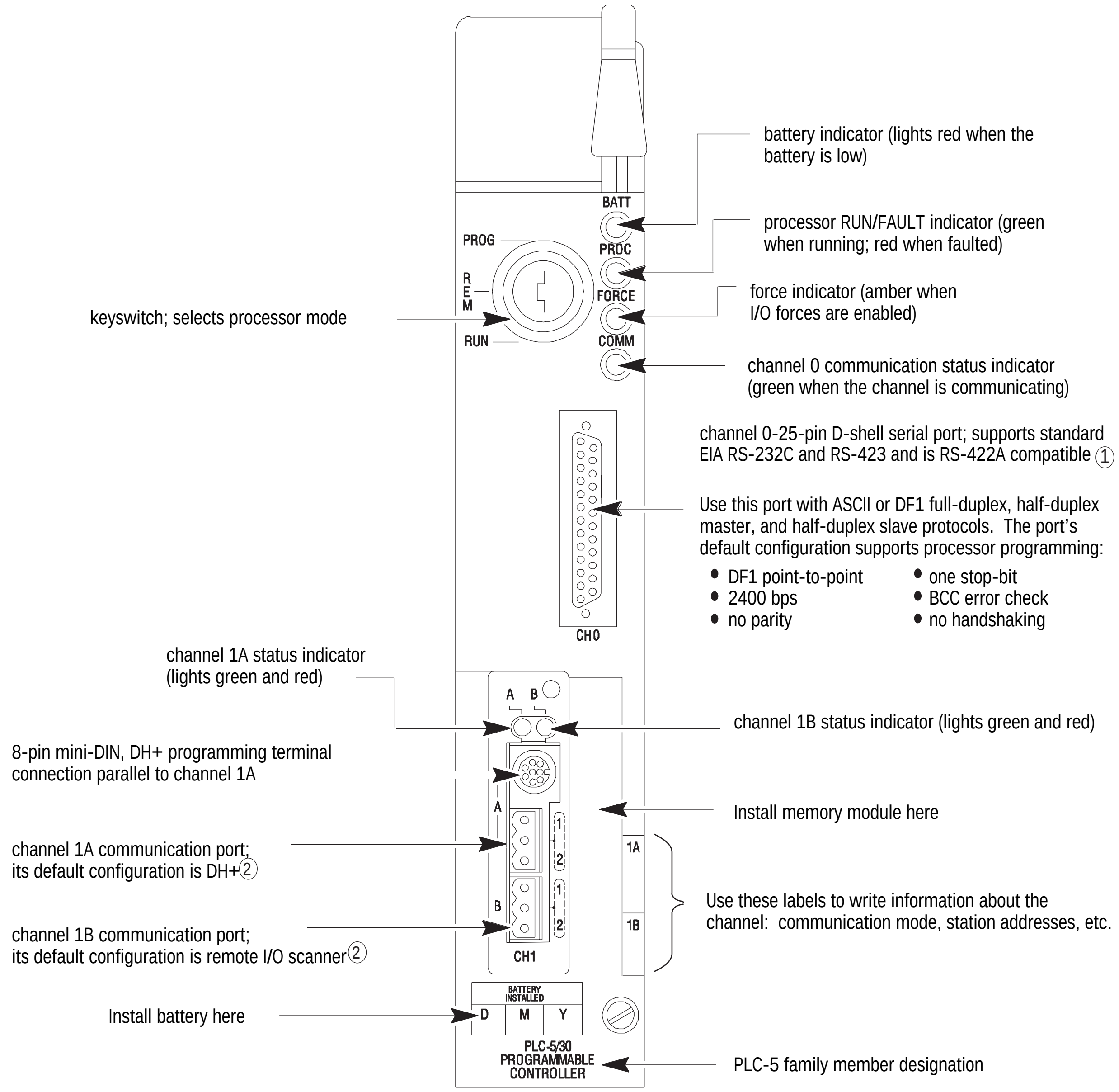
Figure 1.1
PLC-5/11, -5/20, and -5/26 Processor Front Panels


PLC-5/11 Processor
keyswitch; selects processor mode
channel 0-25-pin D-shell serial port; supports standard EIA RS-232C and RS-423 and is RS-422A compatible ①
Use this port with ASCII or DF1 full-duplex, half-duplex master, and half-duplex slave protocols. The port's default configuration supports processor programming:
• DF1 point-to-point
• 2400 bps
• no parity
• one stop-bit
• BCC error check
• no handshaking
channel 1A status indicator (lights green and red)
8-pin mini-DIN, DH+ programming terminal connection parallel to channel 1A
channel 1A communication port; for the PLC-5/11 processor, the default configuration is DH+②
channel 1A communication port; this 3-pin port is a dedicated DH+ port
battery indicator (red when the battery is low)
processor RUN/FAULT indicator (green when running; red when faulted)
force indicator (amber when I/O forces are enabled)
channel 0 communication status indicator (green when the channel is communicating)
Install memory module here.
Install battery here
channel 1B status indicator (lights green and red)
channel 1B communication port; its default configuration is remote I/O scanner ②
PLC-5 family member designation
BATT
PROC
FORCE
COMM
PROG
REM
RUN
CH0
BATTERY INSTALLED
D M Y
A
B
1
2
PLC-5/11 PROGRAMMABLE CONTROLLER
PLC-5/20 PROGRAMMABLE CONTROLLER

① Channel 0 is optically-coupled (provides high electrical noise immunity) and can be used with most RS-422A equipment as long as:
- termination resistors are not used
- the distance and transmission rate are reduced to comply with RS-423 requirements

② Configure these 3-pin ports for:
- remote I/O scanner
- remote I/O adapter,
- DH+ communication
- unused

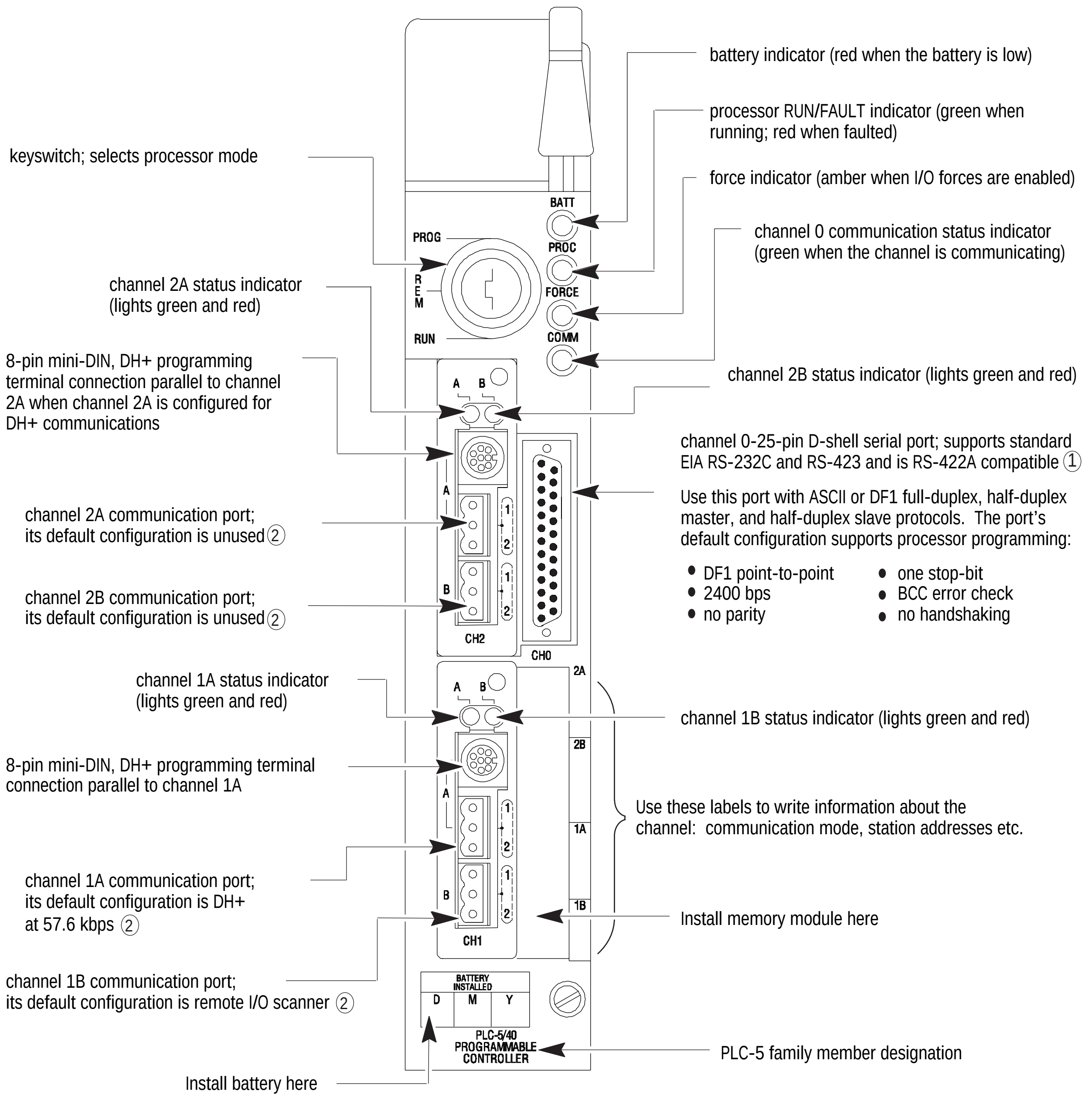
Figure 1.2
PLC-5/30 Processor Front Panel


battery indicator (lights red when the battery is low)
processor RUN/FAULT indicator (green when running; red when faulted)
force indicator (amber when I/O forces are enabled)
channel 0 communication status indicator (green when the channel is communicating)
channel 0-25-pin D-shell serial port; supports standard EIA RS-232C and RS-423 and is RS-422A compatible (1)
Use this port with ASCII or DF1 full-duplex, half-duplex master, and half-duplex slave protocols. The port's default configuration supports processor programming:
• DF1 point-to-point
• 2400 bps
• no parity
• one stop-bit
• BCC error check
• no handshaking
keyswitch; selects processor mode
BATT
PROC
FORCE
COMM
PROG
R E M
RUN
CH0
channel 1A status indicator (lights green and red)
channel 1B status indicator (lights green and red)
8-pin mini-DIN, DH+ programming terminal connection parallel to channel 1A
Install memory module here
channel 1A communication port; its default configuration is DH+(2)
channel 1B communication port; its default configuration is remote I/O scanner(2)
Use these labels to write information about the channel: communication mode, station addresses, etc.
A B
1A
1B
CH1
BATTERY INSTALLED
D M Y
Install battery here
PLC-5/30 PROGRAMMABLE CONTROLLER
PLC-5 family member designation

(1) Channel 0 is optically-coupled (provides high electrical noise immunity) and can be used with most RS-422 equipment as long as:
- termination resistors are not used
- the distance and transmission rate are reduced to comply with RS-423 requirements

(2) Configure these 3-pin ports for:
- remote I/O scanner,
- remote I/O adapter,
- DH+ communication
- unused

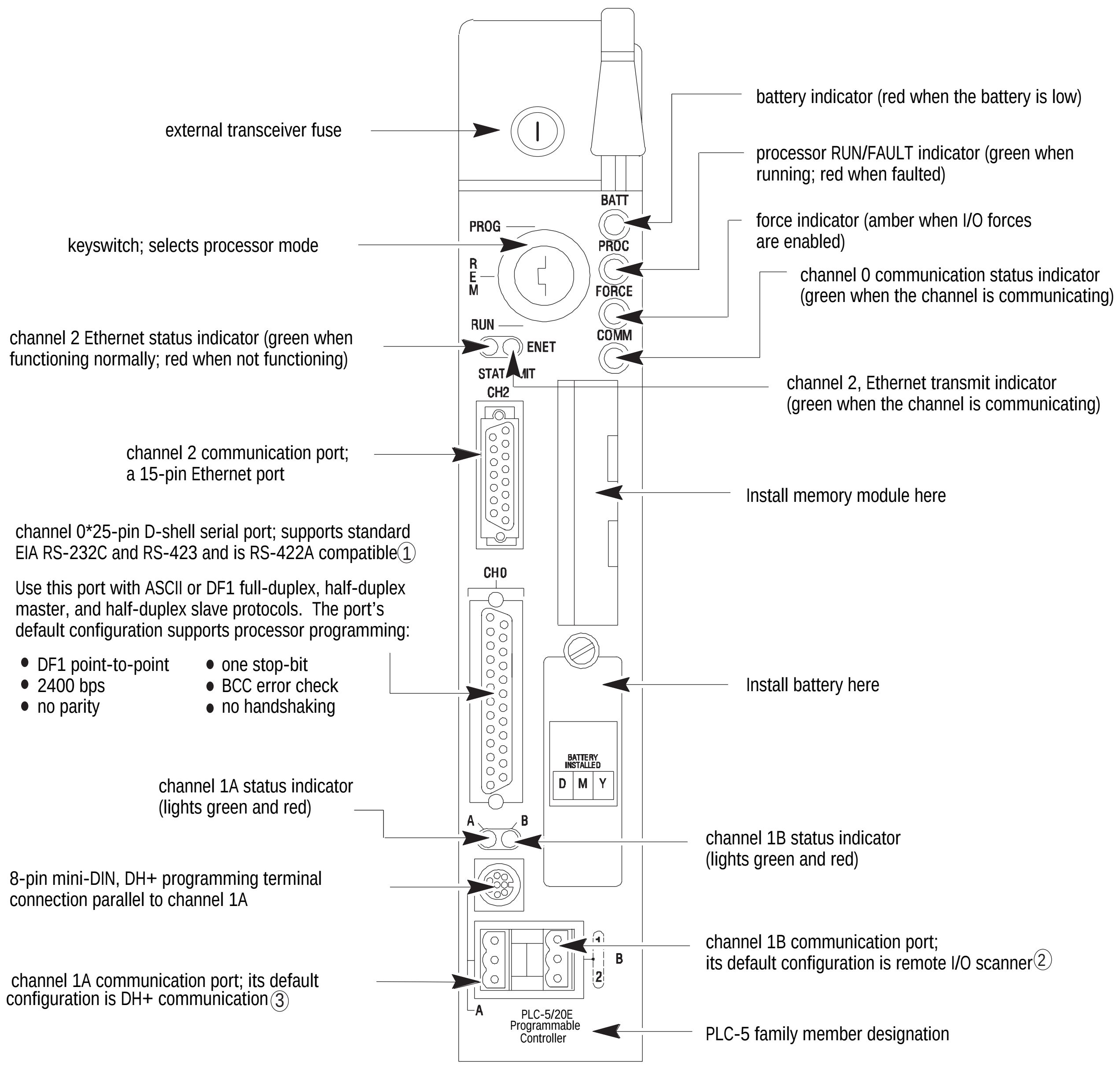
Figure 1.3
PLC-5/40, -5/46, -5/60, -5/80, and -5/86 Processor Front Panels


battery indicator (red when the battery is low)
processor RUN/FAULT indicator (green when running; red when faulted)
keyswitch; selects processor mode
force indicator (amber when I/O forces are enabled)
channel 0 communication status indicator (green when the channel is communicating)
BATT
PROG
PROC
REM
FORCE
RUN
COMM
channel 2A status indicator (lights green and red)
8-pin mini-DIN, DH+ programming terminal connection parallel to channel 2A when channel 2A is configured for DH+ communications
channel 2B status indicator (lights green and red)
A B
channel 0-25-pin D-shell serial port; supports standard EIA RS-232C and RS-423 and is RS-422A compatible ①
Use this port with ASCII or DF1 full-duplex, half-duplex master, and half-duplex slave protocols. The port's default configuration supports processor programming:
• DF1 point-to-point
• 2400 bps
• no parity
• one stop-bit
• BCC error check
• no handshaking
channel 2A communication port; its default configuration is unused ②
channel 2B communication port; its default configuration is unused ②
CH2
CH0
2A
channel 1A status indicator (lights green and red)
channel 1B status indicator (lights green and red)
8-pin mini-DIN, DH+ programming terminal connection parallel to channel 1A
2B
Use these labels to write information about the channel: communication mode, station addresses etc.
1A
channel 1A communication port; its default configuration is DH+ at 57.6 kbps ②
1B
Install memory module here
CH1
channel 1B communication port; its default configuration is remote I/O scanner ②
BATTERY INSTALLED
D M Y
PLC-5/40 PROGRAMMABLE CONTROLLER
PLC-5 family member designation
Install battery here

① Channel 0 is optically-coupled (provides high electrical noise immunity) and can be used with most RS-422A equipment as long as:
- termination resistors are not used
- the distance and transmission rate are reduced to comply with RS-423 requirements

② Configure these 3-pin ports for:
- remote I/O scanner,
- remote I/O adapter,
- DH+ communication
- unused

Figure 1.4
PLC-5/20E Processor Front Panel



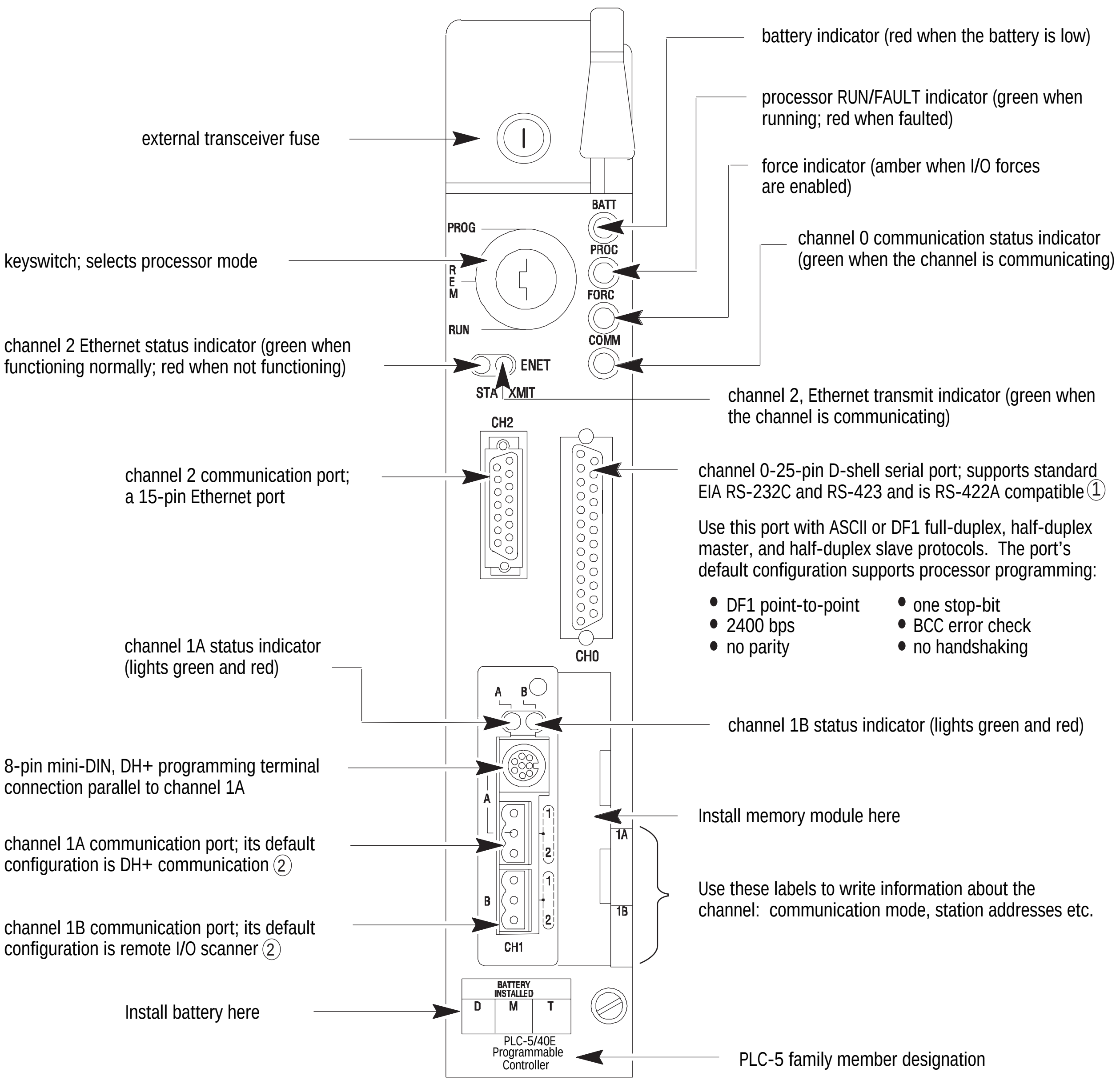
external transceiver fuse
battery indicator (red when the battery is low)
processor RUN/FAULT indicator (green when running; red when faulted)
BATT
PROG
keyswitch; selects processor mode
PROC
REM
force indicator (amber when I/O forces are enabled)
FORCE
channel 0 communication status indicator (green when the channel is communicating)
RUN
COMM
channel 2 Ethernet status indicator (green when functioning normally; red when not functioning)
ENET
STAT XMIT
CH2
channel 2, Ethernet transmit indicator (green when the channel is communicating)
channel 2 communication port; a 15-pin Ethernet port
Install memory module here
channel 0*25-pin D-shell serial port; supports standard EIA RS-232C and RS-423 and is RS-422A compatible①
CH0
Use this port with ASCII or DF1 full-duplex, half-duplex master, and half-duplex slave protocols. The port's default configuration supports processor programming:
● DF1 point-to-point
● 2400 bps
● no parity
● one stop-bit
● BCC error check
● no handshaking
Install battery here
BATTERY INSTALLED
D M Y
channel 1A status indicator (lights green and red)
A B
channel 1B status indicator (lights green and red)
8-pin mini-DIN, DH+ programming terminal connection parallel to channel 1A
channel 1B communication port; its default configuration is remote I/O scanner②
1 2 B
channel 1A communication port; its default configuration is DH+ communication③
A
PLC-5/20E Programmable Controller
PLC-5 family member designation

① Channel 0 is optically-coupled (provides high electrical noise immunity) and can be used with most RS-422A equipment as long as:
- termination resistors are not used
- the distance and transmission rate are reduced to comply with RS-423 requirements

② Configure these 3-pin ports for:
- remote I/O scanner
- remote I/O adapter
- DH+ communication
- unused

③ Configure this 3-pin port for:
- remote I/O adapter
- DH+ communication

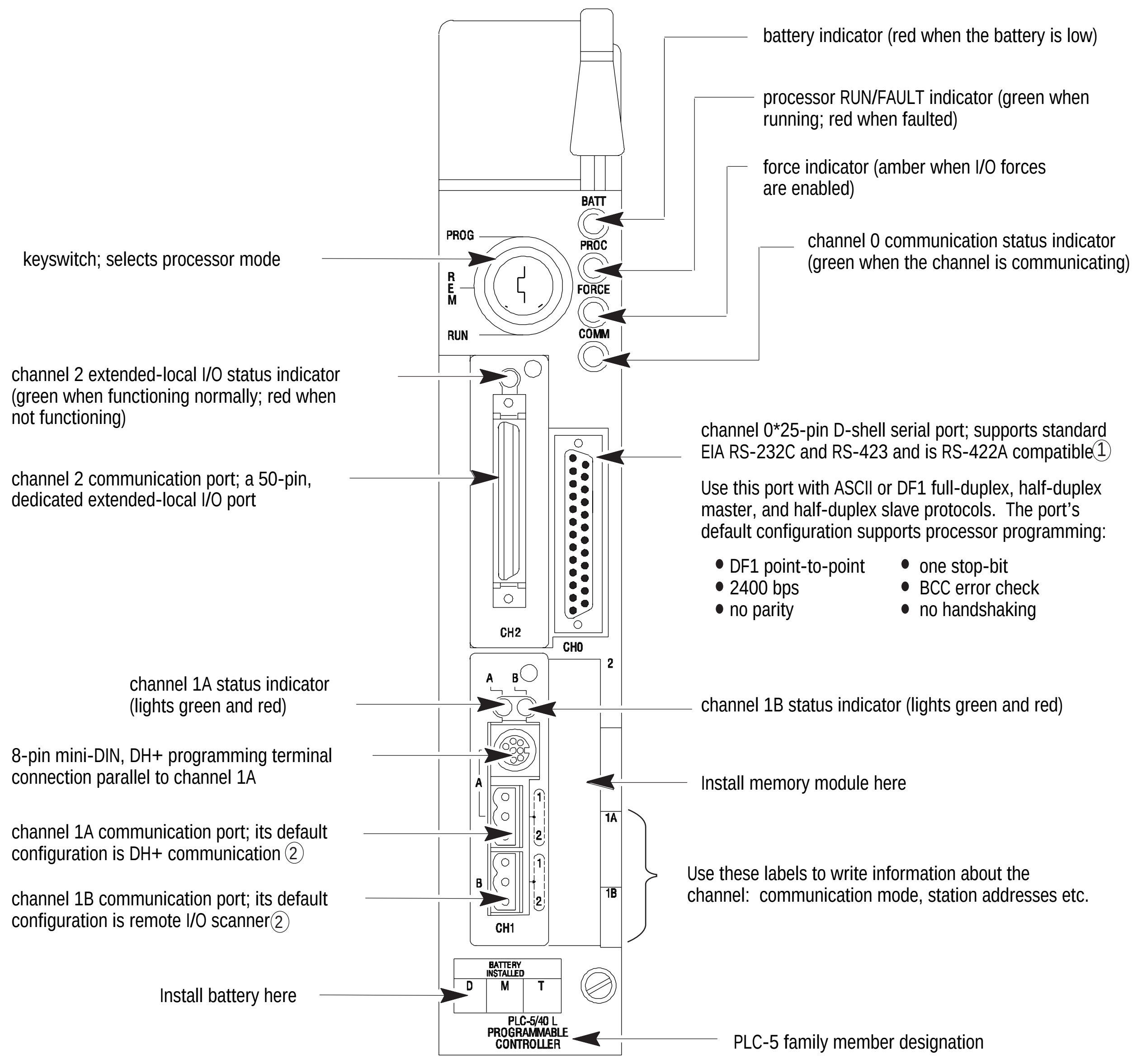
Figure 1.5
PLC-5/40E and -5/80E Processor Front Panels


battery indicator (red when the battery is low)
processor RUN/FAULT indicator (green when running; red when faulted)
external transceiver fuse
force indicator (amber when I/O forces are enabled)
BATT
PROG
PROC
channel 0 communication status indicator (green when the channel is communicating)
keyswitch; selects processor mode
R E M
FORC
RUN
COMM
channel 2 Ethernet status indicator (green when functioning normally; red when not functioning)
ENET
STA
XMIT
channel 2, Ethernet transmit indicator (green when the channel is communicating)
CH2
channel 2 communication port; a 15-pin Ethernet port
channel 0-25-pin D-shell serial port; supports standard EIA RS-232C and RS-423 and is RS-422A compatible (1)
Use this port with ASCII or DF1 full-duplex, half-duplex master, and half-duplex slave protocols. The port's default configuration supports processor programming:
• DF1 point-to-point
• 2400 bps
• no parity
• one stop-bit
• BCC error check
• no handshaking
channel 1A status indicator (lights green and red)
CH0
A B
channel 1B status indicator (lights green and red)
8-pin mini-DIN, DH+ programming terminal connection parallel to channel 1A
A
Install memory module here
1A
channel 1A communication port; its default configuration is DH+ communication (2)
1 2
Use these labels to write information about the channel: communication mode, station addresses etc.
B
1B
channel 1B communication port; its default configuration is remote I/O scanner (2)
CH1
BATTERY INSTALLED
D M T
Install battery here
PLC-5/40E Programmable Controller
PLC-5 family member designation

(1) Channel 0 is optically-coupled (provides high electrical noise immunity) and can be used with most RS-422A equipment as long as:
- termination resistors are not used
- the distance and transmission rate are reduced to comply with RS-423 requirements

(2) Configure these 3-pin ports for:
- remote I/O scanner
- remote I/O adapter
- DH+ communication
- unused

Figure 1.6
PLC-5/40L and -5/60L Processor Front Panels



battery indicator (red when the battery is low)

processor RUN/FAULT indicator (green when running; red when faulted)

force indicator (amber when I/O forces are enabled)

channel 0 communication status indicator (green when the channel is communicating)

keyswitch; selects processor mode

channel 2 extended-local I/O status indicator (green when functioning normally; red when not functioning)

channel 0*25-pin D-shell serial port; supports standard EIA RS-232C and RS-423 and is RS-422A compatible[1]

Use this port with ASCII or DF1 full-duplex, half-duplex master, and half-duplex slave protocols. The port's default configuration supports processor programming:

- DF1 point-to-point
- 2400 bps
- no parity
- one stop-bit
- BCC error check
- no handshaking

channel 2 communication port; a 50-pin, dedicated extended-local I/O port

channel 1A status indicator (lights green and red)

channel 1B status indicator (lights green and red)

8-pin mini-DIN, DH+ programming terminal connection parallel to channel 1A

Install memory module here

channel 1A communication port; its default configuration is DH+ communication [2]

Use these labels to write information about the channel: communication mode, station addresses etc.

channel 1B communication port; its default configuration is remote I/O scanner[2]

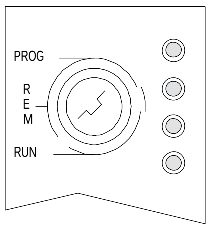
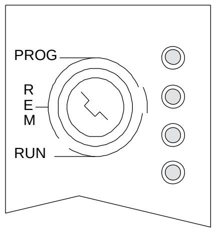
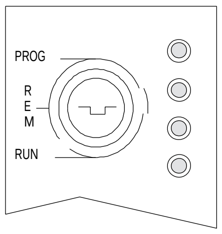
Install battery here

PLC-5 family member designation

[1] Channel 0 is optically-coupled (provides high electrical noise immunity) and can be used with most RS-422A equipment as long as:
- termination resistors are not used
- the distance and transmission rate are reduced to comply with RS-423 requirements

[2] Configure these 3-pin ports for:
- remote I/O scanner,
- remote I/O adapter,
- DH+ communication
- unused

Use the keyswitch to change the mode in which a processor is operating.

If you want to:	Turn the keyswitch to:
• Run your program. Outputs are enabled. (Equipment being controlled by the I/O addressed in the ladder program begins operation.) • Force I/O. • Save your programs to a disk drive (during operation). • Enable outputs. • Edit data table values. **Notes:** • You cannot create or delete a program file, create or delete data files, edit online, or change the modes of operation through the programming software while in run mode. • You can prevent forcing and data table changes by usingRSLogix5 programming software to set user control bit S:26/6.	RUN 
• Disable outputs (outputs are turned off). • Create, modify, and delete ladder files, SFC files, or data files. • Download to/from a memory module. • Save/restore programs. **Notes:** • The processor does not scan the program. • You cannot change the mode of operation through the programming software while in program mode.	PROG (program) 
Change between remote program, remote test, and remote run modes through the programming software. **Remote run** • Enable outputs. • Save/restore programs. • Edit while operating. **Remote program** See the program-mode description above. **Remote test** • Execute ladder programs with outputs disabled. • *Cannot* create or delete ladder programs or data files. • Save/restore programs. • Edit while operating.	REM (remote) 

Programming Features

This table highlights the programming features of a PLC-5 processor.

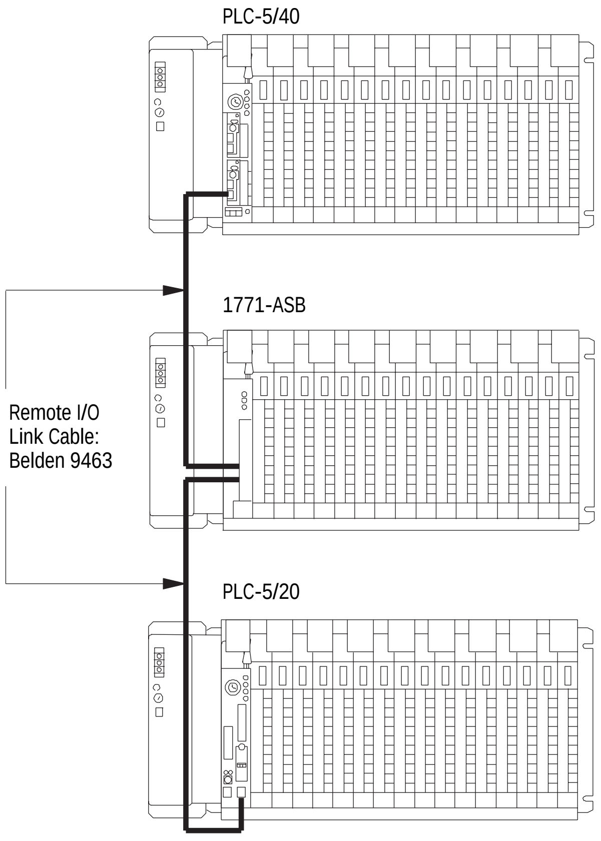
This capability:	Lets you:
Ladder logic	program using a language that is representative of relay logic. Choose this language • if you are more familiar with ladder logic than with programming languages such as BASIC Your plant personnel may be more familiar with ladder logic; consider their needs as well. • performing diagnostics • programming discrete control
Subroutines	store recurring sections of program logic that can be accessed from multiple program files. A subroutine saves memory because you program repetitive logic only once. The JSR instruction directs the processor to go to a separate subroutine file within the logic processor, scan that subroutine file once, and return to the point of departure.
Sequential Function Charts (SFCs)	use sequence-control language to control and display the state of a sequential process. Instead of using one long ladder program for your application, divide the logic into steps and transitions. A step corresponds to a control task; a transition corresponds to a condition that must occur before the programmable controller can perform the next control task. The display of these steps and transitions lets you see what state the machine process is in at a given time via a flowchart form. SFCs offer constructs that enable execution of multiple paths of logic, or a single selected path of logic, as well as the ability to jump forwards and backwards. Troubleshooting can be reduced to a small routine of logic instead of an entire ladder file. SFCs are best for defining the order of events in a sequential process.
Structured text	program using a language similar to BASIC. Choose structured text if you are: • more familiar with programming languages such as BASIC than with ladder logic • using complex mathematical algorithms • using program constructs that repeat or “loop” • creating custom data-table monitoring screens
Main Control Programs (MCPs)	separate sequential logic from ladder logic and structured text as a way of modularized your process and making troubleshooting easier. Use several main control programs (MCPs) to define one main control program for each particular machine or function of your process. MCPs accommodate independent or non-sequential activities. A main control program can be an SFC file numbered 1-999 or a ladder-logic file or structured-text program numbered 2-999. One data table is used by all MCPs (i.e., you do not have a separate data table for each MCP).

Using a PLC-5 Processor Channel as a Remote I/O Scanner

Configure a remote I/O channel for scanner mode to read and write I/O information between a PLC-5 processor and an I/O device remotely located from the processor.

A processor with a channel configured for scanner mode acts as a supervisory processor for other processors that are in adapter mode as well as remote I/O adapter modules. The scanner-mode PLC-5 processor can:

- gather data from node adapter devices in remote I/O racks
- process I/O data from 8-, 16-, or 32-point I/O modules
- address I/O in 2-, 1-, or 1/2-slot I/O groups
- support a complementary I/O configuration
- support block-transfer in any I/O chassis

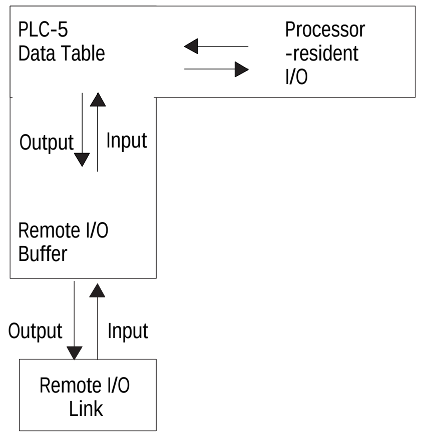


The scanner-mode PLC-5 processor:

- transfers discrete data and block-transfer data to/from modules in remote I/O racks as well as to/from processors in adapter mode.
- scans ***remote I/O buffers*** asynchronously to the program scan.
- updates the ***input/output image data table*** from the remote I/O buffer(s) synchronously to the program scan

PLC-5 data table is updated synchronously to program scan (at housekeeping).

Remote I/O buffers are updated asynchronously to the program scan.



A PLC-5 processor transfers I/O data and status data using:

• discrete transfers	data transfers of 8 words per rack occur automatically on the remote I/O network
• block-transfers	special data transfers that require ladder logic instructions to achieve the transfer allow a transfer of a maximum of 64 words of data also used to communicate information between a scanner channel and an adapter-mode processor channel

MORE

For more information about using the processor as a remote I/O scanner, see chapter 6.

Using a PLC-5 Processor Channel as a Remote I/O Adapter

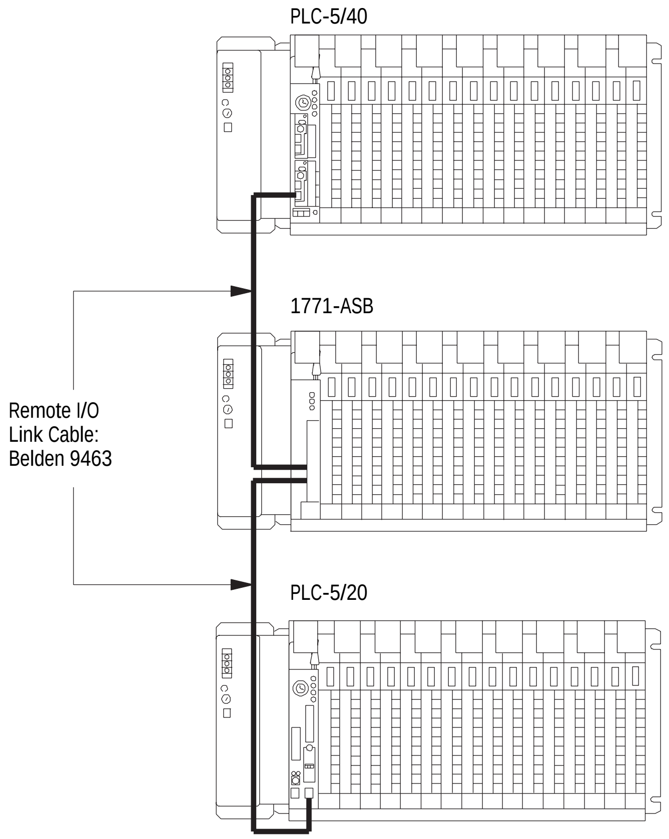
Configure a PLC-5 processor channel for adapter mode when you need predictable, real-time exchange of data between a distributed control adapter-mode PLC-5 processor channel and a supervisory processor. The remote I/O adapter channel exchanges data with a supervisory processor.

In this example, a PLC-5/40 processor channel is the supervisory (scanner-mode) processor of the 1771-ASB module and the PLC-5/20 processor.

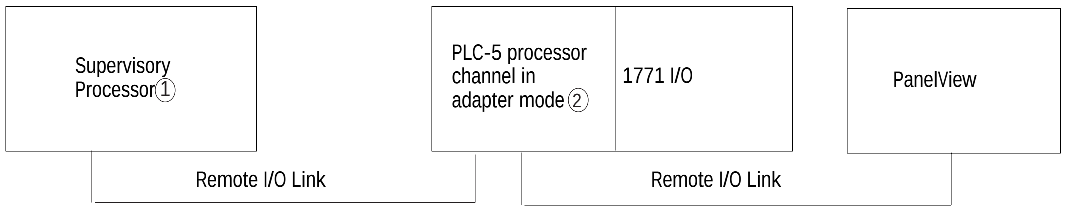
Connect the processors via the remote I/O link.

You can monitor status between the supervisory processor and the adapter-mode PLC-5 processor channel at a consistent rate (i.e., the transmission rate of the remote I/O link is unaffected by programming terminals and other non-control-related communications).

The adapter-mode PLC-5 processor can monitor and control its processor-resident local I/O while communicating with the supervisory processor via a remote I/O link.



For Enhanced and Ethernet PLC-5 processor channels in adapter mode, you do not need ladder logic in the adapter processor for block-transfer instructions. You define the block-transfers via an adapter configuration screen and by defining block-transfer files.



① The following programmable controllers can operate as supervisory processors:

PLC-2/20 and PLC-2/30 processors
PLC-3 and PLC-3/10 processors
PLC-5/15 and PLC-5/25E processors
All Enhanced and Ethernet PLC-5 processors; separate channels can be configured for a remote I/O scanner and an adapter
PLC-5/V30, PLC-5/V40, PLC-5/V40L, and PLC-5/V80 processors
PLC-5/250 processors

② All PLC-5 family processors, except the PLC-5/10, can operate as remote I/O adapter modules

MORE

For more information about using the processor as a remote I/O adapter, see chapter 7.

Using a PLC-5/40L, -5/60L Processor as an Extended-Local I/O Scanner

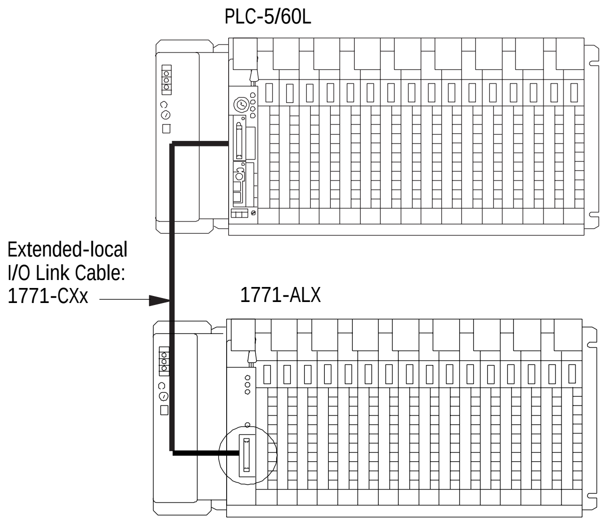
Use the extended-local I/O link when you need I/O updates more quickly than is possible from remote I/O link. An extended-local I/O link provides faster scan and update time than a remote I/O link. The extended-local I/O link is limited to 30.5 cable-m (100 cable-ft). If an I/O chassis is located more than 30.5m from the processor, you must use a remote I/O link.

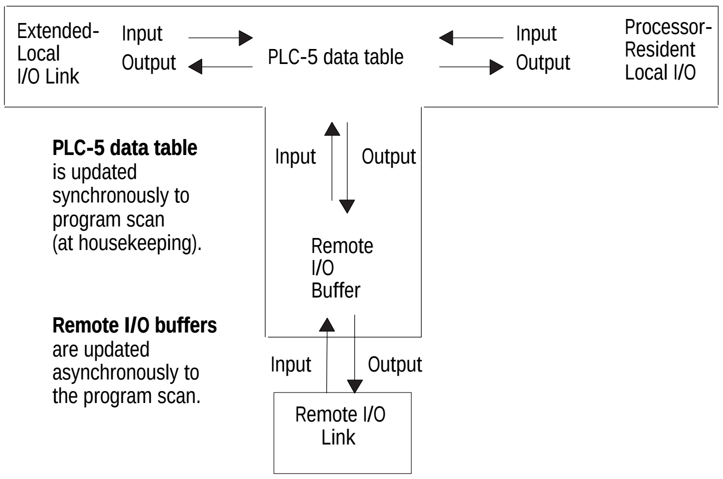
A PLC-5/40L or -5/60L processor (channel 2) and an extended-local I/O adapter module (1771-ALX) form an extended-local I/O link.

The extended-local I/O link is a parallel link that enables a PLC-5/40L or -5/60L processor to scan a maximum of 16 extended-local I/O chassis.

Due to the cabling design, you can remove an adapter module from a chassis on the extended-local I/O link without disrupting communication to other chassis on the extended-local I/O link.

Important: The PLC-5/40L and -5/60L processors cannot be used as extended-local I/O adapters.





PLC-5 data table is updated synchronously to program scan (at housekeeping).

Remote I/O buffers are updated asynchronously to the program scan.

MORE

For more information about using extended-local I/O, see chapter 8.

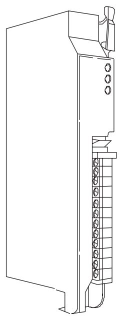
Selecting and Placing I/O

Using This Chapter

Selecting I/O Modules

Select I/O modules to interface your PLC-5 processor with machines or processes that you determine while analyzing your plant operation.

Use the following list and table as guidelines for selecting I/O modules and operator control interface(s).



- How much I/O is required to control your process(es)?
- Where will you concentrate I/O points for portions of an entire process when the entire process is distributed over a large physical area?
- What type of I/O is required to control your process(es)?
- What is the required voltage range for each I/O module?
- What is the backplane current required for each I/O module?
- What are the noise and distance limitations for each I/O module?
- What isolation is required for each I/O module?

Table 2.A
Guidelines for Selecting I/O Modules

Choose this type of I/O module:	For these types of field devices or operations (examples):	Explanation:
Discrete input module and block I/O module	Selector switches, pushbuttons, photoelectric eyes, limit switches, circuit breakers, proximity switches, level switches, motor starter contacts, relay contacts, thumbwheel switches	Input modules sense on/off or opened/closed signals. Discrete signals can be either ac or dc.
Discrete output module and block I/O module	Alarms, control relays, fans, lights, horns, valves, motor starter, or solenoids	Output module signals interface with on/off or opened/closed devices. Discrete signals can be either ac or dc.
Analog input module	Temperature transducers, pressure transducers, load cell transducers, humidity transducers, flow transducers, and potentiometers	Convert continuous analog signals into input values for the PLC® processor.
Analog output module	Analog valves, actuators, chart recorders, electric motor drives, analog meters	Interpret PLC processor output to analog signals (generally through transducers) for field devices.
Specialty I/O modules	Encoders, flow meters, I/O communication, ASCII, RF type devices, weigh scales, bar-code readers, tag readers, display devices	Are generally used for specific applications such as position control, PID, and external device communication.

Selecting I/O Module Density

The density of an I/O module is the number of processor input or output image-table bits to which it corresponds. A bidirectional module with 8 input bits and 8 output bits has a density of 8. I/O module density helps determine your I/O addressing scheme. See chapter 4 for more information about I/O addressing.

Use these guidelines for selecting I/O module density:

Table 2.B
Guidelines for Selecting I/O Module Density

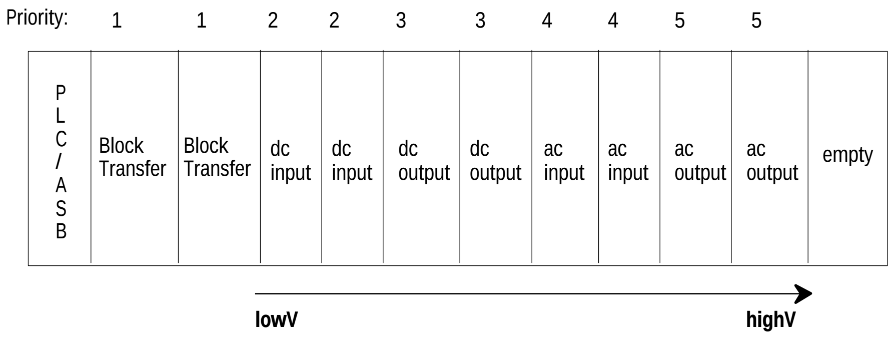
Choose this I/O density:	If you:
8-point I/O module	• currently use 8-point modules • need integral, separately-fused outputs • want to minimize cost per module
16-point I/O module	• currently use 16-point modules • need separately-fused outputs with a special wiring arm
32-point I/O module	• currently use 32-point modules • want to minimize number of modules • want to minimize the space required for I/O chassis • want to minimize cost per I/O point

Placing I/O Modules in a Chassis

Place I/O modules in a chassis depending on the electrical characteristics of the module. The placement is made left to right, with the left-most position being closest in the chassis to the PLC-5 processor or the I/O adapter module. The placement order is as follows:

Module placement priority:

1. block-transfer modules (all types)
2. dc input modules
3. dc output modules
4. ac input modules
5. ac output modules



Place block-transfer modules according to these guidelines:

- Place as many modules as possible for which you need fast block-transfer times in your processor-resident local I/O chassis.
- Place modules in which block-transfer timing is not as critical in remote I/O chassis.
- Ac output modules should always be the furthest I/O modules away from any block-transfer modules in the same chassis.

Place input and output modules according to these guidelines:

- left to right
- lowest voltage to highest voltage

For optimal speed using discrete I/O, use the following module-placement priority scheme:

1. processor chassis
2. extended-local I/O chassis
3. remote I/O chassis

Notes:

Placing System Hardware

Using This Chapter

Determining the Proper Environment

Place the processor in an environment with conditions that fall within these guidelines:

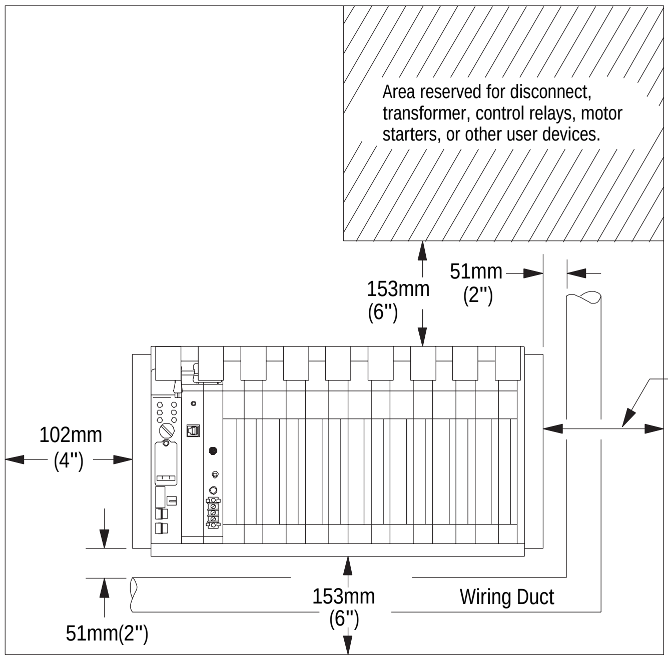
Environmental Condition:	Acceptable Range:
Operating temperature	0 to 60° C (32 to 140° F)
Storage temperature	-40 to 85° C (-40 to 185° F)
Relative humidity	5 to 95% (without condensation)

Separate your programmable controller system from other equipment and plant walls to allow for convection cooling. Convection cooling draws a vertical column of air upward over the processor. This cooling air must not exceed 60° C (140° F) at any point immediately below the processor. If the air temperature exceeds 60° C, install fans that bring in filtered air or recirculate internal air inside the enclosure, or install air-conditioning/heat-exchanger units.

To allow for proper convection cooling in enclosures containing a processor-resident chassis and remote I/O chassis, follow these guidelines.

Minimum spacing requirements for a processor-resident chassis:

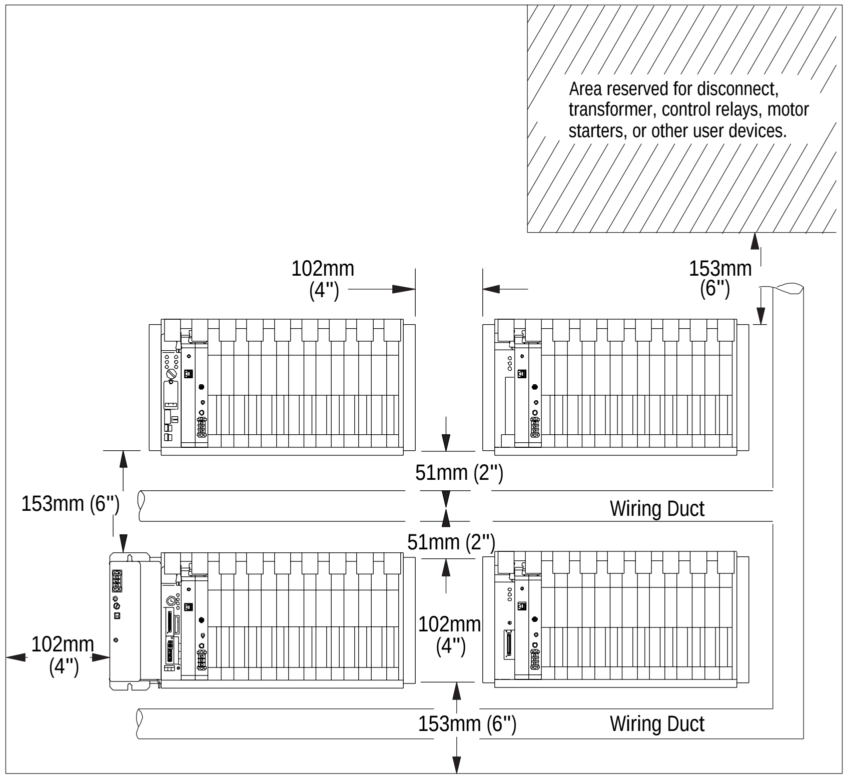
- Mount the I/O chassis horizontally.
- Allow 153 mm (6 in) above and below the chassis.
- Allow 102 mm (4 in) on the sides of each chassis.
- Allow 51 mm (2 in) vertically and horizontally between any chassis and the wiring duct or terminal strips.
- Leave any excess space at the top of the enclosure, where the temperature is the highest.


Area reserved for disconnect, transformer, control relays, motor starters, or other user devices.
153mm (6")
51mm (2")
102mm (4")
102mm (4")
153mm (6")
Wiring Duct
51mm(2")

13081

Minimum spacing requirements for a remote I/O and extended-local I/O chassis:

- Mount the I/O chassis horizontally.
- Allow 153 mm (6 in) above and below all chassis. When you use more than one chassis in the same area, allow 152.4 mm (6 in) between each chassis.
- Allow 102 mm (4 in) on the sides of each chassis. When you use more than one chassis in the same area, allow 101.6 mm (4 in) between each chassis.
- Allow 51 mm (2 in) vertically and horizontally between any chassis and the wiring duct or terminal strips.
- Leave any excess space at the top of the enclosure, where the temperature is the highest.



18749

Protecting Your Processor

You provide the enclosure for your processor system. This enclosure protects your processor system from atmospheric contaminants such as oil, moisture, dust, corrosive vapors, or other harmful airborne substances. To help guard against electromagnetic interference (EMI) and radio frequency interference (RFI), we recommend a steel enclosure.

Mount the enclosure in a position where you can fully open the doors. You need easy access to processor wiring and related components so that troubleshooting is convenient.

When you choose the enclosure size, allow extra space for transformers, fusing, disconnect switch, master control relay, and terminal strips.

Avoiding Electrostatic Damage

ATTENTION: Under some conditions, electrostatic discharge can degrade performance or damage the processor module. Read and observe the following precautions to guard against electrostatic damage.ESD protection

- Wear an approved wrist strap grounding device when handling the processor module.
- Touch a grounded object to discharge yourself before handling the processor module.
- Do not touch the backplane connector or connector pins.

Laying Out Your Cable Raceway

The raceway layout of a system reflects where the different types of I/O modules are placed in I/O chassis. Therefore, you should determine I/O-module placement prior to any layout and routing of wires. When planning your I/O-module placement, however, segregate the modules based on the conductor categories published for each I/O module so that you can follow these guidelines. These guidelines coincide with the guidelines for “the installation of electrical equipment to minimize electrical noise inputs to controllers from external sources” in IEEE standard 518-1982.

To plan a raceway layout, do the following:

- categorize conductor cables
- route conductor cables

Categorize Conductors

Segregate all wires and cables into categories as described in the *Industrial Automation Wiring and Grounding Guidelines*, publication 1770-4.1. See the installation data for each I/O module that you are using for information about its classification.

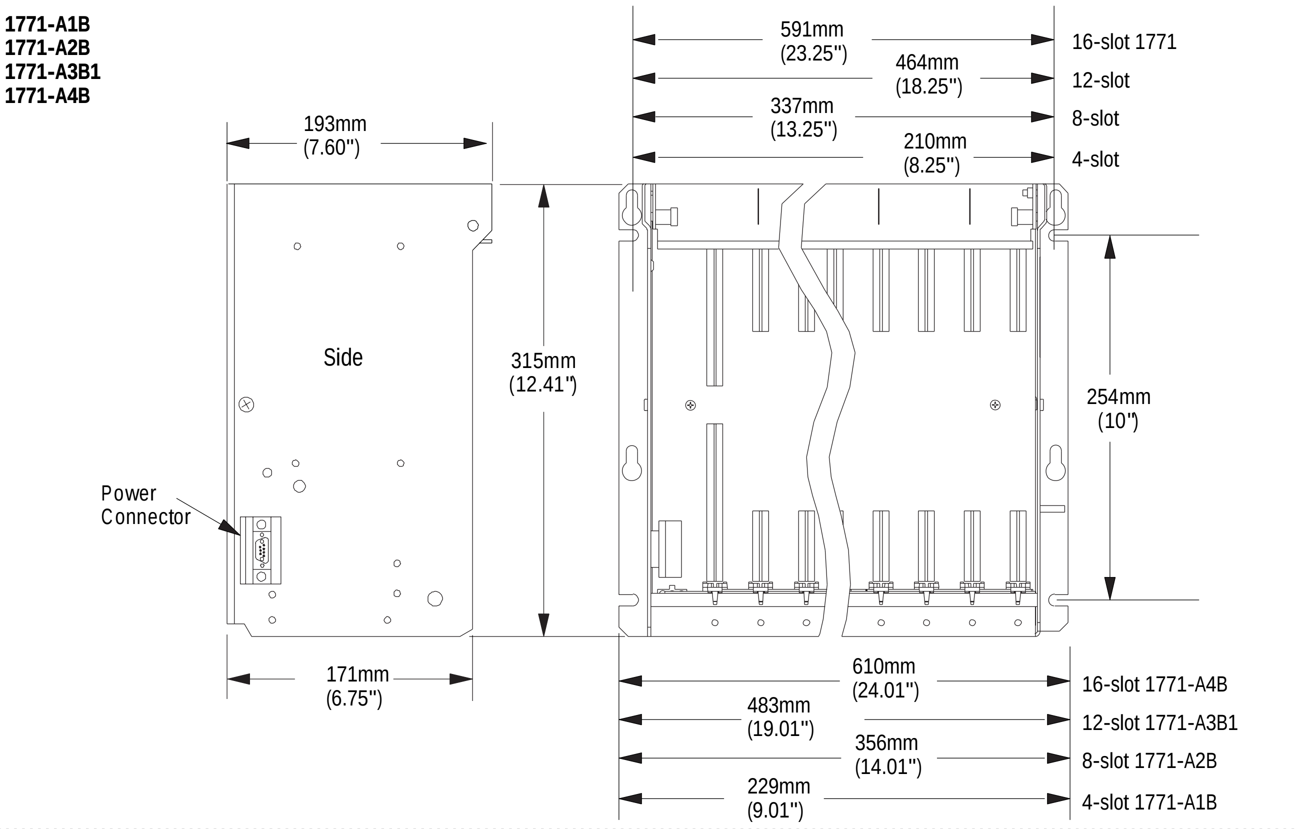
Route Conductors

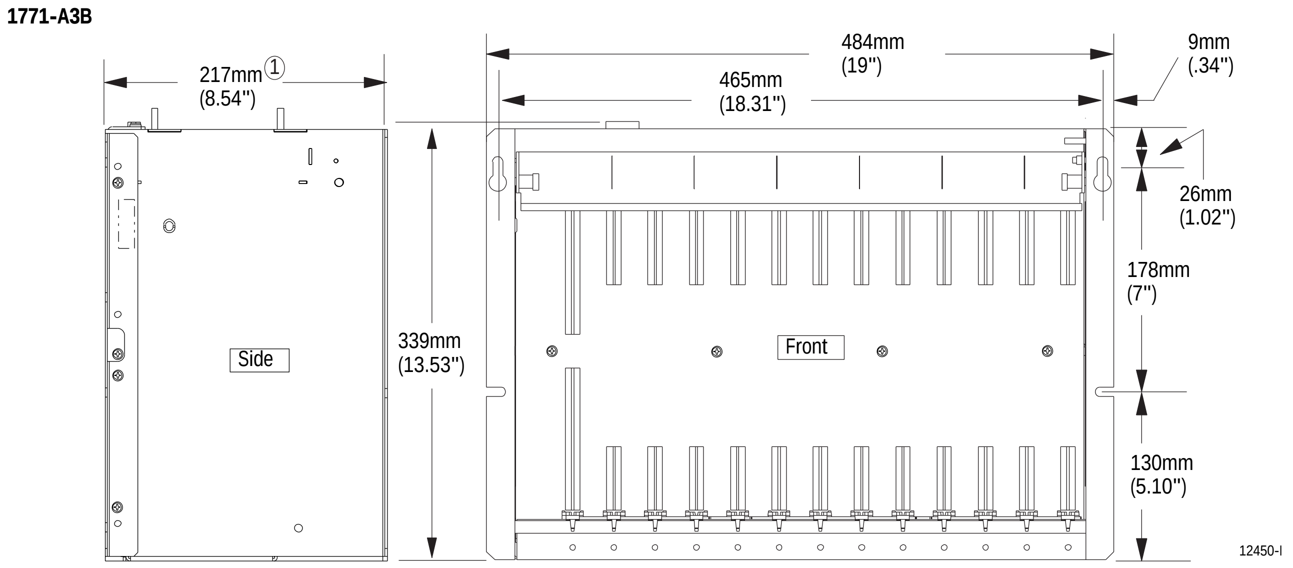
To guard against coupling noise from one conductor to another, follow the general guidelines for routing cables described in the *Industrial Automation Wiring and Grounding Guidelines*, publication 1770-4.1. You should follow the safe grounding and wiring practices called out in the National Electrical Code (NEC, published by the National Fire Protection Association, in Quincy, Massachusetts), and local electrical codes.

Laying Out Your Backpanel Spacing

Use 6.35 mm (0.25 inch) mounting bolts to attach the I/O chassis to the enclosure backpanel.

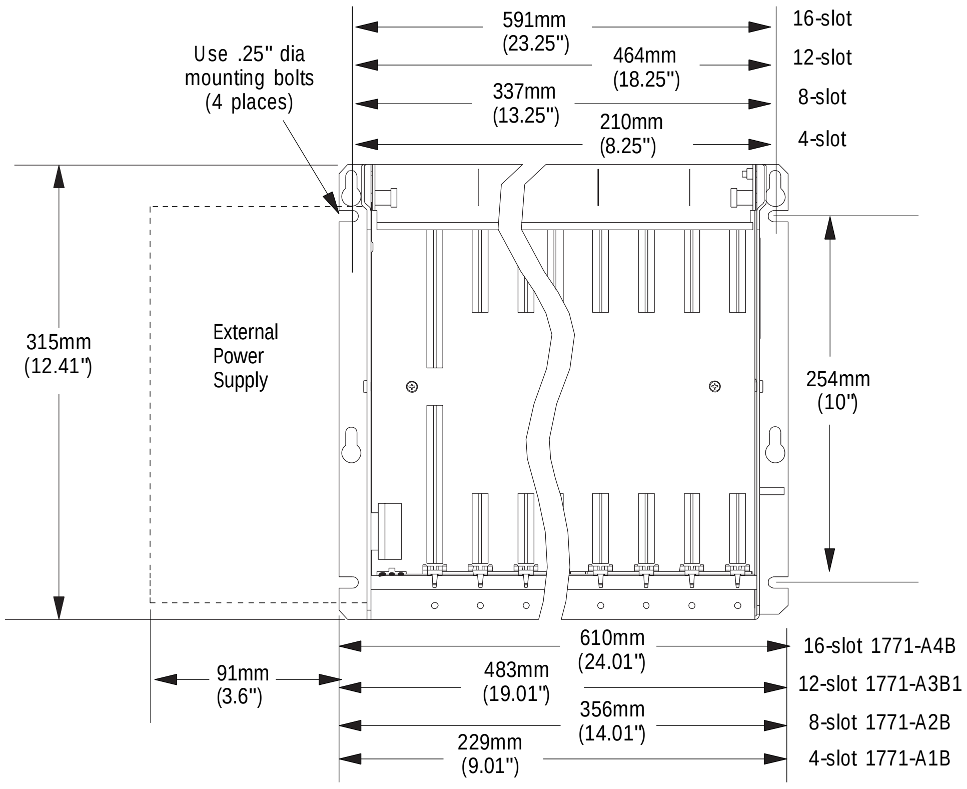
Figure 3.1
Chassis Dimensions (Series B)





① Total maximum depth dimension per installation will be dependent upon module wiring and connectors.

Figure 3.2
I/O Chassis and External Power Supply Dimensions



Clearance depth is 204mm (8") for 8 I/O connection points per module.

12451-I

Grounding Your System

For this grounding configuration:	Refer to:
remote I/O system grounding	Figure 3.3
extended-local I/O grounding	Figure 3.4

MORE

For more information on proper grounding guidelines, see the *Industrial Automation Wiring and Grounding Guidelines*, publication 1770-4.1.

Figure 3.3
Recommended Grounding Configuration for Remote I/O Systems

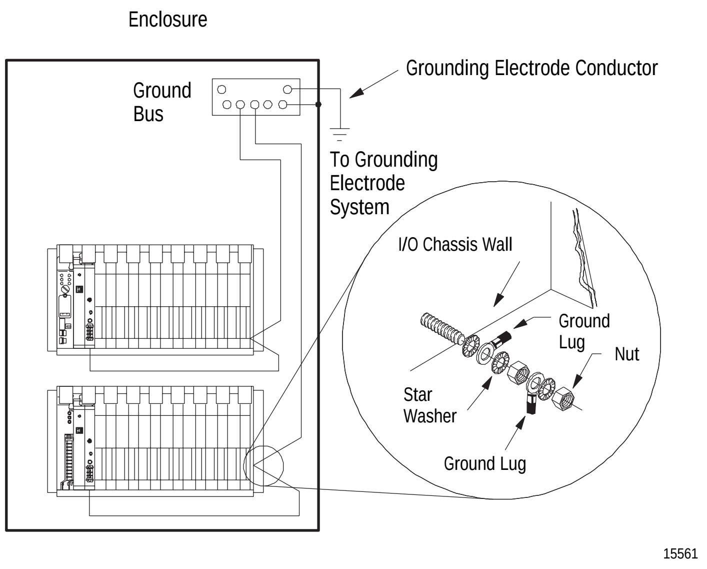
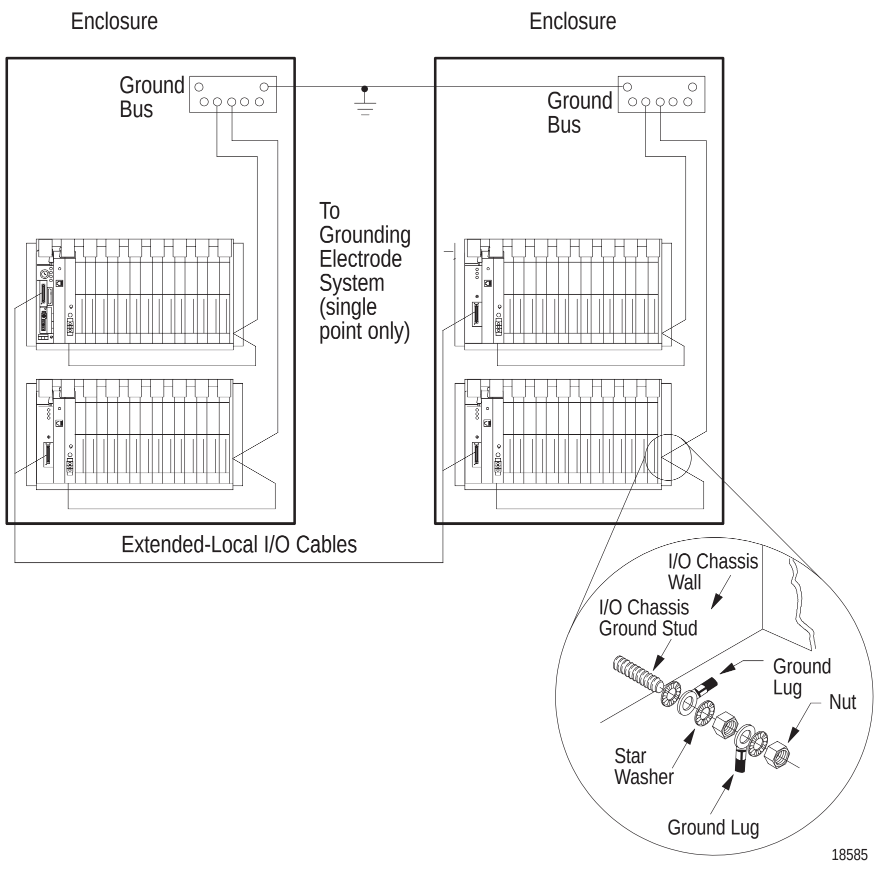


Figure 3.4
Required Grounding Configuration for Extended-Local I/O Systems



Notes:

Addressing I/O and Processor Memory

Using This Chapter

I/O Addressing Concept

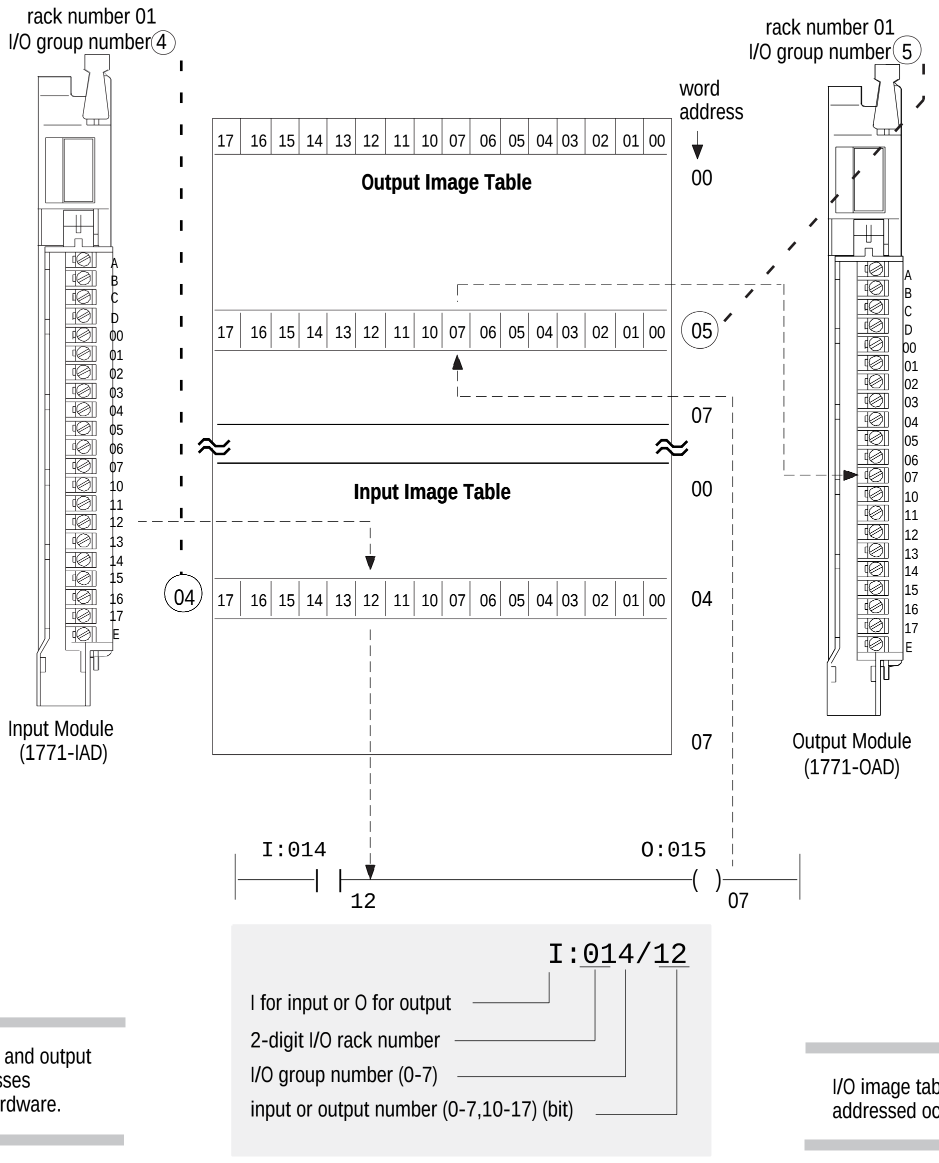
Since the main purpose of a programmable controller is to control inputs and outputs of field devices like switches, valves, and thermocouples, these inputs and outputs must occupy a location in the processor memory so that they can be addressed in your control program. Each terminal on an input or output module that can be wired to a field device occupies a bit within processor memory. The part of processor memory that houses I/O addresses is the **input image table** and the **output image table**.

I/O addressing helps connect the physical location of an I/O module terminal to a bit location in the processor memory. I/O addressing is just a way to segment processor memory. The segmentation is as follows:

Classification:	Term:	Relation to processor memory:
A specific terminal on an I/O module that occupies a space in processor memory	terminal or point	The density of an I/O module, i.e., 8-point, 16-point, 32-point, directly relates to the amount of memory (bits) the module occupies in processor memory. For example, a 16-point input module occupies 16 bits in the processor's input image table.
I/O terminals that when combined occupy 1 word in processor's input image table **and** 1 word in the processor's output image table.	I/O group	16 input bits = 1 **word** in processor's input image table 16 output bits = 1 **word** in the processor's output image table
Processor memory needs to be grouped so that related I/O groups can be considered a unit.	I/O rack	128 input bits and 128 output bits **or** 8 input words and 8 output words **or** 8 I/O groups Each PLC-5 processor has a finite amount of racks it can support. For example, a PLC-5/30 can support 8 I/O racks. The processor always occupies one I/O rack for itself, rack 0 by default.

Figure 4.1 shows the relationship between an I/O terminal and its location in processor memory.

Figure 4.1
I/O Addressing as It Relates to an I/O Terminal



Notice how input and output image file addresses correspond to hardware.

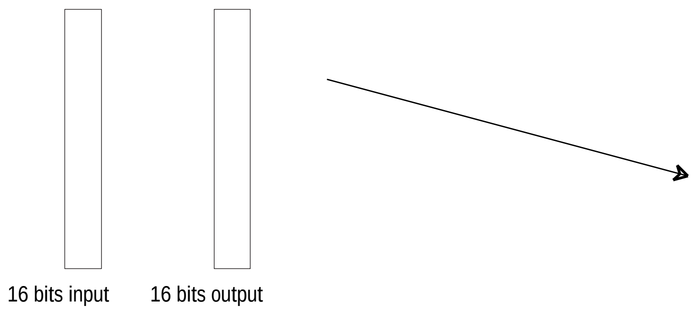
I/O image table is addressed octally.

Now that you are familiar with how processor memory is segmented to address a specific I/O terminal, the next section explains available addressing modes. These modes let you define the relationship between an I/O chassis slot and an I/O group (16 input bits and 16 output bits).

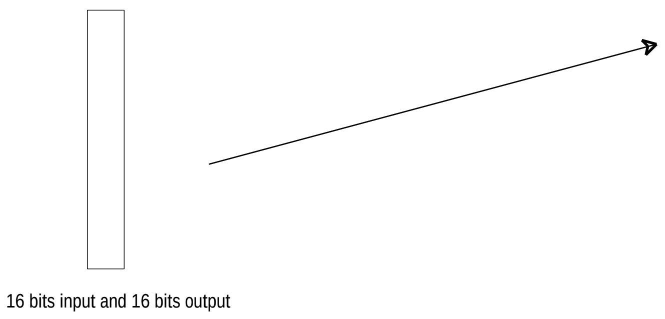
Choosing an Addressing Mode

For each I/O chassis in your system, you must define how many I/O chassis slots make up an I/O group (1 word each in the input image table and output image table); this choice is the chassis' addressing mode. Choose from among these available modes:

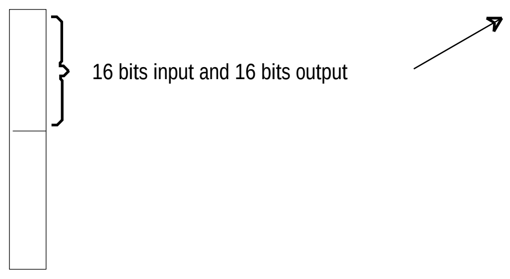
- **2-slot addressing**
  2 I/O chassis slots = 1 I/O group = 1 input image word and 1 output image word = 16 input bits and 16 output bits.

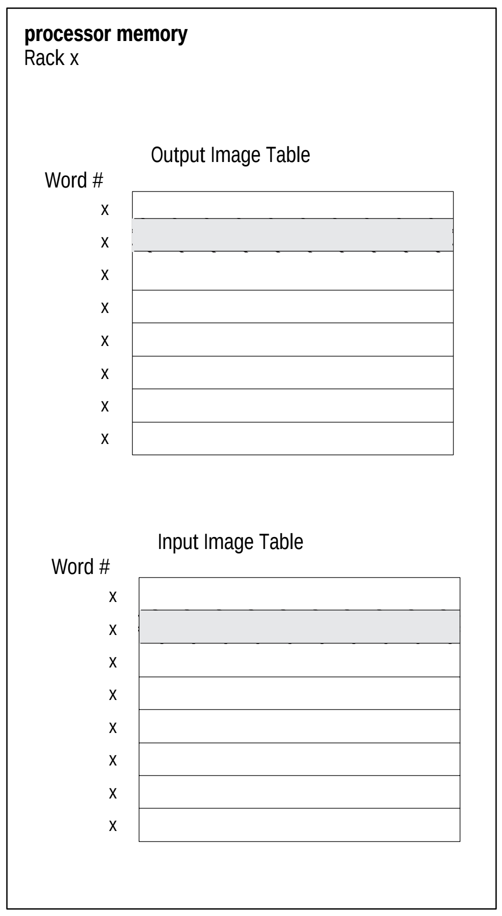


- **1-slot addressing**
  1 I/O chassis slot = 1 I/O group = 1 input image word and 1 output image word = 16 input bits and 16 output bits.



- **1/2-slot addressing**
  1/2 of an I/O chassis slot = 1 I/O group = 1 input image word and 1 output image word = 16 input bits and 16 output bits.

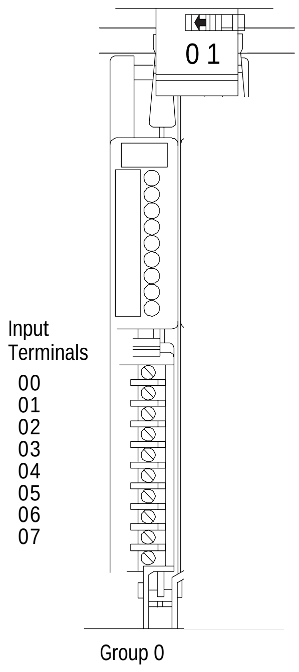




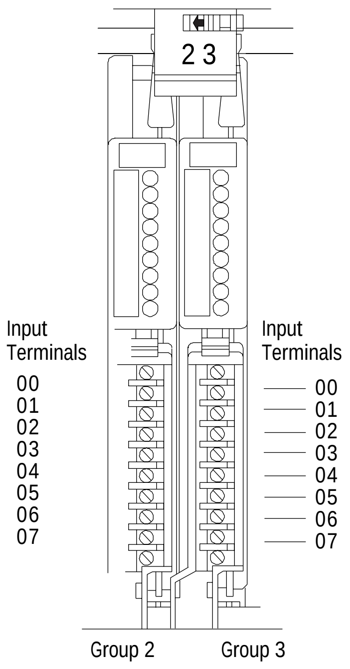
When you place your I/O modules in the I/O chassis slots, the module's density determines how quickly I/O groups form. For example, let's choose 1-slot addressing and see how 8-, 16-, and 32-point I/O modules fill processor memory.

18-and 16-point Example

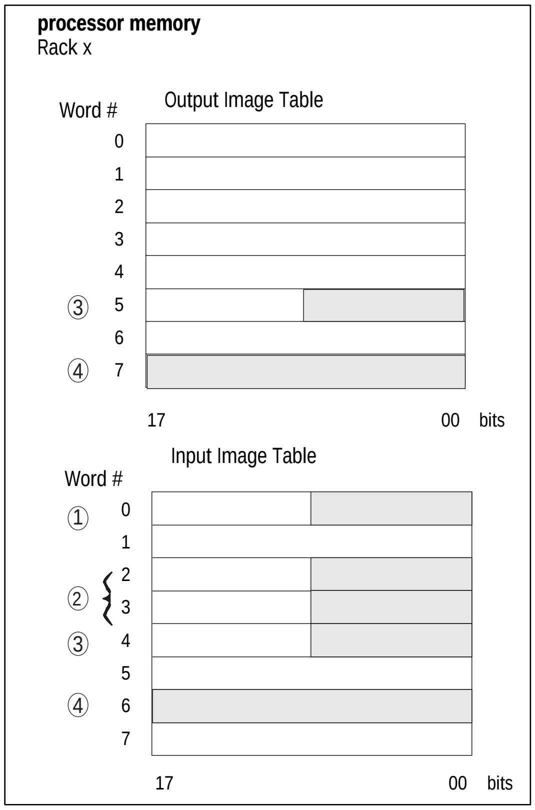
1-slot addressing (1 I/O chassis slot = 1 I/O group = 1 input image word and 1 output image word = 16 input bits and 16 output bits.)

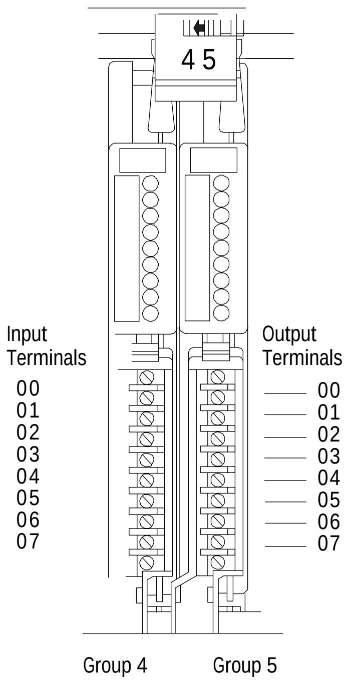


An 8-point I/O module occupies 8 bits in a word. See ①

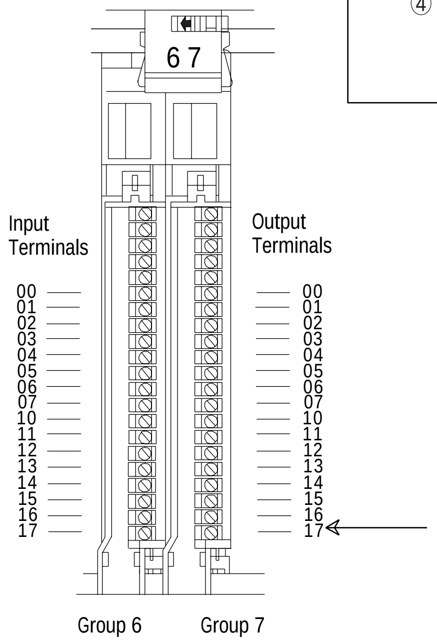


Two 8-point input modules occupy 8 bits of each group. See ②





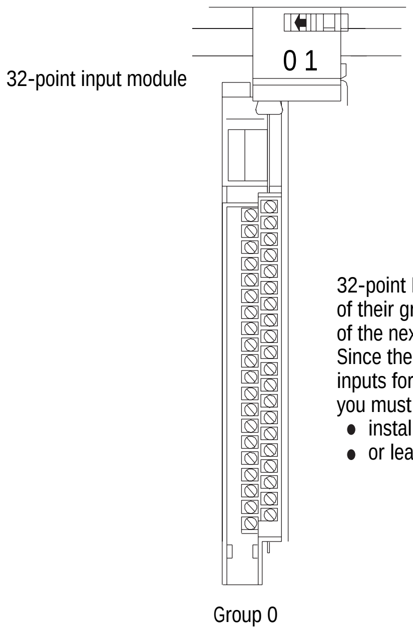
An 8-point input module in group 4 occupies the first eight bits of input word 4. The 8 point output module occupies the first 8-output bits in output word 5. See ③



16-point I/O modules occupy 16 bits, an entire word, in the image table. See ④

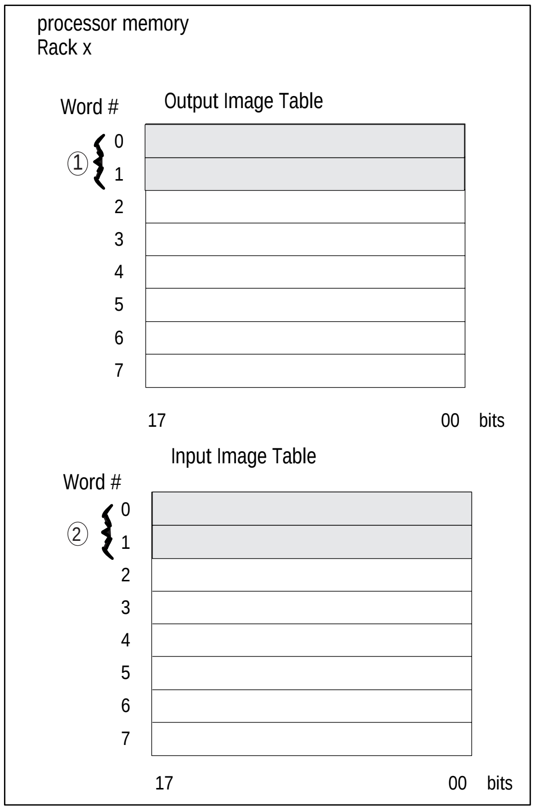
If you were to address the device attached to this output circuit in your control program, the address would be O:xx7/17.

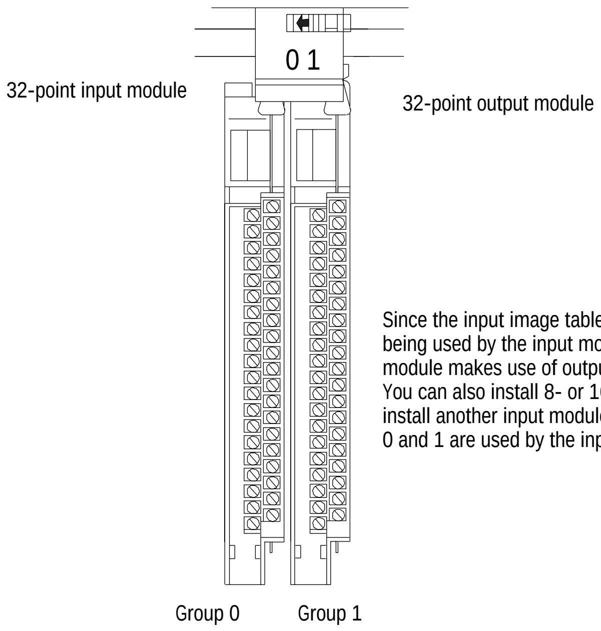
32-point Example



1-slot addressing (1 I/O chassis slot = 1 I/O group = 1 input image word and 1 output image word = 16 input bits and 16 output bits.)

32-point I/O modules use the entire word of their group and borrow the entire word of the next group. See ①.
Since the module is in group 0 and the inputs for group 0 and group 1 are used, you must:
- install an output module in group 1
- or leave the slot empty

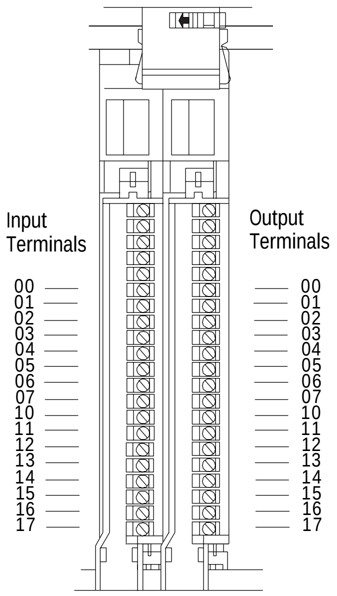




Since the input image table for group 1 is unavailable because it is being used by the input module of group 0, installing a 32-point output module makes use of output image table of group 0 and 1. See ②.
You can also install 8- or 16-point output modules. But you cannot install another input module since all the input image space for groups 0 and 1 are used by the input module of group 0.

When planning your system design, consider the densities of the I/O modules you are using and choose an addressing mode that most efficiently uses processor memory.

An example of efficient I/O image table use.

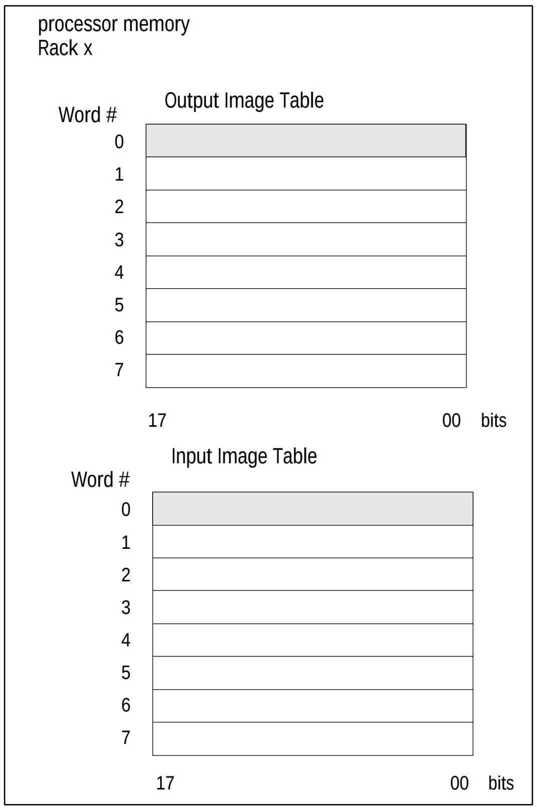


Group 0

16-point I/O modules occupy 16 bits, an entire word, in the image table.

Installing as a pair a 16-point input module and a 16-point output module efficiently uses the image table.

2-slot addressing (2 I/O chassis slot = 1 I/O group = 1 input image word and 1 output image word = 16 input bits and 16 output bits.)



Define the addressing mode for each I/O chassis by setting the chassis backplane switch assembly. For more information, see chapter 23.

Addressing Block-Transfer Modules

Block-transfer modules occupy 8 bits in the processor's I/O image table. Since all block-transfer modules are bidirectional, they cannot be used to complement either input or output modules.

To address:	Use the:
single slot modules	assigned I/O rack and group number of the slot in which the module resides and 0 for the module number When using 1/2-slot addressing, use the assigned rack number and the **lowest** group number and 0 for the module number.
double-slot modules	assigned rack number and the **lowest** group number and 0 for the module number

Addressing Summary

Use this table as a quick reference for addressing.

Addressing Mode:	Guidelines:
2-slot	• Two I/O module slots = 1 group • Each physical 2-slot I/O group corresponds to one word (16 bits) in the input image table and one word (16 bits) in the output image table • When you use 16-point I/O modules, you must install as a pair an input module and an output module in an I/O group; if you use an input module in slot 0, you must use an output module in slot 1 (or it must be empty). This configuration gives you the maximum use of I/O. • You cannot use a block-transfer module and a 16-point module in the same I/O group because block-transfer modules use 8 bits in both the input and output table. Therefore, 8 bits of the 16-point module would conflict with the block-transfer module. • You cannot use 32-point I/O modules. • Assign one I/O rack number to eight I/O groups.
1-slot	• One I/O module slot = 1 group • Each physical slot in the chassis corresponds to one word (16 bits) in the input image table and one word (16 bits) in the output image table • When you use 32-point I/O modules, you must install as a pair an input module and an output module in an even/odd pair of adjacent I/O group; if you use an input module in slot 0, you must use an output module in slot 1 (or it must be empty). This configuration gives you the maximum use of I/O. • Use any mix of 8- and 16-point I/O modules, block-transfer or intelligent modules in a single I/O chassis. Using 8-point modules results in fewer total I/O. • Assign one I/O rack number to eight I/O groups.
1/2-slot	• One half of an I/O module slot = 1 group • Each physical slot in the chassis corresponds to two words (32 bits) in the input image table and two words (32 bits) in the output image table • Use any mix of 8-, 16-, and 32-point I/O or block-transfer and intelligent modules. Using 8-point and 16-point I/O modules results in fewer total I/O. • With the processor-resident local rack set for 1/2-slot addressing, you cannot force the input bits for the upper word of any slot that is empty or that has an 8-point or 16-point I/O module. For example, if you have an 8-point or a 16-point I/O module in the first slot of your local rack (words 0 and 1 of the I/O image table, 1/2-slot addressing), you cannot force the input bits for word 1 (I:001) on or off. • Assign one I/O rack number to eight I/O groups.

Assigning Racks

The number of racks in a chassis depends on the chassis size and the addressing mode:

If using this chassis size:	2-slot addressing, results in:	1-slot addressing, results in:	1/2-slot addressing, results in:
4-slot	1/4 rack	1/2 rack	1 rack
8-slot	1/2 rack	1 rack	2 racks
12-slot	3/4 rack	1-1/2 racks	3 racks
16-slot	1 rack	2 racks	4 racks

Design Tip

When assigning rack numbers, use the following guidelines:

- One I/O rack number is eight I/O groups, regardless of the addressing mode that you select.
- You can assign from **one to four racks in your processor-resident local chassis (128 inputs and 128 outputs)** depending on the chassis size and addressing mode. You cannot split a processor-resident local I/O rack over two or more chassis or assign unused processor-resident local I/O groups to remote I/O racks.
- The default address of the processor-resident local rack is 0.
- You cannot split racks across remote I/O and extended-local I/O links. For example, if an 8-slot extended-local I/O chassis is configured as I/O groups 0-3 of I/O rack 2, an 8-slot remote I/O chassis cannot be configured as I/O groups 4-7 of I/O rack 2. For more information about addressing extended-local I/O, see chapter 8.
- When using complementary I/O addressing, treat complementary rack addresses individually when grouping racks; primary rack numbers are separate from complement rack numbers.
- If you are not using the autoconfiguration function, group together 1/4-racks and 1/2-racks of each logical rack on the configuration screen of your programming software. Do not intersperse these with other rack numbers. For example, your programming software has a screen with the following information for defining racks:

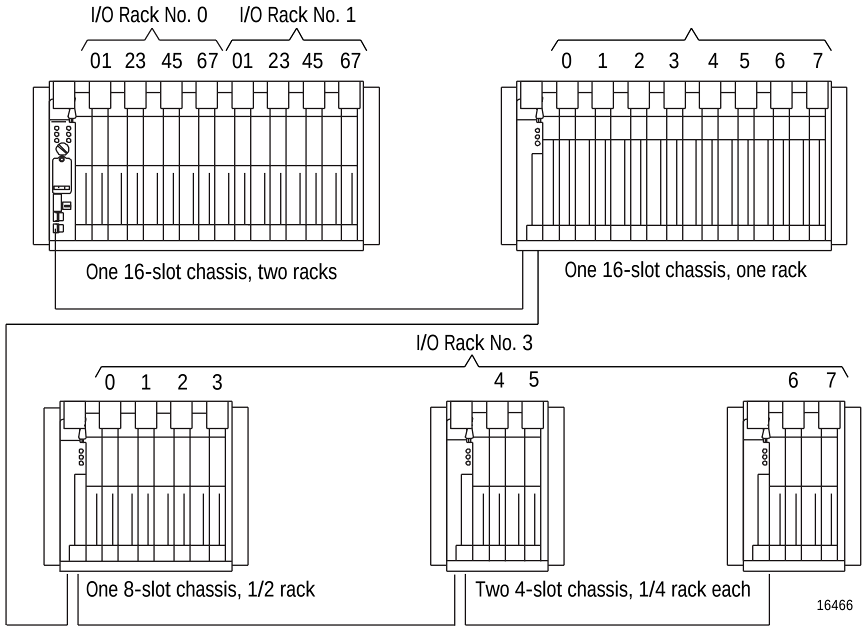
Group together 1/4 racks and 1/2 racks →

```
  Rack      Starting   Rack    Range       Fault   Inhibit   Reset   Retry
 Address     Group     Size

   1           0        1/4    010-011                I        0       0
   1           2        1/4    012-013                0        0       0
   1           4        1/4    014-015                0        0       0
   2           0        1/4    020-021                0        0       0
   2           2        1/4    022-023                0        0       0
   2           4        1/2    024-027                0        0       0
   3           0        1/4    030-031                0        0       0
  17           0        FULL   170-177                0        0       0
```

Design Tip

When assigning remote I/O rack numbers, use these guidelines:

- A single remote I/O scanner channel can support up to 32 devices but only 16 rack numbers. For more information, see chapter 6.
- Limit the number of remote I/O rack numbers to those that your PLC-5 processor can support.
- The PLC-5 processor and the 1771-ASB adapter module automatically allocate the next higher rack number(s) to the remaining I/O groups of the chassis. For example, if you select 1/2-slot addressing for your processor-resident local chassis and you are using a 16-slot (1771-A4B) chassis, the processor will address racks 0, 1, 2, and 3 in this chassis.
- You can assign a remote I/O rack to a fraction of a chassis, a single I/O chassis, or multiple I/O chassis:



I/O Rack No. 0
I/O Rack No. 1
01 23 45 67 01 23 45 67
0 1 2 3 4 5 6 7
One 16-slot chassis, two racks
One 16-slot chassis, one rack
I/O Rack No. 3
0 1 2 3
4 5
6 7
One 8-slot chassis, 1/2 rack
Two 4-slot chassis, 1/4 rack each
16466

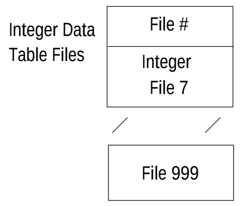
Understanding PLC-5 Processor Memory

PLC-5 memory is divided into two basic areas:

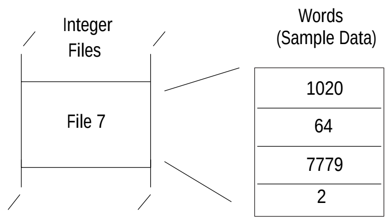
Storage areas	Description
Data	All of the data the processor examines or changes is stored in files in data storage areas of memory. These storage areas store: • Data received from input modules • Data to be sent to output modules; this data represents decisions made by the logic • Intermediate results made by the logic • Preloaded data such as presets and recipes • Control instructions • System status
Program Files	You create files for program logic, depending on the method you are using: ladder logic, sequential function charts, and/or structured text. These files contain the instructions to examine inputs and outputs and return results.

Understanding Data Storage (Data-Table Files)

The processor divides data storage into:

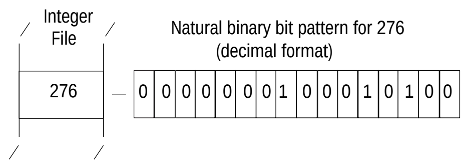


- **Types** that let you specify different formats and ranges to accommodate different types of data. For more information on the different types of data files, see Table 4.A on page 12.

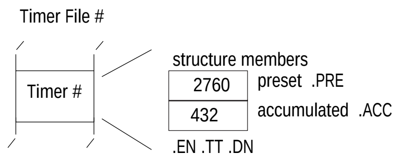


- You can create multiple **files** of a given type. Files let you group and organize logically related data. When you need to access data, you specify the file in which the data is stored.

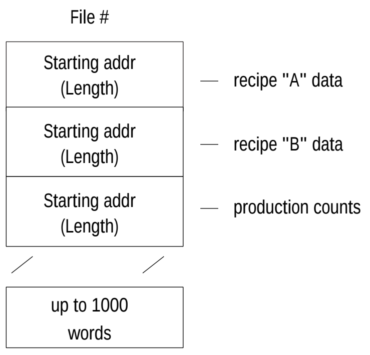
- Some types of files are made up of 16-bit **words**. Floating-point words are 32 bits. When you need to access this data, you specify it with a formatted address.



- Each word contains multiple **bits**. This is the smallest division of data. A bit contains a value of zero or one. When you need to access this data, you specify it with a formatted address.



- Some types of files are divided into **structures** used to control instruction elements. These structures are subdivided into **members** at the bit or word level. When you need to access this data, you specify it with a formatted address.



You can also organize data within files into **data blocks** to group and organize logically related data. When you need to access this data, you specify only the starting address within the file (and length) instead of each individual address.

When you organize data, group data by similar kind, such as:

- results of calculations
- batch recipes

Because of the structure of block-transfer instructions, you must group data such as:

- inputs from analog modules
- outputs to analog modules

Design Tip

You might also want to leave room for future expansion when grouping data. Do this by leaving gaps between:

- data blocks within a file
- groups of sequentially numbered files
- modules in an I/O chassis

Important: If you plan to edit your program online in Run mode, you must allocate unused data table files/elements and program files because you cannot create user memory while in run mode. Each unused file, however, uses 6 words of overhead memory for each data/program file you skip. Use care when leaving gaps.

Design Tip

Follow these guidelines when organizing data files:

- Group large amounts of related data into files.
- Address the data files from 3-999 as needed.
  (See Table 4.A on page page 12).
- Address the words needed in each data file consecutively from 0-999 (0-1999 for some data types in series E, revision D processors and later).
- Address the words of I/O image data according to how you configured your I/O:
  - 0-37 (octal) for PLC-5/11, -5/20, -5/20E
  - 0-77 (octal) for PLC-5/30
  - 0-177 (octal) for PLC-5/40, -5/40L, -5/40E
  - 0-277 (octal) for PLC-5/60, -5/60L, -5/80, -5/80E
- When addressing I/O image bits, address them 00-07 or 10-17 (octal).
- When organizing bit data, address the bits in each word 0-15 (decimal) for binary or integer files.

Each data table file allocates 6 words of memory when you create the file. This is in addition to any data stored in the file.

Addressing File Types

The following two tables show the available file types and the amount of memory used by each.

Table 4.A
Data Table File Types and Memory Usage for PLC-5 Processors Series E/Revision D and Later

File Type	File-Type Identifier	File Number	Maximum Size of File 16-bit words and structures				Memory Used in Overhead for each File (in 16-bit words)	Memory Used (in 16-bit words) per Word, Character, or Structure
			PLC-5/11, -5/20, -5/20E	PLC-5/30	PLC-5/40, -5/40E, -5/40L	PLC-5/60, -5/60L, -5/80, -5/80E		
Output image	O	0	32	64	128	192	6	1/word
Input image	I	1	32	64	128	192	6	1/word
Status	S	2	128	128	128	128	6	1/word
Bit (binary)	B	3[1]	2000 words				6	1/word
Timer	T	4[1]	6000 words/2000 structures				6	3/structure
Counter	C	5[1]	6000 words/2000 structures				6	3/structure
Control	R	6[1]	6000 words/2000 structures				6	3/structure
Integer	N	7[1]	2000 words				6	1/word
Floating-point	F	8[1]	4000 words/2000 structures				6	2/structure
ASCII	A	3-999	2000 words				6	1/2 per character
BCD	D	3-999	2000words				6	1/word
Block-transfer	BT	3-999	12000 words/2000 structures				6	6/structure
Message	MG	3-999	32760 words/585 structures[2]				6	56/structure
PID	PD	3-999	32718 words/399 structures[2]				6	82/structure
SFC status	SC	3-999	6000 words/2000 structures				6	3/structure
ASCII string	ST	3-999	32760 words/780 structures[2]				6	42/structure
Unused	--	9-999	6				6	0

1.This is the default file number and type. For this file type, you can assign any file number from 3 through 999.
2.The maximum size of a data table file is 32K words. The maximum size of the entire data table is 64K words

Table 4.B
Data Table File Types and Memory Usage for PLC-5 Processors Series E/Revision C and Earlier

File Type	File-Type Identifier	File Number	Maximum Size of File 16-bit words and structures				Memory Used in Overhead for each File (in 16-bit words)	Memory Used (in 16-bit words) per Word, Character, or Structure
			PLC-5/11, -5/20, -5/20E	PLC-5/30	PLC-5/40, -5/40E, -5/40L	PLC-5/60, -5/60L, -5/80, -5/80E		
Output image	O	0	32	64	128	192	6	1/word
Input image	I	1	32	64	128	192	6	1/word
Status	S	2	128	128	128	128	6	1/word
Bit (binary)	B	3[1]	1000 words				6	1/word
Timer	T	4[1]	3000 words/1000 structures				6	3/structure
Counter	C	5[1]	3000 words/1000 structures				6	3/structure
Control	R	6[1]	3000 words/1000 structures				6	3/structure
Integer	N	7[1]	1000 words				6	1/word
Floating-point	F	8[1]	2000 words/1000 structures				6	2/structure
ASCII	A	3-999	1000 words				6	1/2 per character
BCD	D	3-999	1000words				6	1/word
Block-transfer	BT	3-999	6000 words/1000 structures				6	6/structure
Message	MG	3-999	32760 words/585 structures[2]				6	56/structure
PID	PD	3-999	32718 words/399 structures[2]				6	82/structure
SFC status	SC	3-999	3000 words/1000 structures				6	3/structure
ASCII string	ST	3-999	32760 words/780 structures[2]				6	42/structure
Unused	--	9-999	6				6	0

1.This is the default file number and type. For this file type, you can assign any file number from 3 through 999.
2.The maximum size of a data table file is 32K words. The maximum size of the entire data table is 64K words

Table 1.C
Valid Data Types/Values Are:

This data type/value:	Accepts any:
Immediate (program constant)	Value between -32768 and 32767 (Constants greater than 1024 use 2 storage words of memory; floating point constants use 3 words of memory.)
Integer	Integer data type: integer, timer, counter, status, bit, input, output, ASCII, BCD, control (e.g., N7:0, C4:0, etc.)
Float	Floating point data type (valid range is $\pm$ 1.175494e^{-38} to $\pm$3.402823e^{+38}) with 7-digit precision
Block	Block-transfer data type (e.g., BT14:0) or integer data type (e.g., N7:0)
Message	Message data type (e.g., MG15:0) or integer data type (e.g., N7:0)
PID	PID data type (e.g., PD16:0) or integer data type (e.g., N7:0)
String	String data type (e.g., ST12:0)
SFC status	SFC status data type (e.g., SC17:0)

Understanding Program-File Storage

Create program files based on the programming method you are using. This table lists the number of words used by each type of program file:

Program File	Number of Words Used
Ladder	6/file + 1/word
SFC	6/file
Structured Text	6/file + 1/word

The more program files that you create, the longer the processor takes to perform certain tasks, e.g., going to run mode, performing online editing, saving a program. Also, certain instructions (JMP, LBL, FOR, and NXT) have longer execution times in higher program file numbers.

Series E PLC-5 processors support 2000 program files to allow for more SFC steps in your program. SFC step/transition program files are typically shorter in length. This enhancement will effectively double your SFC step/transition size.

Each program file you create is allocated 6 words of memory. This memory is in addition to any programming within the file. If you create the maximum program file number 1999, this allocates 12,000 words of memory to the program files, which reduces the amount of processor memory left for programming.

Addressing

Valid formats for addressing data files are:

If you want to access:	Use this addressing format:	And see page:
Input or output bit in the I/O image table	I/O image address	4-15
Bit, word, sub-member, data block, file, or I/O image bit	Logical address	4-16
A component within a logical address by substituting the value in another address	Indirect address	4-18
An address offset by some number of elements	Indexed address	4-19
A substitute name for an address	Symbolic address	4-20

MORE

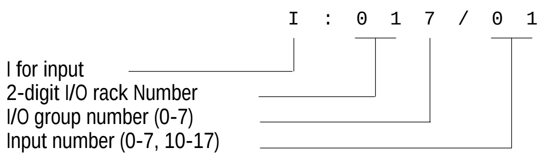
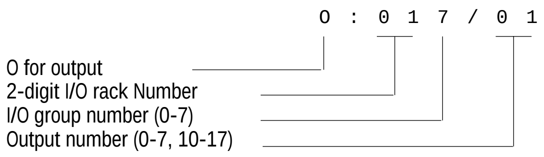
For more information about entering addresses, see the documentation for your programming software.

Specifying I/O Image Addresses

The I/O image address corresponds to the physical location of the I/O circuit in the I/O chassis:

a	I/O address identifier	I = input device
		O = output device	

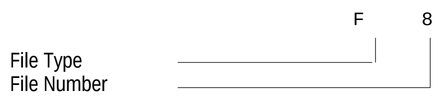
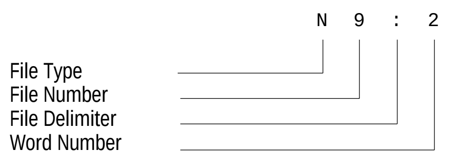
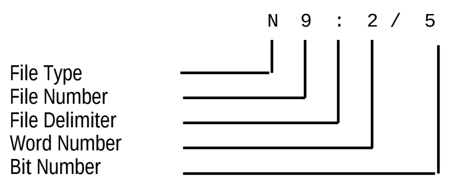
bb	I/O Rack number	PLC-5/11, -5/20, -5/20E PLC-5/30 PLC-5/40, -5/40L, -5/40E PLC-5/60, -5/60L, -5/80, -5/80E	00-03 (octal) 00-07 (octal) 00-17 (octal) 00-27 (octal)
c	I/O Group number	0-7 (octal)	
dd	Terminal (bit) number	00-17 (octal)	

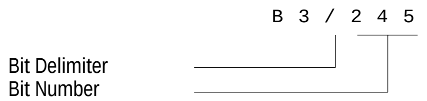
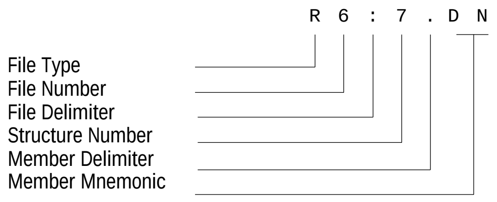
To specify this address:	Example:
Input Image Bit	
Output Image Bit	

Specifying Logical Addresses

The format of a logical address corresponds directly to the location in data storage: `# X F : e . s / b`

Where:	**Is the:**				
#	File address. Omit for bit, word, and structure addresses (also indicates indexed addressing, see next page)				
X	File type:	B—binary	N—integer	T—timer	MG—message
		C—counter	O—output	A—ASCII	PD—PID
		F—floating point	R—control	D—BCD	SC—SFC status
		I—input	S—status	BT—block-transfer	ST—ASCII string
F	File number:	0—output			
		1—input			
		2—status			
		3-999—any other type			
:	colon or semicolon delimiter separates file and structure/word numbers				
e	Structure/word number:	0-277	octal for input/output files		
	up to:	0-127	decimal for the status file		
		0-999	for all the file types except MG, PD, and ST files		
.	Period delimiter is used only with structure-member mnemonics in counter, timer and control files				
s	Structure/member mnemonic is used only with timer, counter, control, BT, MG, PD, SC, and ST files				
/	Bit delimiter separates bit number				
b	Bit number:	00-07 or 10-17 for input/output files			
		00-15 for all other files			
		00-15,999 for binary files when using direct bit address			

To specify the address of a:	**Use these parameters:**
File	
Word within an integer file	 N 9 : 2 File Type File Number File Delimiter Word Number
Bit within an integer file	 N 9 : 2 / 5 File Type File Number File Delimiter Word Number Bit Number

To specify the address of a:	Use these parameters:
Bit within a binary file	 Binary files are bit stream continuous files, and therefore you can address them in two ways: by word and bit, or by bit alone.
Bit within a structure file	 R 6 : 7 . D N File Type File Number File Delimiter Structure Number Member Delimiter Member Mnemonic

You can also use mnemonics to address members at the word or bit level. The available mnemonics depend on the type of data (timer, counter, or control) and the program instruction. For example:

Instruction Type	Word Level		Example	Bit Level		Example
Timer TON, TOF, RTO	preset accumulated	.PRE .ACC	T4:1.PRE	enable timing done	.EN .TT .DN	T4:0.EN

For information about the mnemonics for a specific instruction, see chapter 22, or the *PLC-5 Programming Controller Instruction Set Reference*, publication 1785-6.1.

Specifying Indirect Addresses

ATTENTION: When using indirect addressing, make sure that the indirect address points to a valid data file or element. During Run mode, if ladder execution encounters an invalid or out-of-range indirect address, a run-time error occurs and the processor halts.

The processor uses the value from the pointer address to form the indirect address. You can use ladder logic to change the value stored at that substitute address.

When you specify indirect addresses, follow these guidelines:

- You can indirectly address the file number, word number, or bit number.
- The substitute address must be one of the following types: N, T, C, R, B, I, O, or S. Any T, C, or R address must be a word-length sub-member address, such as T4:0.ACC.
- Enter the pointer address in brackets [ ].

Example	Variable	Explanation
N[N7:0]:0	File number	The file number is stored in integer address N7:0.
N7:[C5:7.ACC]	Structure number	The word number is the accumulated value of counter 7 in file 5.
B3/[I:017]	Bit number	The bit number is stored in input word 17.
N[N7:0]:[N9:1]	File and word number	The file number is stored in integer address N7:0 and the word number in integer address N9:1.

ATTENTION: If you indirectly address the input or output image table, the value you specify in the integer file that you use for the indirection (the pointer) is converted to octal when executed by the instruction.

For example, if you enter O:[N7:0] and N7:0 contains the value 10, the value at N7:0 is converted to octal and the resulting address is O:012, not O:010.

To monitor for invalid indirect addresses, condition the rung with the indirect address with a limit test of the indirect address to ensure that the address stays within the intended range. This is especially advisable if the PLC-5 processor has no control over the indirect address, such as the value is determined by values from an I/O module or a peer processor.

Specifying Indexed Addresses

The processor starts operation at the base address plus the offset. Store the offset value in the offset word in the processor's status file. You can manipulate the offset word in your ladder logic.

The indexed address symbol is the # character. Place the # character immediately before the file-type identifier in a logical address. Enter the offset value in the status file S:24. All indexed instructions use S:24 to store an offset.

When you specify indexed addresses, follow these guidelines:

- Make sure the index value (positive or negative) does not cause the indexed address to exceed the file-type boundary.

⚠ **ATTENTION:** The processor does not check indexed addresses to make sure that the addresses do not cross data-table file boundaries (e.g., N7 to F8). You could even modify the status file, which is physically the last data table file. But if the indexed address exceeds the data-table area of memory, the processor initiates a run-time error and sets a major fault.

- When an instruction uses more than two indexed addresses, the processor uses the same index value for each indexed address.
- Set the offset word to the index value you want immediately before enabling an instruction that uses an indexed address.

⚠ **ATTENTION:** Instructions with a # sign in an address manipulate the offset value stored at S:24. Make sure you monitor or load the offset value you want prior to using an indexed address. Otherwise unpredictable machine operation could occur with possible damage to equipment and/or injury to personnel.

The following MVM example uses an indexed address in the source and destination addresses. If the offset value is 10 (stored in S:24), the processor manipulates the data stored at the base address plus the offset.

```
MVM
MASKED MOVE
Source         #N7:10
Mask         00110011
Destination    #N11:5
```

Value	Base Address	Offset Address
Source	N7:10	N7:20
Destination	N11:5	N11:15

Specifying Symbolic Addresses

When you specify symbolic address, follow these guidelines:

- Start the name with an alphabetic character (not a number).
- The symbol must begin with a letter and can contain as many as 10 of the following characters:
  - A-Z (upper and lower case)
  - 0-9
  - underscore (_)
- You can substitute a symbolic address for word or bit addresses.

Important: Symbols are a feature of the programming software (not the processor) and are stored in a database on the hard disk of the personal computer you are using. If you use a terminal other than the one on which you defined the symbols, you will not have access to the symbol database.

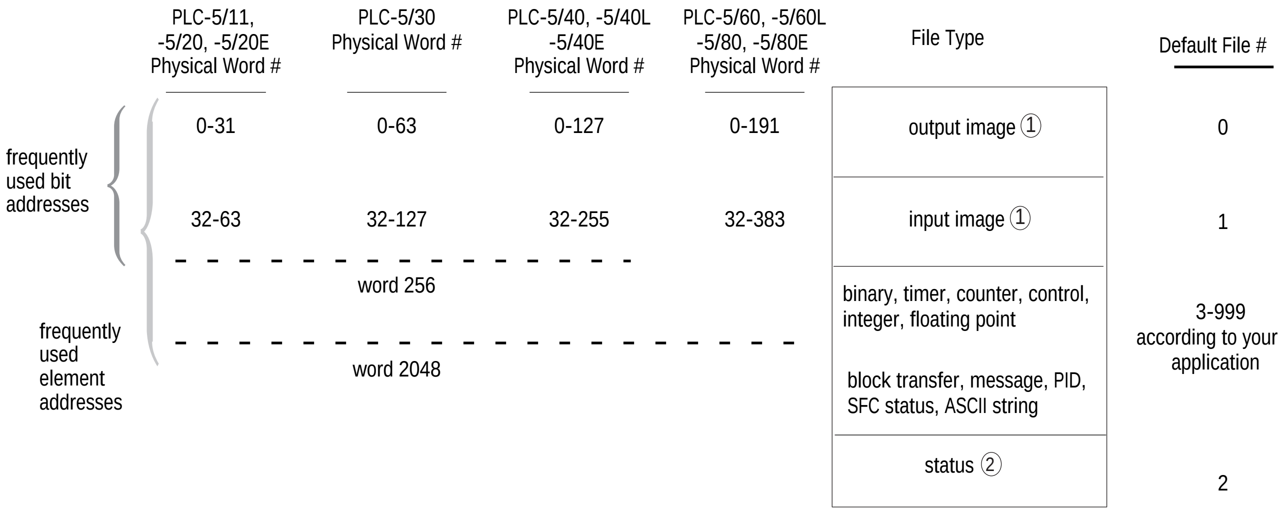
Example	Logical Address	Symbolic Address
Input image (bit)	I:015/00	LS1
	I:015/03	AUTO1
	I:015/06	SW1
Output image (bit)	O:013/00	M1
	O:013/02	CL1
	O:013/04	L1
Word	F10:0	Calc_1
	F10:1	Calc_2

Design Tip

Optimizing Instruction Execution Time and Processor Memory

For the best instruction-execution performance, store your most frequently used addresses as follows:

- Address bit instructions between the end of the input image file and physical word 256. Because, bit addresses located in words greater than 256 require one extra word in the processor's memory for storage and execute 0.16ms slower than bit addresses stored in words 0-255.
- Address element instructions between the end of the input image and physical word 2048. Because, addresses stored in words greater than 2048 require more words in the processor's memory for storage.



PLC-5/11, -5/20, -5/20E Physical Word #
PLC-5/30 Physical Word #
PLC-5/40, -5/40L -5/40E Physical Word #
PLC-5/60, -5/60L -5/80, -5/80E Physical Word #
File Type
Default File #
frequently used bit addresses
frequently used element addresses
0-31
0-63
0-127
0-191
output image ①
0
32-63
32-127
32-255
32-383
input image ①
1
word 256
binary, timer, counter, control, integer, floating point
3-999 according to your application
word 2048
block transfer, message, PID, SFC status, ASCII string
status ②
2

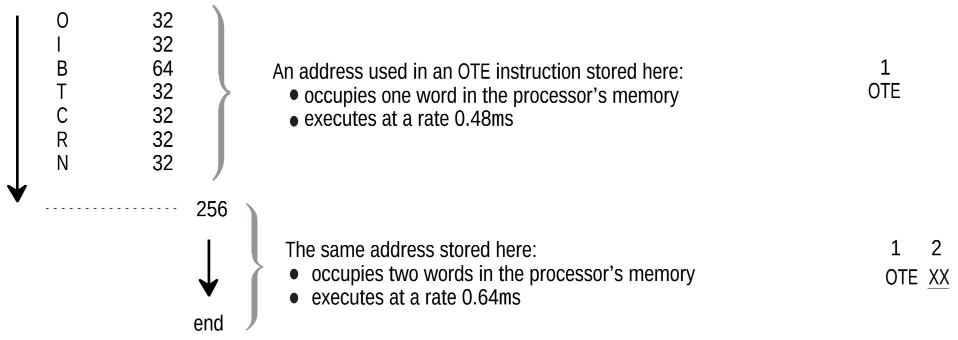
① The minimum size of the file is 32 words.

② The status file is always the last physical file in the data table.

The following examples illustrate these concepts:

Bit address example

If your data table map looks like this:



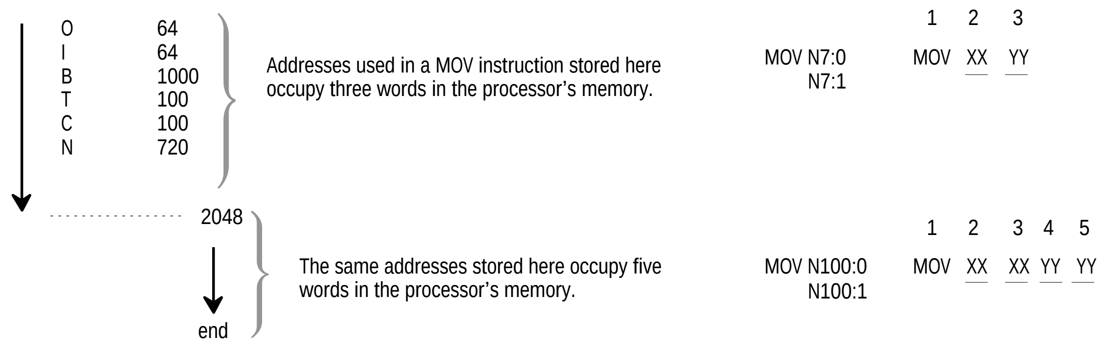
An address used in an OTE instruction stored here:
- occupies one word in the processor's memory
- executes at a rate 0.48ms

The same address stored here:
- occupies two words in the processor's memory
- executes at a rate 0.64ms

This example uses the instruction timing and memory usage tables in chapter 22. Consult these tables for information about other instructions.

Element address example

Your data table map looks like this:


O 64
I 64
B 1000
T 100
C 100
N 720
Addresses used in a MOV instruction stored here occupy three words in the processor's memory.
MOV N7:0
N7:1
1 2 3
MOV XX YY
2048
end
The same addresses stored here occupy five words in the processor's memory.
MOV N100:0
N100:1
1 2 3 4 5
MOV XX XX YY YY

This example uses the instruction timing and memory usage tables in chapter 22. Consult these tables for information about other instructions.

Effectively Using I/O Memory

The PLC-5 processor automatically allocates both an input and output memory location to each I/O location. I/O modules generally only use either the inputs or the outputs. To more effectively use I/O memory, you can use these methods of placing I/O modules.

Use:	Application:
2-slot	Install 16-point I/O modules as an input module and output module pair in an I/O group. For example, if you place an input module in slot 0, place an output module in slot 1.
1-slot	Install 32-point I/O modules as an input module and an output module pair in an I/O group. For example, if you place an input module in slot 0, place an output module in slot 1.
complementary I/O chassis	You configure complementary chassis with a primary and complement chassis pair. You complement the I/O modules I/O group for I/O group between the two chassis. The I/O modules in the complementary chassis perform the opposite function of the corresponding modules in the primary chassis. By designating a PLC-5 scanner channel as complementary, you can complement racks 1-7. A channel configured for complementary I/O can't scan racks greater than 7. Those PLC-5 processors that can address rack numbers greater than 7 can address these racks on another scanner channel which has not been configured as complementary. The remote I/O link device (such as a1771-ASB adapter) must also be configured for complimentary. For more information see the *PLC-5 Reference Guide: Configuring Complementary I/O for PLC-5 Processors*, publication 1785-6.8.3

Communicating with Processor-Resident I/O

Using This Chapter

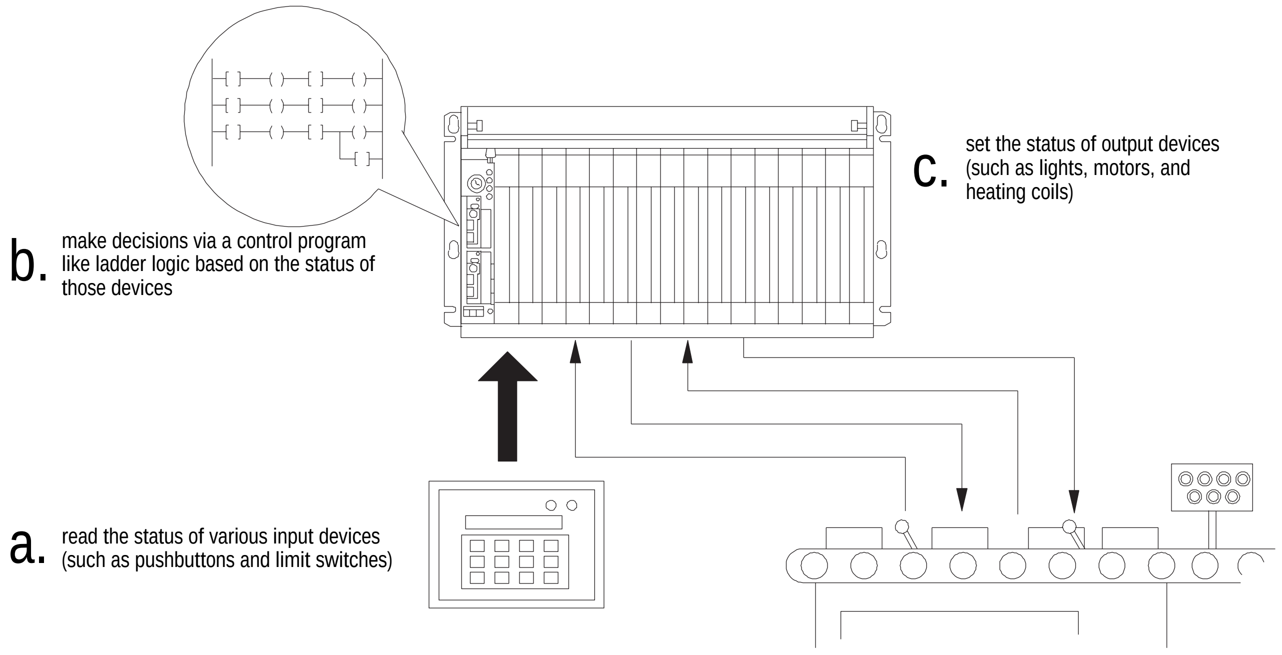
This chapter explains how to configure the processor to communicate with resident I/O:

1. Set the I/O chassis switch for the addressing mode.
2. Set the rack address.

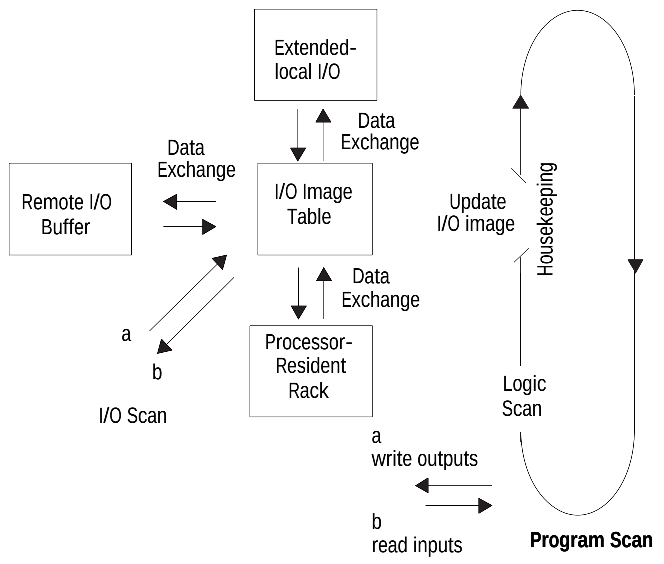
   The rack address defaults to 0. If you want to change the rack address to 1, set bit S:26/2.

Introduction to PLC-5 Processor Scanning

The basic function of a programmable-controller system is to:


c. set the status of output devices (such as lights, motors, and heating coils)
b. make decisions via a control program like ladder logic based on the status of those devices
a. read the status of various input devices (such as pushbuttons and limit switches)
20221

The processor performs two primary operations:
- program scanning where
  - logic is executed
  - housekeeping is performed
- I/O scanning - where input data is read and output levels are set

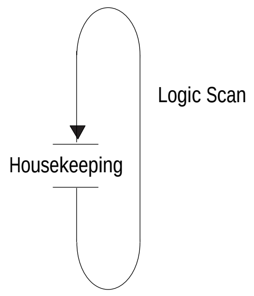


During **logic scan**, inputs are read from and outputs are written to the I/O image table.

During **housekeeping**, data exchange occurs between the I/O image table and the remote I/O buffer, extended local I/O, and processor-resident rack.

Program Scanning

The program scan is the time it takes the processor to execute the logic program once, perform housekeeping tasks, and then start executing logic again.



The processor continually performs a logic program scan and housekeeping. Housekeeping activities for PLC-5 processors include:

- performing processor internal checks
- updating the input image table with:
  - processor-resident input module data
  - remote input module data as contained in the remote I/O buffer
  - extended local I/O input module data
- sending output image table data to:
  - processor-resident output modules
  - remote I/O buffer
  - extended local I/O output modules

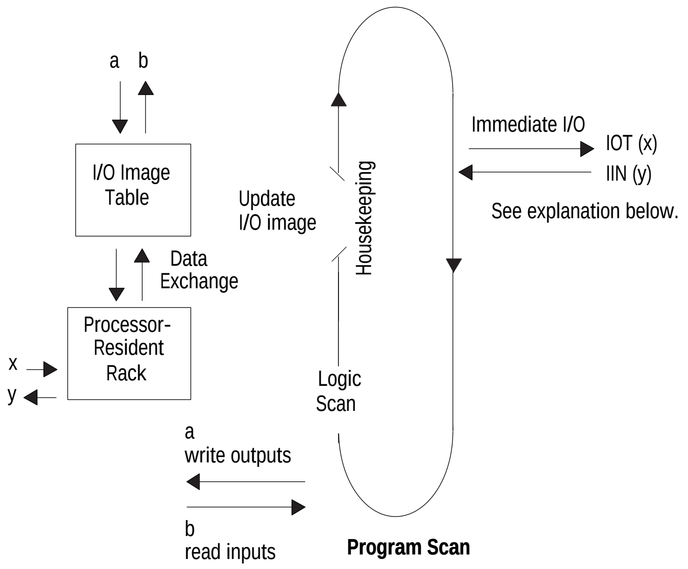
Transferring Data to Processor-Resident I/O

A PLC-5 processor transfers discrete and block-transfer data with processor-resident I/O.

Transferring Discrete Data to Processor-Resident I/O

The processor scans processor-resident local I/O synchronously and sequentially to the program scan.

The processor-resident rack exchanges discrete I/O information with the I/O image table during housekeeping.



Transferring Immediate I/O Requests

The processor responds to immediate input (IIN) and immediate output (IOT) requests during the logic scan. The logic scan is suspended at the request for immediate input/output data. The logic scan resumes after obtaining the data and fulfilling the request.

IIN data transfers directly to and IOT data transfers directly from I/O modules in processor-resident I/O and extended-local I/O chassis. With remote I/O, only the remote I/O buffer is updated. For more information, see chapter 22.

Design Tip

When you place I/O modules, do not place a block-transfer module next to or in the same module group as an input module that you plan to use for immediate I/O. Place the modules in non-adjacent slots. Placing input modules for immediate I/O next to block-transfer modules can result in a -5 Block-Transfer Read error.

If your application cannot support this configuration, condition the immediate I/O instructions with the control bits of the adjacent block-transfer module. This technique helps make certain that an adjacent block-transfer module is not performing a block-transfer while an immediate I/O instruction is executing in its adjacent input module.

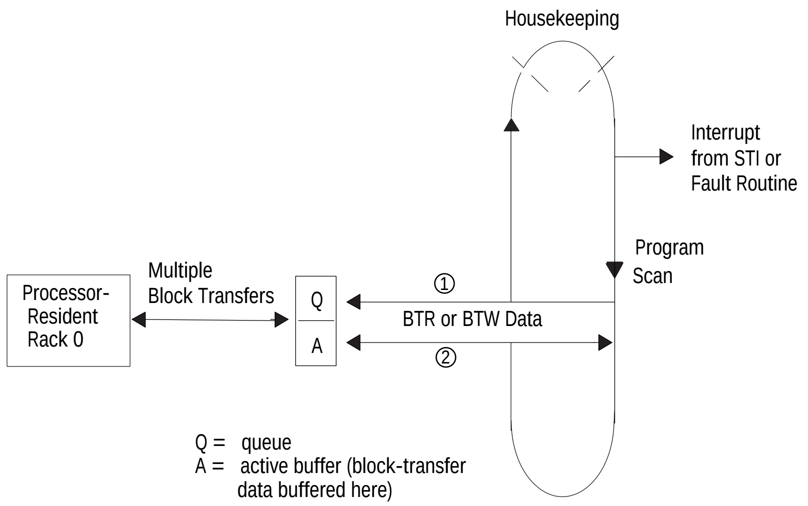
Transferring Block-Transfer Data to Processor-Resident I/O

The processor performs block-transfers at the same time as it scans the program.

Block-transfers to processor-resident local I/O follow these procedures:

- Block-transfer requests are queued for the addressed processor-resident local I/O rack.
- The active buffer continuously handles all block-transfer modules whose block-transfer instructions were enabled in the program scan via the queue scan in the order the requests were queued.
- Block-transfers of I/O data can finish and the done bit can be set anytime during the program scan.

The processor runs all enabled block-transfers of I/O data to processor-resident I/O continuously as each block-transfer request enters the active buffer.


Housekeeping
Interrupt from STI or Fault Routine
Program Scan
Processor-Resident Rack 0
Multiple Block Transfers
Q
A
①
BTR or BTW Data
②
Q = queue
A = active buffer (block-transfer data buffered here)

Configuring the System for Processor-Resident I/O

To configure the system for processor-resident local I/O, you need to set the I/O chassis switch to indicate the rack-addressing mode. The addressing mode determines the number of processor-resident rack numbers used based on the number of slots in the chassis. For more information on addressing modes, see chapter 4. To set the I/O chassis switch, see chapter 23.

The processor-resident rack address defaults to rack 0. If needed, you can set it for rack 1 by setting user control bit 2 (S26:2) on the processor configuration screen in your programming software. If you select rack 1 as the processor-resident rack, rack 0 becomes unavailable for your system.

Communicating with Remote I/O

Using This Chapter

This chapter explains how to configure the processor to communicate with remote I/O:

1. Select which channel to configure as a scanner.

2. Define the I/O status file.

Use a unique, unused integer file. You must define an I/O status file if you want to autoconfigure your system.

3. Define a diagnostic file.

Use a unique, unused integer file.

4. Define the scan list.

Selecting Devices That You Can Connect

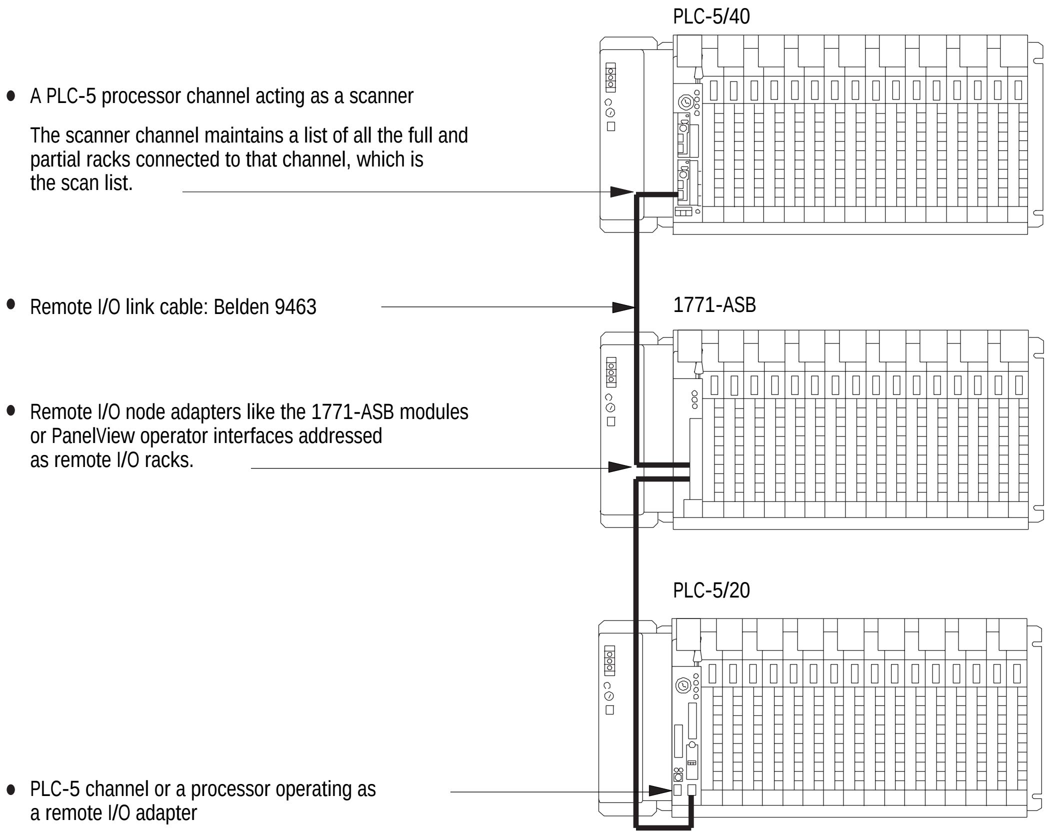
The following table lists some of the devices you can use on a remote I/O link:

Category:	**Product:**	**Catalog Number:**
Other Processors (in adapter mode)	enhanced PLC-5 processors	1785-L*xx*B
	Ethernet PLC-5 processors	1785-L*xx*E
	ControlNet PLC-5 processor	1785-L*xx*C
	VMEbus PLC-5 processors	1785-V*xx*B
	extended-local PLC-5 processors	1785-L*xx*L
	classic PLC-5 processors	1785-LT*x*
Other Processors (in adapter mode)	Direct Communication Module for SLC Processors	1747-DCM
To Remote I/O	SLC 500 Remote I/O Adapter Module	1747-ASB
	1791 Block I/O	1791 series
	Remote I/O Adapter Module	1771-ASB
	1-Slot I/O Chassis with Integral Power Supply and Adapter	1771-AM1
	2-Slot I/O Chassis with Integral Power Supply and Adapter	1771-AM2
	Direct Communication Module	1771-DCM
Operator Interfaces	DL40 Dataliner	2706-*xxxx*
	RediPANEL	2705-*xxx*
	PanelView Terminal	2711-*xxx*
Drives	Remote I/O Adapter for 1336 AC Industrial Drives	1336-RIO
	Remote I/O Adapter for 1395 AC Industrial Drives	1395-NA

Introduction to Remote I/O

A remote I/O system lets you control I/O that is not within the processor's chassis. A PLC-5 processor channel, in scanner mode, transfers discrete and block-transfer data with remote I/O devices.

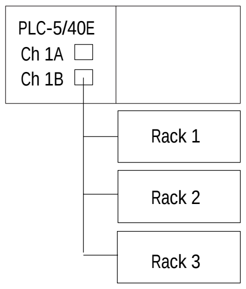
An example remote I/O system looks like this:



- A PLC-5 processor channel acting as a scanner

  The scanner channel maintains a list of all the full and partial racks connected to that channel, which is the scan list.
- Remote I/O link cable: Belden 9463
- Remote I/O node adapters like the 1771-ASB modules or PanelView operator interfaces addressed as remote I/O racks.
- PLC-5 channel or a processor operating as a remote I/O adapter

The remote I/O scanner channel keeps a list of all of the devices connected to each remote I/O link called a scan list. An example channel scan list looks like this:



Ch 1B Scan List

Rack AddresS	Starting Group	Rack Size	Range
1	0	Full	010-017
2	0	1/2	020-023
3	0	Full	030-037

In this example, channel 1B continually scans the three racks in its scan list and places the data in the remote I/O buffer in the processor. The processor updates its own buffer and the I/O image table. During housekeeping, the two buffers are updated by exchanging the input and output data with each other.

For more information on scan lists, see page 6-9.

.

Follow these steps for setting up a remote I/O system:

Step:		See:
1.	configure the remote I/O adapter devices	the device's user manual
2.	layout and connect the remote I/O link cable	• page 6-4 for design • chapter 3 for cable routing information • your processor's installation information (For enhanced PLC-5 processors, see publication 1785-10.4; for Ethernet PLC-5 processors publication 1785-10.5)
3.	configure the scanner channel	page 6-6

Designing a Remote I/O Link

Designing a remote I/O link requires applying:

- remote I/O link design guidelines
- cable design guidelines

Design Tip

Link Design Guidelines

Keep these rules in mind as you design remote I/O links:

- All devices connected to a remote I/O link must communicate using the same communication rate, either 57.6, 115.2, or 230.4 kbps. Choose a rate that all devices support.
- Two or more channels of the same processor operating in scanner mode cannot scan the same partial or full rack address. Assign unique partial and full racks to each channel used in remote I/O scanner mode.
- You can split rack addresses between scanner channels; however, issues arise when performing block-data transfer. See page 6-15.
- A scan list can have a maximum of 16 rack numbers or a maximum of 32 physical devices connected to it using 82-Ω termination resistors. See page 6-9 for more information about scan lists.

Design Tip

Cable Design Guidelines

Specify 1770-CD (Belden 9463) cable. Connect a remote I/O network using a daisy chain or trunk line/drop line configuration.

Trunk line/drop line considerations:

When using a trunk line/drop line configuration, use 1770-SC station connectors and follow these cable-length guidelines:

- trunk line-cable length*depends on the communication rate of the link
- drop-cable length*30.4 m (100 cable-ft)

For more information about designing trunk line/drop line configurations, see the Data Highway/Data Highway Plus/Data Highway II/Data Highway-485 Cable Installation Manual, publication 1770-6.2.2.

Verify that your system's design plans specify cable lengths within allowable measurements.

Important: The maximum cable length for remote I/O depends on the transmission rate. Configure all devices on a remote I/O link to communicate at the same transmission rate.

For daisy chain configurations, use this table to determine the total cable length you can use.

Table 6.A
Choose the correct cable length

A remote I/O link using this communication rate:	Cannot exceed this cable length:
57.6 kbps	3,048 m (10,000 ft)
115.2 kbps	1,524 m (5,000 ft)
230.4 kbps	762 m (2,500 ft)

Important: If you select the baud rate as 230.4 kbps, and you are using the serial port or a PLC-5 coprocessor, use channel 2 for better overall system performance.

For proper operation, terminate **both** ends of a remote I/O link by using the external resistors shipped with the programmable controller. Selecting either a 150Ω or 82Ω terminator determines how many devices you can connect on a single remote I/O link.

If your remote I/O link:	Use this resistor rating:	The maximum number of ***physical*** devices you can connect on the link:	The maximum number of racks you can scan on the link:
operates at 230.4 kbps	82Ω	32	16
operates at 57.6 kbps or 115.2 kbps **and** no devices listed in Table 6.B are on the link			
contains any device listed in Table 6.B	150Ω	16	16
operates at 57.6 kbps or 115.2 kbps, and you do not require the link to support more than 16 physical devices.			

Table 6.B
I/O Link Devices that Require 150Ω Termination Resistors

Device Type:	Catalog Number:	Series:
Scanners	1771-SN 1772-SD, -SD2 1775-SR 1775-S4A, -S4B 6008-SQH1, -SQH2	All
Adapters	1771-AS	
	1771-ASB	A
	1771-DCM	All
Miscellaneous	1771-AF	

Configuring a Processor Channel as a Scanner

Use this table to help you determine the processor channels you can configure as a remote I/O scanner:

Processor:		Channels that support remote I/O scanner:
PLC-5/11		1A
PLC-5/20	PLC-5/20E	1B
PLC-5/30 PLC-5/40L PLC-5/60L	PLC-5/40E PLC-5/80E	1A, 1B
PLC-5/40 PLC-5/60 PLC-5/80		1A, 1B, 2A, 2B

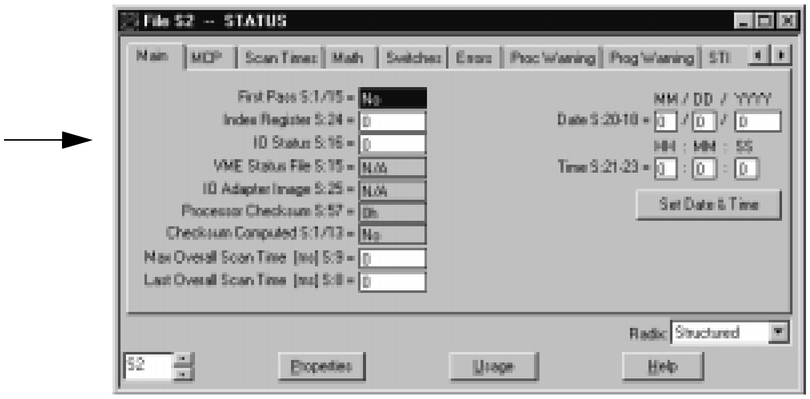
To configure a processor channel as a scanner, you:

- define an I/O status file, which stores information about the racks connected to the processor, by using the processor configuration screen in your programming software
- specify the scanner's communication rate and diagnostic file and define a scan list by using the scanner mode channel configuration screen in your programming software

Define an I/O Status File

The I/O status file stores data for the processor's I/O rack configuration tables. The I/O status from each remote I/O rack requires two words. These two words store the reset, present, inhibit, and fault bits for each rack.

To define an I/O status file, enter an unused integer file number (9-255) in the I/O status file field (S:16) of the processor configuration screen. If you do not want to use I/O rack configuration tables, enter 0. However, if you want to use the autoconfiguration option to create your scan list, you must define an I/O status file. Use the processor configuration screen in your programming software:



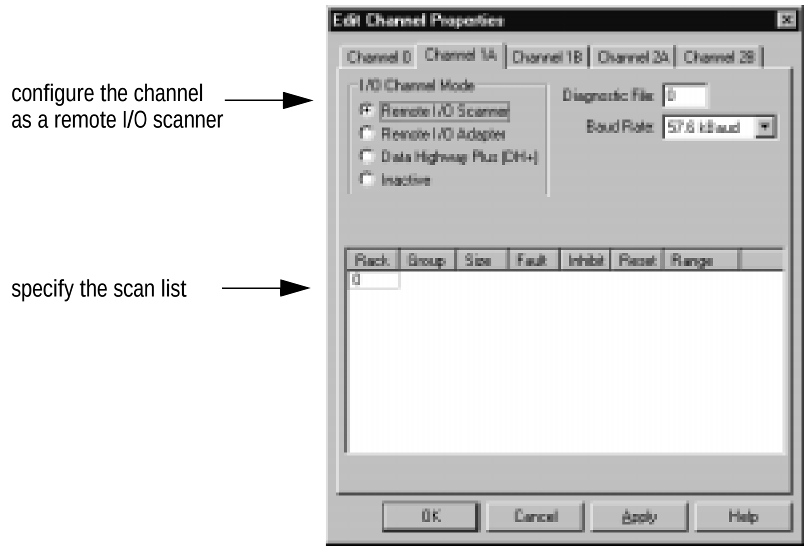
First Pass S:1/15 = No
Index Register S:24 = 0
IO Status S:16 = 0
VME Status File S:15 = N/A
IO Adapter Image S:25 = N/A
Processor Checksum S:57 = 0h
Checksum Computed S:1/13 = No
Max Overall Scan Time [ms] S:9 = 0
Last Overall Scan Time [ms] S:8 = 0

MM / DD / YYYY
Date S:20-18 = 0 / 0 / 0
HH : MM : SS
Time S:21-23 = 0 : 0 : 0

Radix: Structured
S2

Specify Channel Configuration Information

Use the scanner mode configuration screen in your programming software to configure a channel for scanner mode.



I/O Channel Mode: [✓ Remote I/O Scanner] Remote I/O Adapter Data Highway Plus (DH+) Inactive

Diagnostic File: 0

Baud Rate: 57.6 kBaud

Rack	Group	Size	Fault	Inhibit	Reset	Range
0						

In this field:	Define:	By doing the following:
Diagnostic file	The file containing the channel's status information: • messages received • messages sent • messages received with error • unable to receive • sent with error • rack retries	Cursor to the field, type an integer file number (9-999) **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used integer file. Unpredictable machine damage can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want to get status information for that channel.
Baud rate	The communication rate for the remote I/O scanner mode link	Cursor to the field and select the desired rate. Available rates are: 57.6, 115.2, and 230.4 kbps.

Specify the Scan List

A scan list is a map of the I/O devices being scanned by the scanner channel. For the channel to communicate with the I/O devices connected to it, you must create a scan list.

To:	Do the following:
Create a scan list	Make sure the processor is in Remote Program or Program mode. **1.** Make sure that you defined an I/O status file on the processor configuration screen (see page 6-7). **2.** Accept any edits made to the channel configuration. **3.** Use the autoconfiguration function If you have errors when you accept edits, clear the scan list and accept edits again. If you encounter the error message “Resource not Available,” you have not defined an I/O status file. Define the I/O status file and try automatic configuration again.
Insert an entry into the scan list	Make sure the processor is in Remote Program, Program, or Remote Run mode. **1.** Position the cursor at the place on the scan list where you want to insert an entry. **2.** Insert an entry into the list and enter the appropriate values for the list. **Important:** If incorrect information is entered for an entry, the processor will not display the new configuration when you save edits.
Delete an entry for the scan list	Make sure the processor is in Remote Program, Program, or Remote Run mode. **1.** Position the cursor at the place on the scan list where you want to delete an entry. **2.** Delete the entry from the list. **Important:** If incorrect information is entered for an entry, the processor will not display the new configuration when you save edits.

A scan list includes the following:

For this field:	A scan list contains:
Rack address	1-3 octal (PLC-5/11, -5/20, -5/20E processors) 1-7 octal (PLC-5/30 processors) 1-17 octal (PLC-5/40, -5/40L, -5/40E processors) 1-27 octal (PLC-5/60, -5/60L, -5/80, -5/80E processors) If complementary I/O is enabled, a C appears before the complemented rack address.
Starting group	0, 2, 4, or 6
Rack size	1/2, 1/4, 3/4, or FULL
Range	Automatically calculated based on rack address, starting module group and chassis size. An asterisk (*) after a range indicates the last valid rack entry.

Design Tip

If you need multiple updates to an I/O device during an I/O scan, you can enter a logical address in the scan list more than one time. Do not assign the same partial or full rack address to more than one channel in scanner mode. Each channel must scan unique partial and/or full rack addresses.

Keep these limitations in mind when creating/modifying a scan list:

- The automatic configuration always displays the actual hardware configuration, except for racks that have their global-rack inhibit bit set. In this case, the global-rack bit overrides the automatic configuration. You must first clear the global-rack inhibit and then select autoconfigure.

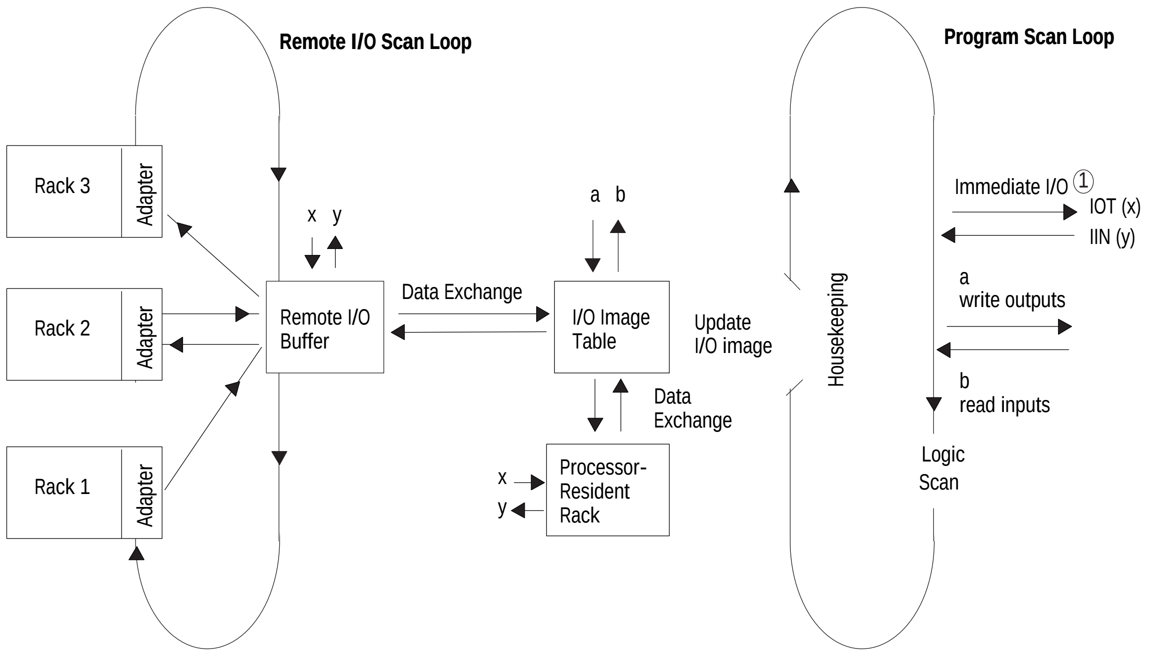
  Clear global-rack inhibit bits for the channel that scans the racks that you want to resume scanning. Use the scanner mode status screen in your programming software.

- If you change a channel configuration from adapter or DH+ mode to scanner mode, use the clear list function of your programming software to clear the scan list. In any other instance where you need to clear entries from the scan list, use the delete-from-list function to delete the entries one at a time.

Communicating to a Remote I/O Node Adapter

A scanner channel exchanges discrete data with remote I/O node adapters like 1771-ASB modules via the remote I/O buffer.

Figure 6.1
Remote I/O Scan and Program Scan Loops



The remote I/O scan is the time it takes for the processor to communicate with all of the entries in its rack scan-list once. The remote I/O scan is independent of and asynchronous to the program scan.

During housekeeping:
- Data exchange between the I/O image table, the processor-resident rack, and the remote I/O buffer occurs.
- The remote I/O buffer is updated.

Remember that the I/O scanner is constantly updating the remote I/O buffer asynchronously to the program scan.

① In remote racks, immediate I/O data transfers update the remote I/O buffer.

Important: The remote I/O scan for each channel configured for scanner mode is independent and asynchronous to the remote I/O scan for any other channel.

For the scanner channel to communicate with the 1771-ASB adapter modules, do the following:	For more information, see:
1. Set the I/O chassis backplane switch for each chassis that houses an adapter module. **2.** Set the switches on the adapter module itself.	chapter 23
3. Connect the remote I/O cable.	your processor installation instructions

Troubleshooting Remote I/O Communication Difficulties

Follow these steps to make sure the processor can communicate with devices on remote I/O links.

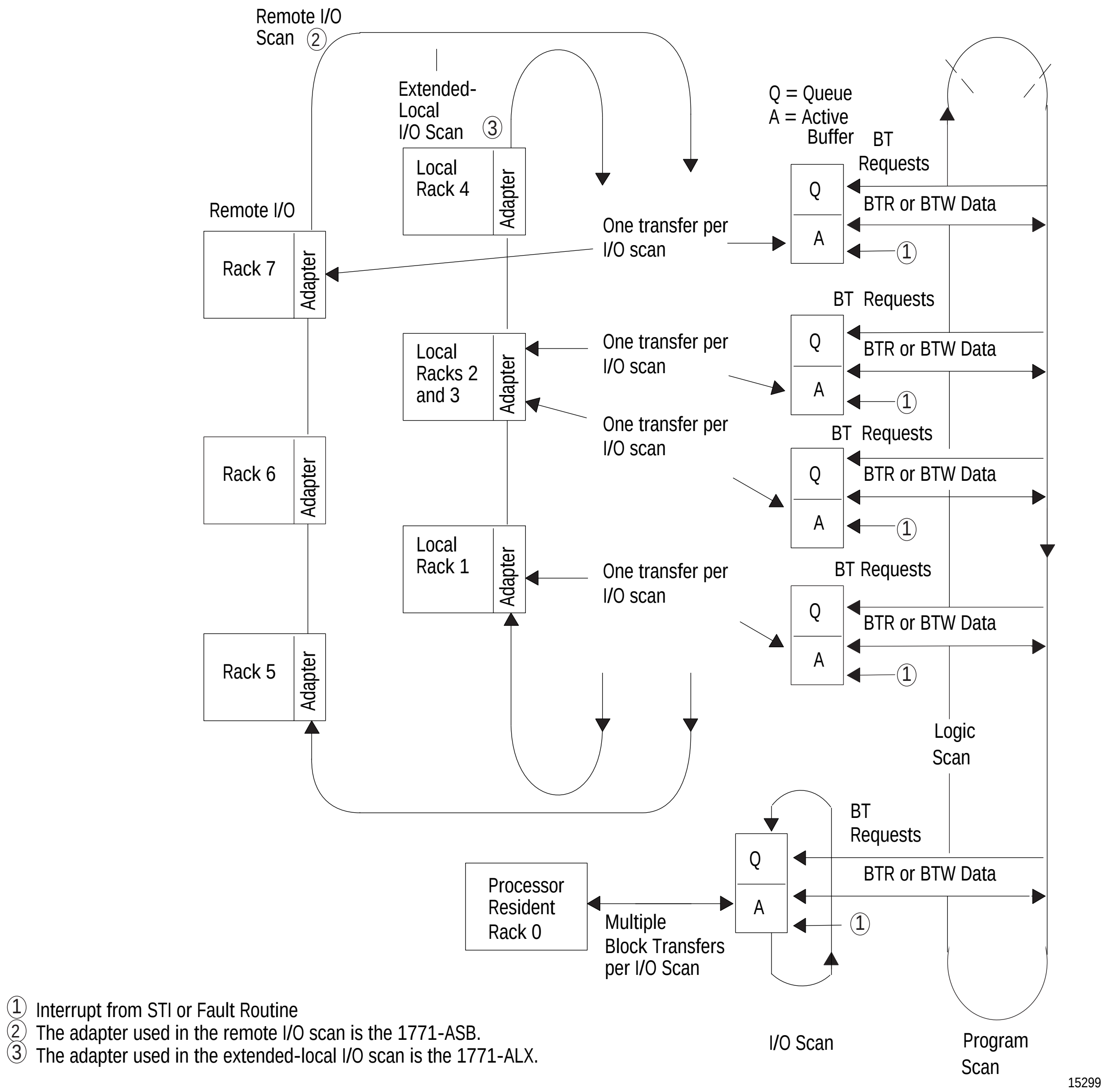
1. Put the processor in program mode. Go into the memory map and find two unused file numbers. The processor will use these files. Do not create the files, just record which file numbers you will use.
2. Go to the processor status screen and make sure all rack inhibit bits are zeroed (0).
3. Go to the processor configuration screen and assign one of the previous file numbers to be the I/O status file (see page 6-7).
4. Go to the channel configuration screen for the appropriate channel and assign the remaining file number (from above) to be the channel diagnostic file (see page 6-8).
5. Perform an autoconfigure and confirm that all the racks were found and listed in the I/O scan list.
6. Check all I/O rack retry counters in channel status to make sure there are no communications problems.

If you follow the above steps and there are still remote I/O communications problems, it is possible that the I/O status file is corrupt. Try assigning a brand new I/O status file and repeat the steps above. Also, confirm that the I/O image tables exist for the racks you are having difficulty communicating with.

Transferring Block Data

In addition to discrete data, the processor can also exchange block data with remote I/O. Block-transfer instructs the processor to interrupt normal I/O scanning and transfer as many as 64 words of data to/from a selected I/O module. Figure 6.2 shows how the scanner-mode processor handles a block-transfer.

Figure 6.2
Block-Transferring Data to Processor-Resident Local, Extended-Local, and Remote I/O



① Interrupt from STI or Fault Routine
② The adapter used in the remote I/O scan is the 1771-ASB.
③ The adapter used in the extended-local I/O scan is the 1771-ALX.

As shown in Figure 6.2, the processor has the following storage areas for block-transfers:

Maximum Number of Active Buffers Per Remote I/O Channel

PLC-5/60, -5/60L, -5/80, -5/80E	23
PLC-5/40, -5/40L, -5/40E	31
PLC-5/30	39
PLC-5/20, -5/20E	43
PLC-5/11	43

Storage area:	**Description:**
active buffers	store initialized block-transfer requests for a channel The adjacent table lists the maximum active buffers for each enhanced and Ethernet PLC-5 processor. The processor places a block-transfer request **directly** into the active buffer only if: a buffer is available and no block-transfers to the slot is in the queue.
waiting queues	store block-transfer requests that cannot be placed into the active buffer because: • all of the channel's active buffers are being used • the slot addressed by the block-transfer is currently processing a block-transfer

Placing the processor in program mode, cancels block-transfers in the active buffers and in the waiting queues.

Once a block-transfer to a slot completes, the processor checks the queue to see if a block-transfer addressed to the slot is waiting. If one exists, the processor moves it to the active buffer.

Since a processor can request a block-transfer from every slot in a chassis concurrently, the adapter device chooses the order in which the block-transfers execute on the chassis. Block-transfer requests are processed differently in fault routines, selectable timed interrupt routines (STI), and processor input interrupt routines (PII); see chapters 16, 18, and 19 respectively for more information.

Block-Transfer Minor Fault Bits

This minor fault:	Description:
S:17/0 Block-transfer queue full to remote I/O	There is a possibility that the PLC-5 processor might temporarily be unable to initiate multiple consecutive user-programmed block-transfers. For any block-transfer which temporarily can't be processes, the PLC-5 processor sets minor fault bit S:17/0 and skips that block-transfer instruction. This condition is self-correcting, but bit S:17/0 remains set until you reset it. You can avoid this minor fault be separating block-transfer instruction rungs with other rungs.
S:17/1 through S:17/4 Queue full - channel xx	The PLC-5 processor can process a maximum of 64 remote block-transfers per channel pair (1A/1B or 2A/2B). This maximum includes: • block-transfers that are currently in the active buffer • initialized block-transfers that are waiting for execution in the holding queue Once the 64 block-transfer maximum is reached, the following minor fault bits are set, depending on which channel pair is involved: **Channel pair:** / **Minor fault bits set:** 1A/1B — S:17/1 and S:17/2 2A/2B — S17:3 and S:17/4 The PLC-5 processor won't initialize any remote block-transfer instruction which exceeds the 64 maximum. The .EW, .DN, and .ER bits are reset on any block-transfer which exceeds the 64 maximum. This condition is self-correcting, but the bits remain set until you reset them.
S:10/7 No more command blocks exist	This minor fault bit is normally associated with an application programming problem, but this bit can also be set when using block-transfers if the maximum number of command blocks available in the PLC-5 processor is exceeded. The command blocks are used by both the local and remote block-transfers. **PLC-5 type:** / **Maximum number of command blocks:** PLC-5/11, -5/20, -5/30 — 128 PLC-5/40 — 256 PLC-5/60, -5/80 — 384 This condition generally occurs when a program attempts to repeatedly initialize block-transfers which have not yet completed with a .DN or .ER bit. This condition is self-correcting, but bit S:10/7 remains set until you reset it.

Block-Transfers of Remote I/O Data

Block-transfers of I/O data to remote I/O follow these guidelines:

- Block-transfer data exchange and the program scan run independently and concurrently. Once block-transfers are initiated, the processor performs them asynchronously to the program scan.
- During every remote I/O scan, the processor performs a maximum of one block-transfer per entry in the scan list.

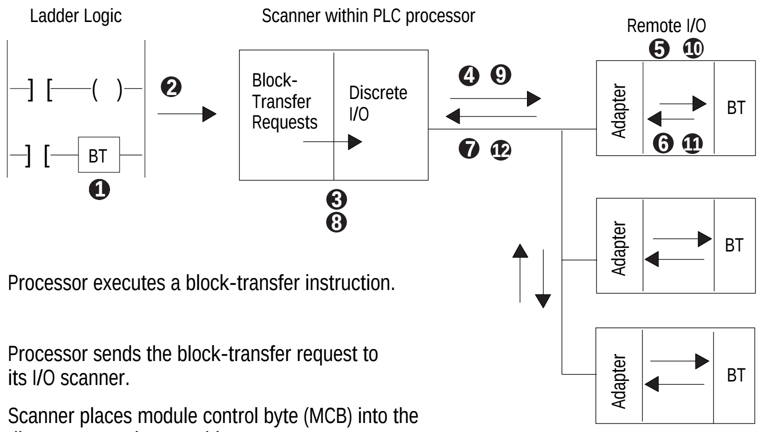
Important: If you split remote rack addresses between scanner channels, block-transfers to lower priority scanner channels do not function properly. Scanner channels have priority according to the following order: 1A, 1B, 2A, then 2B.

For example: if you configure channels 1B and 2A as remote scanners and split rack #2 between them, block-transfers to channel1B (the higher priority channel) will be completed, but block-transfers to the second half of rack #2 (2A, the lower priority channel) will not be completed.

Although splitting remote I/O racks across scanner channels does not affect discrete transfers, I/O status bits such as Fault and Present may not indicate the correct status.

Figure 6.3 shows the remote I/O block-transfer sequence.

Figure 6.3
Block-Transfer Sequence



❶ Processor executes a block-transfer instruction.

❷ Processor sends the block-transfer request to its I/O scanner.

❸ Scanner places module control byte (MCB) into the discrete output image table.

❹ Scanner sends MCB as part of the discrete I/O update to the adapter.

❺ The adapter module sends the block-transfer request to the block-transfer module.

❻ The block-transfer module returns a module status byte (MSB) to the adapter.

❼ MSB returned to the scanner in addition to the discrete I/O by the adapter.

❽ The scanner forms a block-transfer packet.

❾ The scanner sends the block-transfer packet to the adapter for the block-transfer module (the packet includes data if it is a block-transfer write).

❿ The adapter passes the block-transfer packet to the block-transfer module.

⓫ The block-transfer module sends status to the adapter (will also send data if it is a block-transfer read).

⓬ The adapter passes status to the I/O scanner; if the request is a block-transfer read, adapter sends data.

Block-Transfer Sequence with Status Bits

Figure 6.4 describes the different states of the block-transfer status bits.

Figure 6.4
Block-Transfer Status Bit States

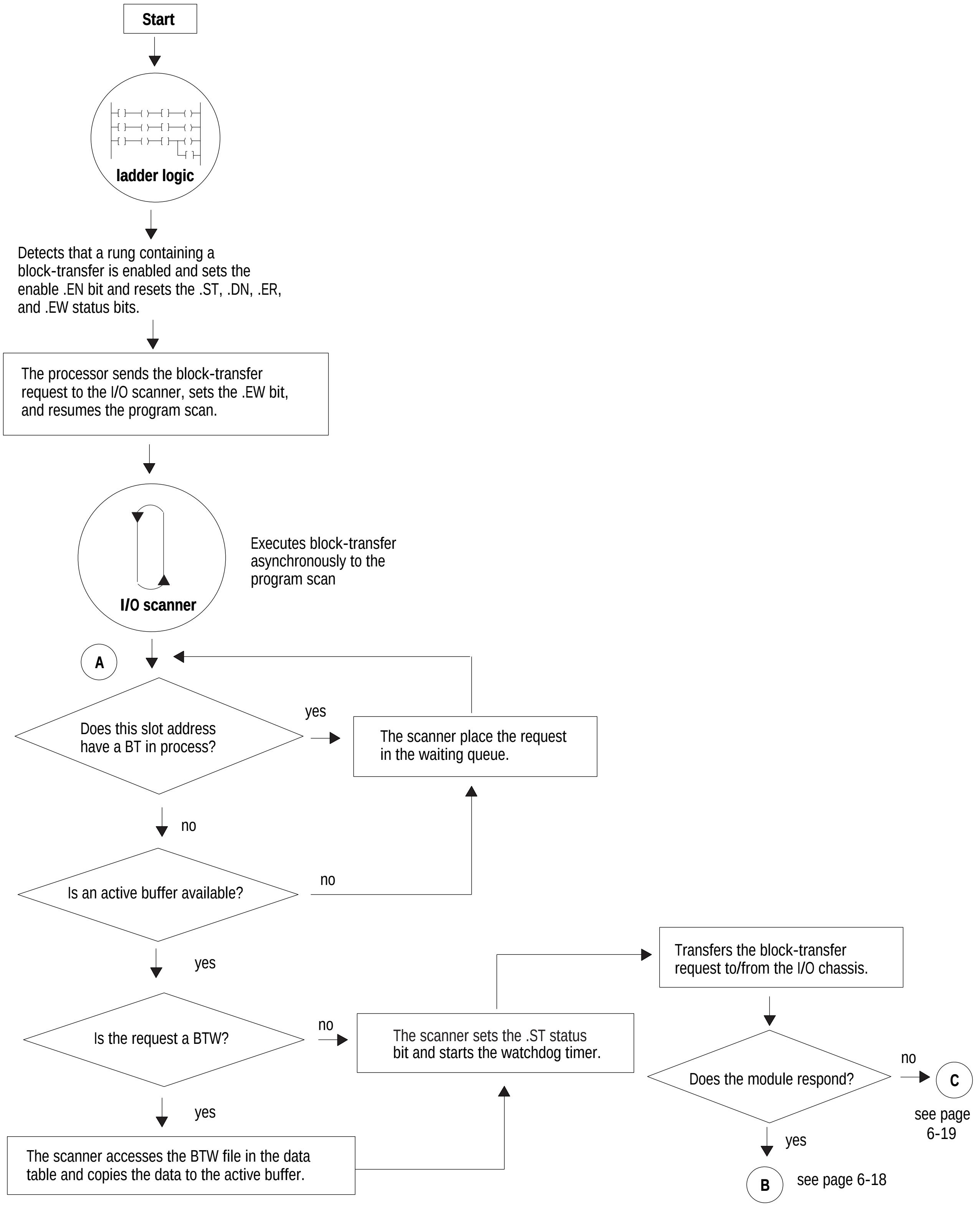

Start
ladder logic
Detects that a rung containing a block-transfer is enabled and sets the enable .EN bit and resets the .ST, .DN, .ER, and .EW status bits.
The processor sends the block-transfer request to the I/O scanner, sets the .EW bit, and resumes the program scan.
I/O scanner
Executes block-transfer asynchronously to the program scan
A
Does this slot address have a BT in process?
yes
The scanner place the request in the waiting queue.
no
Is an active buffer available?
no
yes
Is the request a BTW?
no
The scanner sets the .ST status bit and starts the watchdog timer.
yes
The scanner accesses the BTW file in the data table and copies the data to the active buffer.
Transfers the block-transfer request to/from the I/O chassis.
Does the module respond?
no
C
see page 6-19
yes
B
see page 6-18

Figure 6.4 (continued):
The block-transfer module responds

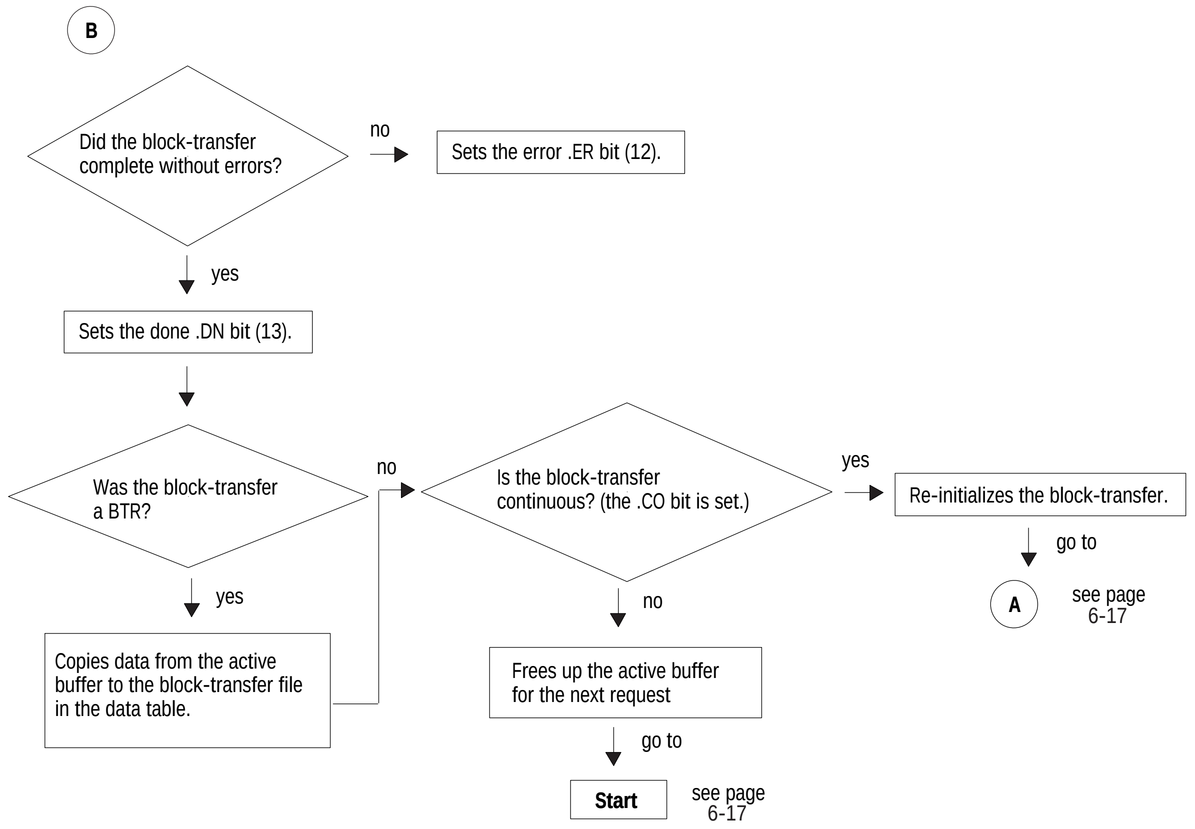
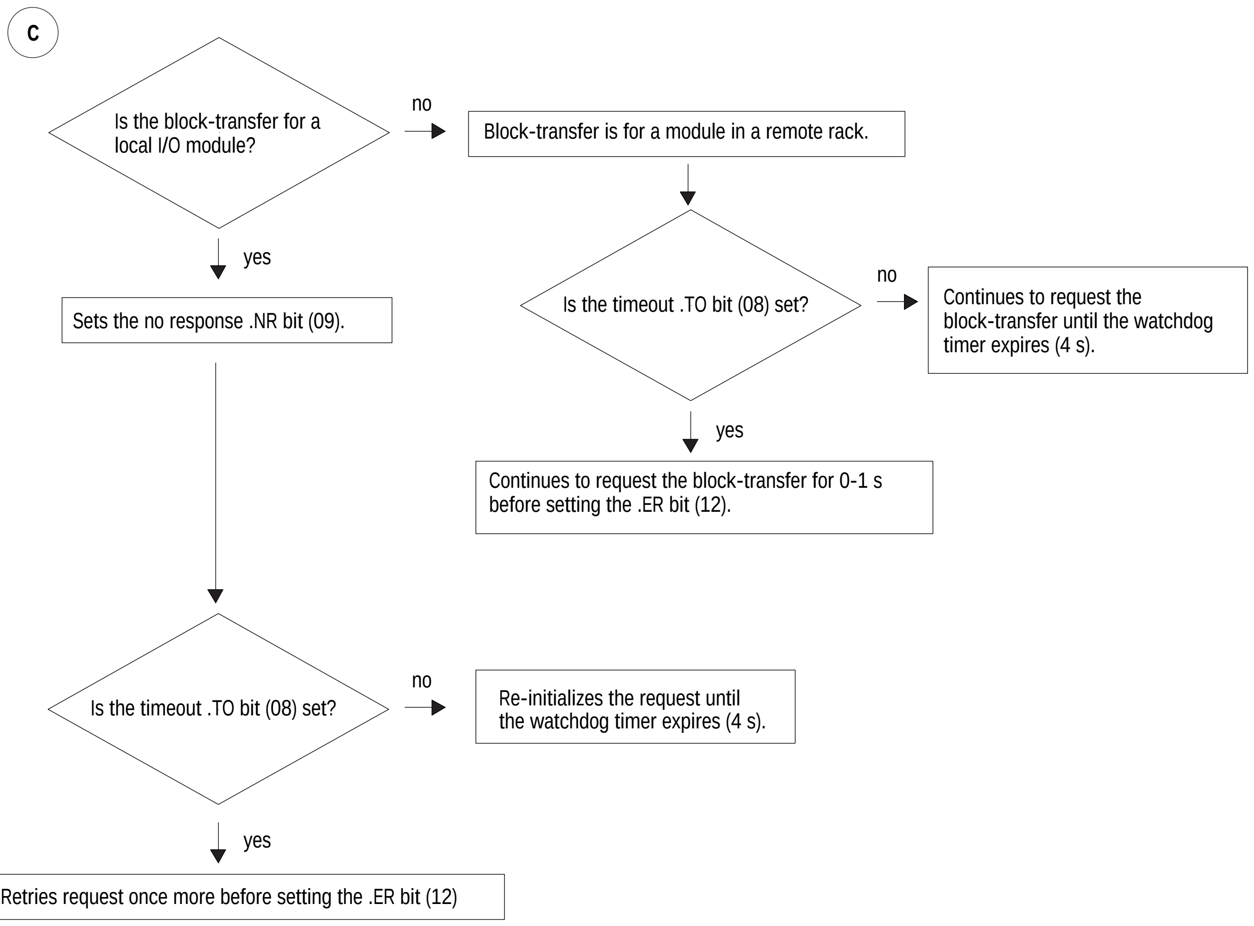

B
Did the block-transfer complete without errors?
no
Sets the error .ER bit (12).
yes
Sets the done .DN bit (13).
Was the block-transfer a BTR?
no
Is the block-transfer continuous? (the .CO bit is set.)
yes
Re-initializes the block-transfer.
go to
A
see page 6-17
yes
Copies data from the active buffer to the block-transfer file in the data table.
no
Frees up the active buffer for the next request
go to
Start
see page 6-17

Figure 6.4 (continued):
The block-transfer module does NOT respond


C
Is the block-transfer for a local I/O module?
no
Block-transfer is for a module in a remote rack.
yes
Sets the no response .NR bit (09).
Is the timeout .TO bit (08) set?
no
Continues to request the block-transfer until the watchdog timer expires (4 s).
yes
Continues to request the block-transfer for 0-1 s before setting the .ER bit (12).
Is the timeout .TO bit (08) set?
no
Re-initializes the request until the watchdog timer expires (4 s).
yes
Retries request once more before setting the .ER bit (12)

MORE

For a list of block-transfer error codes, see the PLC-5 Programming Software Instruction Set Reference, publication 1785-6.1.

Block-Transfer Programming Considerations

Read this section for information about general programming considerations and considerations for processor-resident local racks.

Design Tip

General Considerations

The following are general programming considerations when you are block-transferring I/O data.

- When performing block-transfers (processor-resident local or remote I/O) in any PLC-5 processor, clear the output image table corresponding to the block-transfer module rack location before changing to run mode. If you do not clear the output image table, then you encounter block-transfer errors because unsolicited block-transfers are being sent to the block-transfer module (i.e., if a block-transfer module is installed in rack 2, group 4, clear output word O:024 to 0; do not use the word for storing data).
- If you use remote block-transfer instructions and have the timeout bit (.TO) set to 1, then the processor disables the 4-second timer and continues to request the block-transfer for 0-1 seconds before setting the error (.ER) bit.
- A PLC-5 processor with at least one channel configured as an adapter could incur a non-recoverable fault when you switch it from run to program mode.

To avoid this possibility, program the scanner to request only two or three block-transfers from the PLC-5 adapter at one time by conditioning the block-transfer instructions with the done/error bits.

For Processor-Resident Local Racks

The following are programming considerations when you are block-transferring data in a processor-resident local rack.

- Within the processor-resident local rack, limit the number of continuous-read block-transfers to 16 transfers of 4 words each or 8 transfers of 64 words each. If you attempt to exceed this block-transfer limit, a checksum error (error code -5) occurs.
- Block-transfer instructions to any of the following modules residing in the processor-resident local rack result in frequent checksum errors.
  - 1771-OFE1, -OFE2, and -OFE3 modules; all versions prior to series B, revision B
  - 2803-VIM module, all versions prior to series B, revision A
  - IMC-120, all versions

- To eliminate the checksum errors, replace your modules with the current series and revision. If replacement is not possible:

  1. Go to the processor configuration screen in your programming software.

  2. With the processor in program mode, set user control bit 4 (S:26/4) to 1 (the local block-transfer compatibility bit).

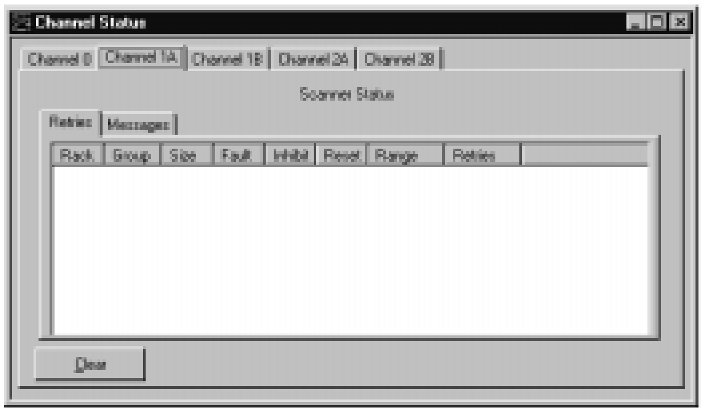
  3. Change the processor mode from program to run.

- Do not program IIN or IOT instructions to a module in the same physical module group as a BT module unless you know a block-transfer is not in progress. If you must do this, then use an XIO instruction to examine the .EN bit of the block-transfer instruction to condition the IIN and IOT.

Monitoring Remote I/O Scanner Channels

To monitor channels configured as a scanner, use the scanner mode status screen in your programming software. This screen displays the data in the diagnostic file you defined on the scanner mode configuration screen (page 6-8).

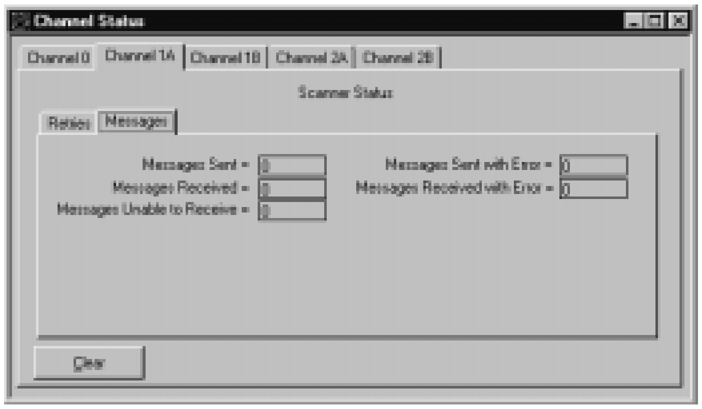
Monitoring transmission retries



Status Field:	Location:	Description:
Retries Tab		
Retry	word 5 etc. word 69	Displays the number of retries for the corresponding rack entry. Entry 1 etc. Entry 64
Rack Address		This field indicates the rack number of the remote racks being scanned by the scanner channel: can only scan rack 3 (PLC-5/11 processor) 1-3 octal (PLC-5/20, -5/20E processor) 1-7 octal (PLC-5/30 processors) 1-17 octal (PLC-5/40, -5/40L, 5/40E processors) 1-27 octal (PLC-5/60, -5/60L, -5/80, -5/80E processors) If complementary I/O is enabled (on the scanner mode configuration screen), the complement of a rack is identified with a C to the left of the rack address column on the status screen.

Status Field:	Location:	Description:
Starting Group		This field indicates the first I/O module group in the rack that the processor scans.
Rack Size		This field displays the portion of the I/O rack addressed by each chassis. Configurations can be 1/4, 1/2, 3/4, or FULL as long as the total sum of the rack does not exceed 8 I/O groups.
Range		This field displays the rack address and module groups being scanned for a rack in the scan list. An asterisk (*) after a range indicates that it is the last valid rack entry.
Fault		An F displayed in this field indicates that the corresponding chassis is faulted. When a fault indicator appears, the system sets the associated fault bit in the global rack fault status on the processor status screen in your programming software. When the global rack fault bit is set, all configuration information starting at the faulted quarter is lost. When a rack faults, F is displayed. If both the fault and inhibit bits are set for a rack, no rack exists at that I/O group.
Inhibit		Inhibit a rack by cursoring to the `Inhibit` field of the rack you want to inhibit and enter `1` When a chassis is inhibited the processor stops scanning it. You can inhibit an entire rack by setting the global rack-inhibit bit for that rack on the processor status screen. All chassis within that rack are inhibited, and an I appears in the `Inhibit` field, indicating the rack was globally inhibited.
Reset		Reset a rack by cursoring to the `Reset` field of the rack you want to reset and type `1` When a chassis is reset, the processor turns off the outputs of the chassis regardless of the last-state switch setting. You can reset an entire rack by setting the global rack-reset bit on the processor status screen. All chassis within that rack are reset, and an R appears in the `Reset` field indicating the rack was globally reset.
Retry		This field displays the number of times the rack was re-scanned.

Monitoring messages



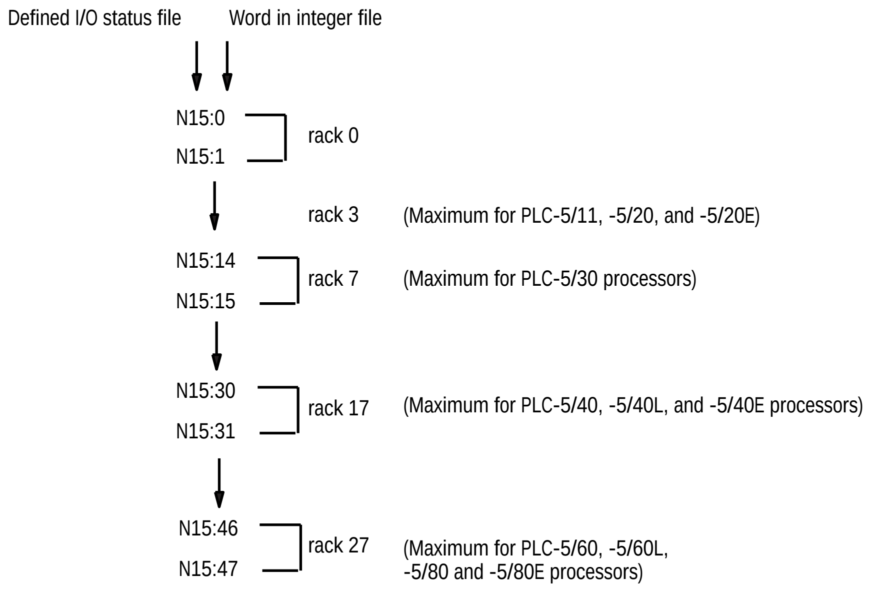
Messages Sent = 0
Messages Received = 0
Messages Unable to Receive = 0
Messages Sent with Error = 0
Messages Received with Error = 0

Status Field:	Location:	Description:
Messages Tab (Messages = SDA messages + SDN messages)		
Messages sent	word 1	Displays the number of messages sent by the channel.
Messages sent with error	word 3	Displays the number of messages containing errors sent by the channel.
Messages received	word 0	Displays the number of error-free messages received by the channel.
Messages received with error	word2	Displays the number of messages containing errors received by the channel (such as bad CRC).
Messages unable to receive	word 4	Displays the number of messages received with protocol-related problems (such as a bad block-transfer status byte with both read and write bits set).

Addressing the I/O Status File

During program execution you can address words and fault bits within the I/O status file. Figure 6.5 shows the arrangement of the words in the I/O status file for a given remote or extended local I/O rack. The example status file used for the figures in this section is integer file 15.

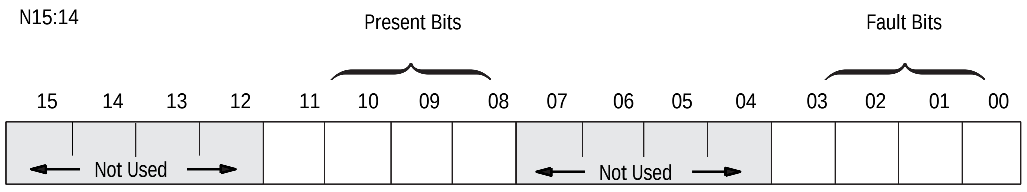
Figure 6.5
Word Arrangement in the I/O Status File


Defined I/O status file
Word in integer file
N15:0
N15:1
rack 0
rack 3
(Maximum for PLC-5/11, -5/20, and -5/20E)
N15:14
N15:15
rack 7
(Maximum for PLC-5/30 processors)
N15:30
N15:31
rack 17
(Maximum for PLC-5/40, -5/40L, and -5/40E processors)
N15:46
N15:47
rack 27
(Maximum for PLC-5/60, -5/60L, -5/80 and -5/80E processors)

The first word for a rack contains present and fault bits, the second word contains reset and inhibit bits. Figure 6.6 shows present, fault, reset, and inhibit bit layouts for rack 7 in the I/O status file.

Important: Setting inhibit bits in the I/O status file does not update inhibit bits in the processor status file.

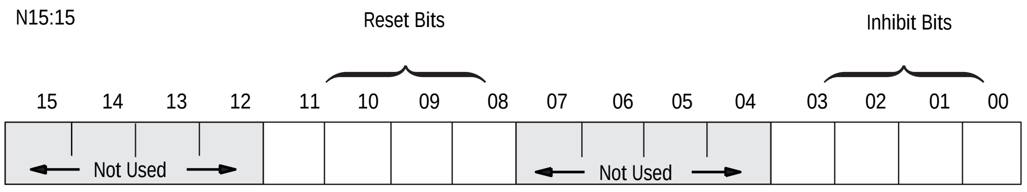
Figure 6.6
Bit Layout Diagrams for the First Word Allotted to a Remote I/O Rack or an Extended-Local I/O Rack



This bit:	Corresponds to:
Fault bits	
00	first 1/4 rack starting I/O group 0
01	second 1/4 rack starting I/O group 2
02	third 1/4 rack starting I/O group 4
03	fourth1/4 rack starting I/O group 6

Present bits	
08	first 1/4 rack starting I/O group 0
09	second 1/4 rack starting I/O group 2
10	third 1/4 rack starting I/O group 4
11	fourth1/4 rack starting I/O group 6

Figure 6.7
Bit Layout Diagrams for the Second Word Allotted to a Remote I/O Rack or an Extended Local I/O Rack



This bit:	Corresponds to:
Inhibit bits	
00	first 1/4 rack starting I/O group 0
01	second 1/4 rack starting I/O group 2
02	third 1/4 rack starting I/O group 4
03	fourth1/4 rack starting I/O group 6

Reset bits	
08	first 1/4 rack starting I/O group 0
09	second 1/4 rack starting I/O group 2
10	third 1/4 rack starting I/O group 4
11	fourth1/4 rack starting I/O group 6

ATTENTION: When you use a ladder program or the software to inhibit and reset an I/O rack, you must set or clear the reset and inhibit bits that correspond to each quarter rack in a given chassis. Failure to set all the appropriate bits could cause unpredictable operation due to scanning only part of the I/O chassis.

Notes:

Communicating with a PLC-5 Adapter Channel

Using This Chapter

This chapter explains how to configure the processor to communicate with an adapter channel:

1. Select which channel to configure as an adapter.
2. Define a diagnostic file.

   Use a unique, unused integer file.

3. Define the rack address, starting group and rack size.

   The default rack address is rack 3.

4. Define the discrete transfer files (enter as decimal numbers).

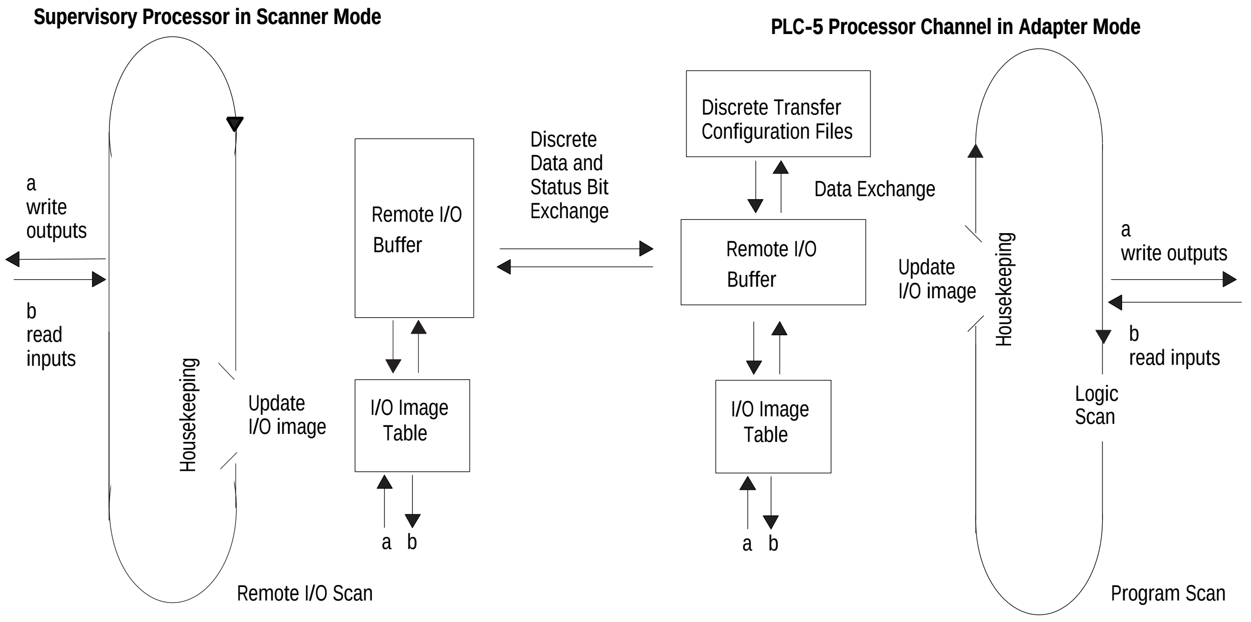
   The input source is where the supervisory processor’s output discrete bits go (default is 001:024 - the decimal representation of rack 3). The output source is where the supervisory processor’s input bits go (default is 000:024).

5. Create the necessary block-transfer control files (one BTR and one BTW).
6. Configure the block-transfers so the supervisory processor knows where to address block-transfers.

Configuring Communication to a PLC-5 Adapter Channel

Because a PLC-5 processor adapter channel is more intelligent than a 1771-ASB module, data communication and configuration tasks are handled differently for adapter channels.

The supervisory processor or scanner channel and the adapter-mode processor channel automatically transfer discrete data and status between themselves via the supervisory processor's remote I/O scan.



During each remote I/O scan, the supervisory processor transfers 2, 4, 6, or 8 words*depending on whether the adapter-mode processor is configured as a 1/4, 1/2, 3/4, or full rack.

The adapter-mode processor transfers 2, 4, 6, or 8 words*depending on whether it is configured as a 1/4, 1/2, 3/4, or full rack.

Discrete data and block-transfer status bits are transferred between a remote I/O scanner's I/O image table and an adapter channel via the adapter channel's discrete transfer configuration files, which you define on the adapter channel configuration screen.

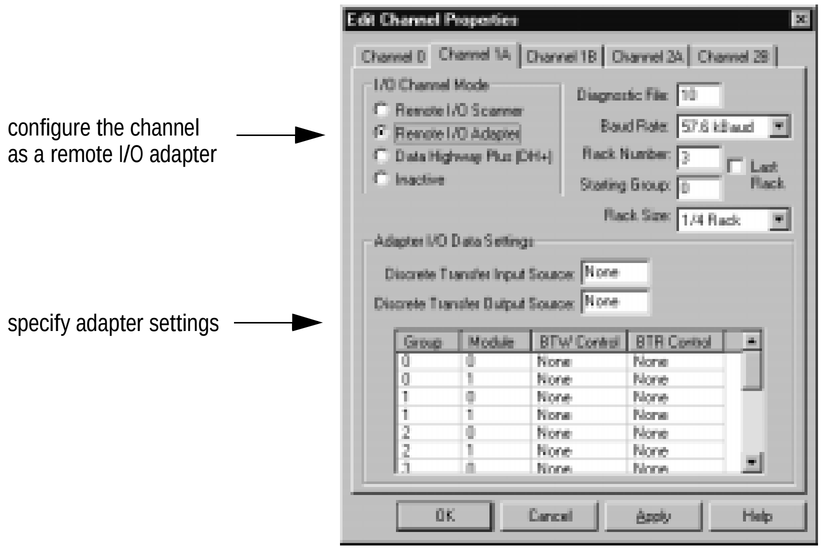
For the scanner channel to communicate with a PLC-5 processor adapter channel, do the following:	For more information, see:
1. Define the communication rate, its address, and rack size (number of words to transfer).	page 7-3
2. Define the discrete transfer configuration files, which are the files from which the adapter processor channel gets the data sent by the supervisory processor and puts data into for the supervisory processor.	page 7-4
3. If you plan to block-transfer data to the adapter channel, define the block-transfer files and configure the block-transfers.	page 7-8
4. Connect the remote I/O cable.	your processor installation instructions

Specify an Adapter Channel’s Communication Rate, Address, and Rack Size

Use this table to help you determine the processor channels you can configure as a remote I/O adapter:

Processor:		**Channels that support remote I/O adapter:**
PLC-5/11		1A
PLC-5/20	PLC-5/20E	1B
PLC-5/30 PLC-5/40L PLC-5/60L	PLC-5/40E PLC-5/80E	1A, 1B
PLC-5/40 PLC-5/60	PLC-5/80	1A, 2A, 1B, 2B

To select a channel as an adapter, use the adapter mode configuration screen in your programming software.



I/O Channel Mode
- Remote I/O Scanner
- [✓ Remote I/O Adapter]
- Data Highway Plus (DH+)
- Inactive

Diagnostic File: 10
Baud Rate: 57.6 kBaud
Rack Number: 3
Last Rack
Starting Group: 0
Rack Size: 1/4 Rack

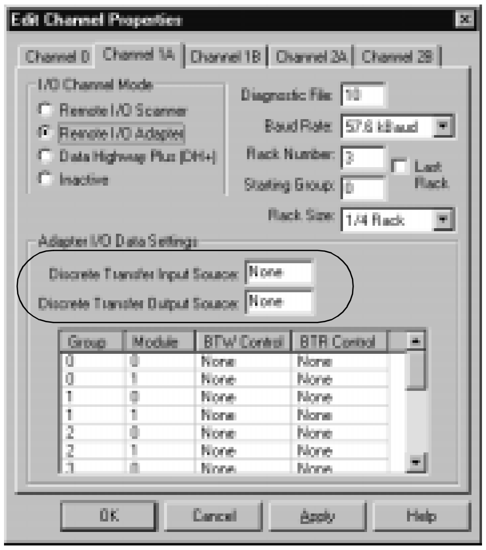
Adapter I/O Data Settings
Discrete Transfer Input Source: None
Discrete Transfer Output Source: None

Group	Module	BTW Control	BTR Control
0	0	None	None
0	1	None	None
1	0	None	None
1	1	None	None
2	0	None	None
2	1	None	None
3	0	None	None

In this field:	**Define:**	**By doing the following:**
Diagnostic file	The file containing the adapter channel’s status information	Cursor to the field and enter an integer file number (9-999). **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used integer file. Unpredictable machine damage can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want to get status information for that channel.
Baud rate	Communication rate for the remote I/O link	Cursor to the field and select the desired rate. Available rates are: 57.6, 115.2, or 230.4 kbps.

In this field:	Define:	By doing the following:
Rack number	The rack address of this PLC-5 processor as it appears to the scanner	Cursor to the field and enter the address. Valid addresses are (based on the scanner this PLC-5 processor communicates with): • 3 octal (PLC-5/11 processors) • 1-3 octal (PLC-5/20, -5/20E processors) • 1-7 octal (PLC-5/30 processors) • 1-17 octal (PLC-5/40, -5/40L, -5/40E processors) • 1-27 octal (PLC-5/60, -5/60L, -5/80, -5/80E processors) The default is rack 3. **Important:** The valid addresses are based on the scanner, not the PLC-5 processor you are configuring. For example, if you are configuring a PLC-5/20, you could enter a rack address between 1-27 if the scanner you will be communicating with is a PLC-5/60.
Last rack	Notifies the supervisory processor that this is the last chassis This information is important when the supervisory processor is a PLC-2 processor.	Select the check box if this is the last rack.
Starting group	The starting group number of the rack	Cursor to the field and enter the number Valid entries are: 0, 2, 4 or 6.
Rack size	The number of I/O words to exchange with the supervisory processor	Cursor to the field and select the rack size, which depends on the starting group you selected above: If you want to communicate using: • 2 words - select 1/4 (starting group 6) • 4 words - select 1/2 (starting group 4) • 6 words - select 3/4 (starting group 2) • 8 words - select FULL (starting group 0) For example, if you choose starting group 6, you can only transfer 2 words. If you choose starting group 4, you can transfer 4 or 2 words.

Specify the Discrete Transfer Configuration Files



I/O Channel Mode: Remote I/O Scanner, [✓ Remote I/O Adapter], Data Highway Plus (DH+), Inactive

Diagnostic File: 10

Baud Rate: 57.6 kBaud

Rack Number: 3

Last Rack: ☐

Starting Group: 0

Rack Size: 1/4 Rack

Adapter I/O Data Settings

Discrete Transfer Input Source: None

Discrete Transfer Output Source: None

Group	Module	BTW Control	BTR Control
0	0	None	None
0	1	None	None
1	0	None	None
1	1	None	None
2	0	None	None
2	1	None	None
3	0	None	None

The discrete transfer configuration files (output source file and the input destination file) are the main vehicles for discrete data and block-transfer status bits exchange between a PLC-5 adapter channel and a scanner channel or a supervisory processor (see Figure 7.1).

The discrete transfer configuration files can be integer, BCD, or binary data file types. Be sure to create the files specified for the input source and output source **prior** to specifying them. If they do not exist at the time of configuration, you will receive an error when trying to accept edits.

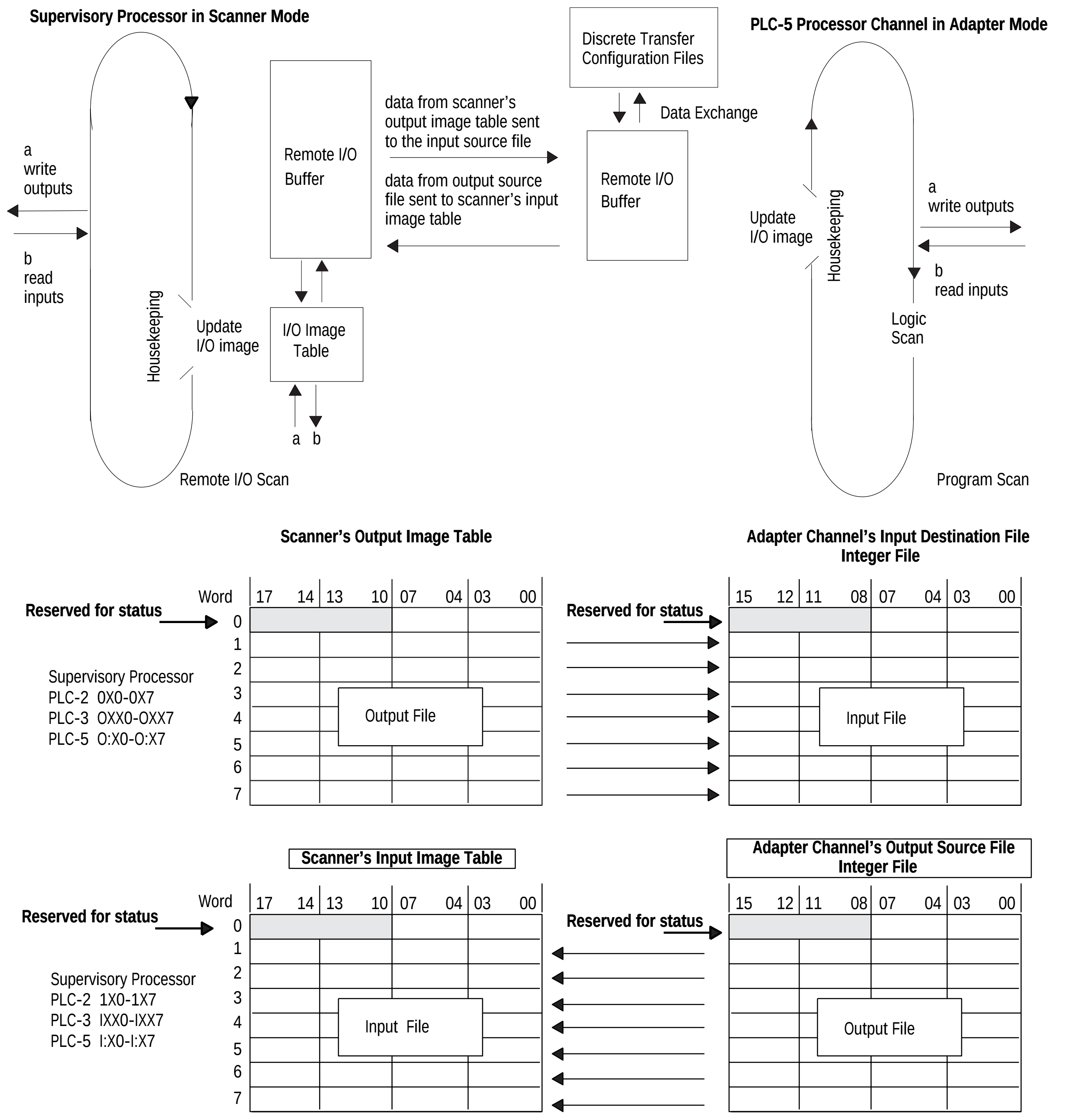
Design Tip

Configure the discrete transfer configuration file as an integer file. Although the PLC-5 processor allows you to use the input or output areas, reserve these for real I/O on scanner channels. In doing so, you are avoiding a possible conflict if you later attempt to add a rack that uses the same I/O image space.

Important: Do not configure the adapter channel's discrete transfer configuration input destination file to be the data table input image. You risk clearing inputs when performing an autoconfiguration for a scanner channel on the same processor.

The adapter inputs will not be updated until a change is detected in the input data being sent by the processor.

Figure 7.1
Discrete data and block-transfer status are exchanged between a scanner and a remote I/O adapter channel via the discrete transfer configuration files.



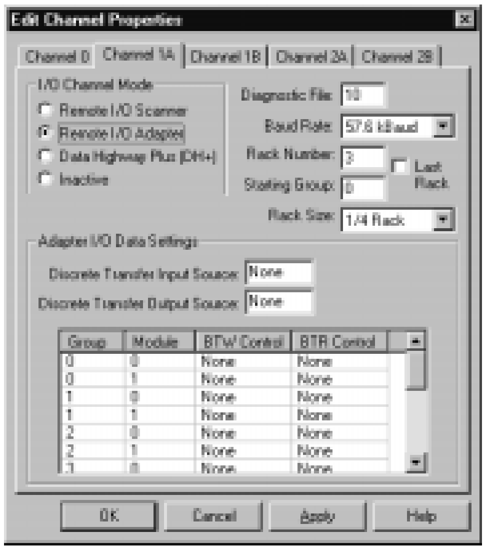
Two, four, six, or eight words of data can be transferred between the scanner and the adapter channel.
The number of words is determined by the rack size specified on the Adapter Channel Configuration screen.

If data from the supervisory processor is intended to control outputs of the adapter-mode processor channel, write ladder logic in the adapter-mode processor to move the data from its input destination file to its output image. Use XIC and OTE instructions for bit data; use move and copy instructions for word data.

If you want the supervisory processor to read data from a data file in the adapter-mode processor, write ladder logic in the adapter-mode processor to move that data to its output source file for transfer to the supervisory processor's input image table.

To create the discrete transfer configuration files, use the adapter mode configuration screen in your programming software.

specify the discrete transfer configuration files



I/O Channel Mode
- ( ) Remote I/O Scanner
- (•) Remote I/O Adapter
- ( ) Data Highway Plus (DH+)
- ( ) Inactive

Diagnostic File: 10
Baud Rate: 57.6 kBaud
Rack Number: 3
Last Rack: ☐
Starting Group: 0
Rack Size: 1/4 Rack

Adapter I/O Data Settings

Discrete Transfer Input Source: None
Discrete Transfer Output Source: None

Group	Module	BTW Control	BTR Control
0	0	None	None
0	1	None	None
1	0	None	None
1	1	None	None
2	0	None	None
2	1	None	None
3	0	None	None

Important: The processor determines the number of words used by the file according to the rack size you specified.

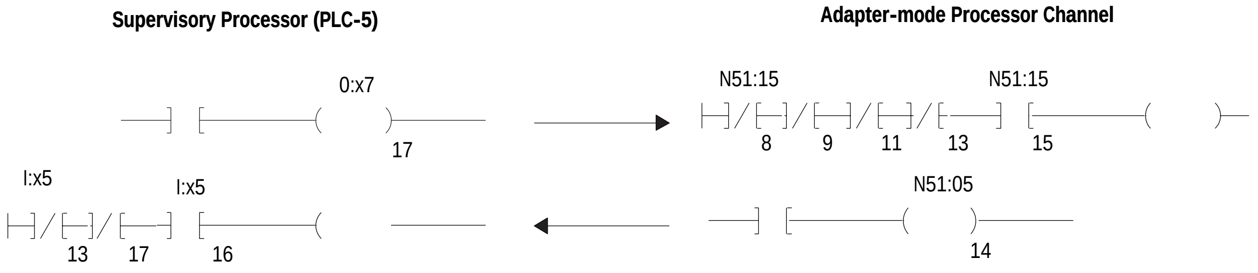
In this field:	Define:	By doing the following:
Input destination	The location where the scanner (host device) places output words into the adapter's input file	**1.** Enter the file number (decimal) of the source data. **2.** Enter the word number (decimal) of the source data. Specify an input image, output image, integer, BCD, or Hex file. For example: if you use file N7:0 and the rack size is FULL, the scanner places the 8 discrete words in file N7 words 0-7 (upper byte of first word is for status).
Output source	The location where the adapter places discrete output words into the scanner's discrete input file	**1.** Enter the file number (decimal) of the source data. **2.** Enter the word number (decimal) of the source data. Specify an input image, output image, integer, BCD, or hex file. For example: if you use file N7:10 and the rack size is FULL, the adapter channel places 8 discrete words in file N7 words 10-17 (upper byte of first word is for status).

MORE

For more information on configuring this file, see the channel configuration documentation for your programming software.

Programming Discrete Transfers in Adapter Mode

Typically, each output instruction in one processor should have a corresponding input instruction in the other processor. The rack number of the adapter mode processor-channel determines the addresses that you use.



- N51 is the adapter-mode processor's discrete transfer configuration file. Input destination and output source entries determine input and output words.
- The ladder logic in the supervisory processor uses the rack number of the adapter-mode processor channel.
- Condition the ladder logic in the adapter processor with the status bits (page).

Programming Block-Transfers of Data to an Adapter Channel

Adapter-mode block-transfers are essentially continuous. As soon as a transfer is completed, another block-transfer occurs immediately in the adapter-mode processor; it then waits (with a buffered snap-shot of data) for the supervisory processor to perform another block-transfer request. Therefore, the data that is transferred after the request is data from the previous transfer. If the supervisory processor performs a block-transfer request from the adapter-mode processor every 500 ms, for example, the data is at least 500 ms old.

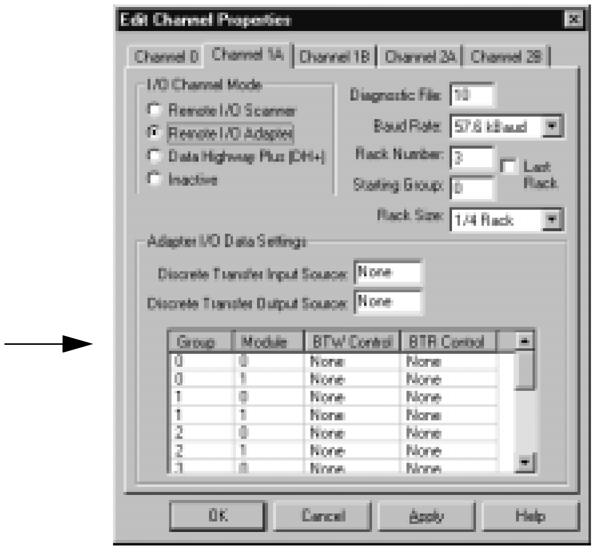
The supervisory processor contains the ladder-logic transfer instructions which controls the actual communication transmission. However, the adapter-mode processor channel controls the:

- actual number of words of data that is transferred
- data table location from which the data is transferred

Important: Do not use ladder-logic block-transfer instructions for the adapter-mode processor channel; you configure the block-transfers from channel configuration screens and data monitor screens.

Configure Block-Transfer Requests

To configure block-transfers to adapter-mode processor channel, use the adapter mode configuration screen in your programming software.



I/O Channel Mode

- Remote I/O Scanner
- [✓ Remote I/O Adapter]
- Data Highway Plus (DH+)
- Inactive

Diagnostic File: 10
Baud Rate: 57.6 kBaud
Rack Number: 3
Last Rack
Starting Group: 0
Rack Size: 1/4 Rack

Adapter I/O Data Settings

Discrete Transfer Input Source: None
Discrete Transfer Output Source: None

Group	Module	BTW Control	BTR Control
0	0	None	None
0	1	None	None
1	0	None	None
1	1	None	None
2	0	None	None
2	1	None	None
3	0	None	None

1. Define the BTW control and BTR control files you need. These control files must already exist (appear on the memory map) or the edit will result in an error. Each control word must contain a unique block-transfer control address to properly transmit block-transfers.

A. Enter the block-transfer file number.

B. Enter the element number.

C. Record on paper the BT files you define. This will help when configuring the BT files through the data monitor.

2. Since the adapter-mode channel controls the location from which data is transferred as well as the amount of data, load data into the block-transfer files by using data monitor screen in your programming software.

A. Specify a BT control file you defined.

B. Enter the transfer length in .RLEN

C. Enter the file and element numbers from which the data is to be transferred in .FILE and .ELEM respectively.

Example:
A block-transfer write of 10 words from file 24, element 10 with BT control file for group 0, module 0 of BT12:000 looks like:

Adapter Mode Configuration screen

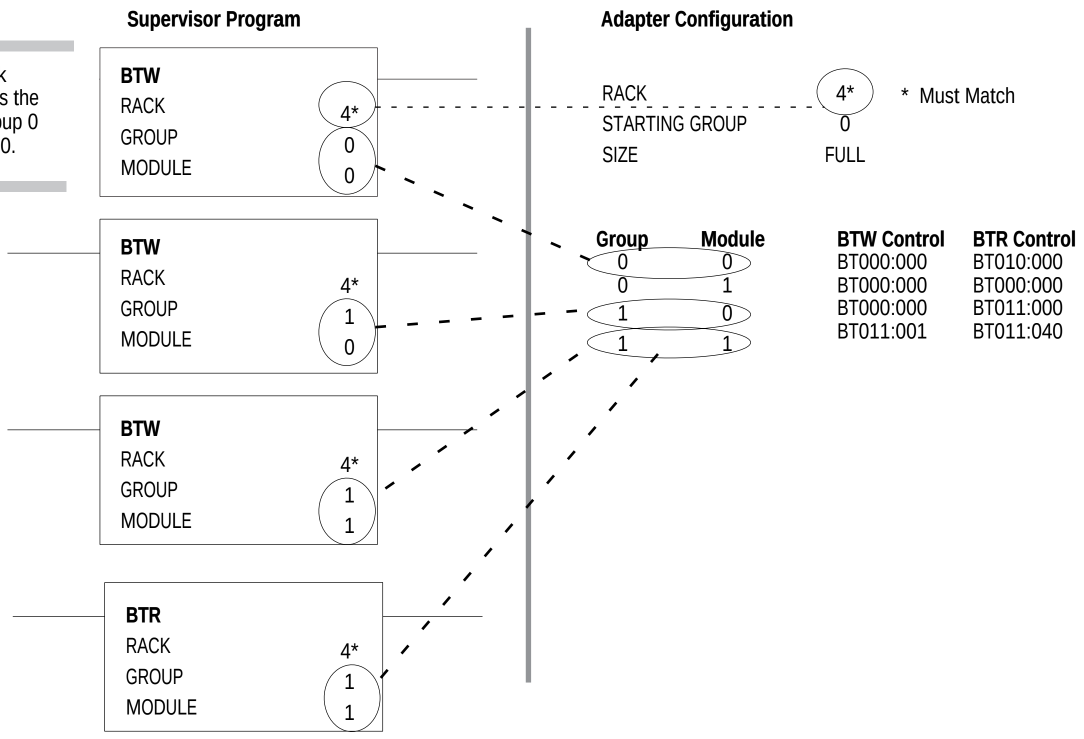
```
          Group  Module      BTW control                     BTR control

            0      0         BT02:000                         BT000:000
```

Data Monitor screen

```
       Address   EN   ST   DN   ER   CO   EW   NR   TO   RW   RLEN   DLEN   FILE   ELEM   R   G   M
     BT12:000     0    0    0    0    0    0    0    0    0    10      0     24     10    0   0   0
```

Program multiple block-transfers to an adapter-mode processor channel by matching block-transfer instructions in the supervisory processor to control files in the adapter.

In this example, the first block transfer in the supervisor uses the BTR control word listed in group 0 module 0, which is BT010:000.



Group	Module	BTW Control	BTR Control
0	0	BT000:000	BT010:000
0	1	BT000:000	BT000:000
1	0	BT000:000	BT011:000
1	1	BT011:001	BT011:040

Block transfer further defined in the adapter-mode processor channel via Data Monitor

Address	EN	ST	DN	ER	CO	EW	NR	TO	RW	RLEN	DLEN	FILE	ELEM	R	G	M
BT10:0	0	0	0	0	0	0	0	0	0	64	0	24	10	0	0	0

BT10:0 points to file 24 and element 10 and has a length of 64 words.

Assuming that file 24 has been created as an integer file, the data written down from the first block-transfer will be found in N24:10 to N24:73. The second block-transfer in the supervisor writes its data to the file to which BT11:0 points, and the third block-transfer writes its data to the file to which BT11:40 points.

You can have up to 15 writes and 15 reads. Each block transfer to a particular group/module location uses the I/O addresses for that rack/group for status bits. These locations are lost to discrete transfer. Therefore, if you configure all available 15 block-transfer read/write pairs, no bits will be available for discrete transfer. See page 7-14 for more information.

Important: Adapter-mode block-transfer reads and block-transfer writes in the same group/module location must have the same length.

If you want to transfer processor-resident local I/O data of the adapter mode processor channel to a supervisory processor or if you want to transfer data from the supervisory processor to processor-resident local I/O of the adapter mode processor channel, you must use MOV or COP instructions within the adapter-mode processor channel to move the data in or out of the data file used in the adapter block-transfer control file.

Example of Block-Transfer Ladder Logic

For block-transfer ladder logic in a:	See:
PLC-5 supervisory processor	Figure 7.2
PLC-5/250 supervisory processor	Figure 7.3

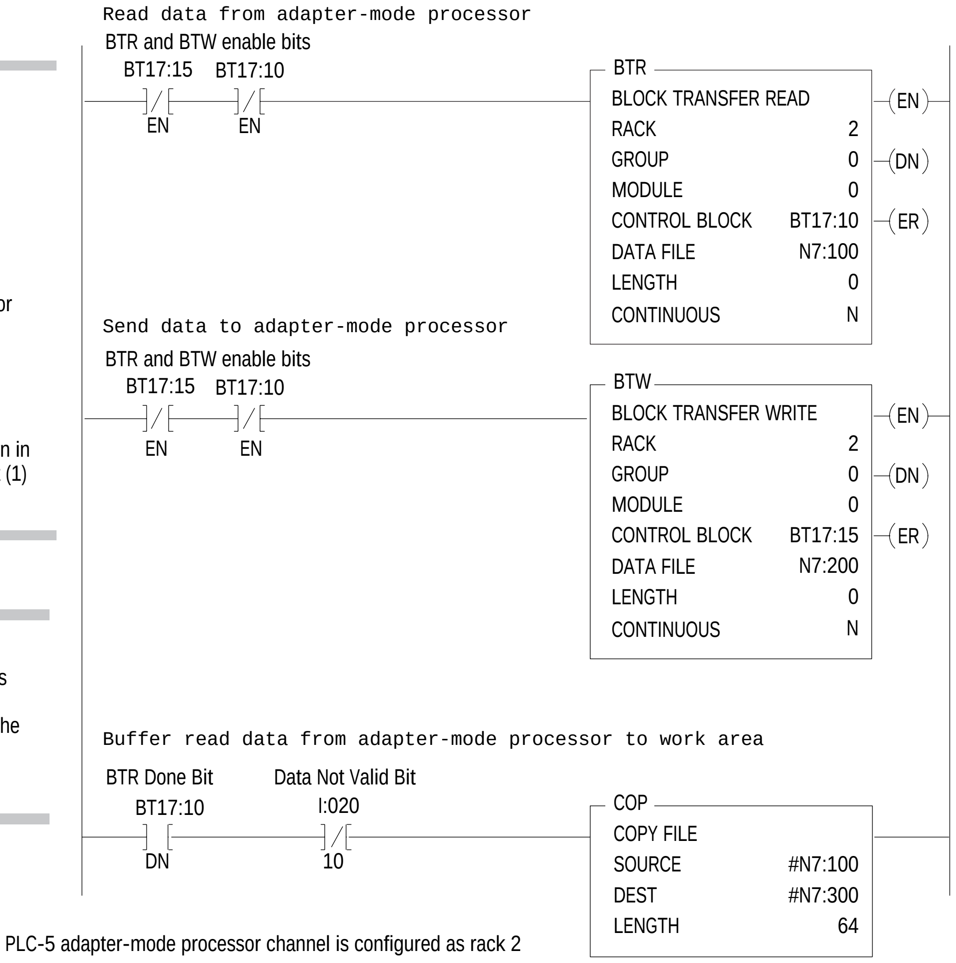
Figure 7.2
Example Bidirectional Repeating Block Transfer in PLC-5 Supervisory Processor

Enter the following parameters in the block-transfer instructions in the supervisory processor.

- Set the length to 0.
- Use the remote I/O rack number for which you configure the adapter-mode processor.
- Use the group and module numbers for which the adapter-mode processor is configured.
- Condition the use of BTR data with a "data valid" bit.

All address comments for contacts shown in the following examples represent the set (1) state of the bit in the PLC-5 processor.

You may have to execute the BTR in the PLC-5 scanner channel twice if the BTR's time delay is greater than 2-3 program scans. If you do not run the BTR twice, the BTR will read old data from the adapter processor.



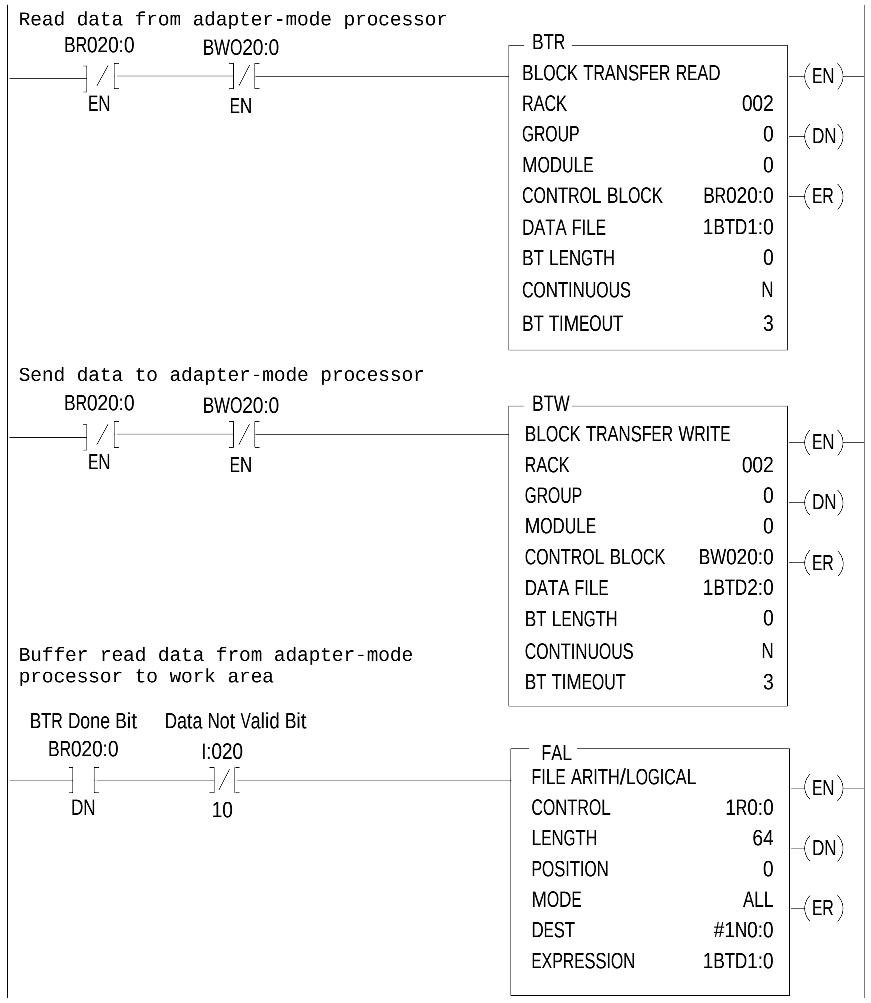
PLC-5 adapter-mode processor channel is configured as rack 2

Figure 7.3
Example Bidirectional Repeating Block Transfer in PLC-5/250 Supervisory Processor

Enter the following parameters in the block-transfer instructions in the supervisory processor.

- Set the length to 0.
- Use the remote I/O rack number for which you configure the adapter-mode processor.
- Use the group and module numbers for which the adapter-mode processor is configured.
- Condition the use of BTR data with a "data valid" bit.

All address comments for contacts shown in the following examples represent the set (1) state of the bit in the PLC-5 processor.

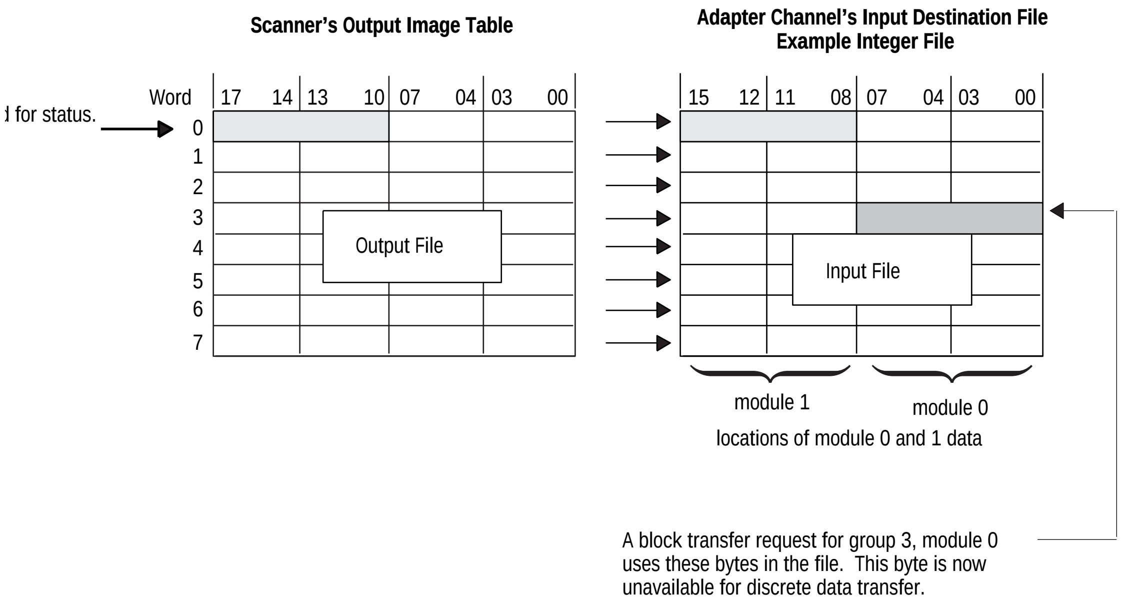


PLC-5 adapter-mode processor is configured for rack 2

Effects of Programming Block-Transfers to an Adapter-Mode Processor Channel on Discrete Data Transfer

Because the discrete transfer configuration files are used for discrete data transfer as well as block-transfer status exchanges between a supervisory processor and adapter-mode processor channel, performing multiple block-transfer to and from the adapter-mode processor channel impacts discrete data transfer.

Each group/module that is programmed as an adapter channel block transfer uses one byte in the adapter channel's input destination file. For example:

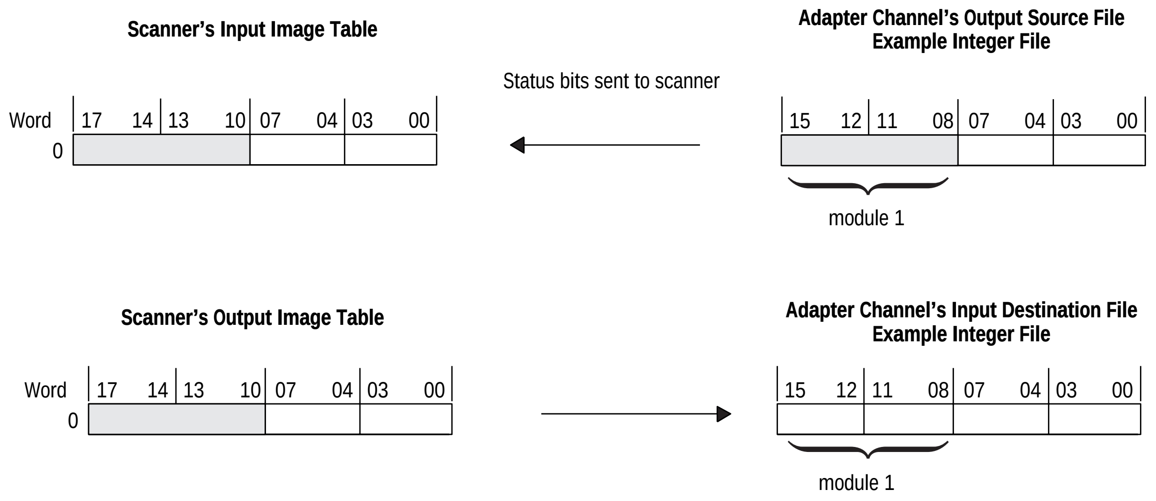

Scanner's Output Image Table
Adapter Channel's Input Destination File
Example Integer File
Word
17 14 13 10 07 04 03 00
15 12 11 08 07 04 03 00
d for status.
0
1
2
3
4
5
6
7
Output File
Input File
module 1
module 0
locations of module 0 and 1 data
A block transfer request for group 3, module 0 uses these bytes in the file. This byte is now unavailable for discrete data transfer.

Use care when planning block-transfer and discrete transfers of data to an adapter-mode processor channel.

ATTENTION: Use caution when performing data transfer. The discrete output data is over-written by the block transfer control on a group/module basis. If you write both types of transfer to the same group slot, unpredictable machine operation and possible damage to equipment or injury to personnel can occur.

Design Tip

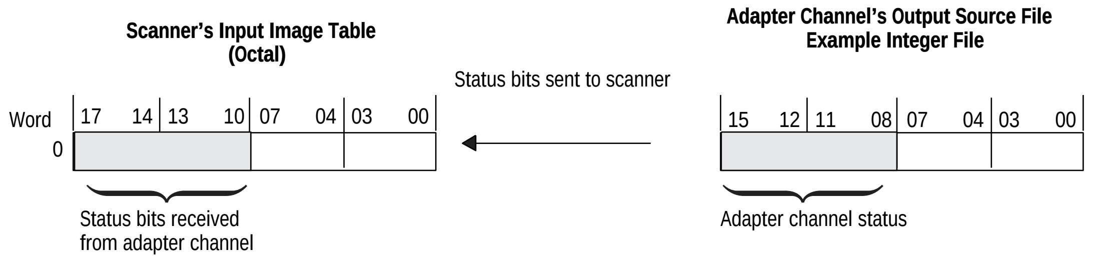
Do not program a block-transfer to group 0, module 1 since this area of the discrete transfer configuration file is used for communication status exchanges between the supervisory processor and the adapter-mode processor channel. For example:



Scanner's Input Image Table
Adapter Channel's Output Source File Example Integer File
Status bits sent to scanner
Word
0
17 14 13 10 07 04 03 00
15 12 11 08 07 04 03 00
module 1
Scanner's Output Image Table
Adapter Channel's Input Destination File Example Integer File
Word
0
17 14 13 10 07 04 03 00
15 12 11 08 07 04 03 00
module 1

Monitoring the Status of the Adapter Channel

The supervisory processor receives status bits from the adapter-mode processor in word 0 of the input image table for the rack that the adapter-mode processor is emulating.



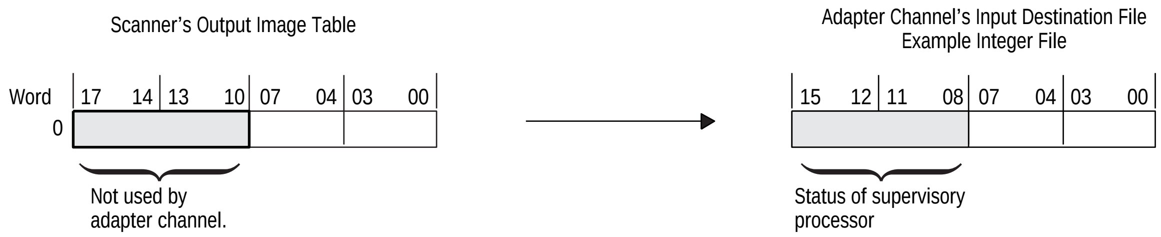
Scanner's Input Image Table (Octal)
Adapter Channel's Output Source File Example Integer File
Status bits sent to scanner
Word
0
17 14 13 10 07 04 03 00
15 12 11 08 07 04 03 00
Status bits received from adapter channel
Adapter channel status

When this bit(s):	Is:	It indicates:
10 octal (8 decimal) **and** 15 octal (13 decimal)	0	adapter-mode processor is in run mode
10 octal (8 decimal) **and** 15 octal (13 decimal)	1	adapter-mode processor is in program or test mode

Write ladder logic in the supervisory processor to monitor the rack-fault bits for the rack that the adapter-mode processor channel is emulating to determine the status of the remote I/O link.

Monitoring the Status of the Supervisory Processor

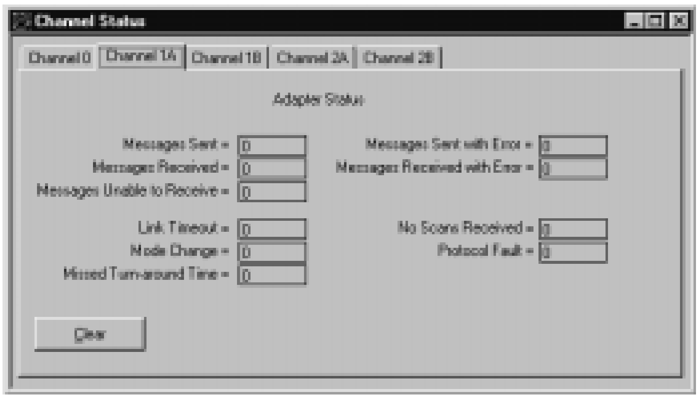
The adapter-mode processor channel reserves bits 10-17 of the first word of the input destination file for status. These bits tell the adapter-mode processor channel the status of the supervisory processor and the integrity of the remote I/O communication link.


Scanner's Output Image Table
Word
0
17 14 13 10 07 04 03 00
Not used by adapter channel.
Adapter Channel's Input Destination File Example Integer File
15 12 11 08 07 04 03 00
Status of supervisory processor

When this bit(s):	Is:	It indicates that the adapter-mode processor:
10 octal (8 decimal)	1	detects a communication failure or receives a reset command from the supervisory processor will be set if either bit 11 octal (9 decimal) or bit 15 octal (13 decimal) is set
11 octal (9 decimal)	1	receives a reset command from the supervisory processor (processor in program or test mode)
13 octal (11 decimal)	1	detects that the supervisory processor has powered up; this bit is reset with the first communication from the supervisory processor
15 octal (13 decimal)	1	detects a communication failure (e.g., no communication activity on the remote I/O communication link within the last 100 ms)

Monitoring Remote I/O Adapter Channels

To monitor channels that are configured to support adapter mode, use the adapter mode status screen. The data displayed is stored in the diagnostic file you defined in the adapter mode configuration screen of your programming software.



Messages Sent = 0
Messages Received = 0
Messages Unable to Receive = 0
Messages Sent with Error = 0
Messages Received with Error = 0
Link Timeout = 0
Mode Change = 0
Missed Turn-around Time = 0
No Scans Received = 0
Protocol Fault = 0

Status Field	Location	Description
Messages sent	word 1	Displays the number of messages sent by the channel.
Messages sent with error	word 3	Displays the number of messages containing errors sent by the channel.
Messages received	word 0	Displays the number of error-free messages received by the channel.
Messages received with error	word 2	Displays the number of messages containing errors received by the channel.
Messages unable to receive	word 4	Displays number of messages that contained protocol errors or packets that were garbled by the adapter.
Link timeout	word 5	Displays the number of times a timeout occurred on the remote I/O link.
No scans received	word 6	Displays the number of times an adapter channel did not receive a packet addressed to itself.
Mode changed	word 7	Displays the number of times the adapter channel switched to online.
Protocol fault	word 8	Displays the number of invalid I/O messages the adapter channel received.
Missed turn-around time	word 9	Displays the number of times the adapter channel took longer than 2 ms to process a message packet. The turn around-time for message packet processing is 2 ms.

Notes:

Communicating with Extended-Local I/O

Using This Chapter

This chapter explains how to configure the processor to communicate with extended-local I/O:

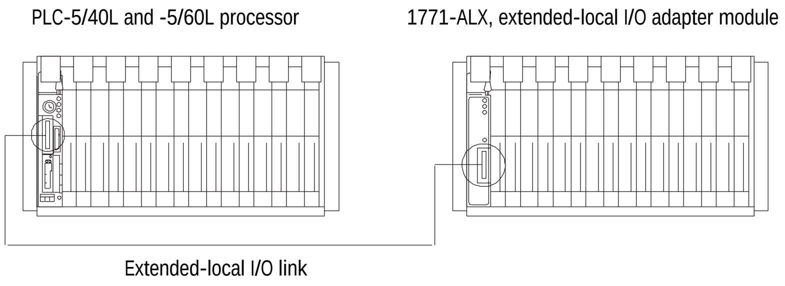
1. Configure channel 2 for extended-local I/O.
2. Define a diagnostic file.

   Use a unique, unused integer file.
3. Define the scan list.

Selecting Devices That You Can Connect

The only products that can form the extended-local I/O link are the PLC-5/40L and -5/60L processors and the extended-local I/O adapter module.

The extended-local I/O processor cannot be an extended-local I/O adapter.


PLC-5/40L and -5/60L processor
1771-ALX, extended-local I/O adapter module
Extended-local I/O link

Cabling

Design Tip

The maximum cable length for an extended-local I/O system is 30.5 cable-m (100 cable-ft). Connect extended-local I/O adapters by using any of these cables:

Cable Length:	Catalog Number:
1 m (3.3 ft)	1771-CX1
2 m (6.6 ft)	1771-CX2
5 m (16.5 ft)	1771-CX5

Important: You cannot connect or splice extended-local I/O cables to form a custom cable length. For example, if you have a distance of 4 m between two extended-local I/O adapters or between a processor and an extended-local I/O adapter, you cannot connect two 2 m cables together. You would have to use the 5 m cable and have the extra 1 m as slack.

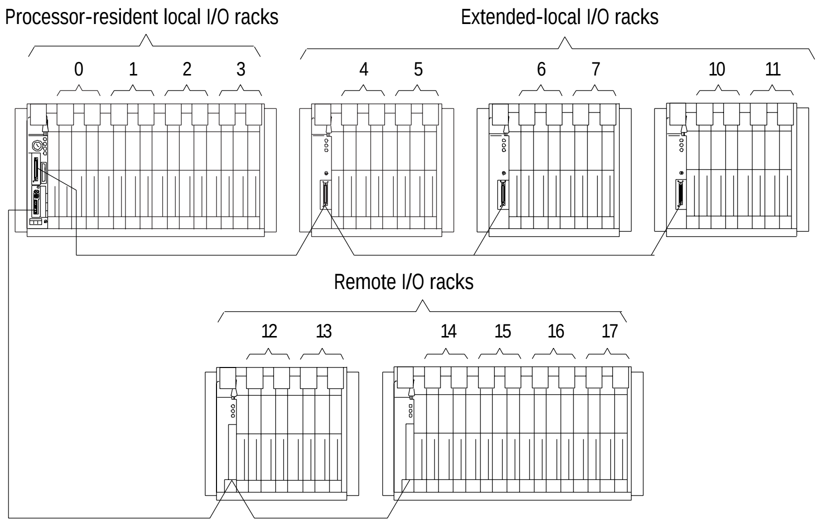
Terminate the link by installing the local I/O terminator (1771-CXT) on the last adapter module. The system will not run without it. The terminator is included with the processor.

Addressing and Placing I/O

Design Tip

When a PLC-5/40L or -5/60L processor is used to scan both extended-local I/O and remote I/O racks, the total of both remote I/O and extended-local I/O racks must not exceed the maximum number of racks allowed for the processor (16 racks for a PLC-5/40L or 24 racks for a PLC-5/60L). Figure 8.1 shows a PLC-5/40L processor controlling both extended-local I/O and remote I/O racks.

Figure 8.1
PLC-5/40L Processor with 16-rack Addressing Capability (Split Between Extended-Local I/O and Remote I/O)



Note: Racks numbers do not need to be consecutive per channel. For example, remote I/O racks can be numbered 6, 7, 14, 15, 16, and 17, while extended-local I/O racks can be numbered 4, 5, 10, 11, 12, and 13. 18584

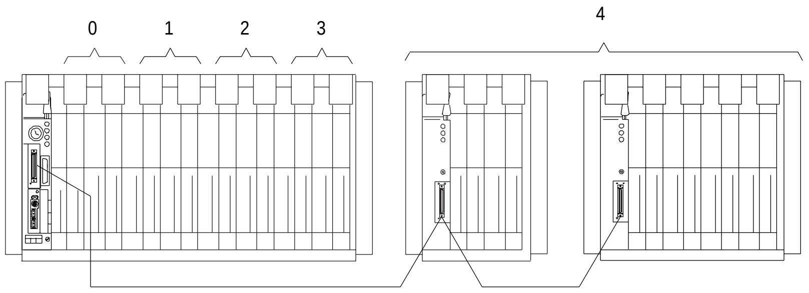
The PLC-5 processor and the 1771-ALX adapter module automatically allocate the next higher rack number(s) to the remaining I/O group(s) of the chassis. For example, if you select 1/2-slot addressing for your processor-resident local chassis and you are using a 16-slot (1771-A4B) chassis, the processor will address racks 0, 1, 2, and 3 in this chassis.

Design Tip

When assigning a rack number to extended local I/O, follow these guidelines:

- Do not split a rack number between extended-local I/O and remote I/O. For example, if you use a partial rack for remote I/O, you cannot use the remaining partial rack for extended-local I/O. See Figure 8.2.
- You can distribute extended-local I/O racks across multiple chassis on the extended-local I/O bus. See Figure 8.2.

Figure 8.2
Extended-local I/O Rack Number Assigned to Multiple I/O Chassis



- You can select a different hardware addressing method for each extended-local I/O chassis in your PLC-5 system.
- You cannot configure more than one rack to have the same starting rack number and module group; that is, you cannot use chassis to chassis complementary I/O.

Design Tip

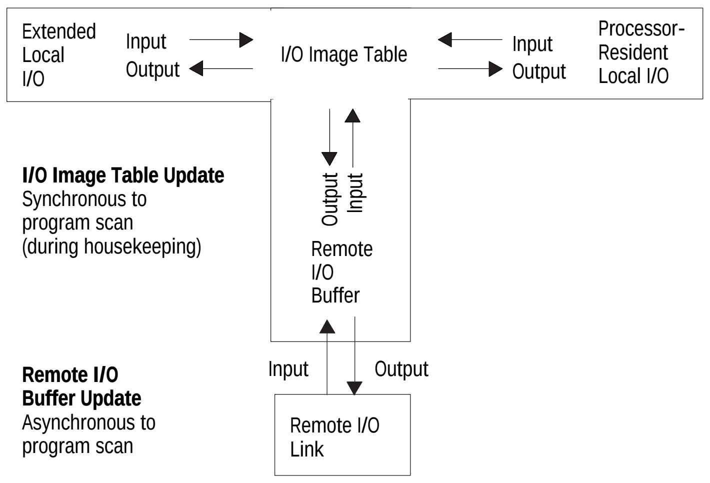
Follow these guidelines when you plan your extended-local I/O system.

- Do not configure processor input interrupts (PIIs) for inputs in an extended-local I/O chassis. The PII inputs must be in the processor-resident local I/O rack.
- You can either use 32-point I/O modules and any addressing method or use 1771-IX or -IY modules and any addressing method in extended- local I/O racks. You must specify the type of I/O modules you are using by setting the configuration plug on the extended-local I/O adapter.
- If you need to use a thermocouple module and 32-point I/O modules in the same I/O chassis, use the 1771-IXE module.

Transferring Data

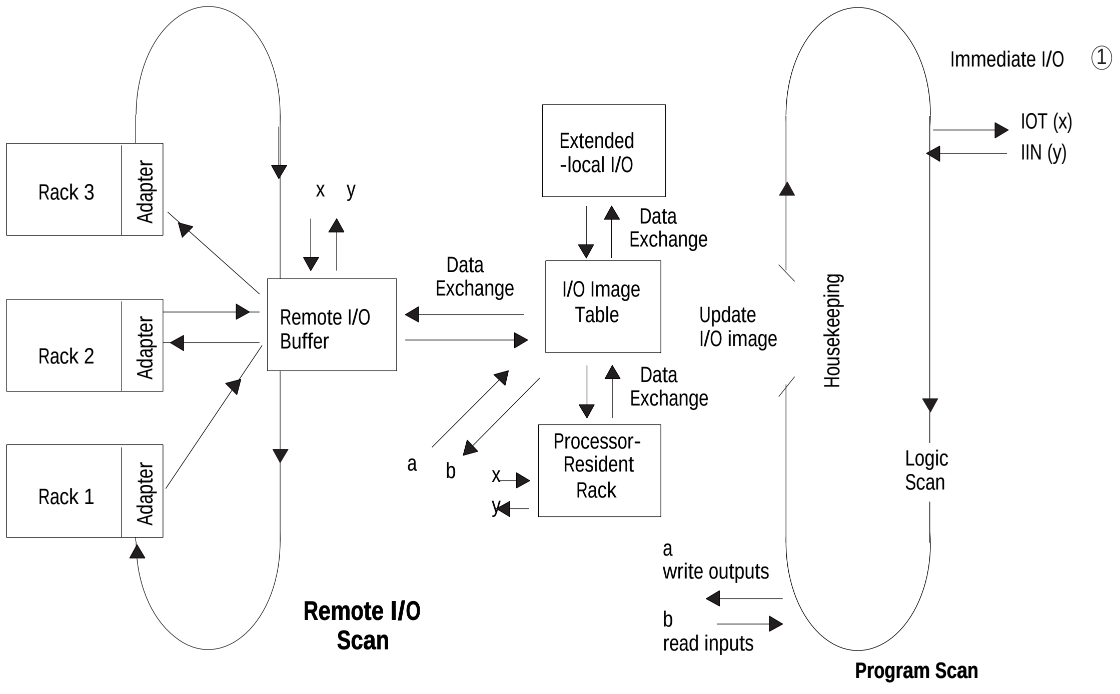
The PLC-5/40L or -5/60L processor can scan processor-resident I/O, extended local I/O, and remote I/O. Figure 8.3 shows how a PLC-5/40L or -5/60L processor accomplishes I/O scanning and update.

Figure 8.3
PLC-5/40L and -5/60L I/O Scanning and Update



Discrete Data Transfer

The processors scan the extended-local I/O chassis during the housekeeping portion of the program scan. Extended-local I/O discrete data is exchanged between the processor's data table image and the I/O in the extended-local I/O chassis.

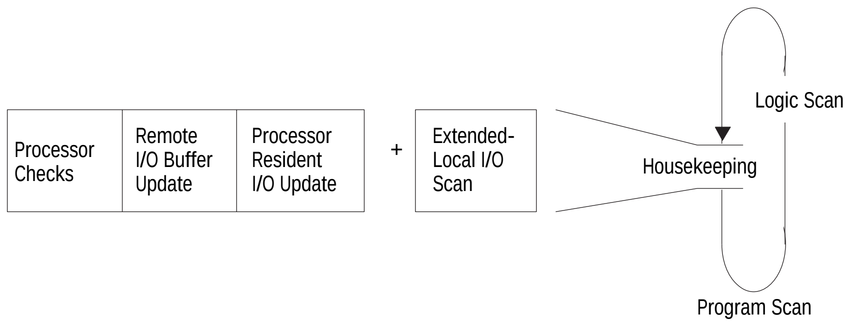


Data exchange occurs during housekeeping. Outputs are written to and inputs read from the I/O image table during the logic scan.

① IIN and IOT data transfer directly to and from I/O modules extended-local I/O chassis.

The time that it takes to scan extended-local I/O chassis is added to the housekeeping time. See Figure 8.4.

Figure 8.4
PLC-5/40L and -5/60L Extended-Local I/O Scan Time



The time in ms that it takes to scan extended-local I/O chassis depends on the number of 1771-ALX adapter modules and the number of extended- local I/O racks. The formula used to calculate the total time to scan extended-local I/O chassis is:

```
extended-local I/O scan time = (0.32 ms x A)+(0.13 ms x L)
```

where:

A = the number of 1771-ALX modules and

L = the number of racks in the extended-local I/O system

Example: If you have three 1771-ALX modules in three chassis and a total of 4 racks, the total time is calculated as follows:

```
extended-local I/O scan time = (0.32 ms x 3)+(0.13 ms x 4)
extended-local I/O scan time = 1.48 ms
```

```
housekeeping time = 1.48 ms (extended-local I/O) + 4.50
ms(other housekeeping)
```

```
housekeeping time = 5.98 ms
```

Transferring Block Data

Requests for block-transfer data occur during the logic scan. Concurrent with the execution of the program logic, block-transfer requests are forwarded to the appropriate 1771-ALX adapter module(s) and data is transferred. A 1771-ALX adapter module may start block-transfer operations to multiple slots and have block-transfer data transactions on-going in parallel within the I/O chassis.

The block-transfer duration shown above does not affect logic scan time. This transfer of data occurs concurrent with execution of program logic.

Block-transfer duration is the time interval between the enabling of the block-transfer instruction and the receipt of the done bit.

Calculating Block-Transfer Completion Time

You can calculate two types of block-transfer timing:

- worst-case calculation for the completion of all block-transfers in the system
- the time to perform a block-transfer for any one block-transfer module in the system

Calculating Worst-Case Completion Time

This formula assumes:

- block-transfer instructions are consecutively placed in the logic program
- block-transfer modules in the I/O chassis are ready to perform when operations are requested

block-transfer duration (ms) = D R

$$D = 2E \quad L + (0.1W)$$

and

$$R = \frac{\text{logic scan + housekeeping}}{\text{logic scan}}$$

or

$$R = 1 \text{ (when } D < \quad \text{logic scan time)}$$

where:

E = number of extended-local I/O chassis with block-transfer modules

L = largest number of block-transfer modules in any extended-local I/O chassis

W = number of words in the longest block-transfer request

Calculating Completion Time for any One Block-transfer

This formula assumes:

- block-transfer instructions are consecutively placed in the logic program
- block-transfer modules in the I/O chassis are ready to perform when operations are requested

block-transfer duration (ms) = D R

$$D = [2E \quad M + (0.1W)]$$

and

$$R = \frac{\text{logic scan + housekeeping}}{\text{logic scan}}$$

or

$$R = 1 \text{ (when } D < \quad \text{logic scan time)}$$

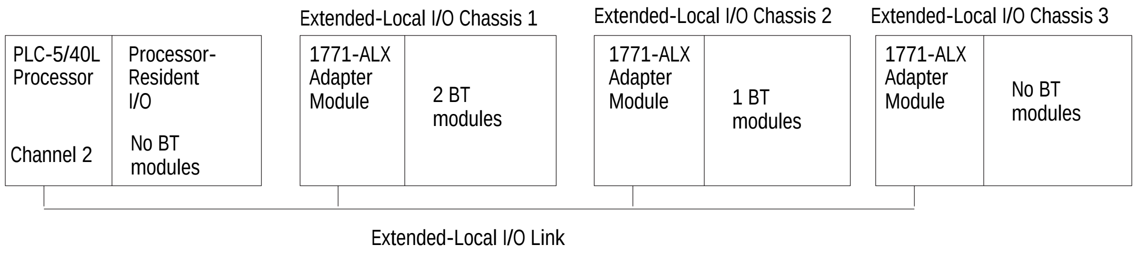
where:

E = number of extended-local I/O chassis with 1771-ALX adapter modules and block-transfer modules

M = number of block-transfer modules in the chassis of the module being calculated

W = number of words in block-transfer request being calculated

Example Calculations:
Here is an example system that provides sample calculations of a worst case block-transfer completion time and the completion time of the modules in chassis 2.



The logic scan completes in 15 ms. Housekeeping completes in approximately 6 ms (as calculated in the formula on page NO TAG). The longest block-transfer request is 20 words.

Worst-case time (T) = D X R

```
D = 2E   L + (0.1W)  and  R = 1   Because 10 < 15 (which is the logic scan)
D (ms) = (2   2)   (2) + (0.1   20 )]
D = 10 ms
T = 10   1
T = 10 ms
```

Completion time (T) for module in chassis 2 transfer: = D X R
Block-transfer length = 20

```
D = 2E   M + (0.1W)    and    R = 1   Because 6 < 15 (which is the logic scan)
D (ms) = (2   2)   (1) + (0.1   20 )]
D = 6 ms
T = 6   1
T = 6 ms
```

Design Tip

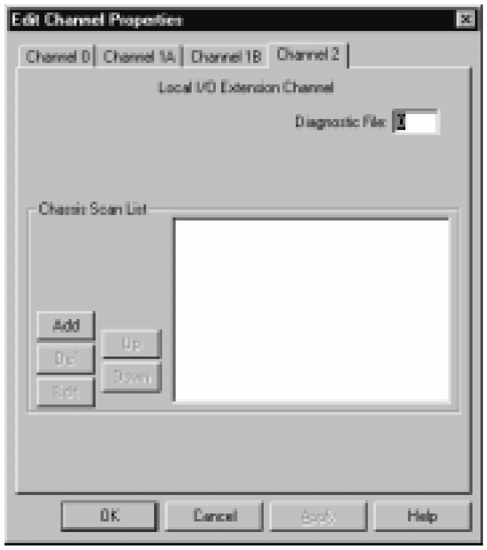
Considerations for Extended-local Racks

The following are programming considerations when you are transferring block data in an extended-local rack:

- Block-transfer instructions to any of the following modules that reside in an extended-local rack will result in frequent checksum errors:
  - 1771-OFE1, -OFE2, and -OFE3 modules, all versions prior to series B, revision B
  - IMC-120 and IMC-123, all versions
- If you are using block-transfer to a 2760-RB module located in the extended-local rack, make sure you **do not set the timeout bit** in the block-transfer control file.

Configuring the Processor as an Extended-Local I/O Scanner

To configure the extended-local I/O (channel 2), use the extended-local I/O configuration screen.



This field:	Specifies:	Configure by doing the following:
Diagnostic file	The file containing the channel's status information	Cursor to the field enter enter an integer file number (9-999). **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used integer file. Unpredictable machine damage can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want to get status information for that channel.
Scan list	The channel I/O configuration	See the next section for information on creating and modifying a scan list.

The scan list for extended-local I/O is similar to the scan list for remote I/O. The differences are:

- The remote I/O scan list displays rack size. The rack size is determined by the chassis size (number of slots) and backplane addressing used by the chassis. Table 8.A explains the relationship among chassis size, backplane addressing and rack size.

Table 8.A
How Chassis Size and Backplane Addressing Determine the Quantity of I/O Racks

If you are using this chassis size:	And 2-slot addressing (single density)	Or 1-slot addressing (double density)	Or 1/2-slot addressing (quad density)
4-slot	1/4 logical rack	1/2 logical rack	1 logical rack
8-slot	1/2 logical rack	1 logical rack	2 logical racks
12-slot	3/4 logical rack	11/2 logical rack	3 logical racks
16-slot	1 logical rack	2 logical racks	4 logical racks

  On the extended-local-I/O scan list, the actual chassis size and backplane addressing is displayed, not the rack size.

- The scan list for extended local I/O has one entry for each physical chassis on the extended local I/O channel.

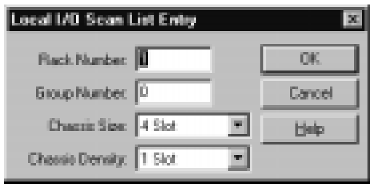
  If an adapter on a remote I/O channel resides in a physical chassis that contains more than one I/O rack, more than one entry appears on the remote I/O scan list for that single chassis.

  Figure 8.5 shows the scan list for both remote I/O and extended local I/O. Each channel shows a 16-slot chassis using 1-slot addressing with a starting address of rack 4, module group 0. This chassis contains logical racks 4-5.

Figure 8.5
Remote I/O Scan List vs Extended-local I/O Scan List

Remote				Extended				
Rack ##	Starting Group	Rack Size	Range	Rack Address	Starting Group	Chassis Size	Backplane Addressing	Range
4	0	FULL	040-047	4	0	16-SLOT	1-SLOT	040-057
5	0	FULL	050-057					

A scan list includes the following:



Rack Number: 1
Group Number: 0
Chassis Size: 4 Slot
Chassis Density: 1 Slot

For this field:	A scan list contains:
Scan rack address	1-17 octal (PLC-5/40L processors) 1-27 octal (PLC-5/60L processors)
Starting group number	0, 2, 4, or 6
Chassis size	4-slot, 8-slot, 12-slot, 16-slot
Backplane addressing	1-slot, 2-slot, or 1/2-slot
Range	Automatically calculated based upon rack address, starting module group and chassis size. An asterisk (*) after a range indicates the last valid rack entry.

Design Tip

Keep the following limitations in mind when creating/modifying your scan list:

- A scan list only can have 16 entries because only 16 adapters can be on channel 2.
- The automatic configuration always displays the actual hardware configuration, except for chassis that have their global inhibit bit set. In this case, that global bit overrides the automatic configuration. You must first clear the global inhibit bits for all chassis on the channel, and then use the autoconfigure function.

  Clear global inhibit bits by using the processor status status screen.

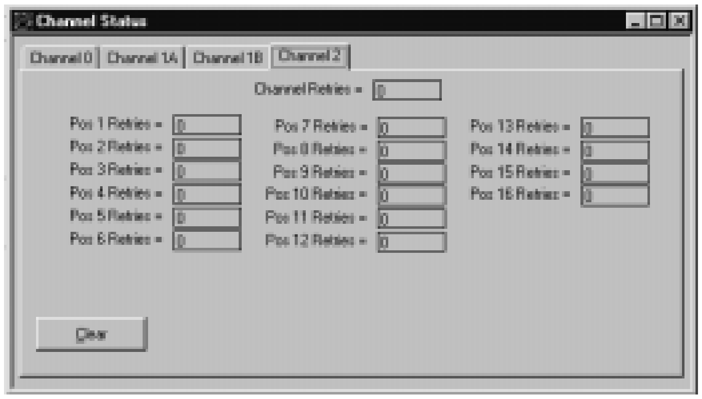
- A scan list can have a maximum of 16 chassis. Entries cannot be repeated on the scan list.

Use the following table for information about creating/modifying your scan list:

To:	Do the following:
Create a scan list	Make sure the processor is in Remote Program or Program mode. **1.** Make sure that you defined an I/O status file on the processor configuration screen. **2**. Accept any edits made to the channel configuration. **3**. Use the autoconfiguration function If you have errors when you accept edits, clear the scan list and accept edits again. If some or all adapters are not on the scan list and should be, check to see that they are powered-up and that the channels are connected properly. Also verify that all switch settings on the adapters are set correctly.
Insert an entry into the scan list	Make sure the processor is in Remote Program, Program, or Remote Run mode. **1.** Position the cursor at the place on the scan list where you want to insert an entry. **2.** Insert an entry into the list and enter the appropriate values for the list. **Important:** If incorrect information is entered for an entry, the processor will not display the new configuration when you save edits.
Delete an entry for the scan list	Make sure the processor is in Remote Program, Program, or Remote Run mode. **1.** Position the cursor at the place on the scan list where you want to delete an entry. **2.** Delete the entry from the list. **Important:** If incorrect information is entered for an entry, the processor will not display the new configuration when you save edits.

Monitoring Extended-Local I/O Status

To monitor extended-local I/O of PLC-5/40L and PLC-5/60L processors, use the extended local I/O status screen in your programming software.



Channel Retries = 0

Pos 1 Retries = 0
Pos 2 Retries = 0
Pos 3 Retries = 0
Pos 4 Retries = 0
Pos 5 Retries = 0
Pos 6 Retries = 0
Pos 7 Retries = 0
Pos 8 Retries = 0
Pos 9 Retries = 0
Pos 10 Retries = 0
Pos 11 Retries = 0
Pos 12 Retries = 0
Pos 13 Retries = 0
Pos 14 Retries = 0
Pos 15 Retries = 0
Pos 16 Retries = 0

Status Field:	Location:	Description:
Channel retry	word 0	Displays the number of times extended local I/O scanner tried and failed to communicate with all adapters on the channel. This value is the sum of all adapter retry counts.
Retry	word 10 word 20 word 30 etc. word 160	Displays the number of retries for the corresponding rack entry (word numbers are in multiples of 10). Entry 1 Entry 2 Entry 3 etc. Entry 16

Important: Setting inhibit bits in the I/O status file does not update inhibit bits in the processor status file.

Notes:

Maximizing System Performance

Using This Chapter

For information about the time that it takes the processor to execute a specific instruction, see chapter 22.

Program Scan

Since the program scan is comprised of the logic scan and housekeeping, any event that impacts the time of one segment affects the program scan.

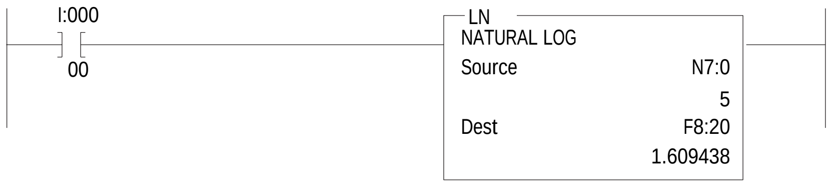
You can monitor the scan time by using the processor status screen in your programming software.

If no change in input status occurs and the processor continues to execute the same ladder logic instructions, the program scan cycle is consistent. In real systems, however, the program scan cycle fluctuates due to the following factors:

- false logic executes faster than true logic
- different instructions execute at different rates
- different input states cause different sections of logic to be executed
- interrupt programs affect program scan times
- editing programs while online affects housekeeping times

Effects of False Logic versus True Logic on Logic Scan Time

The rung below—which changes states from one program scan to the next—will change your scan time by about 400 µs.

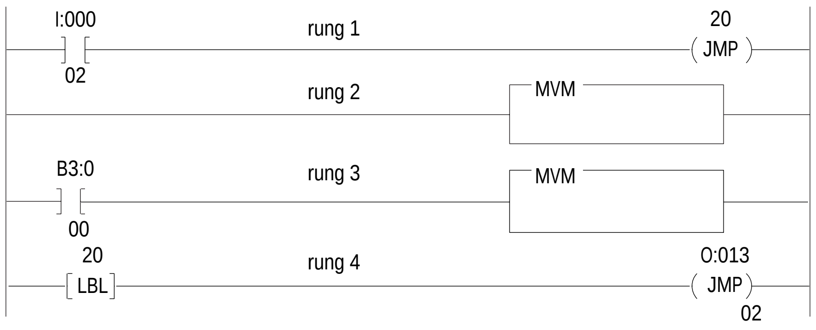


If I:000/00 is:	Then the rung is:
On	True, and the processor calculates the natural log. A natural log instruction takes 409 µs to execute.
Off	False, and the processor scans the rung but does not execute it. It takes only 1.4 µs to only scan the rung.

Other instructions may have a greater or lesser effect.

Effects of Different Input States on Logic Scan Time

You can write your logic so that it executes different rungs at different times, based on input conditions. The amount of logic executed in logic scans causes differences in program scan times. For example, the simple differences in rung execution in the following example cause the program scan to vary.



If I:000/02 is:	Rungs 2 and 3 are:
On	Skipped
Off	Executed

If you use subroutines, program scan times can vary by the scan time of entire logic files.

Effects of Different Instructions on Logic Scan Time

Some instructions have a much greater effect on logic scan time than others based on the time that it takes to execute each instruction.

Program scan time is also affected by the construction of your ladder rungs. The size of the rung and the number of branches can cause the scan time to fluctuate greatly.

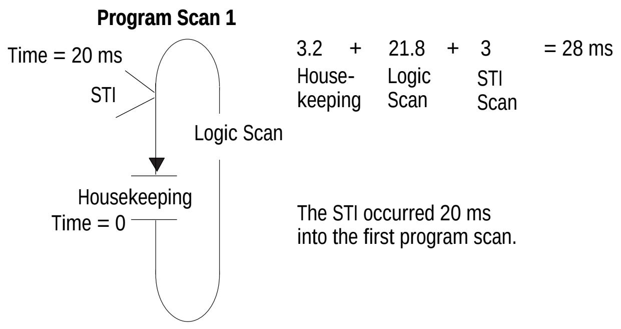
Effects of Using Interrupts on Logic Scan Time

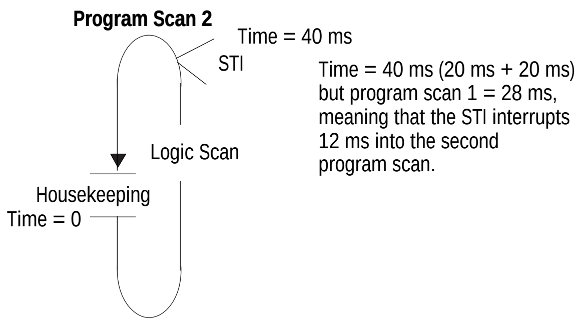
Program scan time is also affected by interrupt programs. An interrupt is a special situation that causes a separate program to run independently from the normal program scan. You define the special event and the type of interrupt that is to occur. For more information on interrupt programs, see chapters 18 and 19.

For example, a selectable timed interrupt (STI) is a program file that you define to execute once every time period. The example shown below has these parameters:

- you configure an STI to execute every 20 ms
- the STI program takes 3 ms to execute
- the logic scan is 21.8 ms
- housekeeping takes 3.2 ms

The first program scan in this example lasts a total of 28 ms. The program scans look like:




Program Scan 2
Time = 40 ms
STI
Time = 40 ms (20 ms + 20 ms) but program scan 1 = 28 ms, meaning that the STI interrupts 12 ms into the second program scan.
Logic Scan
Housekeeping
Time = 0

Because the first program scan takes 28 ms, the STI actually occurs 12 ms into the second program scan (28 + 12 = 40, which is the time for the second STI to occur). This example points out that when the STI time period is different than the program scan time, the STI occurs in different places in the program scan. Also note that, due to fluctuations in program-scan times, multiple STIs may be executed during one scan and no STIs during other scans.

Effects of Housekeeping Time

In PLC-5 processors, basic housekeeping takes 3.5 ms. If it takes the processor 21.8 ms to execute a ladder program, the overall program scan time is 25.3 ms. Any increase in housekeeping affects your program scan.

The following activities can increase housekeeping time:

- editing while in remote run mode
- putting block-transfer modules in the processor-resident chassis
- using the global status flag files

Editing While in Remote Run Mode

The online editing times for ladder programs are as follows:

For this editing operation:	And this type of program:	The times are:
Accept Rung (after inserting, modifying, or deleting a rung edit)	other than the edited file	0.35 ms per 1000 words
	no labels	3 ms + 0.35 ms per 1000 words
	with labels	3.5 ms + 0.35 ms per 1000 words
Test Edits of the program (impacts one program scan)	▨	0.2 ms to change the status of edits from TEST to UNTEST or UNTEST to TEST
Assemble Edits	no edits pending	0.35 ms per 1000 words
	edits pending, no labels	2.0 ms + 1.5 ms per 1000 words
	edits pending, with labels	2.0 ms + 1.9 ms per 1000 words

Important: Editing programs online also delays the execution of PIIs and STIs.

Putting Block-Transfer Modules in Processor-Resident Chassis

Because processor-resident racks cannot be updated until after active block-transfers are completed, putting block-transfer modules in the processor-resident chassis can affect housekeeping by a worst-case time of approximately *100 μs per one word of block-transfer data*. Note that this estimate is based on a worst-case scenario. Typically, the effect, if any, on housekeeping will be minimal.

Using Global Status Flag Files

The global status flag files are updated during housekeeping. This increases housekeeping time as follows:

- each global status flag file on a channel (for example, channel 1A or 1B) adds 3ms
- housekeeping time does not increase more than 6ms, even if there are more than two global status flag files

Design Tip

If you need two global status flag files, split them across two channels.

Calculating Throughput

Throughput is the time that it takes for an output to be energized after its associated input has been energized. You need to consider the following components when evaluating throughput:

- input and output module delay
- I/O backplane transfer
- remote I/O scan time
- processor time

To calculate throughput, use the following equation:

Input Card Delay + I/O Backplane + Worst-Case Remote I/O Scan Time + Worst-Case Processor Time + Worst-Case Remote I/O Scan Time + I/O Backplane + Output Card Delay

Input and Output Modules Delay

All input and output modules have a “delay time,” which is the time that it takes the module to transfer information to/from the I/O backplane through the I/O module to/from the field device.

Depending on the type of modules you are using, these delay times vary; but, the times must be taken into account when calculating system throughput. Choose modules that perform the function that you need with the lowest possible delay times.

I/O Backplane Transfer

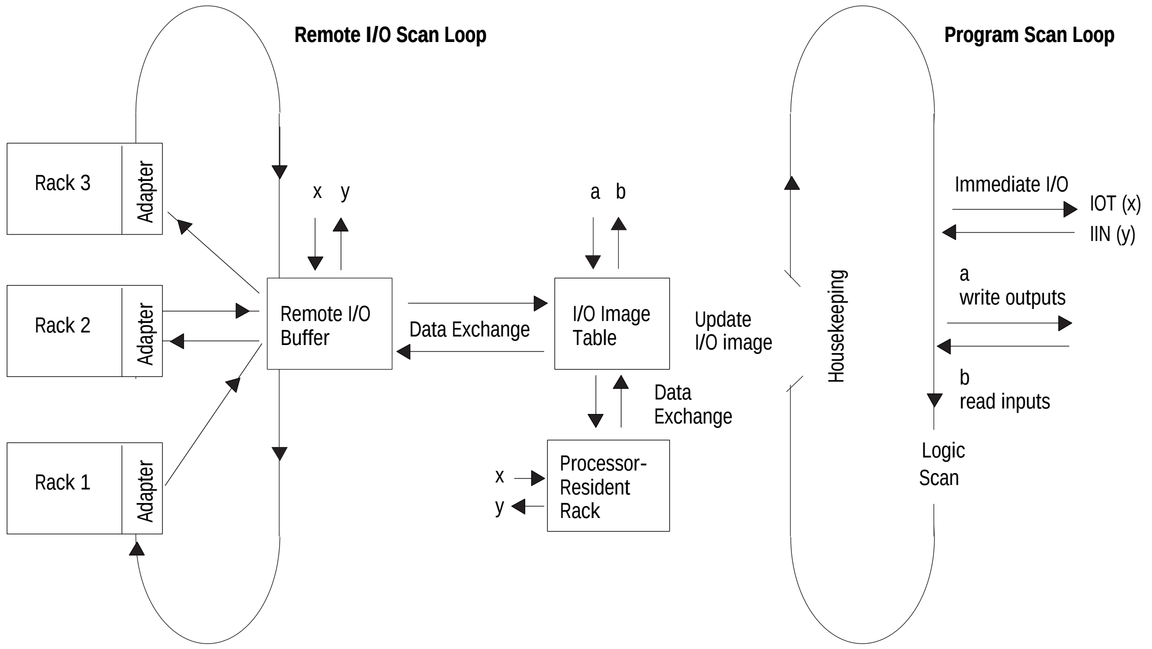
The I/O backplane transfer time is the time it takes for the 1771-ASB adapter module to exchange data with the I/O modules in the same chassis, generally 1-2 ms for a full I/O rack.

This time is fairly insignificant compared to total system throughput, but can be optimized in situations where empty slots or modules that use only backplane power in the chassis exist. For example, if the last four slots of a rack contain a 1785-KA module and power supply (with two empty slots), the 1771-ASB can be configured to ignore those last four slots.

For more information about configuring adapter modules, see the *1771 Remote I/O Adapter Module User Manual*, publication 1771-6.5.83.

Remote I/O Scan Time

The remote I/O scan time is the time it takes for the scanner to communicate with each device in the remote I/O system.



These three factors affect the remote I/O scan time:

- communication rate
- number of rack entries
- block-transfers

Communication Rate

The communication rate determines the time it takes for the scanner to communicate with each individual entry in its scan list. Table 9.A lists the amount of time required to communicate to a device at each communication rate.

Table 9.A
Communication Times at Different Communication Rates

Communication Rate (kbps):	Time (ms):
57.6	10
115.2	7
230.4	3

Note that these are full rack times. Smaller racks will decrease this time.

If four full-rack entries are in the scan list, the I/O scan for that channel at 57.6 kbps is 4 x 10 = 40 ms. If you change the communication rate to 230.4 kbps, the I/O scan decreases to 4 x 3 = 12 ms.

Number of Rack Entries

You determine the total remote I/O scan time in the remote I/O system by this formula:

`total remote I/O scan time` `= # of rack entries X time per rack-entries in the scan list` (see Table 9.A on page 7)

If one channel has twice as many racks as another, for example, the scan time for the first channel is twice as long.

To optimize this scan time, divide your I/O racks between multiple channels. Place your most time-critical I/O on one channel, and non-time-critical I/O on the other channel. Since all I/O channels are independent, a long remote I/O scan on one channel will not affect the remote I/O scan on another channel.

Block-Transfers

A block-transfer is an interruption of the normal remote I/O scan in order to transfer a block of data to a specific I/O module. Most of the time that the processor spends in performing the block-transfer is for the handshaking that occurs between the processor and the block-transfer module. This handshaking is embedded in the discrete I/O transfer and has no effect on the remote I/O scan. The remote I/O scan is affected when the actual data transfer occurs.

The amount of time that the block-transfer interrupts the remote I/O scan depends on the number of words being transferred, the communication rate, and associated overhead:

Use this formula and the table below to calculate block-transfer time:

```
block-transfer time = (number of words being transferred   ms/word based
on the communication rate) + overhead for the communication rate
```

Communication Rate (kbps):	ms/Word:	Overhead (ms):
57.6	.28	3
115.2	.14	2.5
230.4	.07	2

For example, if the communication rate is 115.2 kbps and you want to block-transfer 10 words, the interruption of the remote I/O scan is:

(10 x .14) + 2.5 = 1.4 + 2.5 = 3.9 ms

For the particular remote I/O scan in which the block-transfer takes place, 3.9 ms will be added to the remote I/O scan time.

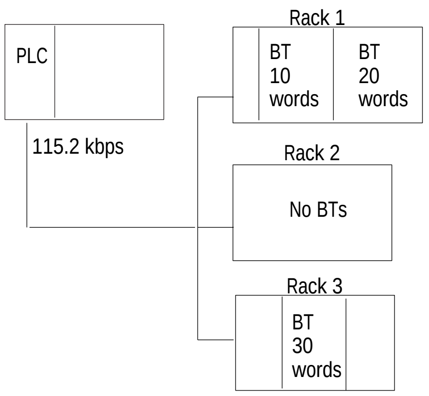
Important: If you select the baud rate as 230.4 kbps, and you are using the serial port or a PLC-5 coprocessor, use channel 2 for better overall system performance.

Calculating Worst-Case Remote I/O Scan Time

Since it is impossible to predict within which remote I/O scan a block-transfer will occur, you only can calculate the worst-case remote I/O scan time. To calculate the worst case time:

1. Determine the normal I/O time (without block-transfers)
2. Add the time of the longest block-transfer to each entry in the scan list. (The processor can only perform one block-transfer per entry in the scan list per I/O scan.)

For example, if your system is:



Worst-case I/O scan:

(3 x 6)	3 racks at 115.2 kbps*normal I/O scan
+ (20 x .14) + 2.5	longest BT in rack 1
+ 0	no BTs in rack 2
+ (30 x .14) + 2.5	longest BT in rack 3

18 + 5.3 + 0 + 6.7 = 30 ms

Optimizing Remote I/O Scan Time

The best way to optimize your scan time is to place your most time-critical I/O on a separate channel from non-critical I/O. If you have only one channel available for I/O, however, you can still optimize the scanning by using the processor's configurable scan list.

In a normal 4-rack system, the scan list would be: rack 1
rack 2
rack 3
rack 4

If you are using 57.6 kbps, the normal I/O scan is 4 racks x 10 ms = 40 ms. Each entry is of equal priority, so each rack is scanned every 40 ms.

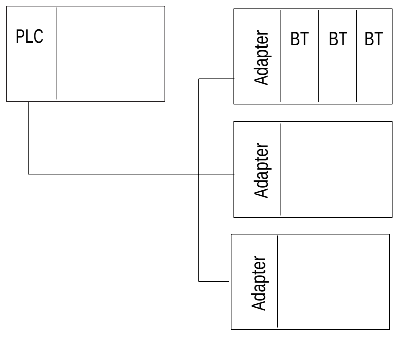
However, if rack 2 has the most time-critical I/O, use the configurable scan list to specify: rack 1
rack 2
rack 3
rack 2
rack 4
rack 2

Using this scan list, rack 2 is scanned every other rack. The list has 6 entries, so the normal I/O scan time is 6 x 10 ms = 60 ms. Since rack 2 is scanned every other rack, however, the rack 2 **effective** scan time is 2 x 10 ms = 20 ms. The remaining racks are scanned every 60 ms. Thus, the tradeoff for the more frequent scanning of rack 2 (every 20 ms) means that the other racks are scanned only every 60 ms.

You can also optimize block-transfers within the channel. You block-transfer to only one block-transfer module per entry in the scan list per I/O scan. If you have three block-transfer modules in one I/O rack, it takes a minimum of three I/O scans to complete the block-transfers to all of the modules:

System Optimized for Discrete-Data Transfer

With this arrangement, only one block-transfer can occur to each BT module for every 3 discrete I/O scans.



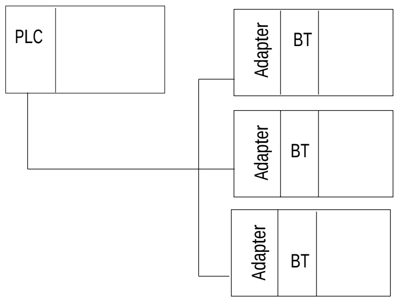
Maximum scan time = 3 discrete scans + 1 block-transfer
= 3D + 1BT

Minimum time to complete a block-transfer to all modules = 3 (3D + 1BT)
= 9D + 3BT

If you place the three block-transfer modules in different racks, however, you can block-transfer to all three modules in one I/O scan. To optimize your system layout for **block-data transfers**, use an arrangement similar to the following:

System Optimized for Block-Data Transfer

With this arrangement, a block-transfer to each BT module can occur in a single discrete I/O scan.



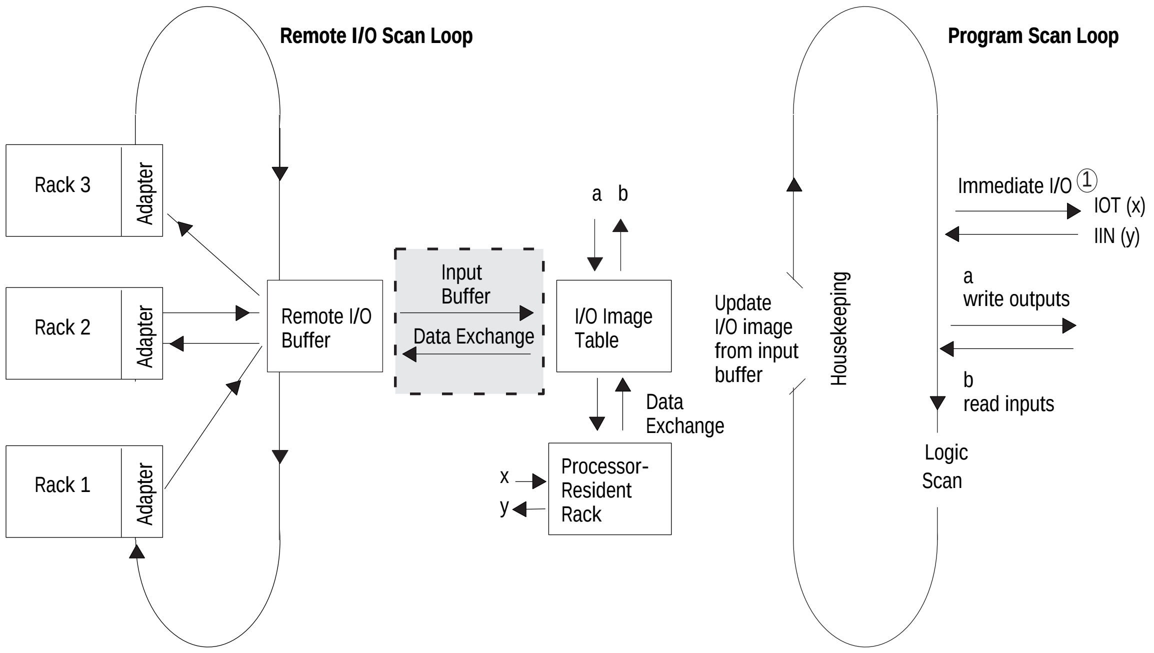
Maximum scan time = 3 discrete scans + 3 block-transfers
= 3D + 3BT

Minimum time to complete a block-transfer to all modules = 1 (3D + 3BT)
= 3D + 3BT

Processor Time

The processor time is the time needed to process the inputs and set the corresponding outputs. This processor time varies for different processors and is based on input buffering, program scan, etc.

In a PLC-5 system, inputs are buffered between the I/O image table and the remote I/O buffer. The movement of inputs from the remote I/O buffer to the input buffer is asynchronous to the movement of data from the input buffer to the input image table.



The worst-case processor time is:

Variable:	Value:
periodic input buffer update from remote I/O buffer	10 ms
one program scan to guarantee inputs received	xx ms
one program scan to guarantee outputs received	xx ms
0.18 ms times number of racks	xx ms
total	

For a 3-rack system with a 20 ms program scan, the worst-case processor time is: $10 + 20 + 20 + (0.18 * 3) = 50.54$ ms.

Example Calculation

Based on the results of each throughput component calculation presented within the chapter, an example of a worst-case update time calculation is:

Variable:	Value:
input card delay	10 ms typical
I/O backplane	1 ms
worst-case remote I/O scan time	30 ms
worst-case processor time	50.54 ms
worst-case remote I/O scan time	30 ms
I/O backplane	1 ms
output card delay	1 ms typical
total	123.54 ms

Performance Effects of Online Operations

The performance of the PLC-5 processor is affected when you perform online operations via a DH+ link to your program files while in Run mode. Affected activities are:

- DH+ messages
- serial port messages
- channel 3A messages
- remote block-transfers

The amount of time that the messaging and block-transfers can be delayed is **proportional to the size (K words) of the ladder file**. Table 9.B lists the performance effects (when using any of the 6200 Series PLC-5 Programming Software releases that support the processor you are using).

Table 9.B
Worst-case Performance Effects When Performing Online Operations While in Run Mode

Effected Data Transfers:	Online Operations via any DH+ Channel:	
	Perform a Page Up/Page Down at the end of a program file:	Insert/Delete ladder rungs:
Remote block-transfers	20 ms/K words	50 ms/Kwords
DH+ messages	20 ms/K words	50 ms/Kwords
Serial port messages	200 ms/K words	50 ms/Kwords
Channel 3A messages	no impact	50 ms/Kwords

You should re-design your programs to avoid possible communication pauses if you currently:

- use large ladder logic program files
- have time critical remote block-transfers and/or serial, DH+, and channel 3A messages
- must edit the program online during run mode

For best processor performance, segment your program files by using modular programming design practices, such as main control programs (MCPs), sequential function charts (SFCs), and the jump to subroutine (JSR) instruction.

Effect of Inserting Ladder Rungs at the 56K-word Limit

This consideration applies to PLC-5/60, -5/60L, -5/80, and -5/80E processors when you are editing a program file that approaches the maximum file limit of 57,344 words.

Performing run-time or program-mode editing of ladder files that approach the maximum program file size of 57,344 words could:

- prevent the rung from being inserted
- cause suspension of the operation by 6200 Series PLC-5 Programming Software (release 4.3 and later)

To avoid or correct this problem, segment your program file using modular programming, such as main control programs (MCPs), sequential function charts (SFCs), and the jump to subroutine (JSR) instruction.

If you cannot segment your program file, save the file often while editing it.

If you encounter the error `Memory Unavailable for Attempted Operation`, then clear processor memory.

Using Program Control Instructions

Scan time can increase based on how you use JMP/LBL instructions and FOR/NXT instructions.

Using JMP/LBL Instructions

Keep in mind these issues when programming JMP/LBL instructions:

Instruction:	Consideration:
JMP	The execution time required for a JMP instruction depends on the program file that contains the JMP instruction. The estimated execution time for a JMP instruction is: 8.9 + (*file_number* – 2) ∗ 0.96 The greater the program file number, the longer it takes to complete a scan of the JMP instruction.
LBL	Each LBL instruction uses 2 words of memory in the program file plus additional memory, depending on the label number itself. Each label number is placed in a label table. Each entry in the label table uses 2 words of memory, starting from label 0. For example, LBL 10 uses 22 (2 words ∗ 11th entry) words of memory in the label table. If you later delete LBL 10, the label table does not deallocate previously used space. The only way to recover this space is to upload and then re-download the program.

Using FOR/NXT Instructions

The FOR/NXT instructions have the same impact on execution time as the JMP instruction. The execution for a FOR/NXT loop depends on the program file that contains the instructions.

The estimated execution time for a FOR/NXT loop is:
8.1 + (*number_of_loops* ∗ 15.9) + (*file_number* – 2) ∗ 0.96

The greater the program file number, the longer it takes to complete the FOR/NXT loops.

Notes:

Communicating with Devices on a DH+ Link

Using This Chapter

Selecting Devices That You Can Connect

You can use a DH+ link for data transfer to other PLC-5 processors or higher level computers and as a link for programming multiple PLC-5 processors. A PLC-5 processor can communicate over a DH+ link with other processors and with a personal computer. You can connect a maximum of 64 stations to a single DH+ link.

Table 10.A
Devices that You Can Connect

Product:	Catalog Number:	Application:	Required Cables:
Data Highway or Data Highway Plus (RS-232C or RS-422-A) Interface Module	1770-KF2	Connects an asynchronous (RS-232C) device to a Data Highway or DH+ network	1770-CD
Data Highway / Data Highway Plus on Broadband	1771-KRF	Media bridge connecting as many as 18 Data Highway networks to communicate over a facility-wide broadband cable system	
Communication Interface Card	1784-KL	Connects the T47 Portable Programming Terminal to DH+	1784-CP 1784-CP2 1784-CP3 1784-CP5 1784-CP6
Data Highway Plus XT/AT Interface Module	1784-KT	Connects IBM XT or AT compatible computers to DH+	
Data Highway Plus PS/2 Interface Module	1784-KT2	Connects IBM PS/2 compatible computers to DH+	
Data Highway Plus to Data Highway Interface Module	1785-KA	Connects a Data Highway network to a DH+ network	1770-CD
DH+ to DH-485 Interface Module	1785-KA5	Connects a DH-485 link to a DH+ link.	
Data Highway Plus RS-232C Interface Module	1785-KE	Connects an asynchronous (RS-232C) device and DH+	
PCMCIA Card	1785-PCMK	Connects PCMCIA Bus notebook computers to DH+	1784-PCM5

Link Design

Specify 1770-CD (Belden 9463) cable. Connect a DH+ network using a daisy chain or trunk line/drop line configuration.

Trunk line/drop line considerations:
When using a trunk line/drop line configuration, use 1770-SC station connectors and follow these cable-length guidelines:

- trunk line-cable length – depends on the communication rate of the link
- drop-cable length – 30.4 m (100 cable-ft)

For more information about designing trunk line/drop line configurations, see the *Data Highway Plus/Data Highway II/Data Highway-485 Cable Installation Manual*, publication 1770-6.2.2.

Verify that your system's design plans specify cable lengths within allowable measurements.

Important: The maximum cable length for DH+ depends on the transmission rate. Configure all devices on a DH+ link to communicate at the same transmission rate.

For daisy chain configurations, use this table to determine the total cable length you can use.

Table 1.B
Choose the correct cable length

A DH+ link using this communication rate:	Cannot exceed this cable length:
57.6 kbps	3,048 m (10,000 ft)
115.2 kbps	1,524 m (5,000 ft)
230.4 kbps	762 m (2,500 ft)

Important: If you select the baud rate as 230.4 kbps, and you are using the serial port or a PLC-5 coprocessor, use channel 2 for better overall system performance.

For proper operation, terminate **both** ends of a DH+ link by using the external resistors shipped with the programmable controller. Selecting either a 150Ω or 82Ω terminator determines how many devices you can connect on a single DH+ link.

If your DH+ I/O link operates at:	Use this resistor rating:
230.4 kbps	82Ω
57.6 kbps or 115.2 kbps	150Ω

Configuring the Channel for DH+ Communication

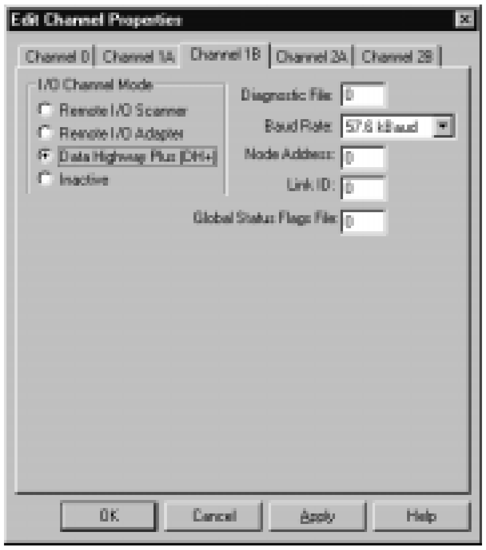
Depending on the processor you are using, you can configure these channels:

Processor:		Channels that support DH+:
PLC-5/11		1A
PLC-5/20		1A (fixed DH+), 1B
PLC-5/30 PLC-5/40L PLC-5/60L	PLC-5/20E PLC-5/40E PLC-5/80E	1A, 1B
PLC-5/40 PLC-5/60 PLC-5/80		1A, 2A, 1B, 2B

Important: To define the DH+ address and baud rate for channel 1A, you must set switch assembly SW1 on the processor; you cannot set this node address through the programming software.

To configure a channel to support a DH+ link, use the DH+ configuration screen in your programming software.

configure the channel for DH+



I/O Channel Mode
- Remote I/O Scanner
- Remote I/O Adapter
- [✓ Data Highway Plus (DH+)]
- Inactive

Diagnostic File: 0
Baud Rate: 57.6 kBaud
Node Address: 0
Link ID: 0
Global Status Flags File: 0

This field:	Specifies:	Configure by doing the following:
Diagnostic file	The file containing the channel's status information	Enter an integer file number (10-999). The system creates an integer file 40 words long. **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used file. Unpredictable machine damage can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want to get status information for that channel.
Baud rate	Communication rate for the current channel	If the DH+ channel is • channel 1A, specify the baud rate by setting SW1 (see chapter 23) • any other channel, select 57.6 kbps, 115.2 kbps, or 230.4 kbps through the programming software
Node address	The station address of your processor	If your DH+ channel is: • 1A—specify the DH+ station number by setting SW1 on your processor (see chapter 23). • anything other than 1A—cursor to Node Address field, enter a value of 0-77 octal, and press **[Enter]**. Each station on a DH+ link must have a unique address.
Link ID	The local link where the channel resides	If your DH+ link is bridged to another Data Highway network, cursor to the field, type a decimal number to identify the protocol link to which the channel is connected, and press **[Enter]**.
Global status flag file	The file where you want to store token pass data	Cursor to the field, type an integer file number (10-999), and press **[Enter]**. The system creates an integer file 64 words long. **ATTENTION:** When you change the processor from run or test to program mode, the processor writes zeroes in the global status flags file. Any information previously in this file is lost. For more information on the global status flags file, see below.

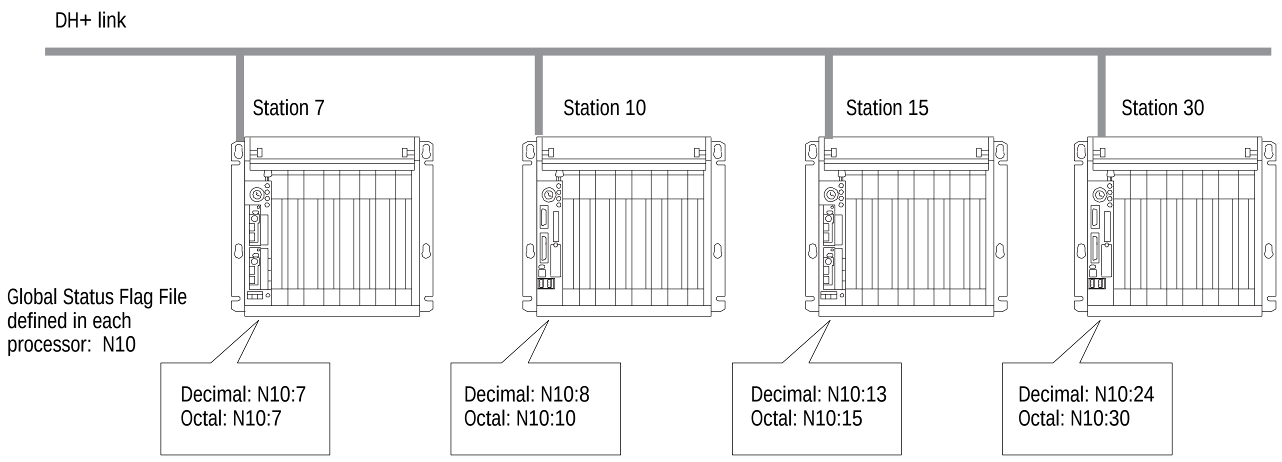
Using the Global Status Flag File

Use the global status flag file to store token pass data. This file stores a 16-bit word of data for each station on the DH+ network. The stations use this file to automatically share data with other stations without requiring user programming.

When a station sends the token to the next station, it, in effect, sends a broadcast message that contains 1 word of information from its own address area in its global status flag file. The data sent out is taken from the word in the global status file that is equal to its own station address. The token is seen by all stations. Each station on the network examines the token and places the word of global status data from the sending station into the word that corresponds to the sending station's address

This process lets each station automatically see the newly updated data. You can create ladder logic to monitor and interpret this data according to your application.

The Global Status Flag data for each node address on your DH+ link is stored in the word address corresponding to the octal node address. For example, if your DH+ link has processors at node addresses 7, 10, 15, and 30 and your global status flag file is N10 for each processor, the global status flag data is stored as follows:



You can specify any integer file in the processor to be the global status flag file; however, for simplicity, specify the same file for all your PLC-5 processors on the DH+ link.
The files are updated during housekeeping.

Design Tip

Make sure that the global status flag file in all of the processors on your DH+ link is as large as the highest node address, so that all of the nodes can communicate with each other. If station 30 is the highest node number, for example, the global status flag file (N10) in each processor must be 24 words long (octal 30 = decimal 24). When you first configure the global status file, it automatically gets 64 words.

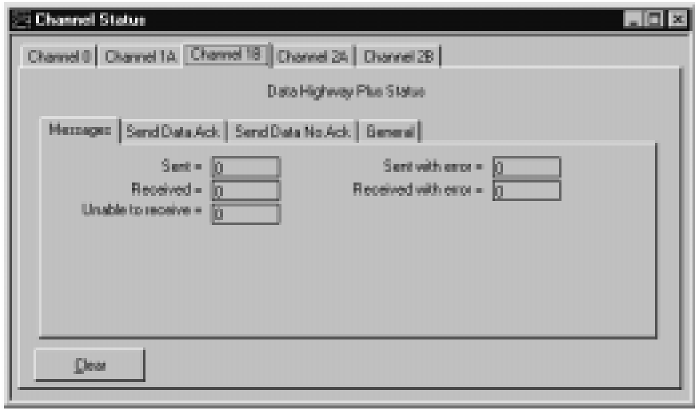
Important: Do not allow either external or internal messages to write into the global status flag file. Writing into the global status file faults the processor.

You can change the radix in the data monitor to display the file address in octal so that you can see the element number of the octal address matching the node address.

Monitoring DH+ Communication Channels

Use the DH+ status screen in your programming software to monitor channels that are configured to support a DH+ link. The data displayed is stored in the diagnostic file defined on the DH+ configuration screen in your programming software. Note that this screen does not display the active node table, which is also stored in the diagnostic file.

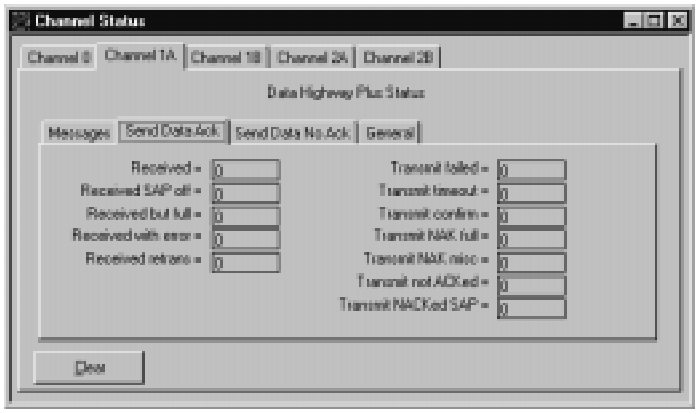
Monitoring messages



Sent = 0
Received = 0
Unable to receive = 0
Sent with error = 0
Received with error = 0

Status Field:	Word(s):	Description:
Sent	5	Total number of messages sent by the station This number is the sum of the send data acknowledge counters (SDA) and send data no acknowledge (SDN) transmit confirm counters.
Sent with error	7	Number of messages sent that were not acknowledged. This number is the sum of the following: • SDA transmit NAK misc • transmit NAK full • SDA transmit NAKed SAP • SDA/SDN retrans • dropped token
Received	4	Number of error-free messages the station has received. This number is the sum of the SDA and SDN received counters.
Received with error	6	Number of invalid messages that the station has received. This number is the sum of the SDA received with error and the SDA received SAP off counters.
Unable to receive	8	Total number of times the station NAKed an incoming message due to the lack of an available buffer. This number should be the same as the SDA received but full counter.

Monitoring Data Sent with Acknowledgment

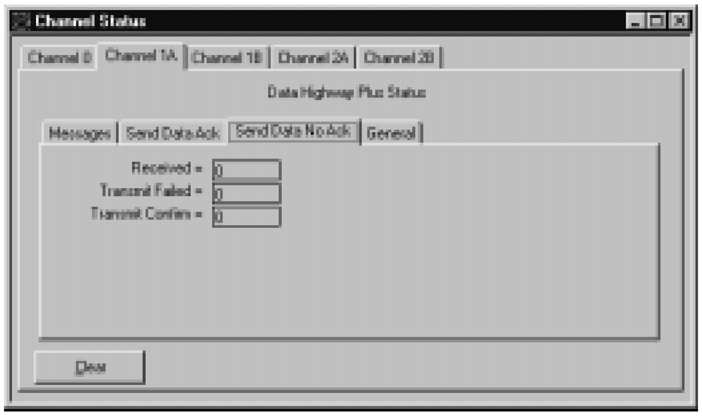


Received = 0
Received SAP off = 0
Received but full = 0
Received with error = 0
Received retrans = 0
Transmit failed = 0
Transmit timeout = 0
Transmit confirm = 0
Transmit NAK full = 0
Transmit NAK misc = 0
Transmit not ACKed = 0
Transmit NACKed SAP = 0

Status Field:	Word(s):	Description:
Received	19	Number of error-free SDA messages that the station received.
Received SAP off	23	Number of SDA messages that the station received but could not process because its service access point (SAP) was off. This counter should always be 0.
Received but full	22	Number of SDA messages that the station could not receive because of lack of memory.
Received with error	20	Number of invalid SDA messages that the station received. Some causes are: • bad CRC • the message has an invalid source address • the message has an unrecognizable control byte • the transmission was aborted This counter indicates noise; increase the cable's shielding from noise.
Received retransmissions	21	Number of times the sending station re-transmitted an SDA message, which was ACKed or NAKed If node sends a message but does not receive an ACK or a NAK response, the node will re-transmit the message. If a node retransmitted a message because the acknowledge response to the first message was lost, the node receiving the message detects the retransmission and sends an acknowledge response. But the receiving node discards the duplicate message. High counts of this counter indicates noise or cable problems; check that the cable is secure and properly shielded from noise.
Transmit failed	29	Number of SDA messages sent by the station that were determined to be in error. This counter is the sum of the SDA transmit not ACKed and SDA transmit timeout counters.
Transmit timeout	26	The number of SDA messages that were sent but not ACKed or NAKed by the receiving station This counter increments even if the message does get through during a retry and if the receiving station is unable to communicate. This counter indicates a noise or a cabling problem (the receiving station is not seeing the messages).
Transmit confirm	24	Number of SDA messages successfully sent to and acknowledged by the addressed station
Transmit NAK full	30	Number of times the station received a NAK to a message because the destination station was full This indicates that messages are being sent to the receiving station faster than the PLC-5 processor can process them. Most likely, more than one station on the DH+ link is sending messages to the same station. Check to see that you are: • not scheduling unnecessary traffic (e.g., your are sending continuous messages when you only need updates once per second) • implementing report-by-exception so that data is sent only if it is new data
Transmit NAK misc.	25	Number of incoming SDA messages that were NAKed due to reasons other than the NAKed full and NAKed inactive counters (e.g., a NAK due to a bad CRC)

Status Field:	Word(s):	Description:
Transmit not ACKed	27	Number of SDA messages that were sent but were not ACKed by the receiving station The following could have occurred: • message could have been NAKed • an invalid ACK was returned • nothing was returned This counter can indicate: • a noise or a cabling problem • the receiving station has been removed from the link • the receiving station cannot communicate
Transmit NAKed SAP	31	Number of SDA messages that were successfully sent to but were NAKed by the addressed station because the SAP specified in the message was illegal This counter should always be 0.

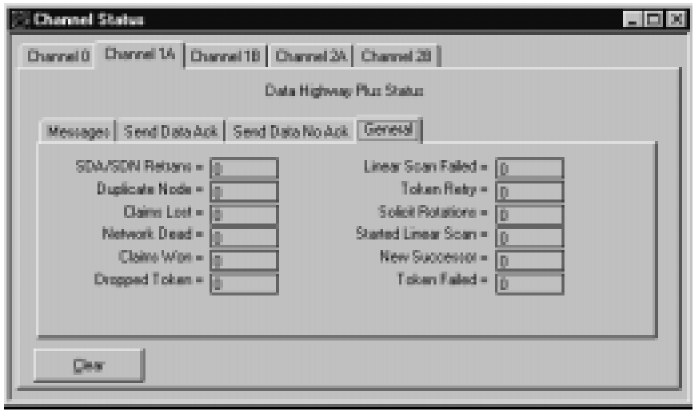
Monitoring Data Sent without Acknowledgment



Received = 0
Transmit Failed = 0
Transmit Confirm = 0

Status Field:	Word(s):	Description:
Received	35	Number of valid SDN messages received
Transmit failed	33	Number of SDN messages sent by the station that were in error This error should never be seen.
Transmit confirm	32	Number of valid SDN messages sent by the station

Monitoring General Status



SDA/SDN Retrans = 0
Duplicate Node = 0
Claims Lost = 0
Network Dead = 0
Claims Won = 0
Dropped Token = 0
Linear Scan Failed = 0
Token Retry = 0
Solicit Rotations = 0
Started Linear Scan = 0
New Successor = 0
Token Failed = 0

Status Field:	**Word(s):**	**Description:**
SDA or SDN transmit retry	28	Total number of SDA or SDN messages that were re-transmitted. Some reasons why the station would retry a message are: • the ACK was lost or corrupted on an SDA message, indicating a possible noise problem • the original message was NACKed
Duplicate node	17	Number of times the station has detected the same station address as itself on the network. As a result, the station goes offline.
Claims lost	11	Number of times the station did not win the claim token sequence. See claims won below for more information.
Network dead	9	Number of times the station detects no traffic on the network. This usually occurs when the station with the token is powered down or is removed from the network. The other stations are waiting for the token to be passed to them. Eventually a network dead situation is declared and a claim token sequence initiated. (See claims won for more information.)
Claims won	10	Number of times the station has won the claim token sequence. All the stations initiated a claim token sequence when a network goes down, is just powered up and the stations on the network detect that no one has the token, or when a station with the token is powered down or removed from the network. A claim token sequence is when all the stations on a network attempt to claim the token. When multiple stations attempt to claim the token, the lowest numbered station wins.
Dropped token	18	Number of times that the station detected that a duplicate node existed on the link and consequently dropped itself off the link A station determines that there is a duplicate node when it detects that the response to a message or solicit successor is incorrect. For example, if a response is received from a station which was not communicated with, then the sending station assumes that the response is for a packet sent by another station with the same node number. Once the station drops itself off the link, it waits indefinitely to be solicited back into the network. It will only be solicited back into the network if the duplicate node is removed from the link, because station numbers that already exist on the link are not solicited into the network.
Linear scan failed	16	Number of times the station solicited every station number without getting a response. See started linear scan below for more information.
Token retry	13	Number of times the station had to re-transmit a token pass. The station re-transmits a token pass if it detects that the station it passed the token to did not receive the token. Noise can cause this to occur.
Solicit rotations	34	Number of times a complete solicit successor of all stations not on the link is completed. A solicit successor occurs during a token pass around the link. Here a station that is currently not on the link is solicited to see if it has been added to the link. During each token pass, a different station number is solicited; solicitation occurs sequentially. A station can only join the link when it is solicited into it.

Status Field:	Word(s):	Description:
Started linear scan	15	Number of times the station has attempted to pass the token to everyone in its active node table and no one has responded. The station will then start a linear scan where it solicits every station number until a station responds.
New successor	12	Number of times the station found a new successor for the token. A new successor occurs when the station detects that a new station with a station number between its and a the station it was passing the token to was added to the link. The station now must past the token to the newly added station.
Token failed	14	Number of times station could not pass token to its listed successor. This usually occurs due to: • the station being removed from the network • noise or cabling problems

Estimating DH+ Link Performance

Many factors affect the performance of your DH+ link, including:

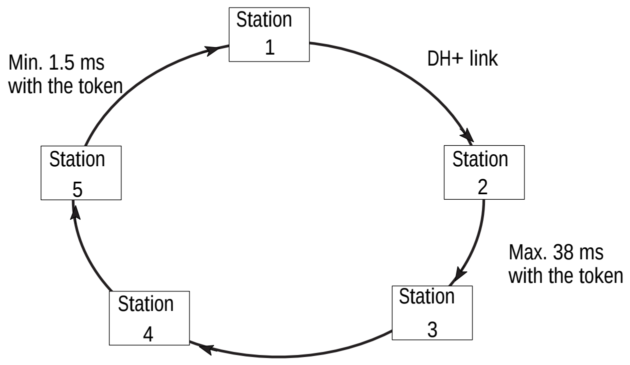
- nodes
- size and number of messages
- message destination
- internal processing time

Nodes

Nodes affect transmission time in the following ways:

- During one complete token rotation, each node on the DH+ link receives the token whether or not it has something to send.
- Each node spends from 1.5 ms (if it has no messages to send) to 38 ms (maximum time allotted) with the token, assuming there are no retries (Figure 10.1)

Figure 10.1
Token Passing



Size and Number of Messages

A PLC-5 processor encodes messages into packets for transmission on the DH+ link. The maximum number of data words in a packet depends on the sending station and command type as shown in the table below.

Sending Station	Command Type	Maximum Packet Size (Data Words)
PLC-5	Typed READ/WRITE	114
PLC-5	Word range READ/WRITE	117
PLC-2	Unprotected READ/WRITE	121

This limit comes from the network protocol, which limits a station to transmitting a maximum of 271 bytes per token pass. A station can send more than one message in a token pass, provided that the total number of combined command and data bytes does not exceed 271.

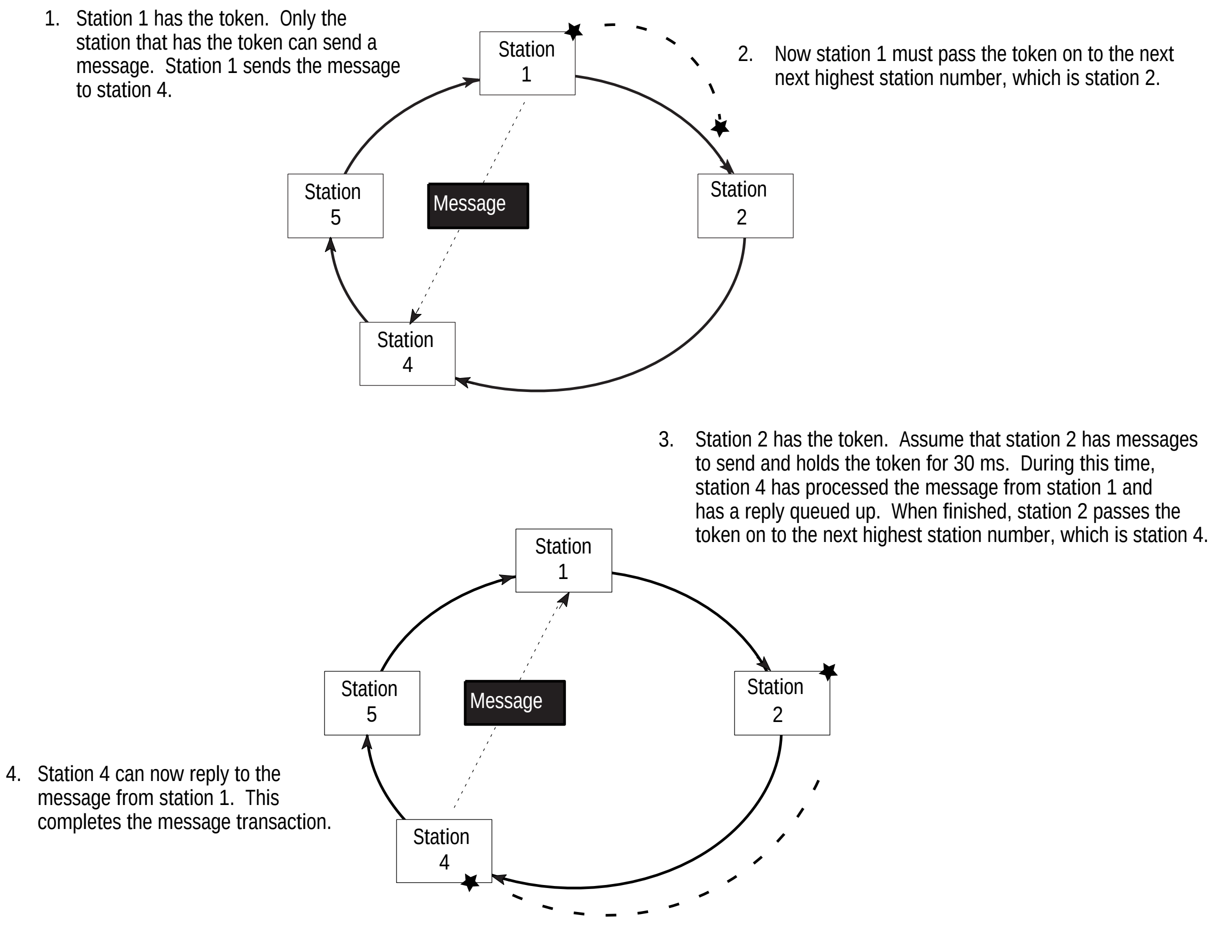
If a message exceeds the maximum packet size allotted, however, the sending station will require more than one token pass to complete the message. For example, if a PLC-5 processor wants to send a 150-word message, it will have to transmit two messages, possibly requiring multiple token rotations.

The number of messages a station has to send also affects throughput time. For example, if a station has three messages queued and a fourth is enabled, the fourth message may have to wait until the previous three are processed.

Message Destination

Throughput times vary depending on whether a receiving station can process the message and generate a reply before it receives the token. Figure 10.2 assumes that station 1 wants to send a message to station 4.

Figure 10.2
Message Destination—Station has adequate time to process a MSG reply



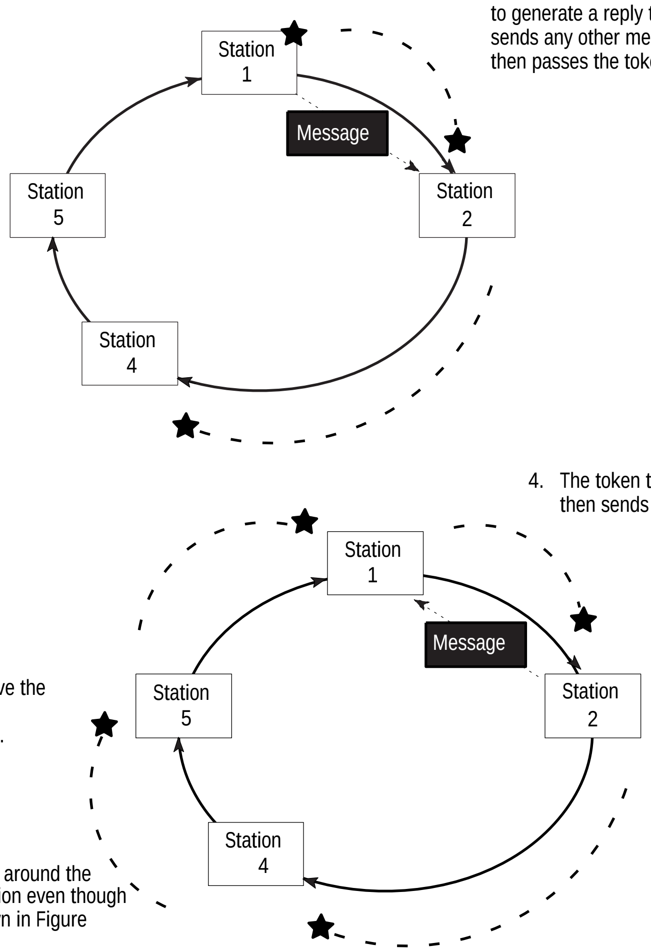
1. Station 1 has the token. Only the station that has the token can send a message. Station 1 sends the message to station 4.
Station 1
2. Now station 1 must pass the token on to the next next highest station number, which is station 2.
Station 5
Message
Station 2
Station 4
3. Station 2 has the token. Assume that station 2 has messages to send and holds the token for 30 ms. During this time, station 4 has processed the message from station 1 and has a reply queued up. When finished, station 2 passes the token on to the next highest station number, which is station 4.
Station 1
Station 5
Message
Station 2
4. Station 4 can now reply to the message from station 1. This completes the message transaction.
Station 4

In Figure 10.2, station 4 has had time to process the message and generate a reply. However, in Figure 10.3, station 2 does not have sufficient time to process a MSG reply.

Figure 10.3
Message Destination—Station has insufficient time to process MSG reply

1. In this figure, we assume that station 1 wants to send the identical message as shown in Figure but to station 2. Station 1 has the token. Station 1 sends the message to station 2 and then passes the token on to station 2.

2. Now station 2 has the token but has not had time to generate a reply to station 1. So station 2 sends any other messages it has queued and then passes the token on to station 4.



4. The token then returns to station 2, which then sends its reply to station 1.

3. Stations 4, 5, and 1 all receive the token in order and send any messages they have queued.

In this example, it took an extra token pass around the network to complete the message transaction even though the message was identical to the one shown in Figure 10.2.

.

Internal Processing Time

Internal processing time depends on how busy a given processor on the network is when sending or receiving a message.

For example, processor A has just received a READ request from processor B on the network. If processor A already has three messages of its own to send, the reply to the READ request from processor B will have to wait until the station completes the processing of the messages queued ahead of it.

Test Setup

One to 22 PLC-5 processors were used with one personal computer online. Each PLC-5 processor executes 1K of ladder logic.

Initial testing was done with one PLC-5 processor writing data to another PLC-5 processor. The response time was recorded. Additional PLC-5 processors were added to the network, each writing the same amount of data to a PLC-5 processor at the next highest station address. Four separate tests were run using data transmissions of 50, 100, 250, and 500 words.

Average DH+ Link Response Time Test Results

This section shows graphically the results of testing performed on a DH+ link where the number of stations and words sent in the message varies.

Figure 10.4 shows the average response time of messages of varying sizes on a DH+ link with a varying numbers of stations. It also gives you an idea of the typical response time you can expect on a given DH+ link.

Figure 10.4
Average Response Time for all PLC-5 Processors

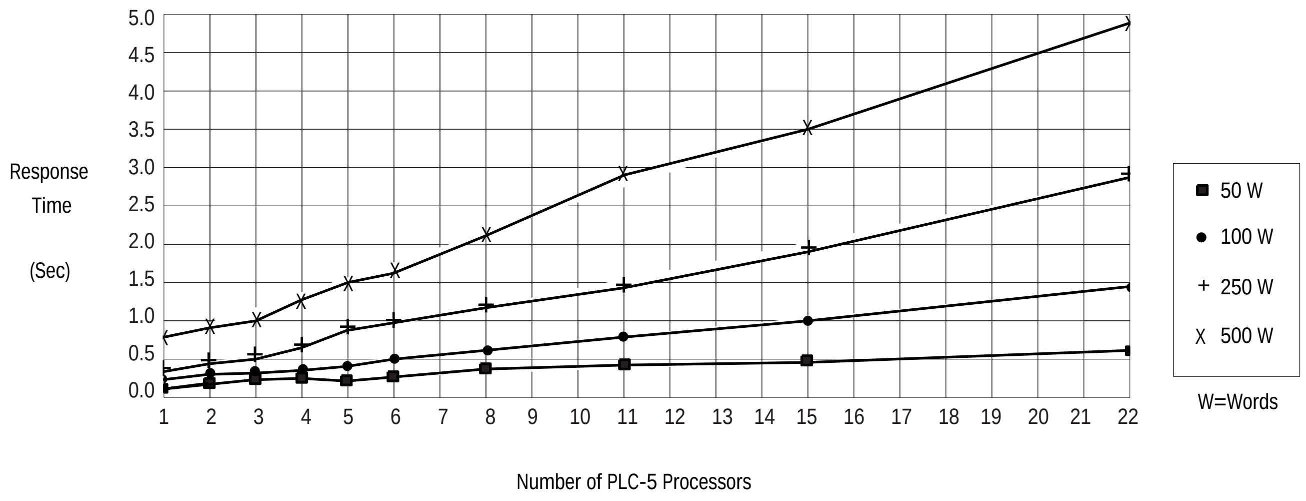
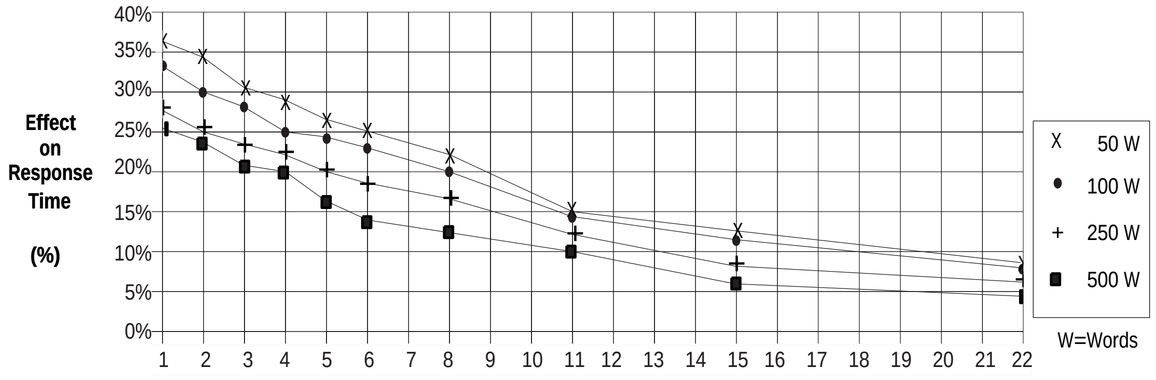


Figure 10.5 shows the effect of a personal computer on message response time under various configurations.

Figure 10.5
Response Time Increase (%) Due to the Effects of a Personal Computer



Number of PLC-5 Processors

Application Guidelines

Consider the following application guidelines when configuring a DH+ link for your system.

Design Tip

- Minimize the number of DH+ nodes to achieve acceptable response times. Keep in mind the size and frequency of messages exchanged between devices.
- Limit the number of nodes on your network when you are trying to achieve fastest control response time. You can establish separate DH+ networks to bring-on additional stations. Use a bridge to connect the DH+ links.
- When you connect a computer to the link for operator interface or a third-party serial device to the DH+ link, select the fastest possible serial interface communication rate.
- Do not add or remove nodes from the network during machine or process operation. If the network token resides with a device that is removed, the token may be lost to the rest of the network. The network is automatically re-established, but it could take several seconds. Control would be unreliable or interrupted during this time.
- A DH+ link has a 90 s timeout period; however, you can include watchdog timers in logic programs for DH+ transfer of data (to provide an orderly shutdown if failure occurs).
- When possible, do not program processors online during machine or process operation. This could result in long bursts of DH+ activity that could increase response time. See chapter 9 for more information.
- When possible, add a separate DH+ link for programming processors to keep effects of the personal computer from the process DH+ link.

Notes:

Communicating with Devices on a Serial Link

Using This Chapter

If more you are using PLC-5 processors in Supervisory Control and Data Acquisition (SCADA) applications, see:

- *SCADA System Selection Guide*, publication AG-2.1
- *SCADA System Application Guide*, publication AG-6.5.8

Choosing Between RS-232C, RS-422A, and RS-423

The table below summarizes some of the differences between RS-232C, RS-422A, and RS-423 communication modes:

This method:	Is normally used when you:
RS-232C	have a data transmission range of up to 50 ft. (15.2m). Applications requiring longer distances can use modems or line drivers. Use RS-232C for half- or full-duplex communication. For example, computers communicating with processors or modems in SCADA applications.
RS-422A	want to transmit data to RS-422A-compatible devices over ranges greater than RS-232C allows. See Table 11.A on page 11-5. Use RS-422A for point-to-point communication, with one device communicating with as many as 16 other devices.
RS-423	want to transmit data to RS-423-compatible devices over ranges greater than RS-232C allows. See Table 11.A on page 11-5. Use RS-423 for point-to-point communication, with one device communicating with as many as 16 other devices.

Configuring the Processor Serial Port

Channel 0 is the serial port and is configurable for RS-232C, RS-423, or RS-422A compatible communication. Use switch assembly SW2 to specify the serial port configuration.

To set the processor switch, see chapter 23 or look on the side label of the processor, which shows the switches in switch assembly SW2 and a table listing the settings.

Using Channel 0

You can use the processor's serial port (channel 0) to connect the processor to devices that:

- can send and receive ASCII characters by using User mode (ASCII communication)
- communicate using DF1 protocol by using one of three available System modes

User Mode

In user mode, all data are received and sent via a buffer. To access or send this data, use ASCII instructions in your ladder program. The ASCII data a PLC-5 processor sends contain no additional protocol characters.

In user mode, only ASCII instructions can be used. If you try to use a message (MSG) instruction that references the serial port, the error (.ER) bit is set.

Examples of ASCII peripheral devices are:

- ASCII terminals
- Bar code readers
- Allen-Bradley Dataliners
- Weigh scales
- Printers

System Mode

In system mode, the processor interprets a command from another device. Use system mode when you need to communicate with other devices on a link. System mode, with DF1 protocol, is a separate and unique communication link from the DH+ link.

In system mode, you can send data to a device using:

- the message (MSG) instruction; or
- ASCII write instructions (send as an ASCII string)

All data is encapsulated inside a DF1 protocol packet; therefore, the processor can communicate only with peripheral devices that support the DF1 protocol.

Examples of DF1 peripheral devices are:

- personal computers
- communication modules such as 1771-KF2 series C, 1771-KE, 1771-KF, and 1785-KE
- modems

Use this mode:	For:
Point-to-Point	communication between a PLC-5 processor and one other DF1 protocol compatible device In point-to-point mode, a PLC-5 processor uses DF1 full-duplex protocol.
DF1 Master Mode	control of polling and message transmission between the master and each remote node In master mode, a PLC-5 processor uses DF1 half-duplex polled protocol. The master/remote network includes one PLC-5 processor configured as the master node and up to 254 remote nodes. You link remote nodes using modems or line drivers. A master/remote network can have node numbers from 0 to 376 (octal). Node 377 is reserved for broadcast. Each node must have a unique node address. Also, at least 2 nodes must exist to define your link as a network (1 master and 1 remote station are two nodes).
DF1 Slave Mode	using processor as a remote station in a master/slave serial communication network When there are multiple remote stations on the network, you link remote nodes using modems or line drivers. When you have a single remote station on the network, you do not need a modem to connect the remote station to the master; you can configure the control parameter for no handshaking. You can connect from 2 to 255 nodes to a single link. In slave mode, a PLC-5 processor follows DF1 half-duplex protocol. One node is designated as the master and it controls who has access to the link. (For example, a master can be a PLC-5/250 or PLC-5/40 processor or a computer running ControlView SCADA option software. All other nodes are remote stations and must wait for permission from the master before transmitting. The master (except PLC-5/250) can send and receive messages from all nodes on the link and to nodes on other Data Highway links connected to the multidrop; whereas, a remote station can only respond to the master.

Master Station to Remote Station Communication Methods

A PLC-5 master station can communicate with remote stations in two ways:

Method:	Option name:	Principal benefits:
initiating polling packets to remote stations according to their position on a polling list Polling packets are formed independently of any user-programming.	standard communication mode	This is the communication mode used most often in point-to-multipoint configurations. Provides for these capabilities: • remote stations can send messages to the master station (polled report-by-exception) • remote stations can send messages to each other • lets the master station maintain an active node table The poll list resides in a user designated and accessible integer-type data file. You can: • include the master on the poll list • configure the master for between-station polls (master transmits any message that it needs to send before polling the next remote station) • have the master both in the poll list and configured for between-station polls
initiating communication to remote stations using only user-programmed message (MSG) instructions Each request for data from a remote station must be programmed via a message instruction. The master polls the remote station for a reply to the message after waiting a user-configured period of time. The waiting period gives the remote station time to formulate a reply and prepare the reply for transmission. After all of the messages in the master's message-out queue are transmitted, the remote-to-remote queue is checked for messages to send.	message-based communication mode	If your application uses satellite transmission or public switched telephone network transmission, consider choosing message-based. Communication to a remote station can be initiated on an as needed basis. Or choose this method if you need to communicate with non-intelligent remote terminal units (RTUs).

Polling Inactive Priority Stations

Through the channel configuration feature of your programming software, you can choose to poll one or all of the inactive priority stations when the PLC-5 processor is in master mode on channel 0. The default selection is to poll one inactive priority station during each polling sequence.

If you choose to poll all inactive stations, you are alerted immediately when an inactive station becomes active; you do not have to wait for all of the other polling sequences to complete. Polling all inactive stations might slow down channel performance.

Changing Modes

Configure channel 0 of the processor to change communication modes via:

- the channel configuration screen in your programming software (in program mode only) ; or
- communication mode change characters (ASCII control characters) sent remotely to the processor, which switches modes

Cabling

Table 11.A lists the maximum cable lengths you can use with channel 0.

Table 11.A
RS-Port Cable Lengths per Communication Rate

Port:	Transmission Rate(s):	Maximum Cable Length:
RS-232C	All	15 m (50 ft)
RS-422A (compatible)	All	61 m (200 ft)
RS-423	All	61 m (200 ft)

Important: Follow these guidelines:

- When channel 0 is configured for RS-422A compatibility, do not use terminating resistors anywhere on the link.
- When channel 0 is configured for RS-422A (compatible) and RS-423, do not go beyond 61 m (200 ft). This distance restriction is independent from the transmission rate.

For a list of serial programming cables and pin information for channel 0, see chapter 25.

Configuring Channel 0

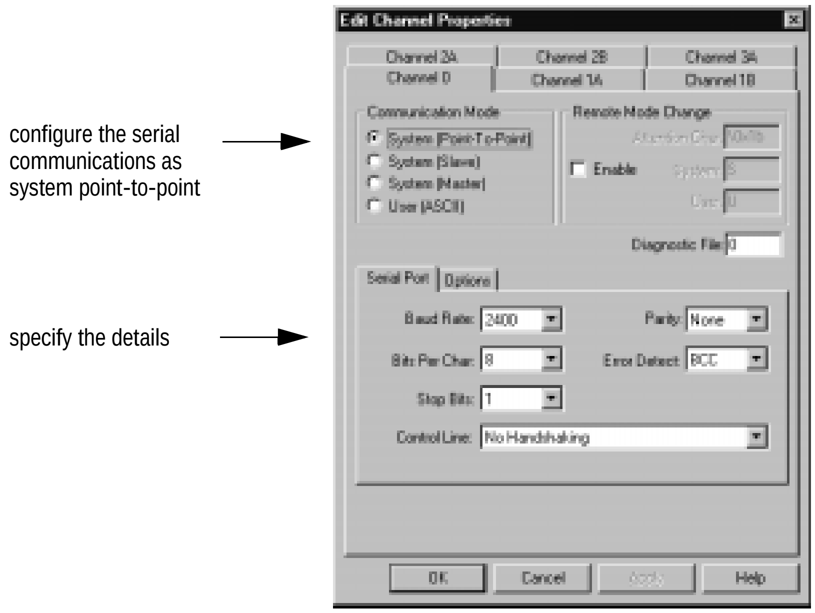
Use switch assembly SW2 processors to specify RS232-C, RS422A (compatible), or RS423 communications for channel 0.

You can configure channel 0 to communicate using these protocols:

If you want to use:		See page:
System mode	DF1 point-to-point	11-6
	DF1 slave	11-8
	DF1 master	11-10
User mode	ASCII	11-15

Configure Channel 0 for DF1 Point-to-Point

To configure channel 0 for DF1 point-to-point communication, use the system mode configuration screen in your programming software.



Communication Mode: (•) System (Point-To-Point) ( ) System (Slave) ( ) System (Master) ( ) User (ASCII)

Remote Mode Change: [ ] Enable Attention Char: \0x1b System: S User: U

Diagnostic File: 0

Serial Port

Baud Rate: 2400 Parity: None

Bits Per Char: 8 Error Detect: BCC

Stop Bits: 1

Control Line: No Handshaking

This field:	Specifies:	Configure by doing the following:
Diagnostic file	The file containing the channel's status information	Enter an integer file number (10-999). **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used integer file. Unpredictable machine operation can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want to get status information for that channel.
Enable	Whether the remote mode change option is enabled	Select ENABLED.
Mode attention char.	The attention character for the system or the user mode character for remote change	Enter a character. If the attention character you want to use is a control character, specify the ASCII equivalent.
System mode char.	The character to be used with the mode attention character (above)	Enter a character. If the attention character you want to use is a control character, specify the ASCII equivalent. When the processor encounters the attention character and the system mode character, the processor sets channel 0 communication to system mode. The remote mode change option must be ENABLED.
User mode char.	The character for the mode attention character (above)	Enter a character. If the attention character you want to use is a control character, specify the ASCII equivalent. When the processor encounters the attention character and the user mode character, the processor sets channel 0 communication to user mode. The remote mode change option must be ENABLED.

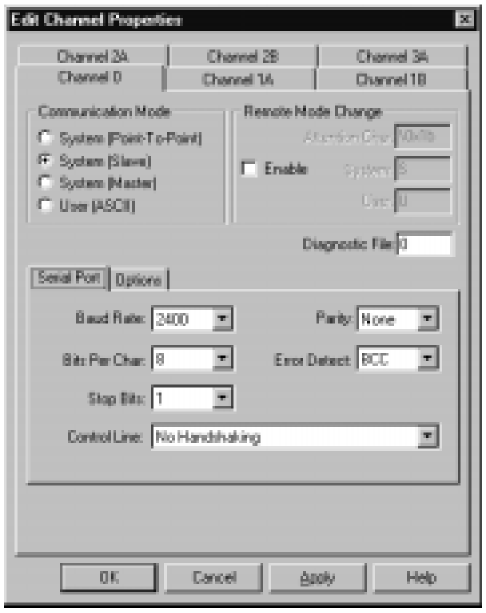
This field:	Specifies:	Configure by doing the following:
Serial settings		
Baud rate	Communication rate for channel 0 Configure all devices in the system for the same communication rate	Select 110, 300, 600, 1200, 2400, 4800, 9600, or 19.2k bps.
Parity	Parity setting for channel 0 Parity provides additional message packet error detection.	Select NONE or EVEN.
Bits per character	Select the number of bits that make up a transmitted character.	Select 7 or 8.
Error detect	Whether you want error detection set to BCC or CRC	Select one of the following: • BCC: the processor sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. • CRC: the processor sends and accepts messages with a 2-byte CRC for error checking. CRC is more complete checking Configure both stations to use the same type of error checking.
Stop bits	Match the number of stop bits to the device with which you are communicating	Select 1, 1.5, or 2.
Control line	Select the mode in which the driver operates.	Select a method appropriate for your system's configuration: • If you are not using a modem, choose NO HANDSHAKING. • If you are using a full-duplex modem, choose FULL-DUPLEX.
Option settings		
Duplicate detect	Whether you want the processor to detect and ignore duplicate messages	Select the desired setting.
ACK timeout	The amount of time you want the processor to wait for an acknowledgment to its message transmission	Enter a value 0-65535. Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2. The recommended time elapse is 1 second. Specify 1 second by typing 50.
MSG appl timeout	The number of seconds within which the reply message must be received before the error bit is set on the message. The timer starts when the ladder program first initiates the message and is restarted if/when the ACK is received	Enter one of the following values: • 1: 30-60 seconds • 2: 60-90 seconds • 3: 90-120 seconds • 4: 120-150 seconds • 5: 150-180 seconds • 6: 180-210 seconds • 7: 210-240 seconds
NAK receive	The number of NAKs your processor can receive in response to a transmitted message	Enter a value 0-255. The recommended setting is 3.
DF1 ENQS	The number of enquiries (ENQs) that you want the processor to send after an ACK timeout	Enter a value 0-255. The recommended setting is 3.

Configure Channel 0 as a Slave Station

To configure channel 0 for DF1 slave communication, use the system mode configuration screen in your programming software.

configure the serial communications as system slave

specify the details



Communication Mode: System (Point-To-Point) [✓ System (Slave)] System (Master) User (ASCII)

Remote Mode Change: Enable [ ] Attention Char: \0x1b System: S User: U

Diagnostic File: 0

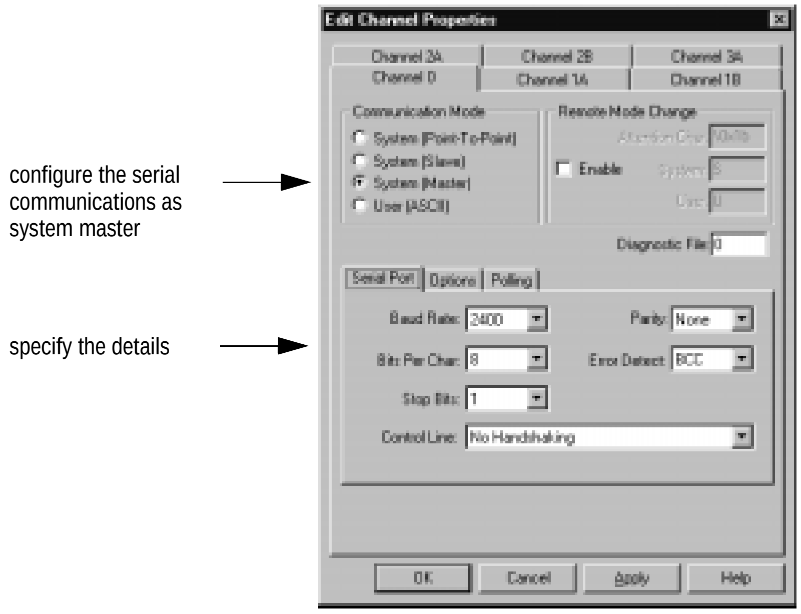
Baud Rate: 2400
Parity: None
Bits Per Char: 8
Error Detect: BCC
Stop Bits: 1
Control Line: No Handshaking

This field:	Specifies:	Configure by doing the following:
Diagnostic file	The file containing the channel's status information	Enter an integer file number (10-999). **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used integer file. Unpredictable machine action can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want to get status information for that channel.
Enable	Whether the remote mode change option is enabled	Select ENABLED or DISABLED.
Mode attention char.	The attention character for the system mode or the user mode character for a remote mode change	Enter a character. If the attention character you want to use is a control character, specify the ASCII equivalent.
System mode char.	The character for the mode attention character (above)	Enter an attention character. If the attention character you want to use is a control character, specify the ASCII equivalent. When the processor encounters the attention character and the system mode character, the processor sets channel 0 communication to system mode. The remote mode change option must be ENABLED.
User mode char.	The character for the mode attention character (above)	Enter a character. If the attention character you want to use is a control character, specify the ASCII equivalent. When the processor encounters the attention character and the user mode character, the processor sets channel 0 communication to user mode. The remote mode change option must be ENABLED.
Serial settings		
Baud rate	Communication rate for channel 0. Configure all devices in the system for the same communication rate	Select 110, 300, 600, 1200, 2400, 4800, 9600, or 19.2k bps.

This field:	Specifies:	Configure by doing the following:
Parity	Parity setting for channel 0 Parity provides additional message packet error detection.	Select NONE or EVEN.
Bits per character	Select the number of bits that make up a transmitted character.	Select 7 or 8.
Error detect	Whether you want error detection set to BCC or CRC	Select one of the following: • BCC: the processor sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. • CRC: the processor sends and accepts messages with a 2-byte CRC for error checking. CRC is more complete checking Configure both stations to use the same type of error checking.
Stop bits	Match the number of stop bits to the device with which you are communicating	Select 1, 1.5, or 2.
Control line	Select the mode in which the driver operates.	Select a method appropriate for your system's configuration: • If you are not using a modem, choose NO HANDSHAKING. • If you are using a full-duplex modem, choose FULL-DUPLEX.
Option settings		
Station address	The station address for channel 0 on the DF1 half-duplex link	Enter a valid DF1 address (0-376 octal).
DF1 retries	The number of times the remote station retries a message before the station declares the message undeliverable	Enter a value 0-255. The recommended setting is 3.
RTS send delay	The amount of time that elapses between the assertion of the RTS signal and the beginning of the message transmission This time allows the modem to prepare to transmit the message. The CTS signal must be high for transmission to occur.	Enter a value 0-255. Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2. The recommended time elapse is 0, unless you are using a modem that automatically returns the CTS as soon as it receives the RTS. If this is the case, enter a delay time to make sure the modem is able to transmit before it attempts to send the message.
RTS off delay	The amount of time that elapses between the end of the message transmission and the de-assertion of the RTS signal. This time delay is a buffer to make sure that the modem has transmitted the message.	Enter a value 0-255.Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2.
ACK timeout	The amount of time you want the processor to wait for an acknowledgment to its message transmission	Enter a value 0-65535. Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2. The recommended time elapse is 1 second. Specify 1 second by typing 50.
Duplicate detect	Whether you want the processor to detect and ignore duplicate messages	Select the desired setting.
MSG application timeout	The number of seconds within which the reply message must be received before the error bit is set on the message The timer starts when the ladder program first initiates the message and is restarted if/when the ACK is received.	Cursor to the field, type in a value 1-7, and press `[Enter]`. Available options are: • 1: 30-60 seconds • 2: 60-90 seconds • 3: 90-120 seconds • 4: 120-150 seconds • 5: 150-180 seconds • 6: 180-210 seconds • 7: 210-240 seconds

Configure Channel 0 as a Master Station

To configure channel 0 for DF1 master communication, use the system mode configuration screen in your programming software.



Communication Mode
- System (Point-To-Point)
- System (Slave)
- [✓ System (Master)]
- User (ASCII)

Remote Mode Change
- Enable: unchecked
- Attention Char: \0x1b
- System: S
- User: U

Diagnostic File: 0

Baud Rate: 2400
Parity: None
Bits Per Char: 8
Error Detect: BCC
Stop Bits: 1
Control Line: No Handshaking

This field:	Specifies:	Configure by doing the following:
Diagnostic file	The file containing the channel's status information	Enter an integer file number (10-999). **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used integer file. Unpredictable machine action can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want to get status information for that channel.
Enable	Whether the remote mode change option is enabled	Select ENABLED or DISABLED.
Mode attention char.	The attention character for the system mode or the user mode character for a remote mode change	Enter a valid attention character. If the attention character you want to use is a control character, specify the ASCII equivalent.
System mode char.	The character for the mode attention character (above)	Enter a valid attention character. If the attention character you want to use is a control character, specify the ASCII equivalent. When the processor encounters the attention character and the system mode character, the processor sets channel 0 communication to system mode. Note that the remote mode change option must be ENABLED.
User mode char.	The character for the mode attention character (above)	Enter a valid attention character. If the attention character you want to use is a control character, specify the ASCII equivalent. When the processor encounters the attention character and the user mode character, the processor sets channel 0 communication to user mode. Note that the remote mode change option must be ENABLED.
Serial settings		
Baud rate	Communication rate for channel 0 Configure all devices in the system for the same communication rate	Select 110, 300, 600, 1200, 2400, 4800, 9600, or 19.2k bps.

This field:	Specifies:	Configure by doing the following:
Parity	Parity setting for channel 0 Parity provides additional message packet error detection.	Select NONE or EVEN.
Bits per character	Select the number of bits that make up a transmitted character.	Select 7 or 8.
Error detect	Whether you want error detection set to BCC or CRC	Select one of the following: • BCC: the processor sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. • CRC: the processor sends and accepts messages with a 2-byte CRC for error checking. CRC is more complete checking Configure both stations to use the same type of error checking.
Stop bits	Match the number of stop bits to the device with which you are communicating	Select 1, 1.5, or 2.
Control line	Select the mode in which the driver operates.	Select a method appropriate for your system's configuration: • If you are not using a modem, choose NO HANDSHAKING. • If you are using a full-duplex modem, choose FULL-DUPLEX.
Option settings		
Station address	The node's address on the DF1 link	Enter a valid DF1 station address. Valid station addresses are: 0-376 octal
DF1 retries	The number of times a message is retried before being declared undeliverable	Enter a valid value 0-255.
RTS send delay	The time delay between the time the RTS is asserted and the beginning of the message transmission This time allows the modem to prepare to transmit the message.	Enter a value 0-255. Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2. The recommended time elapse is 0, unless you are using a modem that automatically returns the CTS as soon as it receives the RTS. If this is the case, enter a delay time to make sure the modem is able to transmit before it attempts to send the message.
RTS off-delay	The time delay between the time the end of the message transmission and the RTS is de-asserted This time delay is a buffer to make sure that the modem has transmitted the message.	Enter a value 0-255. Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2. The recommended time elapse is 0, unless you are using a modem that automatically returns the CTS as soon as it receives the RTS. If this is the case, enter a delay time to make sure the modem is able to transmit before it attempts to send the message.
ACK timeout	The amount of time you want the processor to wait for an acknowledgment from a remote station to its transmitted message before the processor retries the message or the message errors out	Enter a value 0-65535. Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2. The recommended time elapse is 1 second. Specify 1 second by typing 50.
Reply msg wait	The amount of time the master will wait after receiving an ACK (to a master-initiated message) before polling the slave for a reply	Only define this if you are message-based mode. Enter a valid value 0-65535 (in 20ms increments).
MSG application timeout	The number of seconds within which the reply message must be received before the error bit is set on the message The timer starts when the ladder program first initiates the message and is restarted if/when the ACK is received.	Select one of the following: • 1: 30-60 seconds • 2: 60-90 seconds • 3: 90-120 seconds • 4: 120-150 seconds • 5: 150-180 seconds • 6: 180-210 seconds • 7: 210-240 seconds

This field:	Specifies:	Configure by doing the following:
Polling settings		
Polling mode	The current value of the polling mode	Select one of the following: • MESSAGE BASED (ALLOW SLAVE TO INITIATE MESSAGES)—default—this option allows remote station initiated messages to be processed after all master-initiated messages • MESSAGE BASED (DO NOT ALLOW SLAVE TO INITIATE MESSAGES)—remote station-initiated messages will be acknowledged but not processed • STANDARD (MULTIPLE MESSAGE TRANSFER PER NODE SCAN)—the master polls stations based on a list; each station can transmit multiple messages per node scan • STANDARD (SINGLE MESSAGE TRANSFER PER NODE SCAN)—the master polls stations based on a list; each station can transmit only one message per node scan

If you chose standard polling mode:

This field:	Specifies:	Configure by doing the following:
Master message transmit	The current value of channel 0 master message transmit	If you want the master station to: • send all of the master station-initiated MSG instructions to the remote stations before polling the next remote station in the poll list, choose Between Station Polls This method makes certain that master station-initiated messages are sent in a timely and regular manner (after every remote station poll). • only send master station-initiated MSG instructions when the master's station number appears in the polling sequence, choose In Poll Sequence With this method, sending master station-initiated messages are dependent upon where and how often the master station appears in the poll list. To achieve the same goal as the Between Station Polls method, the master-station's address would have to appear after every remote-station's address. The processor sets a minor fault if you are using IN POLL SEQUENCE and the master's station is not in either the normal poll list or the priority poll list.
Normal poll node file	The integer file that contains the addresses of the remote stations you want in the normal poll list	Enter an integer file number 10-999
Normal poll group size	The quantity of active stations located in the normal poll list that you want polled during a scan through the normal poll list before returning to the priority poll list	Enter a valid value 10-999.
Priority poll node file	The integer file that contains the addresses of the remote stations you want in the priority poll list	Enter an integer file number 10-999.
Active station file	The binary file that stores the station addresses of all active stations on the link.	Enter a binary file number 10-999.

To define a polling scheme using standard mode, you must specify the following on the DF1 master configuration screen in your programming software:

Configuration Parameter:	**Definition:**
Polling mode	How you want the master to poll the station lists.
Master message transmit	When you want the master to send messages.
Normal poll file	An integer file in which you place the station addresses of the remote stations. The default size is 64 words.
Priority poll file	An integer file in which you place the addresses of stations from which you need to collect data more frequently. The default size is 64 words.
Normal poll group size	The number of stations that the master polls before it polls a station in the priority poll list.
Active station file	A binary file that stores the station addresses of all active stations on the link. The default size is 18 words. Both the normal poll list and the priority poll list can have active and inactive stations. A station becomes inactive when it does not respond to a master's request for data.

The master station polls the slave station in the following a definitive sequence:

1. All stations in the **active** priority poll file.
2. Specified stations in the active normal poll file. The number of stations polled in this file is determined by the normal poll group size specified on the configuration screen. If the group size was 3, for example, then three stations would be polled in the normal file before the master continues to the next step in the sequence.
3. One station in the inactive poll file after all active stations in the normal poll file have been polled.

To create station lists, place each station address in an individual word in a poll file (normal and/or priority) starting at word 2. The poll file layout is as follows:

This word in a poll file:	Contains this information:
Word 0	total number of stations in the list
Word 1	address location (poll offset) of the station currently being polled For example: a value of 1 means the station address stored in word 2 is being polled, 2 means the address stored in word 3 is being polled, etc. This word is automatically updated by the master station as a new remote station is polled.
Word 2 through word *xx*	remote station address in the order that the station should be polled

To place a station address in a poll file, do the following:

1. Access the data monitor in your programming software.
2. Specify the address of the integer file that is either the normal poll file or priority poll file (e.g., if the normal poll file is N11, then you specify N11:0).
3. Enter the station addresses of the remote stations you want in the poll list starting at word 2. Put them in the order you want them polled.

Important: Station addresses are octal addresses. The poll files are integer files. The default radix is decimal. To properly enter station addresses in a poll file, you must either:

•change the radix of the file to octal

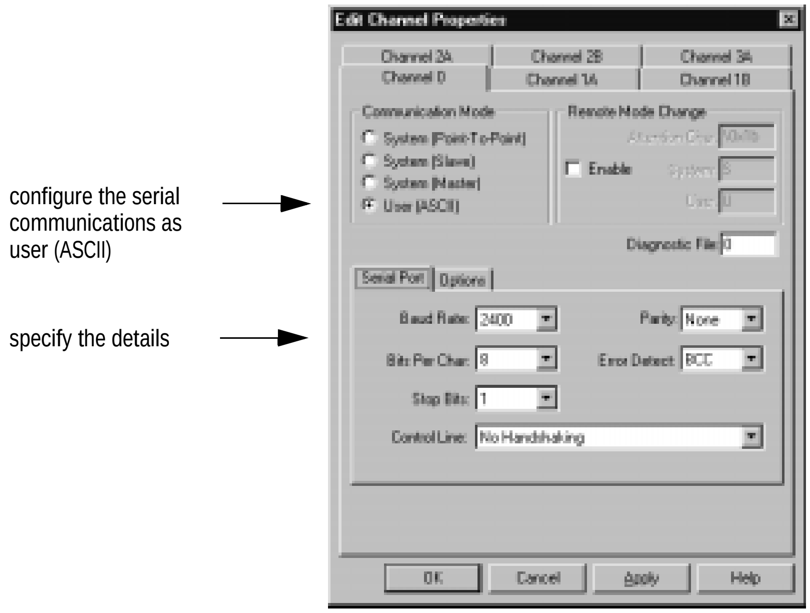
•convert the octal station addresses to decimal before entering the addresses

Below is an example of a station list containing three stations: octal addresses 10, 11, and 12 have been entered. Station 12 (10 decimal) is being polled.

Poll File	Word 0	Word 1	Word 2	Word 3	Word 4
N:11	3	3	08	09	10
N:*xx*	total number of stations	pointer showing the station address being polled (Station 12 in word 4 is being polled.)	address of first station in list	address of second station in list	address of third station in list

Configure Channel 0 for User Mode (ASCII Protocol)

To configure channel 0 for user mode, use the user mode configuration screen in your programming software.



Communication Mode: System (Point-To-Point) / System (Slave) / System (Master) / [✓ User (ASCII)]

Remote Mode Change: Enable ☐ Attention Char: \0x1b System: S User: U

Diagnostic File: 0

Baud Rate: 2400
Parity: None
Bits Per Char: 8
Error Detect: BCC
Stop Bits: 1
Control Line: No Handshaking

This field:	Specifies:	Configure by doing the following:
Diagnostic file	The file containing the channel's status information	Enter an integer file number (10-999). **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used integer file. Unpredictable machine action can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want to get status information for that channel.
Remote mode change	Whether the remote mode change option is enabled	Select ENABLED or DISABLED.
Mode attention char.	The attention character for the system mode or the user mode character	Enter a character. If the attention character you want to use is a control character, specify the ASCII equivalent.
System mode char.	The character for the mode attention character (above)	Enter a character. If the attention character you want to use is a control character, specify the ASCII. When the processor encounters the attention character and the system mode character, the processor sets channel 0 communication to system mode. The remote mode change option must be ENABLED.
User mode char.	The character for the mode attention character (above)	Enter a valid attention character. If the attention character you want to use is a control character, specify the ASCII equivalent. When the processor encounters the attention character and the user mode character, the processor sets channel 0 communication to user mode. The remote mode change option must be ENABLED.
Serial settings		
Baud rate	Communication rate for channel 0 Configure all devices in the system for the same communication rate	Select 110, 300, 600, 1200, 2400, 4800, 9600, or 19.2k bps.

This field:	Specifies:	Configure by doing the following:
Parity	Parity setting for channel 0 Parity provides additional message packet error detection.	Select NONE or EVEN.
Bits per character	Select the number of bits that make up a transmitted character.	Select 7 or 8.
Error detect	Whether you want error detection set to BCC or CRC	Select one of the following: • BCC: the processor sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. • CRC: the processor sends and accepts messages with a 2-byte CRC for error checking. CRC is more complete checking Configure both stations to use the same type of error checking.
Stop bits	Match the number of stop bits to the device with which you are communicating	Select 1, 1.5, or 2.
Control line	Select the mode in which the driver operates.	Select a method appropriate for your system's configuration: • If you are not using a modem, choose NO HANDSHAKING. • If you are using a full-duplex modem, choose FULL-DUPLEX.
Option settings		
RTS send delay	The time delay between the time the RTS is asserted and the beginning of the message transmission	Enter a value between 0 and 255. Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2. The recommended time elapse is 0, unless you are using a modem that automatically returns the CTS as soon as it receives the RTS. If this is the case, enter a delay time to make sure the modem is able to transmit before it attempts to send the message.
RTS off-delay	The time delay between the time the end of the message transmission and the RTS is de-asserted	Enter a value between 0 and 255. Limits are defined in 20 ms intervals. For example to wait 40 ms, type 2. The recommended time elapse is 0, unless you are using a modem that automatically returns the CTS as soon as it receives the RTS. If this is the case, enter a delay time to make sure the modem is able to transmit before it attempts to send the message.
Delete mode	Select how the processor responds to a delete character.	Select Ignore, CRT, or Printer. If you select Ignore, the processor ignores the delete character. If you select CRT or Printer, the processor ignores the character it received immediately before the delete character. The processor then sends a signal to the CRT or printer to erase the deleted character. Select CRT or Printer only if you enable the echo mode.
XON/XOFF	Whether or not you want XON/XOFF enabled	As the processor receives characters, it constantly determines how many more it can receive without losing any. When XON/XOFF is enabled, the processor sends a "stop sending character," XOFF. If the sending device has the XON/XOFF feature, it stops sending characters. When the processor has more room, it will send a "start sending" character (XON). Select ENABLED or DISABLED.
Echo	What the processor should do when it receives an ASCII delete character	If you disable the echo mode, characters received by the processor are sent only to the echo counter and not to an output device, such as a CRT or printer. If you enable the echo mode, the processor sends any characters it receives through an ASCI read or read line instruction to a waiting output device. For example, if you want the processor to print a message to a LED marquee, enable the echo mode.

This field:	Specifies:	Configure by doing the following:
Termination 1 Termination 2	The termination characters you defined	Enter a maximum of two characters (hexadecimal). Use termination characters with the ASCII Read Line instruction or with the Test Buffer for Line (ABL) to indicate a line has been entered. The default character is the ASCII equivalent for [RETURN], 0x0D. You can also use the ASCII equivalent for LINE FEED (0x0A). To specify no character, enter \FF.
Append 1 Append 2	The append characters you defined	Enter a maximum of two characters (hexadecimal). Use append characters with the ASCII Write with Append (AWA) instruction to indicate the end of a line. Append characters are the last characters sent after a line of information. The default characters are the ASCII equivalent for [RETURN] (/0D) and LINE FEED (/0A). To specify no character, enter \FF.

Configure Channel 0 for a Communication Mode Change

You can configure channel 0 so that it switches from one communication mode to another upon receiving a control command. You define a mode attention character and either a system or user mode character.

Character:	Tells the processor to:	Default Character:
Mode attention character	expect a change communication mode command	**[Esc]**
System mode character	switch the communication mode to system mode	**S**
User mode character	switch the communication mode to the user mode	**U**

Every time the processor receives the mode attention character and either a system or user mode character, channel 0's communication mode will be switched to the new mode.

To configure channel 0 for a remote communication-mode change, follow the steps on the left:

If you want to:	Select:
Change the communication mode of channel 0 remotely	ENABLE
Not change the communication mode of channel 0 remotely	DISABLE

Important: Make sure the remote mode change option is disabled if you do not want to change channel 0's communication mode over a remote link. Having the mode disabled prevents an unexpected communication mode change.

The Mode Attention character tells the processor to expect a communication mode change. If you are using a control character, enter the ASCII equivalent in hexadecimal. With other characters, just enter the character. Do one of the following:

Enter the character you want to use to tell the processor to switch communication modes for channel 0. If you are using a control character, use the ASCII equivalent in hexadecimal. With other characters, just enter the character.

Monitoring Channel 0 Status

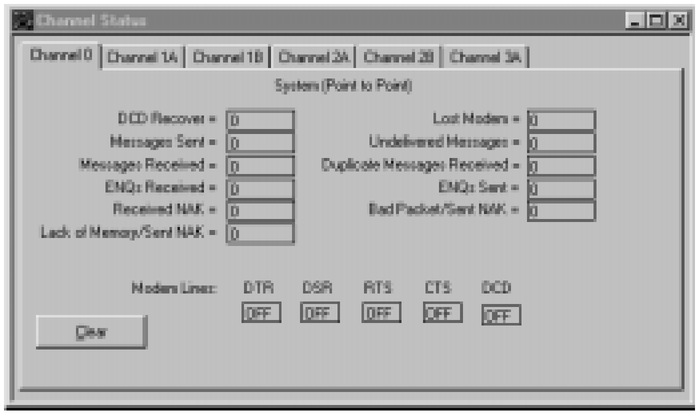
The channel 0 status screens display the information stored in the diagnostic file you specified when you configured channel 0.

If channel 0 is in this mode:	See:
System mode (DF1 point-to-point)	Figure 11.1
System mode (DF1 slave)	Figure 11.2
System mode (DF1 master)	Figure 11.3
User mode (ASCII)	Figure 11.4

Using the System Mode Status Display

This section explains the status data displayed on system mode screens in your programming software:

Figure 11.1
System Mode (DF1 Point-to Point) Status Screen

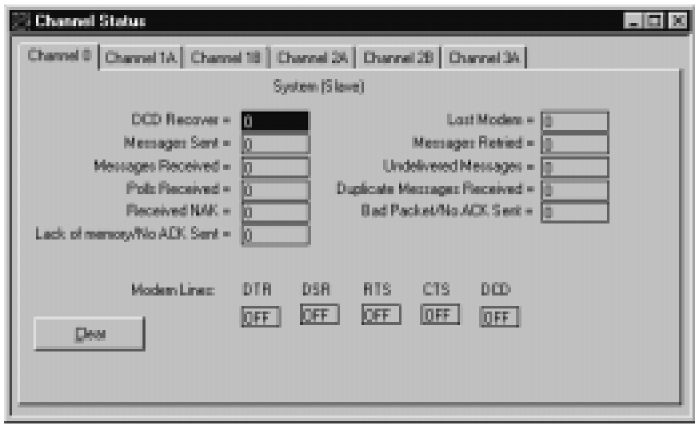


System (Point to Point)

DCD Recover =	0	Lost Modem =	0
Messages Sent =	0	Undelivered Messages =	0
Messages Received =	0	Duplicate Messages Received =	0
ENQs Received =	0	ENQs Sent =	0
Received NAK =	0	Bad Packet/Sent NAK =	0
Lack of Memory/Sent NAK =	0		

Modem Lines:	DTR	DSR	RTS	CTS	DCD
	OFF	OFF	OFF	OFF	OFF

Figure 11.2
System Mode (DF1 Slave) Status Screen

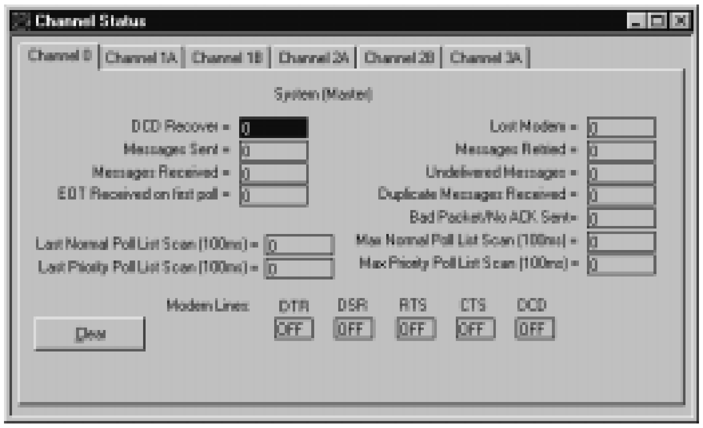


System (Slave)

DCD Recover =	0	Lost Modem =	0
Messages Sent =	0	Messages Retried =	0
Messages Received =	0	Undelivered Messages =	0
Polls Received =	0	Duplicate Messages Received =	0
Received NAK =	0	Bad Packet/No ACK Sent =	0
Lack of memory/No ACK Sent =	0		

Modem Lines:	DTR	DSR	RTS	CTS	DCD
	OFF	OFF	OFF	OFF	OFF

Figure 11.3
System Mode (DF1 Master) Status Screen



Channel Status

Channel 0 | Channel 1A | Channel 1B | Channel 2A | Channel 2B | Channel 3A

System (Master)

DCD Recover = 0
Messages Sent = 0
Messages Received = 0
EOT Received on first poll = 0
Last Normal Poll List Scan (100ms) = 0
Last Priority Poll List Scan (100ms) = 0

Lost Modem = 0
Messages Retried = 0
Undelivered Messages = 0
Duplicate Messages Received = 0
Bad Packet/No ACK Sent= 0
Max Normal Poll List Scan (100ms) = 0
Max Priority Poll List Scan (100ms) = 0

Modem Lines: DTR OFF, DSR OFF, RTS OFF, CTS OFF, DCD OFF

Table 11.B
Descriptions of System Mode Status Screen Fields

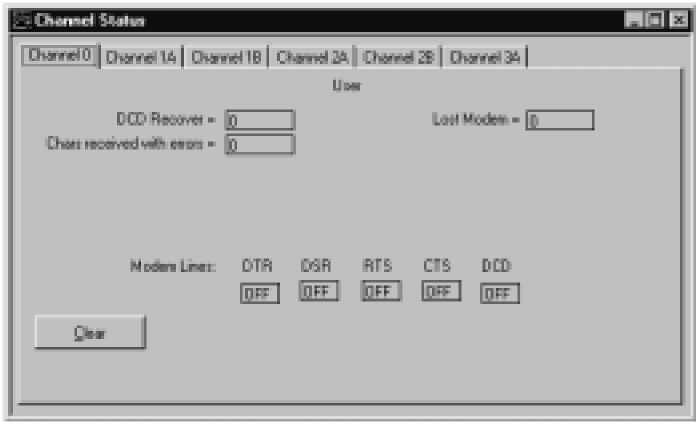
Status field:	Word Bit:	Description:
DCD recover	11	Displays the number of times the processor detects the DCD-handshaking line has gone low to high.
Messages sent	1	Displays the total number of DF1 messages sent by the processor (included message retry).
Messages received	2	Displays the number of messages the processor received with no error.
EOT received on first poll	8	Displays the number of times the master received an EOT in response to the first poll of a station.
Lost modem	12	Displays the number of times a modem was disconnected.
Messages retried	4	For slave and master mode, displays the number of messages resent.
Undelivered messages	3	Displays number of messages that were sent by processor but not received by the destination device.
Duplicate messages received	9	Displays the number of times the processor received a message packet identical to the previous message packet.
Bad packet/no ACK sent	7	Displays the number of incorrect data packets that the processor has received.
Last poll list scan last	5	The time it took to complete the previous scan of the normal station poll list.
Last priority poll list scan last	10	The time it took to complete the previous scan of the priority station poll list.
Max normal poll list scan	6	The maximum time taken to complete a scan of the normal station poll list.
Max priority poll list scan	13	The maximum time taken to complete a scan of the priority station poll list.
ENQs received	6	For point-to-point mode, displays the number of inquiries made by the destination device.
ENQs sent	4	For point-to-point mode, displays the number of inquiries made by the processor.
Received NAK	5	For Point-to-point and slave mode, displays the number of NAK messages received by the processor.
Lack of memory/sent NAK Lack of memory/no ACK sent	8	For point-to-point and slave mode, displays the number of times the processor could not receive a message because it did not have enough memory.
Polling received	6	For slave mode, displays number of times a DF1 master device has polled processor for a message.

Status field:	Word Bit:	Description:
Modem lines		
DTR	0: 4	Displays the status of the DTR handshaking line (asserted by the processor)
DSR	0: 2	Displays the status of the DSR handshaking line (received by the processor)
RTS	0: 1	Displays the status of the RTS handshaking line (asserted by the processor)
CTS	0: 0	Displays the status of the CTS handshaking line (received by the processor)
DCD	0: 3	Displays the status of the DCD handshaking line (received by the processor)

Using the User Mode (ASCII) Status Display

This section describes the user-mode status data displayed on the user mode (ASCII) status screen in your programming software.

Figure 11.4
User Mode Status Screen



DCD Recover = 0
Chars received with errors = 0
Lost Modem = 0
Modem Lines: DTR OFF DSR OFF RTS OFF CTS OFF DCD OFF

Table 11.C
Descriptions of User Mode Status Screen Fields

Status field:	Word Bit:	Description:
DCD recover	11	Displays the number of times the processor detects the DCD-handshaking line has gone low to high.
Character received with error	10	Displays the number of characters the processor received with parity or with errors and discarded
Lost modem	12	Displays the number of times a modem was disconnected.
Modem lines		
DTR	0: 4	Displays the status of the DTR handshaking line (asserted by the processor)
DSR	0: 2	Displays the status of the DSR handshaking line (received by the processor)
RTS	0: 1	Displays the status of the RTS handshaking line (asserted by the processor)
CTS	0: 0	Displays the status of the CTS handshaking line (received by the processor)
DCD	0: 3	Displays the status of the DCD handshaking line (received by the processor)

Communicating with Devices on an Ethernet Network

Using This Chapter

Media and Cabling

Ethernet is a local area network that provides communication between various devices at 10 Mbps. The physical communication media you use can be any standard 802.3 media, including:

- thick-wire coaxial cable (10Base5)
- thin-wire coaxial cable (10Base2)
- twisted pair (10Base-T)
- fiber optic
- broadband

The Ethernet port (channel 2) connects to either a thin-wire, thick-wire, or twisted-pair network via a 15-pin transceiver or Medium Access Unit (MAU) connection. See chapter 25 for detailed information about Ethernet cable connections.

Assigning Your IP Address

Contact your network administrator or the Network Information Center for a unique IP address to assign to your PLC-5/20E, -5/40E, or 5/80E processor.

Network Addressing

Because the Ethernet PLC-5 processor uses the TCP/IP protocol, each processor on the network requires a unique IP address. The IP address is software-configurable using either the BOOTP protocol or your programming software.

If you are using the BOOTP protocol, you must also obtain the hardware Ethernet address. Allen-Bradley assigns each Ethernet PLC-5 processor a hardware Ethernet address at the factory. Look for the address either:

- in the back, upper corner of your module; or
- in the channel 2 configuration screen of your programming software

Configuring Channel 2 for Ethernet Communication

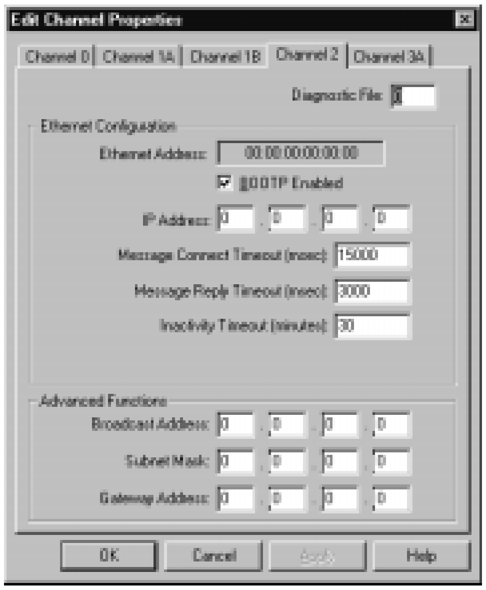
After you assign a unique IP address to your Ethernet PLC-5 processor, you must configure channel 2 so your network recognizes the processor. Configure this channel using one of two methods:

- manually entering module configuration information using the screens within your programming software package
- entering module configuration information using a BOOTP utility (use a BOOTP server on your network to edit the BOOTPTAB file)

Manually Configuring Channel 2

The default for the Ethernet PLC-5 processor is BOOTP enabled. You must first disable BOOTP before you can use the programming software to enter module configuration information.

You can manually configure channel 2 for Ethernet communication using your programming software over a DH+ or serial link



Diagnostic File: 0

Ethernet Configuration

Ethernet Address: 00:00:00:00:00:00

☑ BOOTP Enabled

IP Address: 0 . 0 . 0 . 0

Message Connect Timeout (msec): 15000

Message Reply Timeout (msec): 3000

Inactivity Timeout (minutes): 30

Advanced Functions

Broadcast Address: 0 . 0 . 0 . 0

Subnet Mask: 0 . 0 . 0 . 0

Gateway Address: 0 . 0 . 0 . 0

Enter the IP address and toggle the BOOTP enable field to No. Enter further configuration information in the appropriate fields. See Table 12.A on the following page.

Important: BOOTP enabled is the factory default. You cannot manually change the IP address with your programming software if BOOTP is enabled.

Table 12.A
Ethernet Channel 2 Configuration Fields

This field:	Specifies:	Configure by doing the following:
Diagnostic file	The file containing the channel's status information.	Enter an integer file number (10-999). The system creates an integer file 44 words long. **ATTENTION:** Assign a unique diagnostic file to each channel. Do not assign a diagnostic file that is the I/O status file you assigned or any other used file. Unpredictable machine action can result. **Important:** You must define a diagnostics file for a channel configured for anything but unused (even if you are not using the channel) if you want status information for that channel.
Ethernet Address	The processor's Ethernet hardware address. Display only	Assigned by Allen-Bradley and cannot be changed. Displayed as a set of 6 bytes (in hex), separated by colons.
BOOTP Enable	Whether BOOTP is enabled.	Select NO. **Before** you specify `No`, make sure you have an IP address specified. With BOOTP set to `No`, the processor uses the parameters that you specify locally. To enable BOOTP, see the page 12-5.
IP Address	The processor's Internet address.	Disable BOOTP first. **You cannot manually change the IP address with 6200 software if BOOTP is enabled.** Enter an address in this form: a.b.c.dWhere: a, b, c, d are between 1-254 (decimal) You must specify the IP address to have the processor connect to the TCP/IP network. Do not use 0 or 255 as a, b, c, or d in the IP address.
Message Connect Timeout	The number of milliseconds allowed for an MSG instruction to establish a connection with the destination node.	Enter a timeout period in milliseconds. (The processor rounds to the nearest 250 ms.) The valid range for a timeout period is 0-65,535 ms. The default is 15,000 ms.
Message Reply Timeout	The number of milliseconds the Ethernet interface waits for a reply to a command it initiated (through an MSG instruction).	Enter a timeout period in milliseconds. (The processor rounds to the nearest 250 ms.) The valid range for a timeout period is 0-65,535 ms. The default is 3,000 ms.
Inactivity Timeout	The number of minutes of inactivity before the connection is closed.	Enter a timeout period in minutes. The valid range for a timeout period is 0-65,535 minutes. The default is 30 minutes.
Advanced Functions		
Broadcast Address	The broadcast address to which the processor should respond.	See page 12-8 for information about advanced network functions, including the use of broadcast addressing. This function does not allow for sending one message simultaneously to multiple PLC-5E processors.
Subnet Mask	The processor's subnet mask. The subnet mask is used to interpret IP addresses when the network is divided into subnets.	See page 12-9 for information about subnetworks and gateways.
Gateway Address	The IP address of the gateway that provides a connection to another IP network. This field is required when you communicate with other devices not on a local subnet.	See page 12-9 for information about subnetworks and gateways.
Link ID	A DH+ link number Use the link ID number to identify the network when configuring a ControlLogix system using the Gateway software.	Enter a link ID number. The valid range is 0-199.

Using BOOTP to Enter Configuration Information

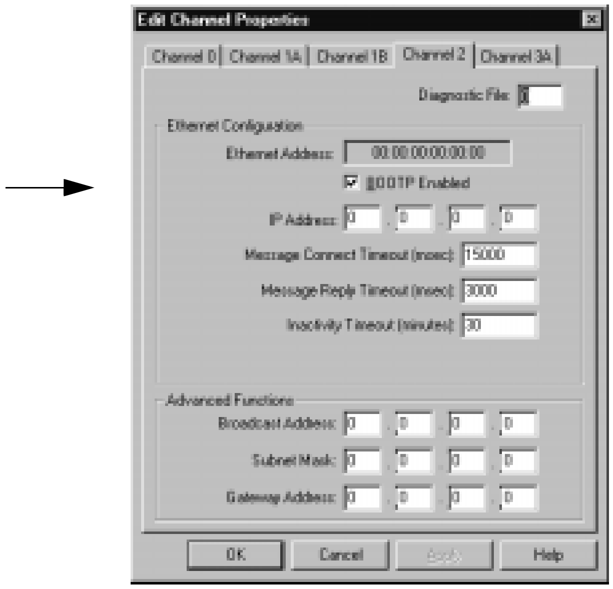
You can also use BOOTP to obtain subnet masks and gateway addresses. See page 12-10.

BOOTP is a protocol that will supply the processor with configuration information at power-up. BOOTP lets you dynamically assign IP addresses to processors on the Ethernet link.

To use BOOTP, a BOOTP server must exist on the local Ethernet subnet. The server is a computer (either a personal computer, VAX, or UNIX system) that has BOOTP-server software installed and reads a text file containing network information for individual nodes on the network.

To enable BOOTP, use the Ethernet channel 2 configuration screen in your programming software. Specify YES for `BOOTP Enable`.

Important: If you change this field from NO to YES, the change does not take effect until you cycle power.



Diagnostic File: 0

Ethernet Address: 00:00:00:00:00:00

☑ BOOTP Enabled

IP Address: 0 . 0 . 0 . 0

Message Connect Timeout (msec): 15000

Message Reply Timeout (msec): 3000

Inactivity Timeout (minutes): 30

Broadcast Address: 0 . 0 . 0 . 0

Subnet Mask: 0 . 0 . 0 . 0

Gateway Address: 0 . 0 . 0 . 0

Specify further configuration information using this screen. See Table 12.A on page 12-4 for other field descriptions.

When BOOTP is enabled, the following events occur at power-up:

- The processor broadcasts a BOOTP-request message containing its hardware address over the local network or subnet.
- The BOOTP server compares the hardware address with the addresses in its look-up table in the BOOTPTAB file.
- The BOOTP server sends a message back to the processor with the IP address and other network information that corresponds to the hardware address it received.

With all hardware and IP addresses in one location, you can easily change IP addresses in the BOOTP configuration file if your network needs change.

If you have BOOTP enabled and the message `BOOTP response not received` appears, check the cabling connections and the BOOTP server system.

Important: If BOOTP is disabled, or no BOOTP server exists on the network, you must use PLC-5 programming software to enter/change the IP address for each processor.

Editing the BOOTPTAB Configuration File

Important: Be sure you know the Ethernet hardware address of the module. You will enter it in this file.

You must edit the `BOOTPTAB` file, which is an ASCII text file, to include the name, IP address, and hardware address for each Ethernet PLC-5 processor you want the server to boot. To edit this file:

1. Open the `BOOTPTAB` file using a text editor.
    - The file contains lines that look like this:

      ```
      #Default string for each type of Ethernet client
      defaults5E: ht=1:vm=rfc1048
      ```

      These are the default parameters for Ethernet PLC-5 processors and must always precede the client lines in the BOOTPTAB file.
    - The file also contains a line that looks like this:

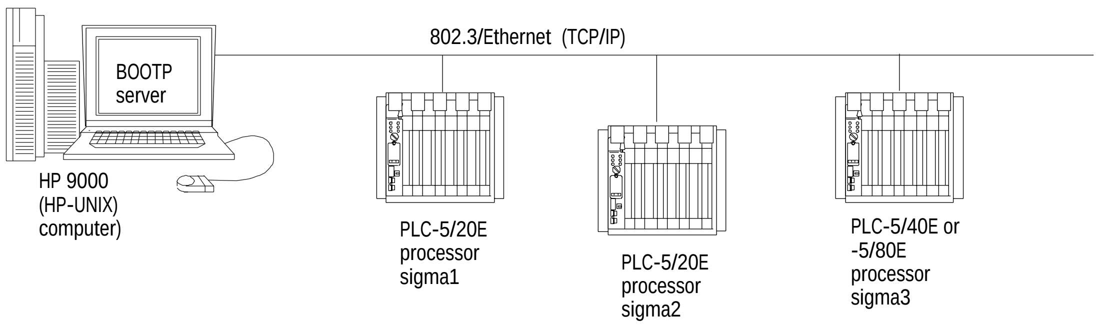
```
plc5name: tc=defaults5E:ip=aa.bb.cc.dd:ha=0000BC1Cxxyy
```

Important: Use this line as the configuration template for Ethernet PLC-5 processors.

2. Make one copy of the Ethernet PLC-5 processor template for every Ethernet PLC-5 processor in your system.
3. Edit each copy of the template as follows:
   - **A.** Replace `plc5name` with the name of the Ethernet PLC-5 processor. Use only letters and numbers; do not use underscores.
   - **B.** Replace `aa.bb.cc.dd` with the IP address to be assigned to the processor.
   - **C.** Replace `xxyy` with the last four digits of the hardware address. Use only valid hexadecimal digits (0-9, A-F); do not use the hyphens that separate the numbers. (You will find the hardware address on a label affixed to the printed circuit board of the Ethernet PLC-5 processor.)
4. Save, close, and make a backup copy of this file.

Example
In this example there are three Ethernet PLC-5 processors and an HP 9000 personal computer. The names and hardware addresses are device specific:

Device	Name	IP Address	Hardware Address
Ethernet PLC-5	sigma1	12.34.56.1	00:00:BC:1C:12:34
Ethernet PLC-5	sigma2	12.34.56.2	00:00:BC:1C:56:78
Ethernet PLC-5	sigma3	12.34.56.3	00:00:BC:1C:90:12


802.3/Ethernet (TCP/IP)
BOOTP server
HP 9000 (HP-UNIX) computer)
PLC-5/20E processor sigma1
PLC-5/20E processor sigma2
PLC-5/40E or -5/80E processor sigma3

Based on this configuration, the BOOTPTAB file looks like:

```
#      Legend:    gw -- gateways
#                 ha -- hardware address
#                 ht -- hardware type1
#                 ip -- host IP address
#                 sm -- subnet mask
#                 vm -- BOOTP vendor extensions format2
#                 tc -- template host

#Default string for each type of Ethernet client
defaults5E: ht=1:vm=rfc1048

#Entries for Ethernet PLC-5 processors:
device1: tc=defaults5E:ip=12.34.56.1:ha=0000BC1C1234
device2: tc=defaults5E:ip=12.34.56.2:ha=0000BC1C5678
device4: tc=defaults5E:ip=12.34.56.3:ha=0000BC1C9012
```

1. 1 = 10MB Ethernet
2. use rfc1048

Run your BOOTP utility to send the configuration information to the Ethernet interface module.

Using Advanced Ethernet Functions

Configure the following advanced communication characteristics using the Ethernet channel 2 configuration screen:

- broadcast address
- subnet mask
- gateway address

If you are using	See page
Broadcast addressing	12-8
Subnet masks and gateways	12-9

Important: If BOOTP is enabled, you can't change any of the advanced Ethernet communications characteristics.

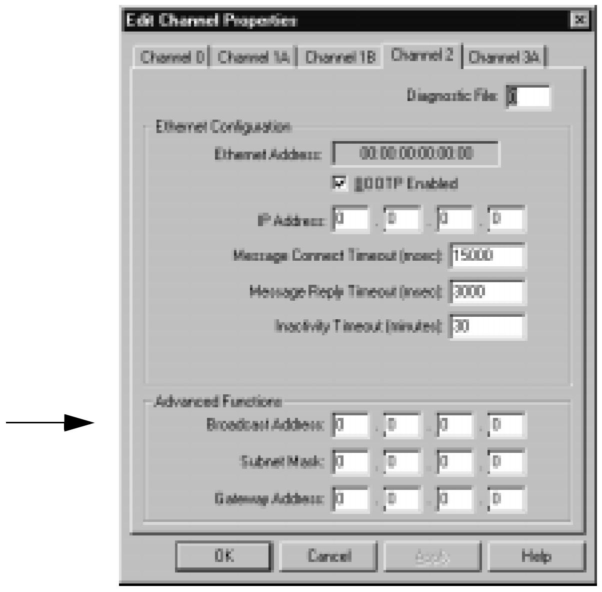
Using Broadcast Addressing

The broadcast address is part of the IP protocol used by a host to send messages to every IP address on the link. This field in the channel 2 configuration screen identifies the address on which the module will receive broadcast messages sent by a host.

Important: The broadcast address is used only for the reception of messages. When used in the context of Ethernet addressing, the broadcast function does **not** refer to ladder-logic messaging.

This function does not allow for sending one message to multiple PLC-5E processors at the same time.

In most cases, you can leave the broadcast address at the default setting.



Diagnostic File: 0
Ethernet Address: 00:00:00:00:00:00
☑ BOOTP Enabled
IP Address: 0 . 0 . 0 . 0
Message Connect Timeout (msec): 15000
Message Reply Timeout (msec): 3000
Inactivity Timeout (minutes): 30
Broadcast Address: 0 . 0 . 0 . 0
Subnet Mask: 0 . 0 . 0 . 0
Gateway Address: 0 . 0 . 0 . 0

Configure this field:	By doing the following:
Broadcast Address	Cursor to the field, and enter an address of the following form: a.b.c.dWhere: a, b, c, d are between 0-255 (decimal) If you change the default and need to reset it, type 0.0.0.0.

Using Subnet Masks and Gateways

If your network is divided into subnetworks that use gateways or routers, you must indicate the following information when configuring channel 2:

- subnet mask
- gateway address

A *subnet mask* is a filter that a node applies to IP addresses to determine if an address is on the local subnet or on another subnet. If an address is located on another subnetwork, messages are routed through a local gateway to be transferred to the destination subnetwork.

MORE

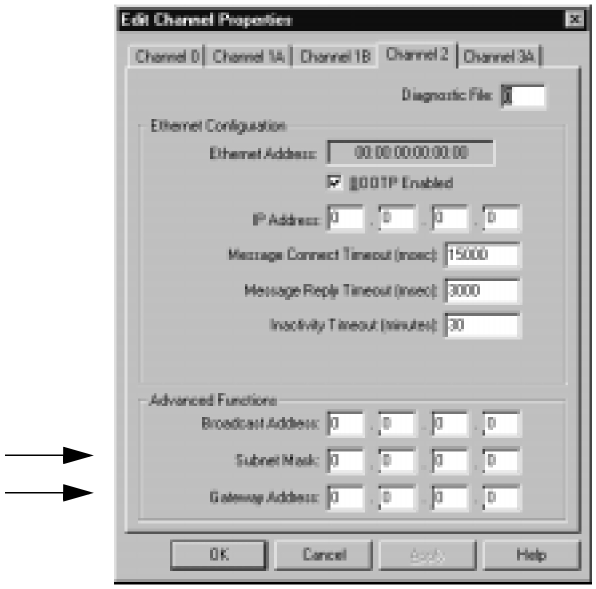
For more information about using subnet masks and gateways, see Comer, Douglas E; *Internetworking with TCP-IP, Volume 1: Protocols and Architecture;* Englewood Cliffs, N.J.: Prentice-Hall, 1990.

If your network is not divided into subnets, then leave the subnet mask field at the default.

If you are	Then	See page
manually configuring channel 2 and have a network with subnets	• be sure the BOOTP enable field is set to `No` • use your programming software to enter the subnet mask and gateway address; see Table 12.B.	12-10
using BOOTP to configure channel 2 and have a network with subnets	• be sure BOOTP is enabled • configure the BOOTPTAB file to include the subnet mask(s) and gateway address(es)	12-11

Manually Configuring Channel 2 for Processors on Subnets

If you are manually configuring channel 2 for a processor located on a subnet, see Table 12.B to configure the subnet mask and gateway address fields for each processor via your programming software.



Diagnostic File: 0

Ethernet Configuration

Ethernet Address: 00:00:00:00:00:00

☑ BOOTP Enabled

IP Address: 0 . 0 . 0 . 0

Message Connect Timeout (msec): 15000

Message Reply Timeout (msec): 3000

Inactivity Timeout (minutes): 30

Advanced Functions

Broadcast Address: 0 . 0 . 0 . 0

Subnet Mask: 0 . 0 . 0 . 0

Gateway Address: 0 . 0 . 0 . 0

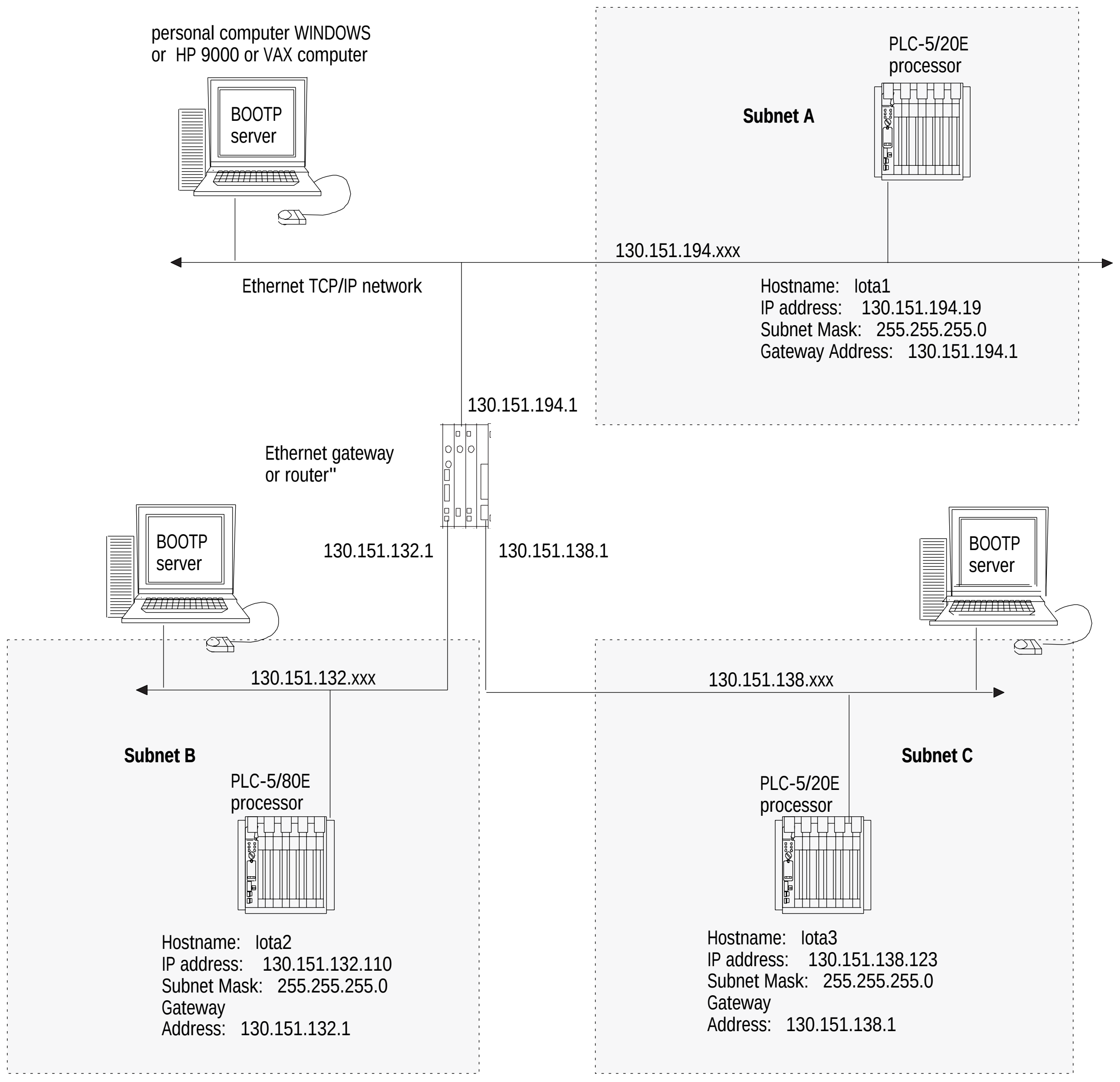
Table 12.B
Ethernet Channel 2 Configuration Screen Advanced Functions

This field:	Specifies:	Configure by doing the following:
Subnet Mask	The processor's subnet mask. The subnet mask is used to interpret IP addresses when the internet is divided into subnets.	Enter an address of the following form: a.b.c.dWhere: a, b, c, d are between 0-255 (decimal) If your network is not divided into subnets, then leave the subnet mask field at the default. If you change the default and need to reset it, type 0.0.0.0.
Gateway Address	The IP address of the gateway that provides a connection to another IP network. This field is required when you communicate with other devices not on a local subnet.	Enter an address of the following form: a.b.c.dWhere: a, b, c, d are between 0-255 (decimal) The default address is `No Gateway`.

Using BOOTP to Configure Channel 2 for Processors on Subnets

Configure the BOOTPTAB file according to the subnet mask and gateway address for each PLC-5E processor on the link. See the example below and the corresponding BOOTPTAB file on the next page.

Important: Because BOOTP requests are seen only on the local subnet, each subnet needs its own BOOTP server and BOOTPTAB file.


personal computer WINDOWS or HP 9000 or VAX computer
BOOTP server
Subnet A
PLC-5/20E processor
130.151.194.xxx
Ethernet TCP/IP network
Hostname: Iota1
IP address: 130.151.194.19
Subnet Mask: 255.255.255.0
Gateway Address: 130.151.194.1
130.151.194.1
Ethernet gateway or router"
BOOTP server
130.151.132.1
130.151.138.1
BOOTP server
130.151.132.xxx
130.151.138.xxx
Subnet B
PLC-5/80E processor
Subnet C
PLC-5/20E processor
Hostname: Iota2
IP address: 130.151.132.110
Subnet Mask: 255.255.255.0
Gateway Address: 130.151.132.1
Hostname: Iota3
IP address: 130.151.138.123
Subnet Mask: 255.255.255.0
Gateway Address: 130.151.138.1

The BOOTPTAB files that correspond to this example looks like:

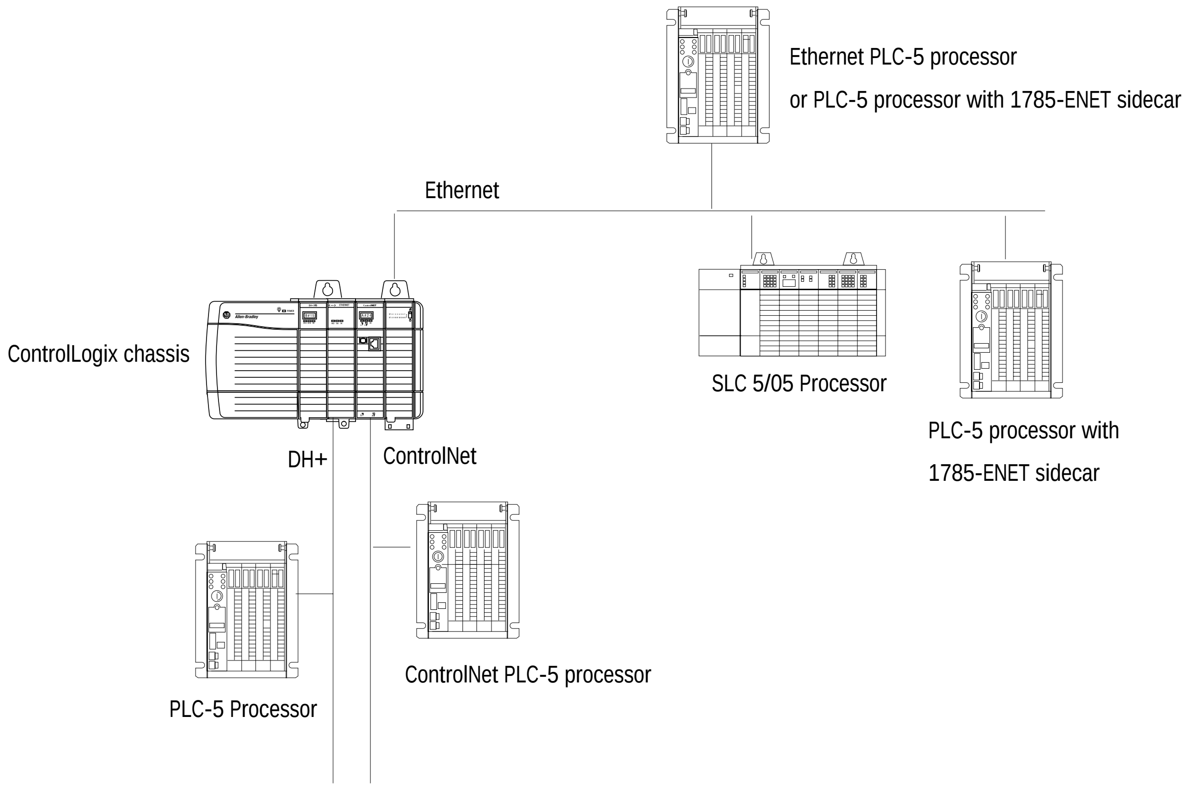
```
#  Legend:   gw -- gateways
#            ha -- hardware address
#            ht -- hardware type
#            ip -- host IP address
#            sm -- subnet mask
#            vm -- BOOTP vendor extensions format
#            tc -- template host

#Default string for each type of Ethernet client
defaults5E: ht=1:vm=rfc1048:sm=255.255.255.0

#Entries for Ethernet PLC-5 processors:
iota1:\
             tc=defaults5E:\
             gw=130.151.194.1:\
             ha=0000BC1C1234:/
             ip=130.151.194.19
```

```
#  Legend:   gw -- gateways
#            ha -- hardware address
#            ht -- hardware type
#            ip -- host IP address
#            sm -- subnet mask
#            vm -- BOOTP vendor extensions format
#            tc -- template host

#Default string for each type of Ethernet client
defaults5E: ht=1:vm=rfc1048:sm=255.255.255.0

#Entries for Ethernet PLC-5 processors:

iota2:\
             tc=defaults5E:\
             gw=130.151.132.1:\
             ha=0000BC1C5678:/
             ip=130.151.132.110
```

```
#  Legend:   gw -- gateways
#            ha -- hardware address
#            ht -- hardware type
#            ip -- host IP address
#            sm -- subnet mask
#            vm -- BOOTP vendor extensions format
#            tc -- template host

#Default string for each type of Ethernet client
defaults5E: ht=1:vm=rfc1048:sm=255.255.255.0

#Entries for Ethernet PLC-5 processors:

iota3:\
             tc=defaults5E:\
             gw=130.151.138.1:\
             ha=0000BC1C9012:/
             ip=130.151.138.123
```

Communicating with ControlLogix Devices

The series E, revision D and later processors can communicate over Ethernet with ControlLogix devices or through a ControlLogix Ethernet (1756-ENET) module to other PLC-5 processors. You need either an Ethernet PLC-5 processor or any PLC-5 processor with a series A, revision E 1785-ENET sidecar module. The following diagram shows an Ethernet PLC-5 processor and the other PLC and SLC processors it can communicate with using a multihop MSG instruction.



To communicate through a ControlLogix 1756-ENET module, you configure the multihop feature of a MSG instruction from the Ethernet PLC-5 processor (or PLC-5 processor with 1785-ENET sidecar module) to the target device. You need RSLogix 5 programming software. For more information, see the MSG instruction in the *PLC-5 Programmable Controller Instruction Set Reference Manual*, publication 1785-6.1.

If you want to go through the ControlLogix 1756-ENET module and out the 1756-DHRIO module to the target device, you:

- use Gateway configuration software to configure the 1756-DHRIO module routing table in the ControlLogix system.
- specify a Link ID number on channel properties for channel 2/3A of the Ethernet PLC-5 processor (or PLC-5 processor with a 1785-ENET sidecar module).

For more information about configuring a PLC-5 channel or specifying the path of the MSG instruction, see the documentation for your programming software.

Interpreting Error Codes

When the processor detects an error during the transfer of message data, the processor sets the .ER bit and enters an error code:

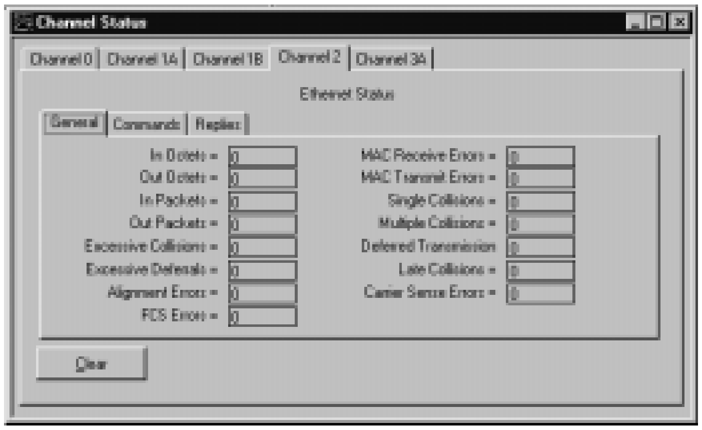
Code - hexadecimal: (word 1 of the control block)	**Description:** (displayed on the data monitor screen)
0010	No IP address configured for the network
0011	Already at maximum number of connections
0012	Invalid internet address or host name
0013	No such host
0014	Cannot communicate with the name server
0015	Connection not completed before user-specified timeout
0016	Connection timed out by the network
0017	Connection refused by destination host
0018	Connection was broken
0019	Reply not received before user-specified timeout
001A	No network buffer space available
0037	Message timed out in local processor
0083	Processor is disconnected
0089	Processor’s message buffer is full
0092	No response (regardless of station type)
00D3	You formatted the control block incorrectly
00D5	Incorrect address for the local data table
1000	Illegal command from local processor
2000	Communication module not working
4000	Processor connected but faulted (hardware)
5000	You used the wrong station number
6000	Requested function is not available
7000	Processor is in program mode
8000	Processor’s compatibility file does not exist

Code - hexadecimal: (word 1 of the control block)	**Description:** (displayed on the data monitor screen)
9000	Remote node cannot buffer command
B000	Processor is downloading so it is inaccessible
F001	Processor incorrectly converted the address
F002	Incomplete address
F003	Incorrect address
F006	Addressed file does not exist in target processor
F007	Destination file is too small for number of words requested
F00A	Target processor cannot put requested information in packets
F00B	Privilege error, access denied
F00C	Requested function is not available
F00D	Request is redundant
F011	Data type requested does not match data available
F012	Incorrect command parameters
F01A	File owner active – the file is being used
F01B	Program owner active – someone is downloading, online editing, or set the program owner with APS in the WHO Active screen

Interpreting Ethernet Status Data

Monitor the status of Ethernet PLC-5 processors by accessing the Ethernet channel 2 status screen of your programming software. The diagnostic counter data displayed is stored in the diagnostic file defined on the Ethernet channel 2 configuration screen.

Monitoring general Ethernet status

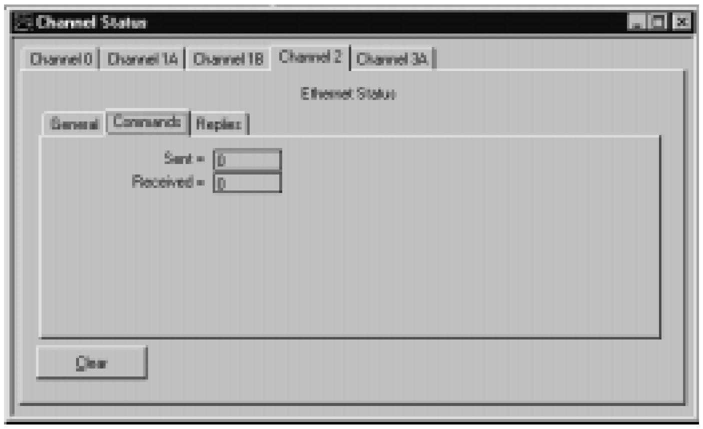


In Octets = 0
Out Octets = 0
In Packets = 0
Out Packets = 0
Excessive Collisions = 0
Excessive Deferrals = 0
Alignment Errors = 0
FCS Errors = 0
MAC Receive Errors = 0
MAC Transmit Errors = 0
Single Collisions = 0
Multiple Collisions = 0
Deferred Transmission 0
Late Collisions = 0
Carrier Sense Errors = 0

Status Field:	**Bytes**	**Displays the number of:**
In Octets	28-31	Octets received on the channel
Out Octets	32-35	Octets sent on the channel
In Packets	36-39	Packets received on the channel, including broadcast packets
Out Packets	40-43	Packets sent on the channel, including broadcast packets

Status Field:	Bytes	Displays the number of:
Excessive collisions	56-59	Frames for which a transmission fails due to excessive collisions
Excessive deferrals	60-63	Frames for which transmission is deferred for an excessive period of time
Alignment errors	44-47	Frames received on the channel that are not an integral number of octets in length
FCS errors	48-51	Frames received on the channel that do not pass the FCS check
MAC receive errors	64-67	Frames for which reception on an interface fails due to internal MAC sublayer receive error
MAC transmit errors	68-71	Frames for which reception on an interface fails due to internal MAC sublayer transmit error
Single collisions	72-75	Successfully transmitted frames for which transmission was delayed because of collision.
Multiple collisions	76-79	Successfully transmitted frames for which transmission was delayed more than once because of collision.
Deferred transmission	80-83	Frames for which the first transmission attempt is delayed because the medium is busy
Late collisions	84-87	Times that a collision is detected later than 512 bit-times into the transmission of a packet
Carrier sense errors	52-55	Times that the carrier sense condition was lost or never asserted while trying to transmit a frame

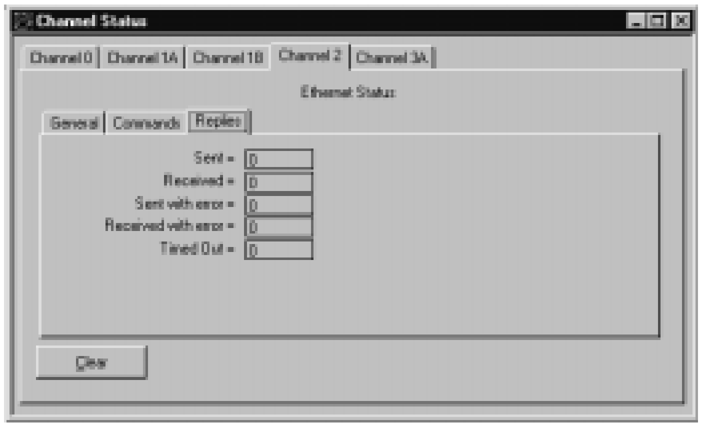
Monitoring Ethernet commands



Sent = 0
Received = 0

Status Field:	Bytes	Displays the number of:
Sent	0-3	Commands sent by the channel
Received	4-7	Commands received by the channel

Monitoring Ethernet replies



Sent = 0
Received = 0
Sent with error = 0
Received with error = 0
Timed Out = 0

Status Field:	Bytes	Displays the number of:
Sent	8-11	Replies sent by the channel
Received	12-15	Replies received by the channel
Sent with error	16-19	Replies containing errors sent by the channel
Received with error	20-23	Replies containing errors received by the channel
Timed out	24-27	Replies not received within the specified timeout period

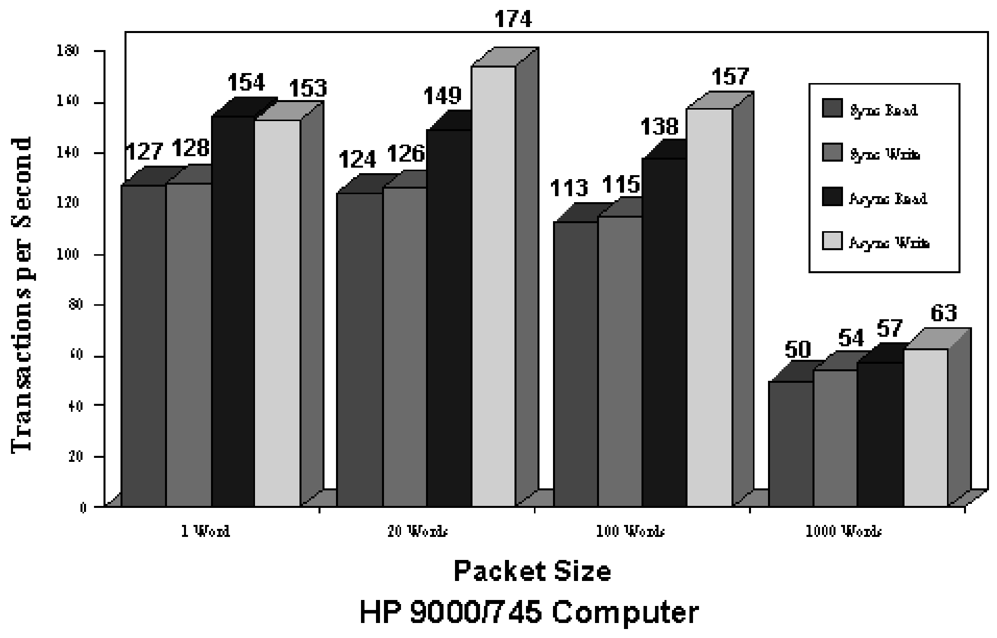
Ethernet PLC-5 Performance Considerations

Actual performance of an Ethernet PLC-5 processor varies according to:

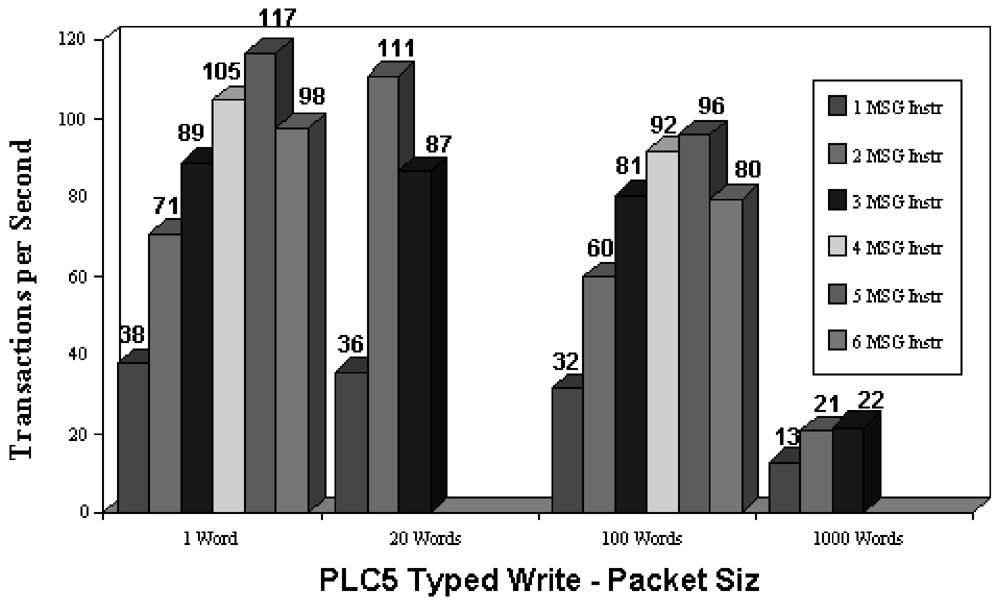
- size of Ethernet messages
- frequency of Ethernet messages
- network loading
- the implementation of and performance of your processor application program

The following charts show performance of the Ethernet PLC-5 processor, depending on packet size.

Performance: Host to Ethernet PLC-5 Processor



Performance: Ethernet PLC-5 Processor to Ethernet PLC-5 Processor



Protecting Your Programs

Using This Chapter

Read this chapter for an overview of:

- defining privilege classes
- assigning a privilege class to a channel or offline file
- assigning a privilege class to a node
- assigning read/write privileges to a program file
- assigning read/write privileges to a data file

Important: To use these options, select the full passwords and privileges feature when you install the software.

MORE

For detailed information about configuring privileges, see the documentation for your programming software.

If your application requires privileges beyond those provided by the enhanced or Ethernet PLC-5 processors, see the *PLC-5 Protected Processor Product Data* for 1785-5/26, -5/46, and -5/86 processors, publication 1785-2.28.

About Passwords and Privileges

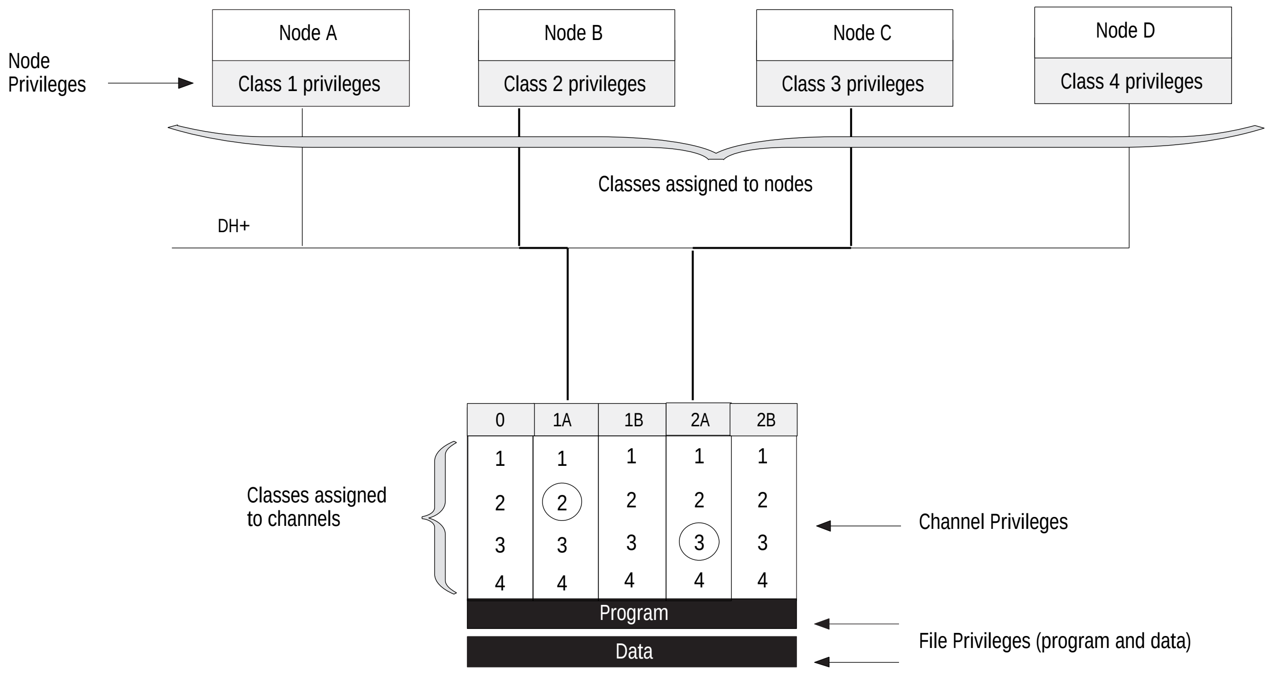
The passwords and privileges function supported by enhanced and Ethernet PLC-5 processors helps you protect your programs by restricting access to processor files and functions.

You can assign a **privilege class** to a node, channel or file. The privilege class defines the level of access (read or write) or type of function (I/O forcing, memory clearing) the PLC-5 processor allows.

This privilege:	Restricts access:
Node	from a particular node to the processor.
Channel	to a particular channel on the processor.
File	to view or change a file.

Important: Node privileges override the default privilege class of the channel.

Figure 13.1
Privileges Supported by Enhanced and Ethernet PLC-5 Processors



In Figure 13.1, the class privileges assigned to each node govern the access the device has to the processor. For example:

- Node B has Class 2 access to channel 1A, based on the node privilege the processor has assigned it
- Node C has Class 3 access to channel 2A, based on the node privilege the processor has assigned it

Important: If node privileges had not been assigned in this example, the node would have had the same privilege class as that assigned to its channel.

Design Tip

Follow these guidelines when using the passwords and privileges:

- You must define the passwords and privileges information for each processor in your system.
- You **cannot** assign default class privileges to channels configured as scanner or adapter. The read/write privileges you see on the channel privileges screen apply to read/write access of the channel configuration screen of that channel. The read/write privileges for each channel's diagnostic file (channel status screen) must be set up through the data table privileges screen. The default privilege fields on the channel privileges screen determine the privilege class of all stations/nodes that are attached through that channel.
- Tell all of the users of your software which privilege class they can use and the appropriate password. If they want to change to a different class (other than the one for which the personal computer is configured), they must enter the new class and password.
- The passwords and privileges feature helps prevent unauthorized or accidental changes to the system. However, the passwords and privileges feature has limitations; it will not prevent acts of malicious tampering nor can it ensure that changes made by an individual with the password will be appropriate for a particular application.

Defining Privilege Classes

You can define four privilege classes (class 1-4), each with its own password. Within each class, you then can assign access to certain operations in the software (such as modifying program or data files, or channel configurations). These privilege classes are the upper level organization for your password structure.

You can define Class 1 to have all privileges, equivalent to a system manager. Then, define the remaining three classes to have fewer privileges.

For example, set your privilege classes as follows on the channel privileges screen of your programming software (an X indicates that the privilege is enabled):

Privileges \ Privilege Class Names	Class1	Class2	Class3	Class4
Modify Privileges	X			
Data Table File Create/Delete	X	X	X	
Program File Create/Delete	X	X	X	
Logical Write	X	X	X	X
Physical Write	X	X	X	X
Logical Read	X	X	X	X
Physical Read	X	X	X	X
Mode Change	X	X	X	X
I/O Force	X	X		
SFC Force	X	X		
Clear Memory	X			
Restore	X			
On-line Editing	X			
Modify passwords	X			

Assigning a Privilege Class to a Channel or Offline File

You can assign a privilege class to all channels (except remote I/O scanner or adapter channels) and offline files. Each channel and offline file has a Class 1 privilege by default.

The read/write privileges you see on the channel privileges screen apply to read/write access to the channel configuration screen of that channel. The read/write privileges for each channel's diagnostic file (channel status screen) must be set up through the data table privileges screen. The default privilege fields on the channel privileges screen determine the privilege class of all stations/nodes that are attached through that channel.

Important: You cannot assign default class privileges to scanner or adapter channels.

Assigning a Privilege Class to a Node

All stations/nodes default to the same privilege class as that of the channel they communicate through. You can give a node its own privilege class if you want it to have a class different from the default privilege assigned to that channel.

Important: Node privileges override the default privilege class of the channel assigned on the channel privileges screen.

Assigning Read/Write Privileges to a Program File

You can assign read and write privileges for each program file in a processor. These privileges limit the access of users to view or change your program files. Two privileges determine whether a user can read or write to a program file:

- the users' privilege class
- whether read and write privileges have been assigned to the program file itself

Assigning Read/Write Privileges to a Data File

You can assign read and write privileges for each data file in a processor. These privileges limit the access of users to view or change data file values. Two privileges determine whether a user can read or write to a data file:

- the users' privilege class
- whether read and write privileges have been assigned to the data file itself

Important: Removing both the read and write access from a data table file prevents you from accessing that file.

Using Protected Processors

To avoid compromising security when importing and exporting files to and from PLC-5/26, -5/46, or -5/86 processors that contain series C/revision H or later firmware, be sure to program with release 5.0 or later of 6200 Programming Software.

Earlier releases of 6200 software will not communicate with series C/revision H and later protected processors and may incorrectly identify them. Likewise, early releases of programming software from other manufacturers will not recognize series C/revision H and later PLC-5 protected processors.1.

MORE

For more information about programming protected processors, see the *PLC-5 Protected Processor Supplement*, publication 1785-6.5.13.

Notes:

Chapter 14

Programming Considerations

Using This Chapter

Forcing

Forcing I/O lets you turn specific input and output bits on or off for testing purposes. Forcing bits on or off or forcing SFC transitions lets you simulate operation or control of a device.

Important: Forcing inputs lets you force the bits in the input image file. Forcing output lets you force the actual output module, leaving the output image table file in its original state.

Important: Forces are held by the processor (and not the personal computer). Forces remain even if the personal computer is disconnected.

ATTENTION: When anything is forced on or off, keep personnel away from the machine area. Forcing can cause unexpected machine motion that could injure personnel.

Forcing Inputs and Outputs

You can forces bits directly from the ladder editor or the force monitor screens in your programming software. After you configure which bits to force, you must enable forces before the forces take effect.

You can only force live I/O points, which are bits in an input or output word that are physically attached to and configured for your system.

With the processor-resident local rack set for 1/2-slot addressing, you cannot force the input bits for the upper word of any slot that is empty or that has an 8-point or 16-point I/O module. For example, if you have an 8-point or a 16-point I/O module in the first slot of your local rack (words 0 and 1 of the I/O image table, 1/2-slot addressing), you cannot force the input bits for word 1 (I:001) on or off.

You can't force:

- output addresses on input instructions
- input addresses on output instructions
- other bit addresses other than inputs and outputs, such as N, B, T, C, etc. addresses

Forcing SFC Transitions

When you monitor an SFC through your programming software, you can force transitions on or off. This lets you override the flow of your SFC for troubleshooting purposes.

Extended Forcing

Extended forcing is useful when using the 1771-SDN module because it allows you to force discrete I/O on a DeviceNet network. Extended forcing is also useful for forcing analog I/O. With series E, revision B and later Enhanced, Ethernet, and ControlNet PLC-5 processors, you gain the capability to force a total of 1,024 block-transfer data words. These words can be either integer, binary, ASCII, or hexadecimal/BCD data type.

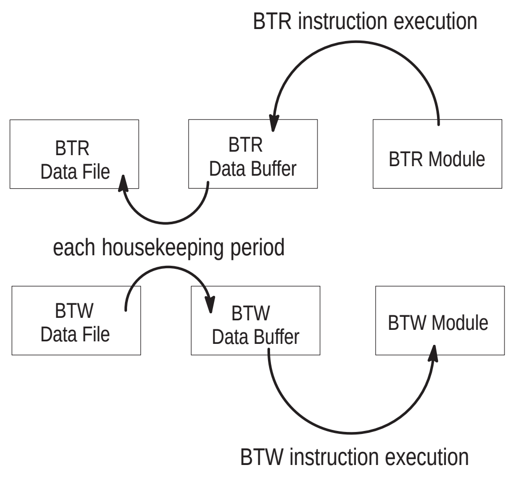
To use the extended forcing feature, you select the block-transfer files that contain words or bits you want to force. You then use your programming software to enter this data along with the associated force values in the extended force configuration table. Once you do this, you can force all data you send or receive via block-transfer instructions.

Extended forcing works with the following block-transfer instructions:

- block-transfer read (BTR)
- block-transfer write (BTW)
- 1771 read command type of the CIO instruction
- 1771 write command type of the CIO instructions

The 1771 read command type of the CIO instruction operates in the same manner as the BTR instruction; the 1771 write command type of the CIO instruction operates in the same manner as the BTW instruction. For simplicity, the following descriptions and examples of extended forcing refer to the BTR instruction (for BTR and 1771 read command type of CIO instructions) and the BTW instruction (for BTW and 1771 write command type of CIO instructions).

You program block-transfer instructions in the same manner, regardless of whether you configure the data file in the extended for configuration table. The following figure shows how block-transfer data table files are updated during housekeeping.



When you use extended forcing, you affect the way your programmable controller system operates, Before you begin to use this forcing feature, read this entire section to understand the effects.

ATTENTION: Any block-transfers or data table locations included in the extended force configuration table will be affected **regardless** of whether forces are enabled.

ATTENTION: Do not use BTR data tables files to store non-block-transfer data. All non-block-transfer data that you include in the extended force configuration table as read data will be forced to zero during housekeeping at the end of each program scan. If your ladder program expects values other than zero for this data, unpredictable machine operation could result.

For BTR instructions using non-configured data tables, the .DN bit indicates when data is valid in the BTR data file. When you configure files in the extended force configuration table, the .DN bit indicates that the data is in the BTR data buffer. The BTR data is not forced and moved into the BTR data file until the next housekeeping period. Delay using the BTR data until the scan after the .DN bit is set.

For BTW instructions using non-configured data tables, the data that is in the BTW data file when the block-transfer is enabled is transferred. When you configure files in the extended force configuration table, the data that in the BTW data buffer when the block-transfer is enabled is transferred. Any new data in the BTW data file is not forced or moved into the BTW data buffer until the next housekeeping period. Delay enabling the BTW instruction until the scan after the BTW data file is updated.

Increased Program Scan Time

When you use the extended forcing feature, program scan time increases proportionately to the number of words you configure in the extended force configuration table. The amount of increase depends on whether you enable or disable forces. Typical increases in program scan time when you configure data table files in the extended force configuration table are:

When forces are:	Scan time increases by this much:	
	per word:	per 1000 words:
enabled	0.003 ms	3.0 ms
disabled	0.0015 ms	1.5 ms

I/O Force Privileges

The I/O forcing privilege lets you enable, disable, or clear all forces in the processor. This privilege now includes extended forcing.

Extended forcing reads force data in a read data file; extended forcing writes force output data, leaving the write data file in its original state.

Important: Forces are held by the processor (and not the personal computer). Forces remain even if the personal computer is disconnected.

Using Protected Processors

If you are using a PLC-5 protected processor, you must configure forcing online since, by their design, protected processors cannot download forcing operations. This protects processor operation from possible force operations programmed in offline mode. For more information about protected processors, see the *PLC-5 Protected Processor Supplement*, publication 1785-6.5.13.

Using Selectable Timed Interrupts (STIs) and Processor Input Interrupts (PIIs)

We recommend that you do not use forcible block-transfer data table files within STIs or PIIS due to the unique data flow of forcible block-transfer data. Block-transfer data that is written out or read in is never valid within the interrupt program file execution itself. Any necessary additional program scan time may defeat the purpose for programming the STI or PII.

Setting Up and Using Extended Forcing

Use your programming software to set up and use the extended forcing feature. The following table lists the software requirements for the extended forcing feature.

With the programming software package:	You need this software release:
RSLogix5	2.0 or later
6200	5.3 or later
A.I. 5	8.03 or later
WinLogic 5	3.22 or later

To set up and use the extended forcing feature, you need to:

1. Select the group of data you want to force.

2. Use the programming software to enter or edit the data you want to force in the extended force configuration table.

3. Use the programming software to enter force values for the specified data table files.

4. Enable or disable the forces.

Step 1 - Select Which Group of Data You Want to Force

Important: Group the data in the extended force configuration table so that you separate read date from write data. If you do not separate read and write data, you encounter error code -3 if:

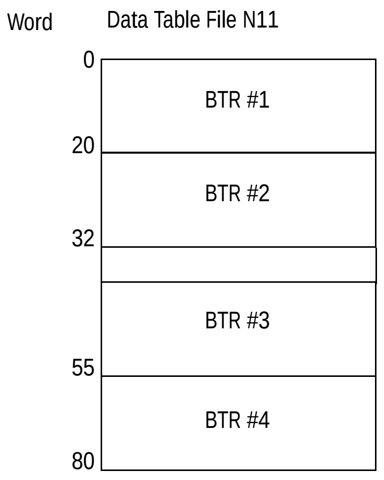
- program a BTW instruction using a data file that you configured in the extended force configuration table as a read application
- program a BTR instruction using a data file that you configured in the extended force configuration table as a write application

You also encounter this error if you try to transfer block-transfer data that crosses the forcible range you configured in the extended force configuration table.

When you select the group of data you want to force, you must select and configure data that corresponds to an entire “chunk” or multiple “chunks” of block-transfer data. For example:

You want to force some data associated with block-transfer read #2 and with block-transfer read #4. To select the data, you could:

- Select all of data file N11
- Select N11 beginning at word 20 for 60 words (i.e., beginning at the start of BTR #2 and ending at the end of BTR #4)
- Make two selections, one beginning at the start of BTR #2 with the size of BTR #2 (N11:20 for 12 words), and one beginning at the start of BTR #4 with the size of BTR #4 (N11:55 for 25 words).



Step 2 - Use the Programming Software to Enter or Edit the Data You Want to Force in the Extended Force Configuration Table

The extended force configuration table lets you specify as many as four groups of block-transfer data words to force. Each group can contain as many as 256 words of block-transfer data. When you plan your forcing, you can group together multiple block-transfer instructions until you reach the 256-word maximum for each group. Keep in mind that the data in each group should be all read data or all write data.

Specify each group by entering the address of the first block-transfer instruction in that group in the extended force configuration table. Use the programming software‘s edit function on the extended force configuration table to clear entries, modify entries, or change block-transfer instructions.

Use your programming software to edit the extended force configuration table:

1. Choose the software option that lets you modify entries in the extended force configuration table.
2. Enter the file number and starting element.
3. Enter the file size (1-256 words).
4. Enter the direction of the instruction (R=read; W=write).

Forced data table files must be of type B, A, N, or D or this error appears: `FORCES MUST BE OF TYPE B, A, N, OR D`

Data files are automatically created and their size automatically increased if necessary. To delete or reduce the size of a data file, you must use the memory map function of the programming software. If you configure a file using the extended force configuration table, you must delete the file from the extended force configuration table before you can delete it from the memory map.

Step 3 - Use the Programming Software to Enter Force Values for the Specified Data Table Files

The block-transfer forcing screens include a function that lets you change the radix among binary, octal, HEX/BCD, and ASCII. If you select the binary radix, the display is similar to the I/O forcing display. The programming software displays forces differently, depending on the selected radix:

Radix:	Force:	Screen display:
binary	no force	. (period)
	off	0
	on	1
other	no force	. (period)
	all bits	forced value
	some bits	BINARY (use binary radix to view the forced bits)

If you enter a force value on the block-transfer force screen, you force the entire word to the value you enter, even if the word was only partially force before.

Step 4 - Enable or Disable the Forces

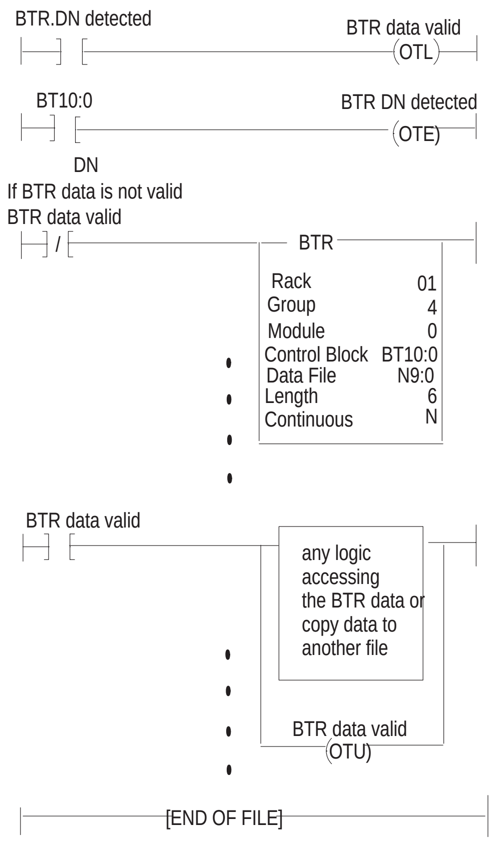
Enabling and disabling extended block-transfer forces is similar to enabling and disabling I/O forces. For more information, see page 14-1.

Using Extended Forcing with Time-Critical Applications

For many applications in which you execute multiple block-transfers on a continuous basis, you do not need any additional programming when using extended forcing. When you configure extended forcing, block-transfer instructions only move data between the block-transfer modules and the block-transfer data buffers. Data is forced and moved during housekeeping. In applications in which you perform a single block-transfer or in which new block-transfer data must be completely transferred in every block-transfer instruction, you must include additional programming to make sure that you are using valid, updated data.

To ensure that the received BTR data table file has been properly updated before you use the data, do the following:

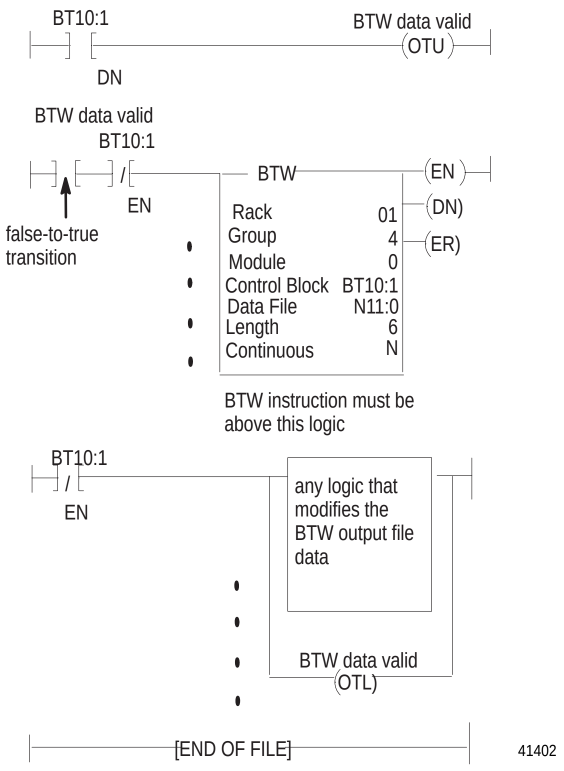
1. Enable the input conditions of the BTR rung.
2. Wait for the BTR done bit to be set.
3. Allow time for housekeeping to force and send the changed data from the block-transfer data buffer to the block-transfer data table file.



After the BTR done bit is set, the valid data in the BT data buffer is copied to the BTR data table file during housekeeping. 41401

To ensure that the intended BTW data table file was properly transferred, do the following:

1. Change the data in the block-transfer output data table.
2. Allow time for housekeeping to force and send the changed data from the block-transfer output data table file to the block-transfer data buffer.
3. Enable the BTW
4. Ensure that data does not change in the block-transfer data table output file until the BTW is complete.



Using Special Programming Routines

Use your design specification to determine if you need one or more of the following special programming routines:

- power-up routines
- fault-driven routines (necessary to safely manage equipment faults)
- time-driven interrupt routines (selectable timed interrupts)
- event-driven interrupt routines (processor input interrupts)

Table 14.A explains when to use these programming features.

Table 14.A
Deciding When to Use Special Routines

If a portion of logic should execute:	Example:	Use:	By doing the following:
Immediately on detecting conditions that require a startup	Restart the system after the system has been shut down	Power-Up Routine	Create a separate file for a controlled start-up procedure for the first time you start a program or when you start a program after system down time. The processor executes the power-up routine to completion.
Immediately on detecting a major fault	Shut down plant floor devices safely upon detecting a major fault or Send critical status to a supervisory processor via DH+ after detecting a major fault	Fault Routine	Create a separate file for a controlled response to a major fault. The first fault detected determines which fault routine is executed. The processor executes the fault routine to completion. If the routine clears the fault, the processor resumes the main logic program where it was interrupted. If not, the processor faults and switches to program mode.
At a specified time interval	Monitor machine position every 250ms and calculate the average rate-of-change or Take a measurement and compare it with a standard every 1.0 seconds	Selectable Timed Interrupt (STI)	Create a separate program file and specify the interrupt time interval. The processor interrupts the main logic program at the specified interval, runs the STI to completion, then resumes the main logic program where it left off. The processor interrupts the main logic program at the specified interval and runs the STIs. When a block-transfer instruction to remote I/O is encountered in an STI, the processor resumes execution of lower priority programs (main logic program) until the block-transfer is completed. When this occurs and you want your STI to run to completion before returning to the main logic program, use UID and UIE instructions in your STI program file.
Immediately when an event occurs	Eject a faulty bottle from a bottling line	Processor Input Interrupt (PII)	Create a separate program file and specify 16 inputs of an input word in the I/O rack. When the event(s) occurs, the processor interrupts the main logic program, runs the PII to completion, then resumes the main logic program where it left off. When a block-transfer instruction to remote I/O is encountered in a PII, the processor resumes execution of lower priority programs (main logic program) until the block-transfer is completed. When this occurs and you want your PII to run to completion before returning to the main logic program, use UID and UIE instructions in your PII program file.

Priority Scheduling for Interrupts and MCPs

PLC-5 processors prioritize when fault routines, interrupts, and main control programs are executed. This prioritization is called "scheduling." The PLC-5 processor considers some scheduling tasks to be of greater importance than others. The scheduling priority of each task is as follows (from highest to lowest):

1. Fault Routine
2. Processor Input Interrupt (PII)
3. Selectable Timed Interrupt (STI)
4. Main Control Program (MCP)

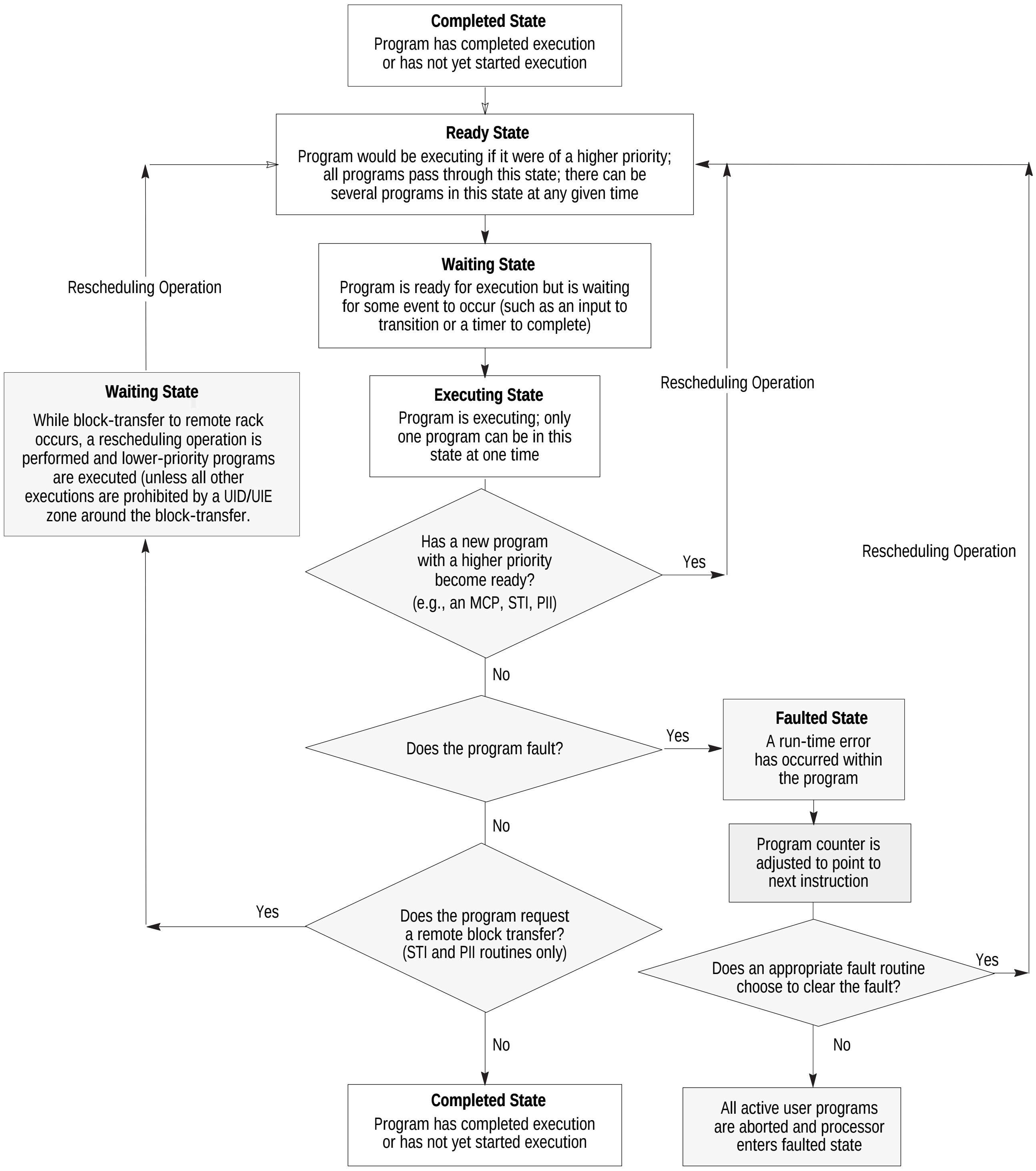
This scheduling determines what controls the program execution path. For example, if a PII is currently executing, it cannot be interrupted by an STI until the PII is completed (since the PII has scheduling priority over the STI). If an MCP is executing and a fault routine is called, however, the MCP's execution will be interrupted because fault routines have priority over the MCPs.

Important: You can temporarily override this priority scheduling by using the UID and UIE instructions. These instructions can be interrupted by a fault routine (see page 14-13).

Fault routines, PIIs, and STIs are interrupt driven. They can execute at any time except during run-time edit operations. MCPs, however, are executed to completion from first user program to last.

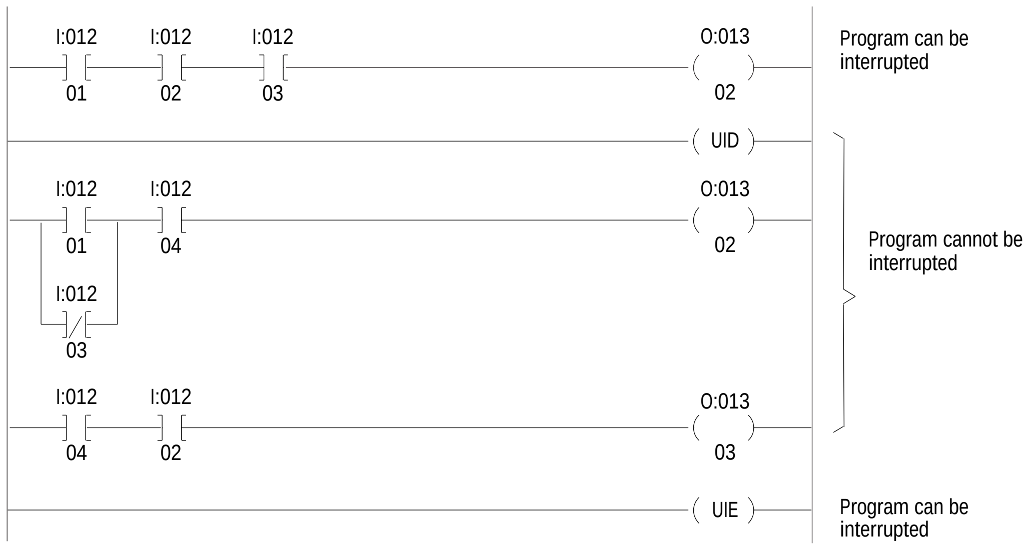
Program Execution States

User programs in the PLC-5 processor are always in one of the following five states: completed, ready, executing, waiting, or faulted.


Completed State
Program has completed execution or has not yet started execution
Ready State
Program would be executing if it were of a higher priority; all programs pass through this state; there can be several programs in this state at any given time
Waiting State
Program is ready for execution but is waiting for some event to occur (such as an input to transition or a timer to complete)
Executing State
Program is executing; only one program can be in this state at one time
Rescheduling Operation
Waiting State
While block-transfer to remote rack occurs, a rescheduling operation is performed and lower-priority programs are executed (unless all other executions are prohibited by a UID/UIE zone around the block-transfer.
Rescheduling Operation
Rescheduling Operation
Has a new program with a higher priority become ready? (e.g., an MCP, STI, PII)
Yes
No
Does the program fault?
Yes
Faulted State
A run-time error has occurred within the program
No
Program counter is adjusted to point to next instruction
Yes
Does the program request a remote block transfer? (STI and PII routines only)
Does an appropriate fault routine choose to clear the fault?
Yes
No
No
Completed State
Program has completed execution or has not yet started execution
All active user programs are aborted and processor enters faulted state

Influencing Priority Scheduling

Use the UID (user interrupt disable) and UIE (user interrupt enable) instructions to influence user program scheduling. They can be used to protect important portions of ladder logic that must be executed through to completion. The UID/UIE instructions are designed to be used in pairs. For example:



After a UID instruction has executed, interrupts are postponed. The interrupt program is placed in the `ready` state. After a UIE instruction has executed, any user programs that are currently in the `ready` state are checked for priority. If the **ready** program is of a higher priority than the currently executing program, the executing program returns to the `ready` status while the interrupt program begins executing. While the processor is executing within a UID/UIE zone, the executing program cannot be interrupted except by a fault routine.

MORE

For more information on programming UID or UIE instructions, see the *PLC-5 Programming Software Instruction Set Reference*, publication 1785-6.1.

Defining and Programming Interrupt Routines

For information about configuring and programming these routines, see the appropriate chapter:

For information about:	See chapter:
Power-up routines	15
Fault routines	16
Main control programs (MCPs)	17
Selectable timed interrupts (STIs)	18
Processor input interrupts (PIIs)	19

Notes:

Preparing Power-Up Routines

Using This Chapter

For information about:	Go to page:
Setting power-up protection	15-1
Allowing or inhibiting startup	15-1
Defining processor power-up procedure	15-2

Setting Power-Up Protection

You can configure your processor so that if a power-loss is experienced while in run mode, the processor does not come back up in run mode. User control bit S:26/1 defines whether power-up protection (e.g., fault routine) is executed upon power-up.

If S:26/1 is:	After power loss, the processor:
Set (1)	Scans the fault routine before returning to normal program scan When set, the processor scans the fault routine once to completion after the processor recovers from a power loss. You can program the fault routine to determine whether the processor's status will let the processor respond correctly to logic and whether to allow or inhibit the startup of the processor.
Reset (0)	Powers up directly at the first rung on the first program file

Set S:26/1 manually from the Processor Status screen, or latch this bit through ladder logic.

Allowing or Inhibiting Startup

Major fault bit S:11/5 controls whether you can power up the processor in run mode after a loss of power. Do not confuse this bit with user control bit S:26/1.

This bit:	Tells the processor:
user control S:26/1	whether or not to scan a fault routine upon power up before returning to normal program scan.
major fault S:11/5	whether or not to fault at the end of scanning the fault routine.

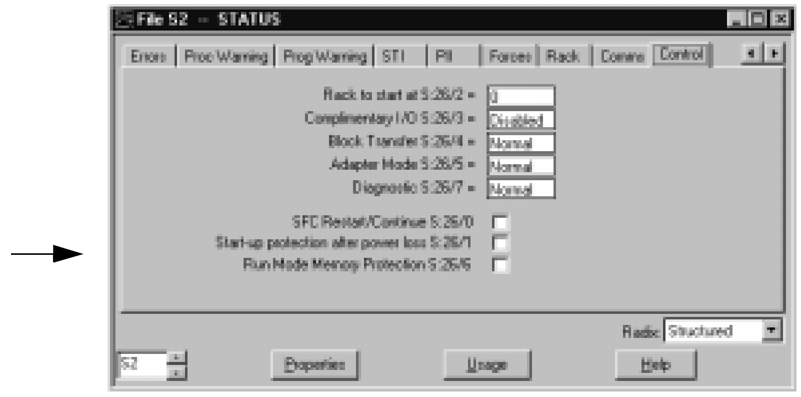
After a power loss is experienced while the processor is in run mode, the processor automatically sets major fault bit S:11/5 if user control bit S:26/1 has been set.

If the fault routine makes S:11/5:	Then the processor:
Set (1)	Faults at the end of scanning the fault routine Leave this bit set to inhibit startup
Reset (0)	Resumes scanning the processor memory file Reset this bit to allow startup

Important: You can use JMP and LBL instructions to scan only the portion of the fault routine associated with a particular fault or power-up condition.

Defining a Processor Power-Up Procedure

The user control bits S:26/0 and S:26/1 define how the processor starts in run mode after a power loss or when you switch to run mode from program or test mode.



Rack to start at S:26/2 = 0
Complimentary I/O S:26/3 = Disabled
Block Transfer S:26/4 = Normal
Adapter Mode S:26/5 = Normal
Diagnostic S:26/7 = Normal
SFC Restart/Continue S:26/0 ☐
Start-up protection after power loss S:26/1 ☐
Run Mode Memory Protection S:26/6 ☐
Radix: Structured
S2

To set and reset bits:

1. Cursor to the bit location.
2. Set by entering 1; reset the bit by entering 0.

Use this bit:	To:
0	Control processors that are using SFCs This bit determines if the SFC restarts or resumes at the last active step after a power loss.
1	Select power-loss protection If this bit is set and a power loss occurs, the processor sets major fault bit 5 and executes a fault routine you define before it returns to normal program scan.

Table 15.A describes the possible start-up routines. For more information about fault routines, see chapter 16.

See chapter 21 for definitions of the user control bits (S:26/0-6).

Table 15.A
Possible Processor Power-Up Routines

If you are:	With:	And you want to:	Then set bit 0 and 1 as shown: 15.............0
Using SFCs	No fault routine	Restart at the first step	xxxxxxxx xxxxxx00
		Restart at the last active step	xxxxxxxx xxxxxx01
Not using SFCs	Fault routine	Start at the first file	xxxxxxxx xxxxxx0x
		Restart using the fault routine file	xxxxxxxx xxxxxx1x
Using a fault file	SFCs	Restart using the fault file and then the first step	xxxxxxxx xxxxxx10
		Restart using the fault file and then the last active step	xxxxxxxx xxxxxx11
Not using a fault file	Not using SFCs	Start at the first file in the processor’s memory.	xxxxxxxx xxxxxx00
Each x indicates a bit that can be 0 or 1 for the status value described.			

Notes:

Preparing Fault Routines

Using This Chapter

Understanding the Fault Routine Concept

Fault routines execute when a PLC-5 processor encounters a major fault during program execution.

Use a fault routine to specify how you want the processor to respond to a major fault. If your processor experiences a fault during program execution, you can tell the processor to interrupt the current program, run your fault routine, and then continue processing the original program.

A fault routine processes the major fault bit found in S:11 and determines the course of program execution based on the fault bit present. Fault routines provide a means to either:

- systematically shut down a process or control operation
- log and clear the fault and continue normal operation

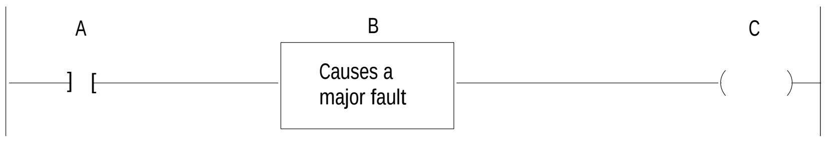
For a detailed list of the words in the processor status file, see chapter 21.

Responses to a Major Fault

When the processor detects a major fault, the processor immediately interrupts the current program. If a fault routine exists (i.e., program file is specified in S:29 as a fault routine), the processor runs that fault routine program for recoverable faults. Depending on the type of fault, the processor then:

- returns to the current ladder program file if the processor can recover from the fault
- enters fault mode if the processor cannot recover from the fault

For example, this rung includes an instruction that causes a major fault:



In this example, the processor runs the fault routine after detecting the fault. If the fault routine resets the faulted bits, the processor returns to the next instruction in the program file that follows the one that faulted and continues executing the remainder of the rung.
If you do not program a fault routine for fault B, the processor immediately faults.

Understanding Processor-Detected Major Faults

In general:

If the processor detects a:	It sets:
major fault	a major fault bit and resets I/O
hardware fault	outputs in 1771-ASB remote I/O racks and/or 1771-ALX extended-local I/O racks are set according to their last state switch setting The outputs remain in their last state or they are de-energized, based on how you set the last state switch in the I/O chassis.

Important: In the PLC-5 processor-resident chassis, outputs are reset regardless of the last state switch setting when one of the following occurs:

•processor detects a major fault

•you set a status file bit to reset a local rack

•you select program or test mode

To decide how to set this switch, evaluate how the machines in your process will be affected by a fault. For example:

- how will the machine react to outputs remaining in their last state or to outputs being automatically de-energized?
- what is each output connected to?
- will machine motion continue?
- could this cause the control of your process to become unstable?

To set this switch, see chapter 23.

Fault in a Processor-Resident or Extended-Local I/O Rack

If a problem occurs with the chassis backplane, the processor sets the appropriate minor fault bit (S:7/0-7) and continues scanning the program and I/O. As soon as this bit is set, the outputs for that rack are reset. However, the processor continues normal run-time operation.

The outputs are enabled again only if the faulted rack condition is cleared. For example, if a local I/O module faults, all outputs in that rack are reset and the processor continues executing the program scan. Outputs will be enabled only after the faulted module is removed.

Your ladder program should monitor the I/O rack fault bits (S:7/0-7) and take the appropriate recovery action.

⚠ **ATTENTION:** If a processor-resident local I/O rack fault occurs and you have no recovery methods, the input image table and outputs for the faulted rack remain in their last state. Potential injury to personnel and damage to the machine may result.

Fault in a Remote I/O Chassis

A remote I/O rack fault can be a loss of communications with the remote I/O device or a problem with the remote I/O device itself. When the processor detects a remote I/O rack fault, the processor sets an I/O rack fault bit in the processor status table. The processor then continues scanning the program and controlling I/O.

The outputs in the faulted rack remain in their last state or they are de-energized, based on how you set the last state switch in the I/O chassis.

⚠ **ATTENTION:** If outputs are controlled by inputs in a different rack and a remote I/O rack fault occurs (in the inputs rack), the inputs are left in their last non-faulted state. The outputs may not be properly controlled and potential injury to personnel and damage to the machine may result. Be sure you have recovery methods.

Outputs in the processor-resident chassis and in any non-faulted remote rack can remain active if a remote I/O rack fault is detected. Make sure to design your program so that the system goes to a known state in the event that outputs in the processor-resident chassis or non-faulted remote racks are controlled by inputs from the faulted remote I/O rack. Your program must be able to account for the inputs remaining in their last state or the program must monitor the rack fault status bits and reset the input image data table to make remote inputs inactive.

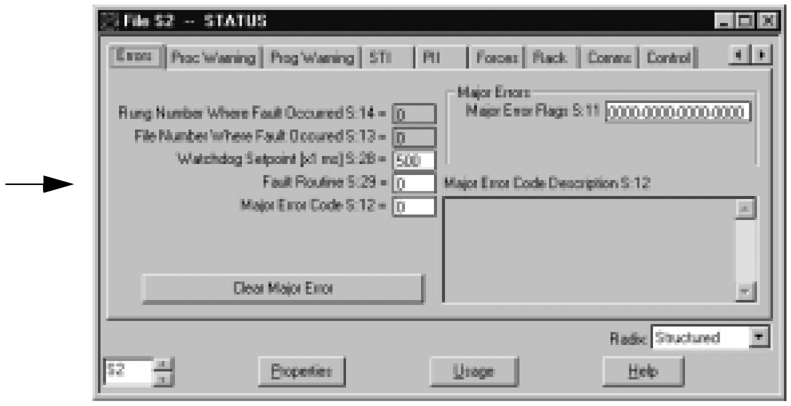
Here are two programming methods you can use:

- In the very first executable instruction, the program monitors the rack fault bits. If any faults bits are set, the program copies zeros (0) to the faulted rack's input image data table. The program must continually copy zeros at the beginning of the program scan to the input image table as long as the fault condition remains because the processor sets the input image bits back to the last state at the end of the program scan.
- In the very first executable instruction, the program monitors the rack fault bits. If any fault bits are set, the program sets the corresponding inhibit bit for the faulted rack. The program must then execute a one time copy of zeros to the faulted rack's input image table to reset all inputs.

Defining a Fault Routine

You can write multiple fault routine programs and store them in multiple fault routine files, but the processor runs only one fault routine program when the PLC-5 processor detects a major fault. You can, however, change the fault routine program that is to be run through ladder logic. If you do not specify a program file number, the processor immediately enters fault mode after detecting a fault.

To define a processor fault routine:



Rung Number Where Fault Occured S:14 = 0
File Number Where Fault Occured S:13 = 0
Watchdog Setpoint [x1 ms] S:28 = 500
Fault Routine S:29 = 0
Major Error Code S:12 = 0
Major Error Flags S:11 0000:0000:0000:0000
Radix: Structured
S2

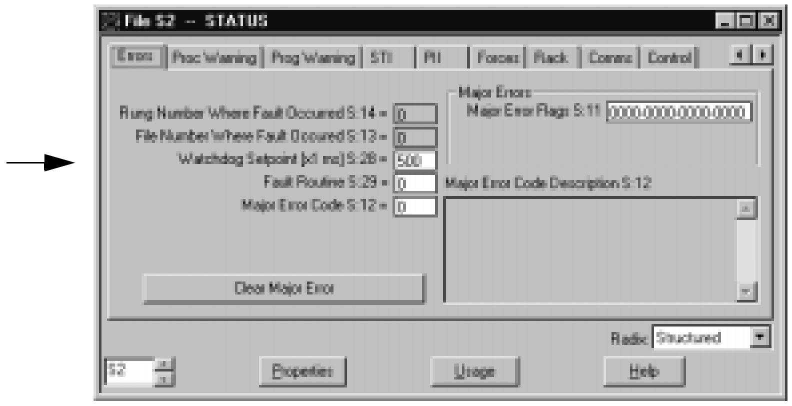
For more information about fault codes, see the documentation for your programming software.

Defining a Watchdog Timer

The watchdog timer (S:28) monitors the program scan. If the scan takes longer than the watchdog timer value, a fault routine is initiated and executed.

The timer is the maximum time (in ms) for the watchdog; or if you use an SFC, it is the maximum time for a single pass through all the active steps.

To define a different value other than the default:



Rung Number Where Fault Occured S:14 = 0
File Number Where Fault Occured S:13 = 0
Watchdog Setpoint [x1 ms] S:28 = 500
Fault Routine S:29 = 0
Major Error Code S:12 = 0
Major Error Flags S:11 0000.0000.0000.0000
Major Error Code Description S:12
Radix: Structured
S2

Important: The watchdog timer can go only as low as 10 ms, even though the programming software allows single-digit inputs.

Avoiding Multiple Watchdog Faults

If you encounter a hardware error or watchdog major fault, it may be because multiple watchdog faults occurred while the processor was busy servicing a ladder-related major fault. The hardware error occurs when the fault queue, which stores a maximum of six faults, becomes full and cannot store the next fault.

Before calling a service representative when you encounter either a hardware error or multiple watchdog faults, check:

If you encounter a:	Then:
watchdog error and a fault bit	Extend the watchdog timer so that the real run-time error is not masked. Check your major fault bits. Ignore the watchdog faults and use any remaining fault bits to help indicate the source of the processor fault.
hardware error	**1.** Power down then power up the processor. **2.** Reload the program. **3.** Set the watchdog timer to a value = 10 current setting **4.** Run the program again.

If you continue to encounter the hardware error, call your Allen-Bradley representative.

Programming a Fault Routine

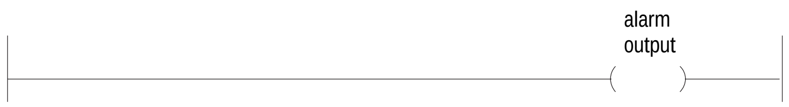
To prepare your fault-routine program, first examine the major fault information recorded by the PLC-5 processor and then decide whether to do the following before the PLC-5 processor automatically goes to fault mode:

- set an alarm
- clear the fault
- execute the appropriate fault routine through ladder logic
- execute the appropriate ladder logic to recover from a fault

Important: If the PLC-5 processor detects a fault in the fault routine (double-fault condition), the PLC-5 processor goes directly to fault mode without completing the fault routine.

Setting an Alarm

If you need an alarm to signal the occurrence of a major fault, put this rung first in your fault routine program:



and combine it with a counter. You can also set an alarm in your fault routine to signal when the fault routine clears a major fault.

Clearing a Major Fault

You can clear a major fault with one of these methods:

- Turn the keyswitch on the PLC-5 processor from REM to PROG to RUN.
- Use the programming software to clear the major fault (if recoverable).

ATTENTION: Clearing a major fault does **not** correct the cause of the fault. Be sure to examine the fault bit and correct the cause of the fault before clearing it.

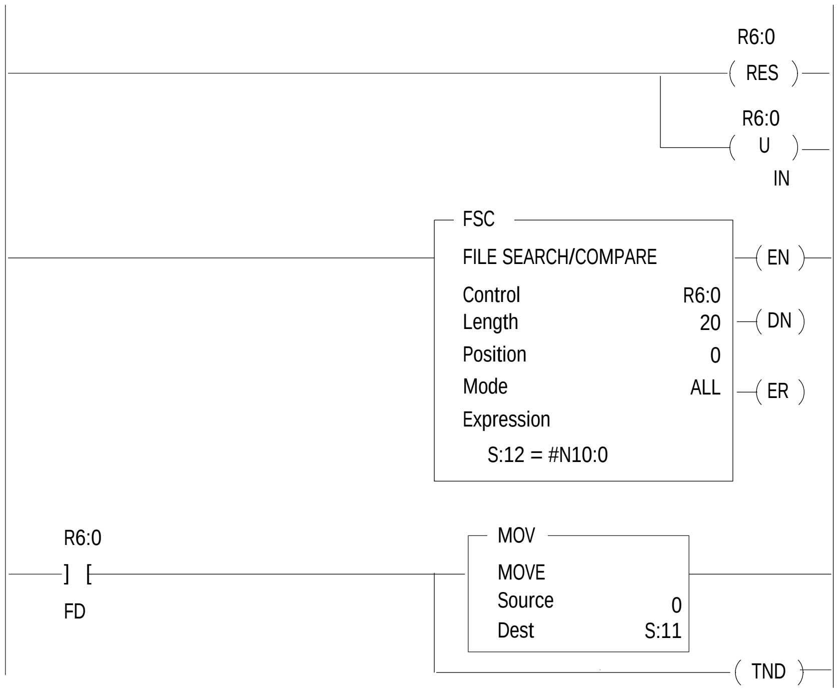
For example, if a major fault is encountered causing bit S:11/2 to be set, indicating a *programming error*, **do not** use a fault routine to clear the fault until you correct your program.

If you decide to clear the fault in the fault routine, follow these steps:

1. Place the ladder logic for clearing the fault at the beginning of the fault routine.
2. Identify the possible major faults.
3. Select only those your application will let you safely clear. These are your ***reference fault codes***.
4. From the fault routine, examine the major fault code that the processor stores in S:12.
5. Use an FSC instruction to compare the fault code to the reference file that contains "acceptable" fault codes (word-to-file comparison).

   If the processor finds a match, the FSC instruction sets the found (.FD) bit in the specified control structure.
6. Use a MOV instruction to clear the fault in S:11. In Figure 16.1, #N10:0 is the reference file.

Figure 16.1
Example of Comparing a Major Fault Code with a Reference



Remainder of fault routine follows

If the fault routine	Then the processor
clears S:11	returns to the program file and resumes program execution.
does not clear S:11	executes the rest of the fault routine and then faults

Important: If the fault routine clears the major fault, the processor completes the fault routine and returns to the next instruction in the program file that follows the one that contained the faulted instruction.

The remainder of the rung is executed and it appears that the fault never occurred. The fault routine execution continues until you correct the cause of the fault.

Follow these guidelines when creating fault routines:

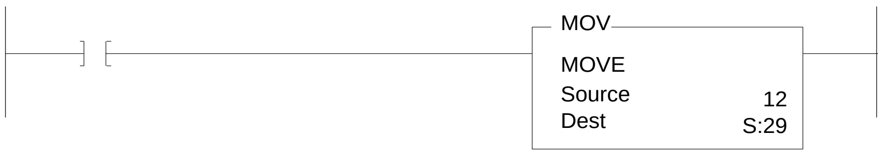
- Store initial conditions and reset other data to achieve an orderly start-up later.
- Monitor the shutdown of critical outputs. Use looping if needed to extend the single fault routine scan time up to the limit of the processor watchdog timer so your program can confirm that critical events took place.

Changing the Fault Routine from Ladder Logic

You can change the specified fault routine from ladder logic by copying a new fault routine file number into word 29 of the processor status file.

Figure 16.2 shows an example rung for changing the fault routine file number.

Figure 16.2
Example of Changing the Fault Routine File Number



ATTENTION: Do not corrupt the program-file number of the fault routine or use the same file for any other purpose. If the file number you specify results in a non-existent fault routine, the processor immediately enters fault mode after detecting a fault. Unexpected machine operation may result with damage to equipment and/or injury to personnel.

Using Ladder Logic to Recover from a Fault

If you have the appropriate fault routine and ladder logic to perform an orderly shutdown of the system, you may want to configure an I/O rack fault as a minor fault. You can program ladder logic in several ways to recover from an I/O rack fault.

Table 16.A
Ways to Recover from a Rack Fault

Method:	Description:
User-generated major fault	The program jumps to a fault routine when a remote I/O rack fault occurs. In other words, if the status bits indicate a fault, you program the processor to act as if a major fault occurred (i.e., jump to the fault routine). You then program your fault routine to stop the process or perform an orderly shutdown of your system. When the processor executes the end-of-file instruction for the fault routine, a user-generated major fault is declared.
Reset input image table	You monitor the status bits and, if a fault is detected, you program the processor to act as if a minor fault occurred. After the status bits indicate a fault, use the I/O status screen in your programming software to inhibit the remote rack that faulted. You then use ladder logic to set or reset critical input image table bits according to the output requirements in the non-faulted rack. If you reset input image table bits, during the next I/O update, the input bits are set again to their last valid state. To prevent this from occurring, your program should set the inhibit bits for the faulted rack. The global inhibit bits control the input images on a rack by rack basis; the partial rack inhibit bits control the input images on a 1/4-rack basis. For more information on these global status bits, see the documentation for your programming software. This method requires an extensive and careful review of your system for recovery operations. For more information on inhibiting I/O racks, see the documentation for your programming software.
Fault zone programming method	Using fault zone programming method, you disable sections of your program with MCR zones. Using the status bits, you monitor your racks; when a fault is detected, you control the program through the rungs in your MCR zone. With this method, outputs within the MCR zone must be non-retentive to be de-energized when a rack fault is detected. For more information on MCR zone programming, see the documentation for your programming software.

Block-Transfers in Fault Routines

If the processor runs a fault routine that contains block-transfer instructions, the processor performs these block-transfers immediately upon completing any block-transfers currently in the active buffer, ahead of block-transfer requests waiting in the queue.

The block-transfers in a fault routine or an STI should be between the processor and local I/O only.

ATTENTION: If you program block-transfer instructions to remote chassis within a fault routine or STI, be aware that the MCP resumes processing while waiting for the block-transfer to complete unless you use a UIE/UID instruction pair.

Testing a Fault Routine

To test a fault routine, use a JSR instruction to jump to the fault routine. Send a fault code as the first input parameter of the JSR instruction. The processor stores the fault code in S:12 and sets the corresponding bit in S:11.

You can detect and set your own faults by using fault codes 0-9 or by using the processor-defined fault codes 10-87.

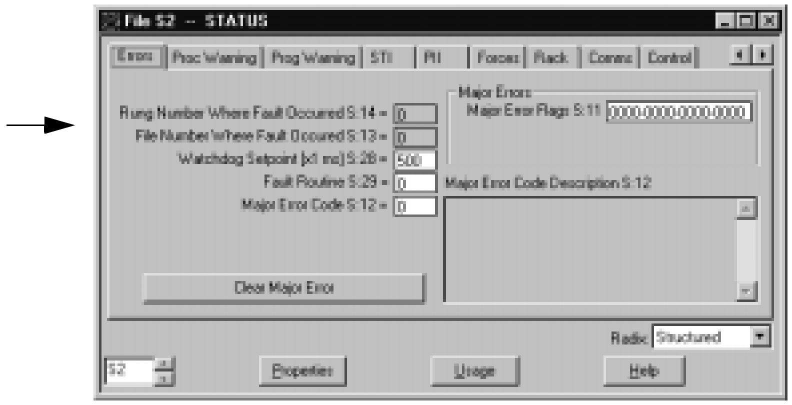
Monitoring Faults

Monitor processor faults using the processor status screen in your programming software.

You can monitor:	Description:	See page:
Minor and major faults	Processor faults are categorized into major and minor faults. The processor displays a unique bit for each fault and displays text that describes the fault.	16-11
Fault codes	Fault codes provide information about processor-defined errors.	21-5
Global status bits	Global status bits are set if a fault occurs in any one of the logical racks.	16-11
Multiple chassis status bits	Multiple chassis status bits are used to monitor the racks in your I/O system.	16-11

Monitoring Major/Minor Faults and Fault Codes

When a fault occurs, the processor status screen in your programming software displays program file and rung number indicators that point to where the fault occurred.



Rung Number Where Fault Occured S:14 = 0
File Number where Fault Occured S:13 = 0
Watchdog Setpoint [x1 ms] S:28 = 500
Fault Routine S:29 = 0
Major Error Code S:12 = 0

Major Errors
Major Error Flags S:11 0000-0000-0000-0000

Major Error Code Description S:12

Radix: Structured

S2

Interpreting Major Faults

Displaying a description of the major faults:	**Clear the faults by:**
• The status text that appears corresponds to the most significant fault when the cursor is not on the major fault status word. • If the cursor is on a major fault word bit and that bit is set, the status text that appears corresponds to the bit that the cursor is on. • If no bits are set, the message area is blank.	• Using the clear major-fault-function on the processor status screen of your programming software. When you clear major faults, the fault code, program file, and rung number fields are also cleared. • Resetting individual bits. If you have more than one major fault and you reset a bit, the status text displays the next major fault message.

For a description of the major faults (S:11), see chapter 21.

Interpreting Minor Faults

Displaying a Description of the Minor Faults:	**Clear the Faults by:**
• The status text that appears corresponds to the most significant fault when the cursor is not on the minor fault status words. • If the cursor is on a minor fault word bit and that bit is set, the status text that appears corresponds to the bit that the cursor is on. • If no bits are set the message area is blank.	• Using the clear minor-fault-function on the processor status screen of your programming software. • Resetting individual bits. If you have more than one minor fault and you reset a bit, the status text displays the next minor fault message.

For a description of the minor faults in word 1 (S:10) and word 2 (S:17), see chapter 21.

Monitoring Status Bits

Two types of status bits display information about your system: global status bits and multiple chassis status bits.

Each bit represents an entire rack, no matter how many chassis make up a rack. (Remember that you can have a maximum of four chassis configured as quarter racks to make up one I/O rack.) These bits are stored in the lower eight bits of words S:7, S:32, and S:34.

The **global status bits** are set if a fault occurs in any one of the racks. See the table below to determine the number of bits.

Processor	Possible I/O Rack Bits
PLC-5/11, -5/20, 5/20E	4
PLC-5/30	8
PLC-5/40, -5/40L, 5/40E	16
PLC-5/60, -5/60L, -5/80, 5/80E	24

The **multiple chassis status bits** are used to monitor the racks in your I/O system. This information is stored in the I/O status file (S:16, low byte) that you specify using the processor configuration screen in your programming software. The software automatically creates an integer data file to store two words of status bits for every rack configured in your system.

MORE

For more information on global status bits and multiple chassis status bits, see the documentation for your programming software.

Chapter 17

Using Main Control Programs

Using This Chapter

Selecting Main Control Programs

You can have as many as 16 control programs active at one time. Each of these programs is called a “main control program” (MCP). You can define one MCP for each particular machine or function of your process. This lets you separate sequential function charts (SFCs), ladder logic, and structured text to better modularize your process and make troubleshooting easier.

Consider using this technique:	If you are:
SFC	defining the order of events in the process
Ladder Logic	• more familiar with ladder logic than with programming languages such as BASIC • performing diagnostics
Structured Text	• more familiar with programming languages such as BASIC than with ladder logic • using complex mathematical algorithms • using program constructs that repeat or “loop” • creating custom data-table monitoring screens

A main control program can be an SFC numbered 1-999; it can also be a ladder or structured-text program numbered 2-999 in any program file. You can use any mix of SFC, ladder, and structured-text programs to define 16 main control programs. One data table is used by all MCPs (i.e., you do not have a separate data table for each MCP).

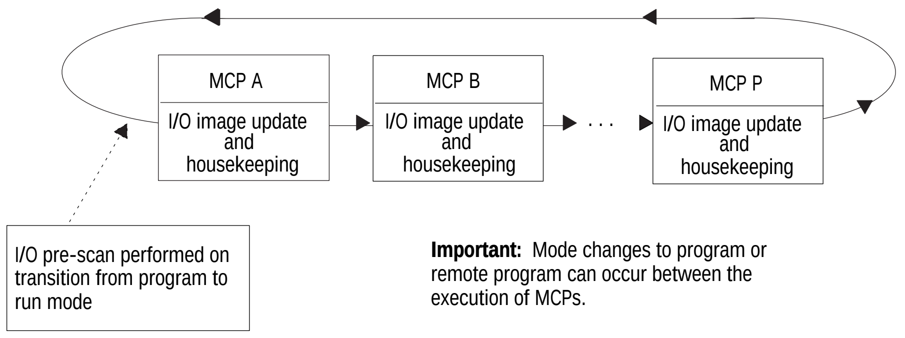
Understanding How the Processor Interprets MCPs

The MCPs are scheduled to execute in the order in which you specify on the Processor Configuration screen. You can configure:

- an I/O image update and housekeeping after each MCP is completed (default parameter), **or**
- the processor to skip the I/O scan and run the next MCP

After the last MCP is completed, all MCPs are then repeated in the same order. Note that the watchdog setpoint covers one scan of all MCPs. Figure 17.1 shows how the processor interprets MCPs when an I/O image update is specified to occur after each MCP is completed.

Figure 17.1
MCP Execution with I/O Update after Each MCP


MCP A
I/O image update and housekeeping
MCP B
I/O image update and housekeeping
. . .
MCP P
I/O image update and housekeeping
I/O pre-scan performed on transition from program to run mode
Important: Mode changes to program or remote program can occur between the execution of MCPs.

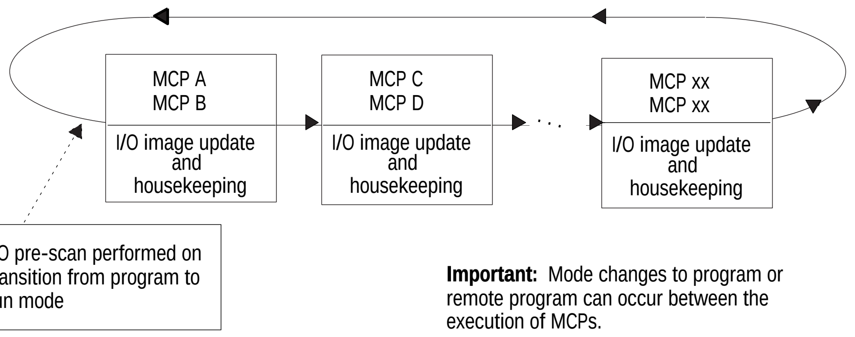
By disabling I/O scans between MCPs, you can gain 2-3 ms of program-scan time per disabled I/O scan. The processor updates your I/O when it reaches the next I/O scan command, which can be:

- an enabled I/O scan between MCPs, **and/or**
- the end of a pass through the entire MCP list.

The processor always performs an I/O scan after a pass through the MCP list.

Figure 17.2 shows how the processor skips I/O scans and moves to the next MCP.

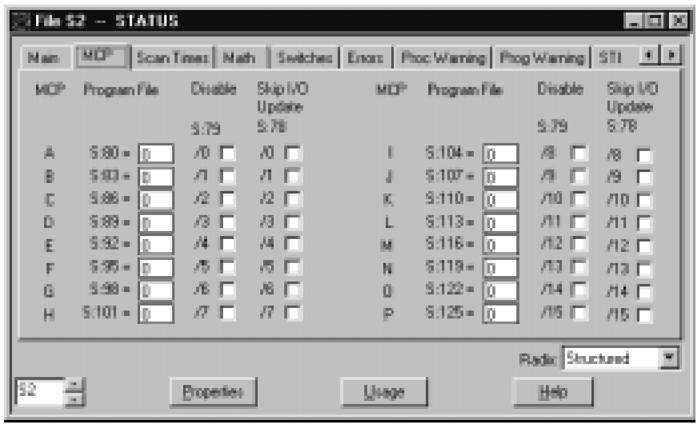
Figure 17.2
MCP Execution with I/O Update Disabled between MCPs


MCP A
MCP B
I/O image update and housekeeping
MCP C
MCP D
I/O image update and housekeeping
. . .
MCP xx
MCP xx
I/O image update and housekeeping
I/O pre-scan performed on transition from program to run mode
Important: Mode changes to program or remote program can occur between the execution of MCPs.

If the MCP is a:	The following occurs:
Ladder-logic program	**1.** All rungs are executed—from the first rung to the last, with all timers, counters, jumps, and subroutines active. **2.** After the END instruction in the ladder program, the processor initiates an I/O update—reading local inputs, writing local outputs, reading remote buffers, and writing remote outputs to the buffer. **3.** The processor starts the next MCP.
Structured-text program	**1.** Code is executed normally. **2.** After the last instruction in the program, the processor initiates an I/O update. **3.** The processor starts the next MCP.
SFC	**1.** Only the active steps are scanned, and transitions from those active steps are examined. **2.** After one complete pass through the active steps, the processor initiates an I/O update. **3.** The processor starts the next MCP.

Configuring MCPs

You configure MCPs on the processor configuration screen in your programming software.



MCP	Program File	Disable S:79	Skip I/O Update S:78	MCP	Program File	Disable S:79	Skip I/O Update S:78
A	S:80 = 0	/0 ☐	/0 ☐	I	S:104 = 0	/8 ☐	/8 ☐
B	S:83 = 0	/1 ☐	/1 ☐	J	S:107 = 0	/9 ☐	/9 ☐
C	S:86 = 0	/2 ☐	/2 ☐	K	S:110 = 0	/10 ☐	/10 ☐
D	S:89 = 0	/3 ☐	/3 ☐	L	S:113 = 0	/11 ☐	/11 ☐
E	S:92 = 0	/4 ☐	/4 ☐	M	S:116 = 0	/12 ☐	/12 ☐
F	S:95 = 0	/5 ☐	/5 ☐	N	S:119 = 0	/13 ☐	/13 ☐
G	S:98 = 0	/6 ☐	/6 ☐	O	S:122 = 0	/14 ☐	/14 ☐
H	S:101 = 0	/7 ☐	/7 ☐	P	S:125 = 0	/15 ☐	/15 ☐

Radix: Structured

S2

In this field:	Do the following:	Status File:
Program file	Specify the program file numbers for MCPs A-P and the order in which the MCPs should be run. This configuration is read before the MCP is executed; if you make a change to the configuration screen regarding an MCP, that change takes effect on the next execution of the MCP. You can change the MCP information on the Processor Configuration screen or through ladder logic. If you specify an MCP file that does not exist or is not a ladder-logic program, structured-text program, or SFC file, a major fault is logged in the status file. A minor fault is also logged if all MCP program files are set to zero. You can have the same program file number specified more than once as an MCP. For example, you may want a program to execute frequently and have a higher priority over other programs. If you do not want to use multiple main programs, program an SFC (program file 1), ladder-logic program (program file 2), or structured-text program (program file 2) and the processor will execute your main program. You do not need to make any entries on the Processor Configuration screen (the processor automatically enters the first configured program file number in the first MCP entry).	S:80-S:127
Disable	By setting or resetting the bit in these fields, you tell the processor to skip over the MCP until the bit is reset. If an MCP program file is inhibited, the processor skips the file and goes to the next program file. **ATTENTION:** If you disable an MCP, outputs remain in the state that they were in during the last scan (i.e., all actions remain active). Make sure that you consider any outputs that might be controlled within that MCP before disabling it. Otherwise, injury to personnel or damage to equipment may result. Disable an MCP if you temporarily want to hold a machine state, regardless of transitions (for example, in machine fault conditions). Disabling an MCP also can help improve scan time; if you know you don't need to run one of your MCPs every scan, you can disable it until you need it. To set and reset the bits for Main Control Programs A-P, cursor to the appropriate field and type 1 to disable (skip) this MCP or 0 to enable (scan) this MCP. If the disable bit is set for all the MCP program files (which indicates that all control programs are to be skipped), a minor fault is logged in the processor status file.	S:79
Skip I/O update	A **1** in this field tells the processor to skip the I/O scan after this MCP. The default **0** tells the processor to perform the I/O scan after the corresponding MCP. To specify the I/O bit, cursor to the appropriate field and enter 0 or 1.	S:78

Important: If you plan to use SFC subcharts, make sure you define something for MCP A - even an empty ladder file is sufficient. If a MCP is undefined, the processor faults on the second SFC scan with major fault code 71 SFC subchart is already executing.

Monitoring MCPs

The program scan times for each MCP are stored in the processor status file, including the previous and maximum scan time. The status file also stores the cumulative scan time, S:8 (the scan time for one complete pass through all MCPs) and the maximum cumulative scan time, S:9.

Using Selectable Timed Interrupts

Using This Chapter

Using a Selectable Timed Interrupt

A selectable timed interrupt (STI) tells the processor to periodically interrupt program execution (due to elapsed time) to run an STI program once to completion. Then, the processor resumes executing the original program file from where it was interrupted. For example, you might want to use an STI to periodically update analog values for a process control loop or send machine data to a host at scheduled intervals.

Design Tip

Writing STI Ladder Logic

Follow these guidelines when you write ladder logic for an STI.

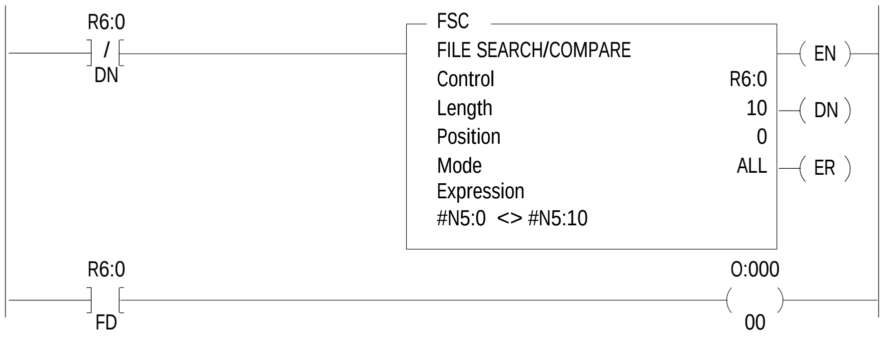
- Store the STI program in its own ladder file.
- Make sure that the interrupt interval you specify (in status word S:30) is longer than the execution time of the STI program. If it is not, an STI overlap can occur and the processor sets a minor fault bit at S:10/2.
- Note that the processor's watchdog timer continues to run while the processor runs an STI program.

Important: If the interrupt occurs during the execution of an instruction, the processor stops executing the instruction, scans the interrupt file once to completion, and then resumes executing the instruction. In effect, STI execution is transparent to program execution time unless you specify too short an interval. An interval that is too short can cause the watchdog timer to time out or cause excessively long program scans.

Online editing affects the performance of an STI routine. The STI cannot interrupt the processor while it is managing its memory due to the online edits being made. The STI input must be on for an amount of time slightly greater than the actual time required to complete the online edits. If not, the STI does not execute.

STI Application Example

Periodically check the status of PLC-5 family processors on the DH+ communication link. Compare the status of each processor with a file of reference data (see rungs below). Set a bit if a mismatch is found. Perform this comparison once every 800 ms. Assume that another active step retrieves status data from the PLC-5 processors with a MSG instruction and loads it into a temporary source file (N5:10).



Block-Transfers in Selectable Timed Interrupts (STIs)

If the processor runs an STI that contains block-transfer instructions, the processor performs these block-transfers immediately on completing any block-transfers currently in the active buffer, ahead of block-transfer requests waiting in the queue.

You can program “immediate” block-transfers to a local I/O chassis using the STI program (i.e., the STI is invoked and the block-transfer occurs immediately). The processor executes the block-transfer immediately, completes the remaining rungs in the STI, then resumes execution of the ladder program.

Design Tip

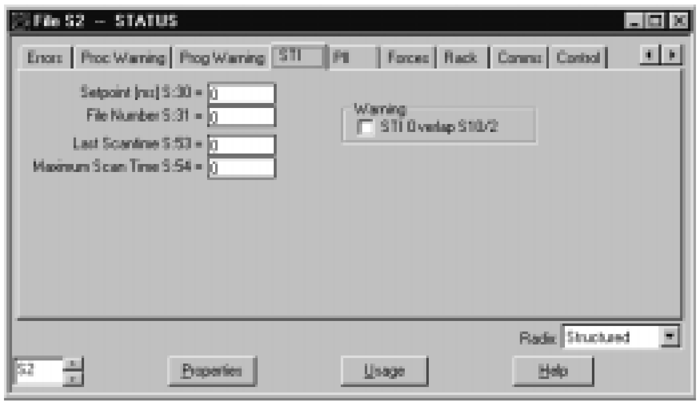
Set the .TO bit on any block-transfer instruction destined for the same slot as the block-transfer in the STI. These block-transfers only try to execute once so as not to keep the STI from completing.

The block-transfers in a fault routine or an STI should only be between the processor and local I/O. Remote block-transfer instructions in an STI cause the processor to resume executing the user program while waiting for the block-transfer to complete. If you want the STI to run to completion before returning to your main logic program, include a UID and UIE instruction pair in your STI program file. Place the block-transfer instruction inside of a UID/UIE pair.

ATTENTION: When the processor runs a fault routine or STI with a block-transfer instruction to a remote chassis, the MCP resumes processing while waiting for the block-transfer to complete unless a UIE/UID instruction pair is used.

Defining a Selectable Timed Interrupt

To configure a selectable timed interrupt, you must specify:



Setpoint [ms] S:30 = 0
File Number S:31 = 0
Last Scantime S:53 = 0
Maximum Scan Time S:54 = 0

Warning
☐ STI Overlap S10/2

Radix: Structured

S2

In this field:	Do the following:	Status File:
Setpoint	Enter the time interval between interrupts (1 to 32767 ms). If you are not using or want to disable an STI, enter zero. **Important:** Remember to specify an interrupt time longer than the STI file execution time. If you do not, the processor sets a minor fault (S:10, bit 2).	S:30
File number	Enter the number of the program file that contains the STI program. If you are not using an STI, enter zero.	S:31

For example, you could enter a `7` in S:31 and a `15` in S:30. This causes the processor to execute ladder file 7 every 15 ms.

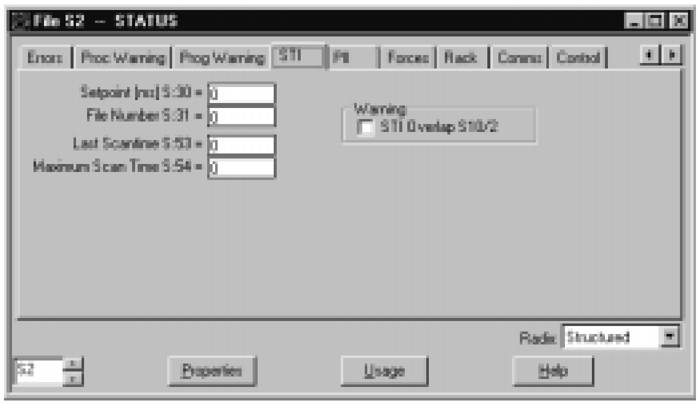
You can use only one STI at any one time. However, you can enable or disable the interrupt, change to a different interrupt file, or change the time between interrupts. Use ladder logic to change the values in word S:30 and word S:31 as needed.

ATTENTION: STI programs lengthen the program scan by an amount equal to the interrupt delay multiplied by the number of times the interrupt occurs during a program scan.

Important: If you disable the STI through ladder logic (write a 0 to S:30), it could take the processor up to 100 ms to re-enable the STI. If you disable the STI (write a 0 to S:31), the processor uses the value in S:30 to determine how often to check for a non-zero value in S:31.

Monitoring Selectable Timed Interrupts

Use the processor status screen in your programming software to monitor STIs.



Setpoint [ms] S:30 = 0
File Number S:31 = 0
Last Scantime S:53 = 0
Maximum Scan Time S:54 = 0

Warning
☐ STI Overlap S10/2

Radix: Structured

S2

In this field:	Do the following:	Status File:
Last scan time	This field displays the time it took for the current or last scan of the STI.	S:53
Maximum scan time	This field displays the longest time that was ever displayed in the Last scan field for the specific STI.	S:54
STI Overlap	This box is checked if an STI overlap occurs. This condition results if the interrupt interval you specify for the setpoint is shorter than the execution time of the STI program.	S:10/2

Using Processor Input Interrupts

Using This Chapter

Using a Processor Input Interrupt

A processor input interrupt (PII) specifies when an event-driven input causes the processor to interrupt program execution and run a PII program file once to completion. Afterwards, the processor resumes executing the program file from where it was interrupted. Use PIIs only for inputs in the processor-resident chassis.

You can use a processor input interrupt (PII) as an event-driven interrupt or in high-speed processing applications. For example, you may need to count inputs quickly to track production, such as in a canning line. Or, use a PII if your application calls for an immediate input update when a part is seen on a conveyor and you need to do an immediate output update to perform the next action. For example, when a part moving down a conveyor line is detected, you may need to stop it so the next piece can be added.

Your PII program can contain immediate update instructions to complete high-speed control functions. As your ladder program is running and the input condition occurs, the processor interrupts program execution and runs the PII program file. Then, the processor resumes executing the program file from the point where it was interrupted.

Design Tip

Writing PII Ladder Logic

Follow these rules when you write ladder logic for a PII.

- Store the PII program in a ladder file.
- Make sure the input condition (to cause the interrupt) doesn't occur faster than the execution time of the PII program. If a second identical input condition occurs before the interrupt program has finished executing for the first input condition, a PII overlap occurs and the processor sets a minor fault bit at S:10/12.

  The timing for a PII is as follows:

  – 1 ms to switch to the PII task

  – PII ladder logic execution time

  – 1 ms to return to executing the control program

  Since you need to allow at least 1 ms to run your PII logic, define a PII time of at least 3 ms to help prevent PII overlaps.
- The processor's watchdog timer continues to run while running a PII program.
- A PII can detect an event within 100 ms; however, you must allow at least 3 ms between successive PII events.

PII Application Examples

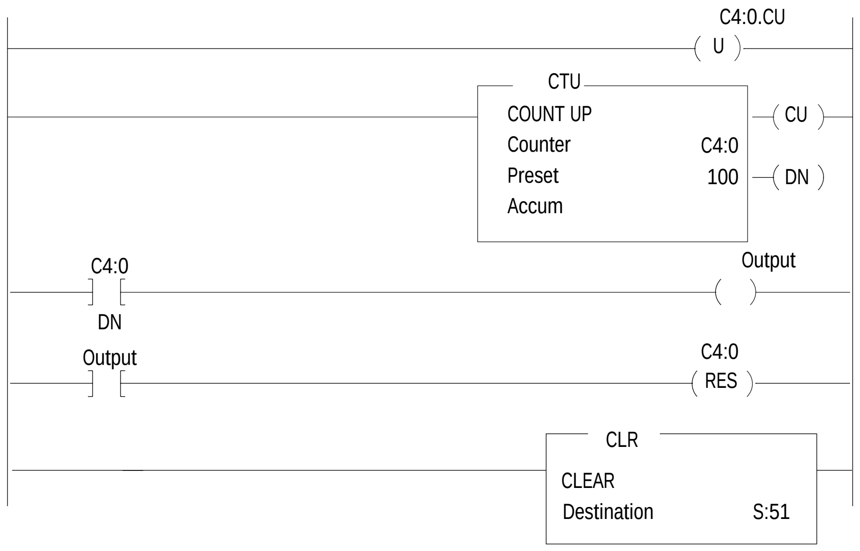
Two ways that you can use a PII program:

Mode:	Description:
Counter	Using counter mode, you make use of the processor's internal counter. You configure the PII with a preset value so that the hardware counts an input condition and then runs the PII when the preset equals the accumulated value. The PII ladder logic only needs to contain the output that you want to occur.
Bit transition	Using bit-transition mode, you configure the PII to occur every time the input condition is true. For example, you want to count tablets as they leave the production line at a rate of 100 tablets per second. The machinery packs 100 tablets per package. Assume an optical switch detects each tablet.

The PII program (Figure 19.1) must:

- count 100 tablets per group
- set an output at the 100th tablet
- reset the counter for the next group

Figure 19.1
Example PII Program



The output image bit remains set until the next count.

Block-Transfers in Processor Input Interrupts (PIIs)

If the processor runs a PII that contains block-transfer instructions, the processor performs these block-transfers immediately on completing any block-transfers currently in the active buffer, ahead of block-transfer requests waiting in the queue.

You can program "immediate" block-transfers to a local I/O chassis using the PII program (i.e., the PII is invoked and the block-transfer occurs immediately). The processor executes the block-transfer immediately, completes the remaining rungs in the PII, then resumes execution of the ladder program.

You can use the PII for a block-transfer to remote I/O. Remote block-transfer instructions in a PII cause the processor to resume executing user programs, including STIs, while waiting for the block-transfer to complete. If you want the PII to run to completion before returning to your main logic program, include a UID and UIE instruction pair in your PII program file. Place the block-transfer instruction inside of a UID/UIE pair.

Important: If the interrupt occurs during the execution of an instruction, the processor stops executing the instruction, scans the interrupt file once to completion, then resumes executing the instruction. In effect, execution of a PII is transparent to program execution time unless you program too many too often. Too many PIIs often can cause the watchdog timer to time out or cause excessively long program scans.

PII configuration changes are not put into effect until the processor goes from program to run or test mode.

Design Tip

Design Considerations

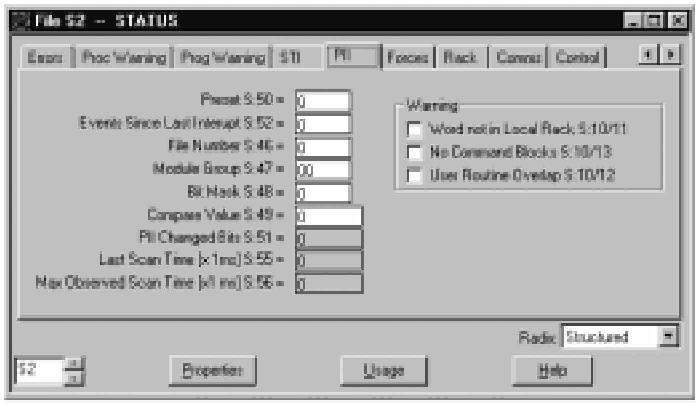
Consider the following guidelines when planning PIIs.

- Do not use 2-slot addressing when using PIIs.
- Do not use 1771-IG or -IGD, 8- and 16-point TTL modules for the PII. Use the 1771-IQ16 input module instead. Since the module's input delay filter is selectable, you can set the delay to 0 or about 200 ms.
- Avoid using a block-transfer module in the processor-resident rack with a PII configured because you could miss an input pulse while a block-transfer of data is in progress. However, if you need to use block-transfers, make sure that a PII input pulse is at least 400 ms, which causes the block-transfer not to affect the PII.
- Online editing affects the performance of a PII routine. A PII cannot interrupt the processor while it is managing its memory due to the online edits being made. The PII input must be on for an amount of time slightly greater than the actual time required to complete the online edits. If not, the PII does not execute.
- Clear S:51 in one of two ways:
  - using a CLR instruction (see Figure 19.1)
  - placing a MOV (move) instruction on the last rung in the PII file. Move a 0 into S:51 to reset the PII bits before finishing the PII file.

Important: If S:51 is not cleared, a PII overlap bit is set on that status page, causing a minor fault.

Defining a Processor Input Interrupt

To define a PII, use the processor configuration screen in your programming software.



Preset S:50 = 0
Events Since Last Interrupt S:52 = 0
File Number S:46 = 0
Module Group S:47 = 00
Bit Mask S:48 = 0
Compare Value S:49 = 0
PII Changed Bits S:51 = 0
Last Scan Time [x 1ms] S:55 = 0
Max Observed Scan Time [x1 ms] S:56 = 0

Warning
☐ Word not in Local Rack S:10/11
☐ No Command Blocks S:10/13
☐ User Routine Overlap S:10/12

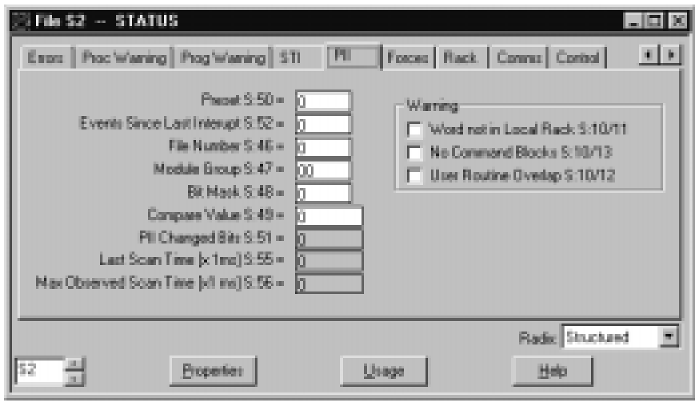
Radix: Structured
S2

In this PII configuration field:	Do the following:	Status File Address:
Preset	Enter a preset value to determine how many conditions you want to occur before the interrupt. Valid range is 0 - 32,767. If you want the interrupt to occur every time, enter a 0 or 1.	S:50
File number	Enter the number of the program file that contains the PII program. This is the only PII parameter that you can change while the processor is in RUN mode.	S:46
Module group	Enter the assigned rack number and I/O group number of the input to monitor (e.g., 21 for rack 2, group 1). Do not enter the address. (Only for inputs in the processor-resident chassis). If the input word number specified is not in the local rack or if there is not an input module in the slot addressed, a minor fault bit (S:10/11) is set at mode transition.	S:47
Bit mask	Each module group (specified in S:47) has a control bit that is used to monitor the input bit. • To monitor the bit, enter a 1. • To ignore the bit, enter a 0.	S:48
Compare value	Each module group (specified in S:47) has a bit that is used when controlling a PII through bit transition. • For a false to true transition to count (bit trigger), enter a 1. • For a true to false transition to count (event trigger), enter a 0.	S:49

Important: If you change the PII configuration while in run mode, you must toggle the mode to program, then back to run mode for the change to take effect.

Monitoring Processor Input Interrupts

Use the processor status screen in your programming software to monitor PIIs.



Preset S:50 = 0
Events Since Last Interrupt S:52 = 0
File Number S:46 = 0
Module Group S:47 = 00
Bit Mask S:48 = 0
Compare Value S:49 = 0
PII Changed Bits S:51 = 0
Last Scan Time [x 1ms] S:55 = 0
Max Observed Scan Time [x1 ms] S:56 = 0

Warning
☐ Word not in Local Rack S:10/11
☐ No Command Blocks S:10/13
☐ User Routine Overlap S:10/12

Radix: Structured
S2

This PII field:	Stores	Status File Address:
Events since last interrupt	Displays the number of PII events (the input conditions that caused the interrupt) since the last interrupt.	S:52
PII changed bits	Displays the bit transitions that caused the interrupt. You can use this information to condition other rungs in your ladder program. If one of these bits is already set (i.e., a previous interrupt set the bit), the processor sets a minor fault (S:10/2) to indicate a possible PII overlap. If you want to monitor this overlap, make sure the last rung in your PII program clears this return mask in the status file.	S:51
Last scan time	Displays the current or last scan time through the PII.	S:55
Max observed scan time	Displays the maximum value that was displayed in the last scan field.	S:56
Word not in local rack	This box is checked if the input word number specified is not on the local rack or if there is not an input module in the slot addressed.	S:10/11
No command blocks	This box is checked if no command blocks exist to get the PII. You can use the processor's internal counter or bit transition to execute the PII.	S:10/13
User routine overlap	This box is checked if a set condition exists in the PII return mask or changed bits (possibly set by a previous interrupt) before completing the currently executing PII routine. PII changed bits are retentive. It may be necessary to place a MOV instruction on the last rung in the PII file. Move 0 in S:51 to reset the PII bits before finishing the PII file. If this is not done, a PII overlap bit will be set on that status page, causing this minor fault.	S:10/12

Use S:51/0-15 within the PII file because these bits are:

- mapped from the actual input module being used for the PII
- retentive

For the PII routine to execute properly, do not use the addresses of the input module's bits within the PII routine.

System Specifications

Processor Specifications

Backplane Current	PLC-5/11, -5/20, -5/26, -5/30 . . . . 2.3A PLC-5/40, -5/46, -5/40L, -5/60, -5/60L -5/80, -5/86 . 3.3A PLC-5/20E, -5/40E, -5/80E. 3.6A
Heat Dissipation	PLC-5/11, -5/20, -5/26, -5/30 . . . . 41.30 BTU/hr PLC-5/40, -5/46, -5/40L, -5/60, -5/60L -5/80, -5/86 . 59.04 BTU/hr PLC-5/20E, -5/40E, -5/80E. . . . . 61.43 BTU/hr
Environmental Conditions	Operating Temperature. . . . . . . 0 to 60° C (32-140° F) Storage Temperature . . . . -40 to 85° C (-40 to 185° F) Relative Humidity . . . . 5 to 95% (without condensation)
Shock	Operating . . . . 30 g peak acceleration for 11±1 ms duration Non-operating . . . . 50 g peak acceleration for 11±1 ms duration
Vibration **(operating and non-operating)**	1 g @ 10 to 500 Hz 0.012 inches peak-to-peak displacement
Time-of-Day Clock/Calendar	Maximum Variations at 60° C . ± 5 min per month Typical Variations at 20° C . . . ± 20 s per month Timing Accuracy. . . . . 1 program scan
Battery	1770-XYC
Memory Modules	1785-ME16 1785-ME64 1785-ME32 1785-M100
Typical Discrete I/O Scan	• 0.5 ms / extended-local I/O • 10 ms / remote I/O adapter communication at 57.6 kbps • 7 ms / remote I/O adapter communication at 115.2 kbps • 3 ms / remote I/O adapter communication at 230.4 kbps
I/O Modules	Bulletin 1771 I/O including 8-, 16-, 32-pt, and intelligent modules

Hardware Addressing	2-slot • Any mix of 8-pt modules • 16-pt modules must be I/O pairs • No 32-pt modules 1-slot • Any mix of 8- or 16-pt modules • 32-pt modules must be I/O pairs 1/2-slot — Any mix of 8-,16-, or 32-pt modules
Ethernet Communications	512 maximum unsolicited definitions
Communication	• DH+ • DH using 1785-KA • Serial • Ethernet (TCP/IP protocol, 15-pin AUI transceiver port) • remote I/O • extended-local I/O (PLC-5/40L and -5/60L processors only)
Location	1771-A1B, -A2B, A3B, -A3B1, -A4B, chassis, left-most slot
Keying	• Between 40 and 42 • Between 54 and 56
Weight	PLC-5/20, -5/26 1.21 kg (2.7 lbs) PLC-5/30 1.20 kg (2.6 lbs) PLC-5/40, -5/46, -5.40L 1.42 kg (3.1 lbs) PLC-5/60, -5/60L 1.42 kg (3.1 lbs) PLC-5/80, -5/86 1.42 kg (3.1 lbs) PLC-5/20E 1.43 kg (3.2 lbs) PLC-5/40E 1.39 kg (3.1 lbs) PLC-5/80E 1.38 kg (3.0 lbs)
Agency Certification (when product or packaging is marked)	• CSA Class I, Division 2, Groups A, B, C, D • UL listed • CE marked for all applicable directives

Processor Specifications (continued)

Processor/ Cat. No.	Maximum User Memory Words	Total I/O Maximum (any mix)	Types of Communication Ports	Maximum Number of I/O Racks (rack addresses)	Maximum Number of I/O Chassis		
					Total	Ext Local	Remote
PLC-5/11 (1785-L11B)	8 K	• 512 (any mix) or • 384 in + 384 out (complementary)	• 1 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible	4 (0-3)	5	0	4 (must be rack 3)
PLC-5/20 (1785-L20B) PLC-5/26 (1785-L26B)	16K	• 512 (any mix) or • 512 in + 512 out (complementary)	• 1 DH+ (Fixed) • 1 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible	4 (0-3)	13	0	12
PLC-5/20E (1785-L20E)	16K	• 512 (any mix) or • 512 in + 512 out (complementary)	• 1 DH+ (Fixed) • 1 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible • 1 channel Ethernet only	4 (0-3)	13	0	12
PLC-5/30 (1785-L30B)	32 K	• 1024 (any mix) or • 1024 in and 1024 out (complementary)	• 2 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible	8 (0-7)	29	0	28
PLC-5/40 (1785-L40B) PLC-5/46 (1785-L46B)	48 K[1]	• 2048 (any mix) or • 2048 in + 2048 out (complementary)	• 4 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible	16 (0-17)	61	0	60
PLC-5/40E (1785-L40E)	48 K[1]	• 2048 (any mix) or • 2048 in + 2048 out (complementary)	• 2 DH+/Remote I/O (Adapter or Scanner) • 1 channel Ethernet only • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible	16 (0-17)	61	0	60
PLC-5/40L (1785-L40L)	48 K[1]	• 2048 (any mix) or • 2048 in + 2048 out (complementary)	• 2 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible • 1 Extended-Local I/O	16 (0-17)	61	16	60
PLC-5/60 (1785-L60B)	64 K[2]	• 3072 (any mix) or • 3072 in + 3072 out (complementary)	• 4 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible	24 (0-27)	93	0	92

Processor/ Cat. No.	Maximum User Memory Words	Total I/O Maximum (any mix)	Types of Communication Ports	Maximum Number of I/O Racks (rack addresses)	Maximum Number of I/O Chassis		
PLC-5/60L (1785-L60L)	64 K[2]	• 3072 (any mix) or • 3072 in + 3072 out (complementary)	• 2 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible • 1 Extended Local I/O	24 (0-27)	81	16	64
PLC-5/80 (1785-L80B) PLC-5/86 (1785-L86B)	100 K[3]	• 3072 (any mix) or • 3072 in + 3072 out (complementary)	• 4 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible	24 (0-27)	93	0	92
PLC-5/80E (1785-L80E)	100 K[3]	• 3072 (any mix) or • 3072 in + 3072 out (complementary)	• 2 DH+/Remote I/O (Adapter or Scanner) • 1 serial port, configurable for RS-232 and 423 and RS-422A compatible • 1 channel Ethernet only	24 (0-27)	65	0	64

[1] The PLC-5/40, -5/40E, -5/40L processors have a limit of 32K words per data table file.
[2] The PLC-5/60 and -5/60L processors have a limit of 56K words per program file and 32K words per data table file.
[3] The PLC-5/80, -5/80E processors have 64K words of total data table space with a limit of 56K words per program file and 32K words per data table file.

Battery Specifications (1770-XYC)

		Worst-case Battery Life Estimates		
Battery used in this processor:	**At this temperature:**	**Power off 100%:**	**Power off 50%:**	**Battery Duration after the LED lights** [1]
PLC-5/11, -5/20, and-5/20E	60°C	256 days	1.4 years	11.5 days
	25°C	2 years	4 years	47 days
PLC-5/30, -5/40, -5/40L, -5/60, -560L, -5/80, -5/40E, and -5/80E	60°C	84 days	150 days	5 days
	25°C	1 year	1.2 years	30 days

[1] The battery indicator (BATT) warns you when the battery is low. These durations are based on the battery supplying the only power to the processor (power to the chassis is off) once the LED first lights.

Memory Backup Devices

You can add an EEPROM to the PLC-5 processor to provide backup memory for your program in case the processor loses power. These memory cards are available:

Catalog Number:	For This Product:	Memory Size:
1785-ME16	Enhanced PLC-5 processors	16K words
1785-ME32	Enhanced PLC-5 processors	32K words
1785-ME64	Enhanced PLC-5 processors	64K words
1785-ME100	Enhanced PLC-5 processors	100K words

Use your programming software to save a program currently in the processor to the EEPROM card. If you restore a program from the EEPROM to processor memory and processor memory is bad, the restore changes the date and time in the processor status file to the date and time the EEPROM was saved. If you restore a program from the EEPROM to processor memory and processor memory is valid, the status file retains its current date and time.

EEPROM Compatibility

EEPROM compatibility is related to:

Area:	Description:
ControlNet PLC-5 processors	EEPROM memory cannot be loaded to a non-ControlNet PLC-5 processor if the EEPROM was saved on a ControlNet PLC-5 processor. EEPROM memory cannot be loaded to a ControlNet PLC-5 processor if the EEPROM was burned on a non-ControlNet PLC-5 processor.
PLC-5 catalog numbers	EEPROM memory can be loaded to a PLC-5 processor if its I/O memory size is greater than or equal to the I/O memory of the PLC-5 processor from which the EEPROM was saved. The I/O memory sizes are: PLC-5/11, -5/20 4 racks PLC-5/30 8 racks PLC-5/40 16 racks PLC-5/60, -5/80 24 racks EEPROM memory can be loaded to a PLC-5 processor if its user memory is greater than or equal to the user memory used on the PLC-5 processor from which the EEPROM was saved. The available user memory is: PLC-5/11 8,192 words PLC-5/20 16,384 words PLC-5/30 32,768 words PLC-5/40 65,536 words PLC-5/80 102,400 words
Firmware release compatibility	EEPROM memory saved on a series D, revision B PLC-5 processor cannot be loaded on a PLC-5 processor with an earlier firmware release. EEPROM memory saved on a series E, revision A PLC-5 processor cannot be loaded on a PLC-5 processor with an earlier firmware release. EEPROM memory saved on a series E, revision B PLC-5 processor cannot be loaded on a PLC-5 processor with an earlier firmware release.

Processor Status File

Processor status data is stored in data file 2.

Important: For more information about any of these topics, see the description in this manual or the documentation for your programming software.

S:0 - S:2

This word:	Stores:
S:0	Arithmetic flags • bit 0 = carry • bit 1 = overflow • bit 2 = zero • bit 3 = sign
S:1	Processor status and flags
S:1/00	RAM checksum is invalid at power-up
S:1/01	Processor in run mode
S:1/02	Processor in test mode
S:1/03	Processor in program mode
S:1/04	Processor uploading to memory module
S:1/05	Processor in download mode
S:1/06	Processor has test edits enabled
S:1/07	Mode select switch in REMOTE position
S:1/08	Forces enabled
S:1/09	Forces present
S:1/10	Processor successfully uploaded to memory module
S:1/11	Performing online programming
S:1/12	Not defined
S:1/13	User program checksum calculated
S:1/14	Last scan of ladder or SFC step
S:1/15	Processor running first program scan or the first scan of the next step in an SFC

This word:	Stores:
S:2	Switch setting information
S:2/00 through S:2/05	Channel 1A DH+ station number
S:2/06	Channel 1A DH+ baud rate 0 57.6 kbps 1 230.4 kbps
S:2/07 S:2/08	Not defined
S:2/09	Last state 0 outputs are turned off 1 outputs retain last state
S:2/11 S:2/12	I/O chassis addressing <u>bit 12</u> <u>bit 11</u> 0 0 illegal 1 0 1/2-slot 0 1 1-slot 1 1 2-slot
S:2/13 S:2/14	Memory module transfer <u>bit 14</u> <u>bit 13</u> 0 0 memory module transfers to processor memory if processor memory is not valid 0 1 memory module does not transfer to processor memory 1 1 memory module transfers to processor memory at powerup
S:2/15	Processor memory protection 0 enabled 1 disable

S:3-10

This word:	Stores:
S:3 to S:6	Active Node table for channel 1A Word Bits DH+ Station # 3 0-15 00-17 4 0-15 20-37 5 0-15 40-57 6 0-15 60-77
S:7	Global status bits: (See also S:27, S:32, S:33, S:34, and S:35) • S:7/0-7 rack fault bits for racks 0-7 • S:7/8-15 unused
S:8	Last program scan (in ms)
S:9	Maximum program scan (in ms)
S:10	Minor fault (word 1) See also S:17
S:10/00	Battery is low (replace in 1-2 days)
S:10/01	DH+ active node table has changed
S:10/02	STI delay too short, interrupt program overlap
S:10/03	memory module transferred at power-up
S:10/04	Edits prevent SFC continuing; data table size changed during program mode; reset automatically in run mode
S:10/05	Invalid I/O status file
S:10/06	reserved
S:10/07	No more command blocks exist to execute block-transfers
S:10/08	Not enough memory on the memory module to upload the program from the processor
S:10/09	No MCP is configured to run
S:10/10	MCP not allowed
S:10/11	PII word number not in local rack
S:10/12	PII overlap
S:10/13	no command blocks exist to get PII
S:10/14	Arithmetic overflow
S:10/15	SFC “lingering” action overlap - step was still active when step was reactivated

S:11

This word:	Stores:
S:11	major fault word
S:11/00	Corrupted program file (codes 10-19). See major fault codes (S:12).
S:11/01	Corrupted address in ladder program (codes 20-29). See major fault codes (S:12).
S:11/02	Programming error (codes 30-49). See major fault codes (S:12).
S:11/03	Processor detected an SFC fault (codes (71-79). See major fault codes (S:12).
S:11/04	Processor detected an error when assembling a ladder program file (code 70); duplicate LBLs found.
S:11/05	Start-up protection fault. The processor sets this major fault bit when powering up in Run mode if the user control bit S:26/1 is set.
S:11/06	Peripheral device fault
S:11/07	User-generated fault; processor jumped to fault routine (codes 0-9). See major fault codes (S:12).
S:11/08	Watchdog faulted
S:11/09	System configured wrong (codes 80-82, 84-88, 200-208). See major fault codes (S:12).
S:11/10	Recoverable hardware error
S:11/11	MCP does not exist or is not a ladder or SFC file
S:11/12	PII file does not exist or is not a ladder file
S:11/13	STI file does not exist or is not a ladder file
S:11/14	Fault routine does not exist or is not a ladder file
S:11/15	Faulted program file does not contain ladder logic

S:12

This word stores the following fault codes:

This fault code:	Indicates this fault:	And the fault is:
00-09	Reserved for user-defined fault codes. You can use user-defined fault codes to identify different types of faults or error conditions in your program by generating your own recoverable fault. To use these fault codes, choose an input condition that decides whether to jump to a fault routine file, then use the JSR instruction as the means to jump to the fault routine file. To use the JSR instruction, enter the fault code number 0-9 (an immediate value) as the first input parameter of the instruction. Any other input parameters are ignored (even if you have an SBR instruction at the beginning of your fault routine file. You cannot pass parameters to the fault routine file using JSR/SBR instructions). You do not have to use the user-defined fault codes to generate your own fault. If you program a JSR with no input parameters, the processor will write a zero to the Fault Code field. The purpose of using the user-defined fault codes is to allow you to distinguish among **different** types of faults or error codes based on the 0-9 fault code numbers. When the input condition is true, the processor copies the fault code number entered as the first input parameter of the JSR instruction into word 12 of the processor status file (S:12), which is the Fault Code field. The processor sets a Major Fault S:11/7 “User-Generated Fault.” The processor then faults unless you clear the Major Fault word (S:11) or the specific fault bit via ladder logic in the fault routine.	*Recoverable:* The fault routine can instruct the processor to clear the fault and then resume scanning the program. A fault routine executes when any of these faults occur.
10	Run-time data table check failed	*Recoverable:* The fault routine can instruct the processor to clear the fault and then resume scanning the program. A fault routine executes when any of these faults occur.
11	Bad user program checksum	
12	Bad integer operand type, restore new processor memory file	
13	Bad mixed mode operation type, restore new processor memory file	
14	Not enough operands for instruction, restore new processor memory file	
15	Too many operands for instructions, restore new processor memory file	
16	Corrupted instruction, probably due to restoring an incompatible processor memory file (bad opcode)	
17	Can’t find expression end; restore new processor memory file	
18	Missing end of edit zone; restore new processor memory file	
19	Download aborted	
20	You entered too large an element number in an indirect address	
21	You entered a negative element number in an indirect address	
22	You tried to access a non-existent program file	
23	You used a negative file number, you used a file number greater than the number of existing files, or you tried to indirectly address files 0, 1, or 2	
24	You tried to indirectly address a file of the wrong type	*Recoverable*
30	You tried to jump to one too many nested subroutine files	*Non-recoverable* The fault routine will be executed but cannot clear major fault bit 2.
31	You did not enter enough subroutine parameters	
32	You jumped to an invalid (non-ladder) file	
33	You entered a CAR routine file that is not 68000 code	

This fault code:	Indicates this fault:	And the fault is:
34	You entered a negative preset or accumulated value in a timer instruction	*Recoverable*
35	You entered a negative time variable in a PID instruction	
36	You entered an out-of-range setpoint in a PID instruction	
37	You addressed an invalid module in a block-transfer, immediate input, or immediate output instruction	
38	You entered a RET instruction from a non-subroutine file	*Non-recoverable* The fault routine will be executed but cannot clear major fault bit 2.
39	FOR instruction with missing NXT	
40	The control file is too small for the PID, BTR, BTW, or MSG instruction	*Recoverable*
41	NXT instruction with missing FOR	*Non-recoverable* The fault routine will be executed but cannot clear major fault bit 2.
42	You tried to jump to a non-existent label	
43	File is not an SFC	
44	Error using SFR. This error occurs if: • you tried to reset into a simultaneous path • you specified a step reference number that is not found or is not tied to a step (it is a transition) • the previous SFR to a different step is not complete	
45	Invalid channel number entered	*Recoverable*
46	Length operand of IDI or IDO instruction is greater than the maximum allowed	
47	SFC action overlap. An action was still active when the step became re-activated	*Non-recoverable*
48-69	Reserved	*Recoverable*
70	The processor detected duplicate labels	
71	The processor tried to start an SFC subchart that is already running	
72	The processor tried to stop an SFC subchart that isn’t running	
73	The processor tried to start more than the allowed number of subcharts	
74	SFC file error detected	
75	The SFC has too many active functions	
76	SFC step loops back to itself.	
77	The SFC references a step, transition, subchart, or SC file that is missing, empty or too small	
78	The processor cannot continue to run the SFC after power loss	
79	You tried to download an SFC to a processor that cannot run SFCs	
80	You have an I/O configuration error	*Recoverable*
81	You illegally set an I/O chassis backplane switch by setting both switch 4 and 5 on	
82	Illegal cartridge type for selected operation. This error also occurs if the processor doesn’t have a memory module, but the backplane switches are set for a memory module. Make sure the backplane switches are correct (set switch 6 ON and switch 7 OFF if the processor doesn’t have a memory module).	
83	User watchdog fault	
84	Error in user-configured adapter mode block-transfer	
85	Memory module bad	

This fault code:	Indicates this fault:	And the fault is:
86	Memory module is incompatible with host	*Recoverable*
87	Scanner rack list overlap	
88	Scanner channels are overloading the remote I/O buffer; too much data for the processor to process. If you encounter fault code 88, be sure you followed the design guidelines listed on page 4-9. Specifically, make sure you: • group together 1/4-racks and 1/2-racks of each logical rack. Do not intersperse these with other rack numbers • if using complementary I/O addressing, treat complementary rack addresses individually when grouping racks; primary rack numbers are separate from complement rack numbers	
90	Sidecar module extensive memory test failed. Call your Allen-Bradley representative for service	
91	Sidecar module undefined message type	
92	Sidecar module requesting undefined pool	
93	Sidecar module illegal maximum pool size	
94	Sidecar module illegal ASCII message	
95	Sidecar module reported fault, which may be the result of a bad sidecar program or of a hardware failure	
96	Sidecar module not physically connected to the PLC-5 processor	
97	Sidecar module requested a pool size that is too small for PC^3 command (occurs at power-up)	
98	Sidecar module first/last 16 bytes RAM test failed	
99	Sidecar module-to-processor data transfer faulted	
100	Processor-to-sidecar module transfer failed	
101	Sidecar module end of scan transfer failed	
102	The file number specified for raw data transfer through the sidecar module is an illegal value	
103	The element number specified for raw data transfer through the sidecar module is an illegal value	
104	The size of the transfer requested through the sidecar module is an illegal size	
105	The offset into the raw transfer segment of the sidecar module is an illegal value	
106	Sidecar module transfer protection violation; for PLC-5/26, -5/46, and -5/86 processors only	
200	ControlNet scheduled output data missed. The processor is unable to transmit the scheduled data it is configured to transmit.	*Recoverable* Check your network for missing terminators or other sources of electrical noise (see the Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1)
201	ControlNet input data missed. The processor is unable to process incoming data from the network	*Recoverable* Check your network for missing terminators or other sources of electrical noise (see the Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1).
202	ControlNet diagnostic data missed.	*Recoverable* Contact your local Allen-Bradley representative if you get this message.

This fault code:	**Indicates this fault:**	**And the fault is:**
203	ControlNet schedule transmit data overflow.	*Recoverable* Contact your local Allen-Bradley representative if you get this message.
204	Too many output connections per NUI.	*Recoverable* Make scheduled outputs with short Requested Packet Intervals longer and reaccept edits for the ControlNet configuration.
205	ControlNet configuration exceeds processor bandwidth. Scheduled connections will be closed. You must cycle power, save with RSNetWorx, or download the program to reopen the connections. Because the configuration software is unable to accurately predict all the resources that the processor will require to execute your ControlNet configuration software (based on the relative loading on the processor), this fault code is used if the processor determines that your configuration (typically when you accept Channel 2 edits) exceeds the processor's available bandwidth. Typical causes of this error code include: • receiving data from the ControlNet network faster than the ControlNet PLC-5 processor can parse it • performing I/O updates too frequently performing immediate COntrolNet I/O ladder instructions too frequently.	*Recoverable* Reduce the number of ControlNet I/O map table entries. Possible ways: • using a discrete rack connection instead of multiple discrete module connections • combining multiple I/O racks into a single I/O rack • putting peer-to-peer data in contiguous blocks in the data table so that less send and receive scheduled messages are required Increase your Network Update Time and/or increase the Requested Packet Intervals for scheduled data transfers in your I/O map table. Increase your ladder program scan by either adding more logic or by increasing the Communications Time Slice (S:77). Reduce the number of immediate ControlNet I/O ladder instructions that are performed.
206	This error code is reserved.	Contact your local Allen-Bradley representative if you get this message.
207	This error code is reserved.	Contact your local Allen-Bradley representative if you get this message.
208	Too many pending ControlNet I/O connections.	*Recoverable* Delete one or more I/O map table entries and reaccept edits for the ControlNet configuration.

S:13-S:24

This word:	Stores:
S:13	Program file where fault occurred
S:14	Rung number where fault occurred
S:15	VME status file
S:16	I/O status File
S:17	Minor fault (word 2) See also S:10.
S:17/00	BT queue full to remote I/O
S:17/01	Queue full - channel 1A; maximum remote block-transfers used
S:17/02	Queue full - channel 1B; maximum remote block-transfers used
S:17/03	Queue full - channel 2A; maximum remote block-transfers used
S:17/04	Queue full - channel 2B; maximum remote block transfers used
S:17/05	No modem on serial port
S:17/06	• Remote I/O rack in local rack table or • Remote I/O rack is greater than the image size. This fault can also be caused by the local rack if the local rack is set for octal density scan and the I/O image tables are smaller than 64 words (8 racks) each.
S:17/07	Firmware revision for channel pairs 1A/1B or 2A/2B does not match processor firmware revision
S:17/08	ASCII instruction error
S:17/09	Duplicate node address
S:17/10	DF1 master poll list error
S:17/11	Protected processor data table element violation
S:17/12	Protected processor file violation
S:17/13	Using all 32 ControlNet MSGs
S:17/14	Using all 32 ControlNet 1771 READ and/or 1771 WRITE CIOs
S:17/15	Using all 8 ControlNet Flex I/O CIOs
S:18	Processor clock year
S:19	Processor clock month
S:20	Processor clock day
S:21	Processor clock hour
S:22	Processor clock minute
S:23	Processor clock second
S:24	Indexed addressing offset
S:25	Reserved

S:26-S:35

This word:	Stores:
S:26	User control bits
S:26/00	Restart/continuous SFC: when reset, processor restarts at first step in SFC. When set, processor continues with active step after power loss or change to RUN
S:26/01	Start-up protection after power loss: when reset, no protection. When set, processor sets major fault bit S:11/5 when powering up in run mode.
S:26/02	Define the address of the local rack: when reset, local rack address is 0. When set, local rack address is 1.
S:26/03	Set complementary I/O (series A only): when reset, complementary I/O is not enabled. When set, complementary I/O is enabled.
S:26/04	Local block-transfer compatibility bit: when reset, normal operation. When set, eliminates frequent checksum errors to certain BT modules.
S:26/05	PLC-3 scanner compatibility bit: when set (1), adapter channel response delayed by 1 ms; when reset (0) operate in normal response time.
S:26/06	Data table-modification inhibit bit. When set (1), user cannot edit the data table or modify forces while the processor keyswitch is in the RUN position. You control this bit with your programming software
S:26/07 through S:26/15	Reserved
S:27	Rack control bits: (See also S:7, S:32, S:33, S:34, and S:35) • S:27/0-7 - - I/O rack inhibit bits for racks 0-7 • S:27/8-15 - - I/O rack reset bits for racks 0-7
S:28	Program watchdog setpoint
S:29	Fault routine file
S:30	STI setpoint
S:31	STI file number
S:32	Global status bits: (See also S:7, S:27, S:33, S:34, and S:35) • S:32/0-7 rack fault bits for racks 10-17 (octal) • S:32/8-15 unused
S:33	Rack control bits: (See also S:7, S:27, S:32, S:34, and S:35) • S:33/0-7 I/O rack inhibit bits for racks 10-17 • S:33/8-15 I/O rack reset bits for racks 10-17
S:34	Global status bits: (See also S:7, S:27, S:32, S:33, and S:35) • S:34/0-7 rack fault bits for racks 20-27 (octal) • S:34/8-15 unused
S:35	Rack control bits: (See also S:7, S:27, S:32, S:33, and S:34) • S:35/0-7 I/O rack inhibit bits for racks 20-27 • S:35/8-15 I/O rack reset bits for racks 20-27

Important: Setting inhibit bits in the processor status file (S:27, S:33, or S:35) does not update inhibit bits in the I/O status file.

S:36-S:78

This word:	Stores:
S:36 - S:45	Reserved
S:46	PII program file number
S:47	PII module group
S:48	PII bit mask
S:49	PII compare value
S:50	PII down count
S:51	PII changed bit
S:52	PII events since last interrupt
S:53	STI scan time (in ms)
S:54	STI maximum scan time (in ms)
S:55	PII last scan time (in ms)
S:56	PII maximum scan time (in ms)
S:57	User program checksum
S:58	Reserved
S:59	Extended-local I/O channel discrete transfer scan (in ms)
S:60	Extended-local I/O channel discrete maximum scan (in ms)
S:61	Extended-local I/O channel block-transfer scan (in ms)
S:62	Extended-I/O channel maximum block-transfer scan (in ms)
S:63	Protected processor data table protection file number
S:64	The number of remote block-transfer command blocks being used by channel pair 1A/1B.
S:65	The number of remote block-transfer command blocks being used by channel pair 2A/2B.
S:66	Reserved.
S:77	Communication time slice for communication housekeeping functions (in ms)
S:78	MCP I/O update disable bits Bit 0 for MCP A Bit 1 for MCP B etc.

S:79-S:127

This word:	Stores:
S:79	MCP inhibit bits Bit 0 for MCP A Bit 1 for MCP B etc.
S:80-S:127	MCP file number MCP scan time (in ms) MCP max scan time (in ms) The above sequence applies to each MCP; therefore, each MCP has 3 status words. For example, word 80: file number for MCP A word 81: scan time for MCP A word 82: maximum scan time for MCP A word 83: file number for MCP B word 84: scan time for MCP B etc.

Chapter 22

Instruction Set Quick Reference

Using This Chapter

If you want to read about:	Go to page:
Relay instructions	22-2
Timer instructions	22-3
Counter instructions	22-4
Compare instructions	22-5
Compute instructions	22-7
Logical instructions	22-14
Conversion instructions	22-15
Bit modify and move instructions	22-16
File instructions	22-17
Diagnostic instructions	22-18
Shift register instructions	22-19
Sequencer instructions	22-20
Program control instructions	22-21
Process control and message instructions	22-23
Block Transfer instructions	22-24
ASCII instructions	22-25
Timing and memory requirements for bit and word instructions	22-28
Timing and memory requirements for file program control, and ASCII instructions	22-37

MORE

Important: For a more detailed description of each of these instructions, see the *PLC-5 Programming Software Instruction Set Reference*, publication 1785-6.1.

Relay Instructions

Instruction	Description	
I:012 —] [— 07	Examine On XIC	Examine data table bit I:012/07, which corresponds to terminal 7 of an input module in I/O rack 1, I/O group 2. If this data table bit is set (1), the instruction is true.
I:012 —]/[— 07	Examine Off XIO	Examine data table bit I:012/07, which corresponds to terminal 7 of an input module in I/O rack 1, I/O group 2. If this data table bit is reset (0), the instruction is true.
O:013 —( )— 01	Output Energize OTE	If the input conditions preceding this output instruction on the same rung go true, set (1) bit O:013/01, which corresponds to terminal 1 of an output module in I/O rack 1, I/O group 3.
O:013 —(L)— 01	Output Latch OTL	If the input conditions preceding this output instruction on the same rung go true, set (1) bit O:013/01, which corresponds to terminal 1 of an output module in I/O rack 1, I/O group 3. This data table bit remains set even if the rung condition goes false.
O:013 —(U)— 01	Output Unlatch OTU	If the input conditions preceding this output instruction on the same rung go true, reset (0) bit O:013/01, which corresponds to terminal 1 of an output module in I/O rack 1, I/O group 3. This is necessary to reset a bit that has been latched on.
01 —(IIN)—	Immediate Input IIN	This instruction updates a word of input-image bits before the next normal input-image update. Address this instruction by rack and group (RRG). For a local chassis, program scan is interrupted while the inputs of the addressed I/O group are scanned; for a remote chassis, program scan is interrupted only to update the input image with the latest states as found in the remote I/O buffer.
01 —(IOT)—	Immediate Output IOT	This instruction updates a word of output-image bits before the next normal output-image update. Address this instruction by rack and group (RRG). For a local chassis, program scan is interrupted while the outputs of the addressed I/O group are updated; for a remote chassis, program scan is interrupted only to update the remote I/O buffer with the latest states as found in the output image.

Timer Instructions

Instruction	Description	
TON TIMER ON DELAY Timer T4:1 Time Base 1.0 Preset 15 Accum 0	Timer On Delay TON Status Bits: EN - Enable TT - Timer Timing DN - Done	If the input conditions go true, timer T4:1 starts incrementing in 1-second intervals. When the accumulated value is greater than or equal to the preset value (15), the timer stops and sets the timer done bit. See page 24-8 for a description of prescan operation for this instruction.
TOF TIMER OFF DELAY Timer T4:1 Time Base .01 Preset 180 Accum 0	Timer Off Delay TOF Status Bits: EN - Enable TT - Timer Timing DN - Done	If the input conditions are false, timer T4:1 starts incrementing in 10 1-ms intervals as long as the rung remains false. When the accumulated value is greater than or equal to the preset value (180), the timer stops and resets the timer done bit. See page 24-8 for a description of prescan operation for this instruction.

Rung Condition	EN 15	TT 14	DN 13	ACC Value	TON Status
False	0	0	0	0	Reset
True	1	1	0	increase	Timing
True	1	0	1	>= preset	Done

Rung Condition	EN 15	TT 14	DN 13	ACC Value	TOF Status
True	1	0	1	0	Reset
False	0	1	1	increase	Timing
False	0	0	0	>= preset	Done

Instruction	Description

```
RTO
RETENTIVE TIMER ON
Timer            T4:10
Time Base          1.0
Preset              10
Accum                0
```

Retentive Timer On
RTO

If the input conditions go true, timer T4:10 starts incrementing in 1-second intervals as long as the rung remains true. When the rung goes false, the timer stops. If the rung goes true again, the timer continues. When the accumulated value is greater than or equal to th preset (10), the timer stops and sets the timer done bit.

Status Bits:
EN - Enable
TT - Timer Timing
DN - Done

Rung Condition	EN 15	TT 14	DN 13	ACC Value	RTO Status
False	0	0	0	0	Disabled
True	1	1	0	increase	Timing
False	0	0	0	maintains	Disabled
True	1	0	1	>= preset	Done

```
   T4:1
──( RES )──
```

Timer Reset
RES

If the input conditions go true, timer T4:1 is reset. This instruction resets timers and counters, as well as control blocks. This is necessary to reset the RTO accumulated value.

Counter Instructions

Instruction	Description

```
CTU
COUNT UP
Counter          C5:1
Preset             10
Accum               0
```

Count Up
CTU

If the input conditions go true, counter C5:1 starts counting, incrementing by 1 every time the rung goes from false-to-true. When the accumulated value is greater than or equal to the preset value (10), the counter sets the counter done bit.

Status Bits:
CU-Count Up
CD-Count Down
DN-Count Up done
OV-Overflow
UN-Underflow

Rung Condition	CU 15	DN 13	OV 12	ACC Value	CTU Status
False	0	0	0	0	Disabled
Toggle True	1	0	0	incr by 1	Counting
True	1	1	0	>= preset	Done
True	1	1	1	>32767	Overflow

See page 24-8 for a description of prescan operation for this instruction.

Instruction	Description

CTD
COUNT DOWN
Counter C5:1
Preset 10
Accum 35

Count Down
CTD

If the input conditions go true, counter C5:1 starts counting, decrementing by 1 every time the rung goes from false-to-true. When the accumulated value is less than the preset value (10), the counter resets the counter done bit.

Status Bits:
CU-Count Up
CD-Count Down
DN-Count Down done
OV-Overflow
UN-Underflow

Rung Condition	CD 14	DN 13	UN 11	ACC Value	CTD Status
False	0	0	0	0	Disabled
False	0	1	0	>= preset	Preload
Toggle True	1	1	0	dec by 1	Counting
True	1	0	0	< preset	Done
True	1	0	1	< -32768	Underflow

See page 24-8 for a description of prescan operation for this instruction.

Compare Instructions

Instruction	Description

LIM
LIMIT TEST (CIRC)
Low limit N7:10
3
Test N7:15
4
High limit N7:20
22

Limit Test
LIM

If the Test value (N7:15) is >= the Low Limit (N7:10) and <= the Hig Limit (N7:20), this instruction is true.

Low Limit	Test	High Limit	LIM
0	0	10	T
-5	5	10	T
5	11	10	F
10	0	0	T
10	5	-5	F
10	11	5	T

MEQ
MASKED EQUAL
Source D9:5
0000
Mask D9:6
0000
Compare D9:10
0000

Mask Compare Equal
MEQ

The processor takes the value in the Source (D9:5) and passes that value through the Mask (D9:6). Then the processor compares the result to the Compare value (D9:10). If the result and this comparisc values are equal, the instruction is true.

Source	Mask	Compare	MEQ
0008	0008	0009	T
0008	0001	0001	F
0087	000F	0007	T
0087	00F0	0007	F

Instruction	Description	
CMP COMPARE Expression N7:5 = N7:10	Compare CMP	If the expression is true, this input instruction is true. The CMP instruction can perform these operations: equal (=), less than (<), less than or equal (<=), greater than (>), greater than or equal (>=), not equal (<>), and complex expressions (up to 80 characters).
xxx xxxxxxxxxxxxx Source A N7:5 3 Source B N7:10 1		

Source A	Source B	EQU	GEQ	GRT	LEQ	LES	NEQ
10	10	T	T	F	T	F	F
5	6	F	F	F	T	T	T
21	20	F	T	T	F	F	T
-30	-31	F	T	T	F	F	T
-15	-14	F	F	F	T	T	T

	Equal to EQU	If the value in Source A (N7:5) is = to the value in Source B (N7:10), this instruction is true.
	Greater than or Equal GEQ	If the value in Source A (N7:5) is > or = the value in Source B (N7:10), this instruction is true.
	Greater than GRT	If the value in Source A (N7:5) is > the value in Source B (N7:10), this instruction is true.
	Less than or Equal LEQ	If the value in Source A (N7:5) is < or = the value in Source B (N7:10), this instruction is true.
	Less than LES	If the value in Source A (N7:5) is < the value in Source B (N7:10), this instruction is true.
	Not Equal NEQ	If the value in Source A (N7:5) is not equal to the value in Source B (N7:10), this instruction is true.

Compute Instructions

Instruction		Description
CPT COMPUTE Dest N7:3 3 Expression N7:4 - (N7:6 * N7:10)	Compute CPT	If the input conditions go true, evaluate the Expression N7:4 - (N7:6 N7:10) and store the result in the Destination (N7:3). The CPT instruction can perform these operations: add (+), subtrac (-), multiply (*), divide (|), convert from BCD (FRD), convert to BCD (TOD), square root (SQR), logical and (AND), logical or (OR), logical r (NOT), exclusive or (XOR), negate (-), clear (0), and move, X to the power of Y (**), radians (RAD), degrees (DEG), log (LOG), natural log (LN), sine (SIN), cosine (COS), tangent (TAN), inverse sine (ASN), inverse cosine (ACS), inverse tangent (ATN), and complex expressio (up to 80 characters) *Note:* Any value entered (i.e., 2.3) expands to 8 characters (2.3000000).
ACS ARCCOSINE Source F8:19 0.7853982 Dest F8:20 0.6674572	Arc cosine ACS	If input conditions go true, take the arc cosine of the value in F8:19 and store the result in F8:20.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	always resets

Instruction		Description
ADD ADD Source A N7:3 3 Source B N7:4 1 Dest N7:12 4	Addition ADD	When the input conditions are true, add the value in Source A (N7:3) to the value in Source B (N7:4) and store the result in the Destination (N7:12).

Status Bit	Description
C	sets if carry is generated; otherwise resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Instruction		Description
ASN ARCSINE Source F8:17 0.7853982 Dest F8:18 0.9033391	Arc sine ASN	When input conditions go true, take the arc sine of the value in F8:17 and store the result in F8:18.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	always resets

Instruction		Description
ATN ARCTANGENT Source F8:21 0.7853982 Dest F8:22 0.6657737	Arc tangent ATN	When input conditions go true, take the arc tangent of the value in F8:21 and store the result in F8:22.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Instruction		Description
AVE AVERAGE FILE File #N7:1 Dest N7:0 Control R6:0 Length 4 Position 0	Average AVE Status Bits: EN - Enable DN - Done bit ER - Error Bit	When the input conditions go from false-to-true, take the average the file #N7:1 and store the result in N7:0.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Instruction	Description
CLR CLR Dest D9:34 0000	Clear CLR

When the input conditions are true, clear decimal file 9, word 34 (s
to zero).

Status Bit	Description
C	always reset
V	always reset
Z	always set
S	always reset

Instruction	Description
COS COSINE Source F8:13 0.7853982 Dest F8:14 0.7071068	Cosine COS

When input conditions go true, take the cosine of the value in F8:13 and store the result in F8:14.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Instruction	Description

```
DIV
DIVIDE
Source A        N7:3
                   3
Source B        N7:4
                   1
Dest           N7:12
                   3
```

Division
DIV

When the input conditions are true, divide the value in Source A (N7:3) by the value in Source B (N7:4) and store the result in the Destination (N7:12).

Status Bit	Description
C	always resets
V	sets if division by zero or overflow; otherwise resets
Z	sets if the result is zero; otherwise resets; undefined if overflow is set
S	sets if the result is negative; otherwise resets; undefined if overflow is set

```
LN
NATURAL LOG
Source          N7:0
                   5
Dest           F8:20
            1.609438
```

Natural log
LN

When input conditions go true, take the natural log of the value in N7:0 and store the result in F8:20.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

```
LOG
LOG BASE 10
Source          N7:2
                   5
Dest            F8:3
           0.6989700
```

When input conditions go true, take the log base 10 of the value in N7:2 and store the result in F8:3.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Instruction | **Description**

```
MUL
MULTIPLY
Source A        N7:3
                   3
Source B        N7:4
                   1
Dest           N7:12
                   3
```

Multiply
MUL

When the input conditions are true, multiply the value in Source A (N7:3) by the value in Source B (N7:4) store the result in the Destination (N7:12).

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

```
NEG
NEGATE
Source          N7:3
                   3
Dest           N7:12
                  -3
```

Negate
NEG

When the input conditions are true, take the opposite sign of the Source (N7:3) and store the result in the Destination (N7:12). This instruction turns positive values into negative values and negative values into positive values.

Status Bit	Description
C	sets if the operation generates a carry; otherwise resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

```
SIN
SINE
Source         F8:11
           0.7853982
Dest           F8:12
           0.7071068
```

Sine
SIN

When input conditions go true, take the sine of the value in F8:11 and store the result in F8:12.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Instruction	Description

```
SQR
SQUARE ROOT
Source            N7:3
                    25
Dest             N7:12
                     5
```

Square Root
SQR

When the input conditions are true, take the square root of the Sour
(N7:3) and store the result in the Destination (N7:12).

Status Bit	Description
C	always resets
V	sets if overflow occurs during floating point to integer conversion; otherwise resets
Z	sets if the result is zero; otherwise resets
S	always reset

```
SRT
SORT
File             #N7:1
Control           R6:0
Length               4
Position             0
```

Sort
SRT

Status Bits:
EN-Enable
DN-Done Bit
ER-Error Bit

When the input conditions go from false-to-true, the values in N7:1 N7:2, N7:3.and N7:4 are sorted into ascending order.

```
STD
STANDARD DEVIATION
File             #N7:1
Dest              N7:0
Control           R6:0
Length               4
Position             0
```

Standard Deviation
STD

Status Bits:
EN - Enable
DN - Done Bit
ER - Error Bit

When the input conditions go from false-to-true, take the standard deviation of the values in file #N7:1 and store the result in the Destination (N7:0).

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	always resets

Instruction	Description	
SUB SUBTRACT Source A N7:3 3 Source B N7:4 1 Dest N7:12 2	Subtract SUB	When the input conditions are true, subtract the value in Source B (N7:4) from the value in Source A (N7:3) and store the result in the Destination (N7:12).

Status Bit	Description
C	sets if borrow is generated; otherwise resets
V	sets if underflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Instruction	Description	
TAN TANGENT Source F8:15 0.7853982 Dest F8:16 1.000000	Tangent TAN	When input conditions go true, take the tangent of the value in F8:15 and store the result in F8:16.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Instruction	Description	
XPY X TO POWER OF Y Source A N7:4 5 Source B N7:5 2 Dest N7:6 25	X to the power of Y XPY	When input conditions go true, take the the value in N7:4, raise it to the power stored in N7:5, and store the result in N7:6.

Status Bit	Description
C	always resets
V	sets if overflow is generated; otherwise resets
Z	sets if the result is zero; otherwise resets
S	sets if the result is negative; otherwise resets

Logical Instructions

Instruction	Description

AND

BITWISE AND	
Source A	D9:3 3F37
Source B	D9:4 00FF
Dest	D9:5 0037

AND

When the input conditions are true, the processor performs an AND operation (bit-by-bit) between Source A (D9:3) and Source B (D9:4) and stores the result in the Destination (D9:5). The truth table for a AND operation is:

Source A	Source B	Result
0	0	0
1	0	0
0	1	0
1	1	1

NOT

NOT	
Source A	D9:3 00FF
Dest	D9:5 FF00

NOT Operation

When the input conditions are true, the processor performs a NOT (takes the opposite of) operation (bit-by-bit) on the Source (D9:3) ar stores the result in the Destination (D9:5). The truth table for a NOT operation is:

Source	Destination
0	1
1	0

OR

BITWISE INCLUSIVE OR	
Source A	D9:3 3F37
Source B	D9:4 00FF
Dest	D9:5 3FFF

OR

When the input conditions are true, the processor performs an OR operation (bit-by-bit) between Source A (D9:3) and Source B (D9:4) and stores the result in the Destination (D9:5). The truth table for a OR operation is:

Source A	Source B	Result
0	0	0
1	0	1
0	1	1
1	1	1

XOR

BITWISE EXCLUSIVE OR	
Source A	D9:3 3F37
Source B	D9:4 3F37
Dest	D9:5 0000

Exclusive OR
XOR

When the input conditions are true, the processor performs an exclusive OR operation (bit-by-bit) between Source A (D9:3) and Source B (D9:4) and stores the result in the Destination (D9:5). The truth table for an XOR operation is:

Source A	Source B	Result
0	0	0
1	0	1
0	1	1
1	1	0

Status Bit	Description
C	always resets
V	always resets
Z	sets if the result is zero; otherwise resets
S	sets if the most significant bit (bit 15 for decimal or bit 17 for octal) is set (1); otherwise resets

Conversion Instructions

Instruction		Description
FRD FROM BCD Source D9:3 0037 Dest N7:12 37	Convert from BCD FRD	When the input conditions are true, convert the BCD value in the Source (D9:3) to a integer value and store the result in the Destination (N7:12). The source must be in the range of 0-9999 (BCD).

Status Bit	Description
C	always resets
V	always resets
Z	sets if the destination value is zero; otherwise resets
S	always resets

Instruction		Description
TOD TO BCD Source N7:3 44 Dest D9:5 0044	Convert to BCD TOD	When the input conditions are true, convert the integer value in Source (N7:3) to a BCD format and store the result in the Destination (D9:5).

Status Bit	Description
C	always resets
V	sets if the source value is negative or greater than 9999 (i.e. outside of the range of 0-9999)
Z	sets if the destinationvalue is zero; otherwise resets
S	always resets

Instruction		Description
DEG RADIANS TO DEGREE Source F8:7 0.7853982 Dest F8:8 45	Convert to Degrees DEG	When the input conditions are true, convert radians (the value in Source A) to degrees and stores the result in the Destination (Source times 180/p).

Status Bit	Description
C	always resets
V	sets if overflow generated; otherwise resets
Z	sets if result is zero; otherwise resets
S	sets if result is negative; otherwise resets

Instruction	Description	
RAD DEGREES TO RADIAN Source N7:9 45 Dest F8:10 0.7853982	Convert to Radians RAD	When the input conditions are true, convert degrees (the value in Source A) to radians and stores the result in the Destination (Source times p/180).

Status Bit	Description
C	always resets
V	sets if overflow generated; otherwise resets
Z	sets if result is zero; otherwise resets
S	sets if result is negative; otherwise resets

Bit Modify and Move Instructions

Instruction	Description	
MOV MOVE Source N7:3 20 Dest F8:12 20.000000	Move MOV	When the input conditions are true, move a copy of the value in Source (N7:3) to the Destination (F8:12), converting from one data type to another This overwrites the original value in the Destination.

Status Bit	Description
C	always resets
V	sets if overflow is generated during floating point-to-integer conversion; otherwise resets
Z	sets if the destination value is zero; otherwise resets
S	sets if result MSB is set; otherwise resets

Instruction	Description	
MVM MASKED MOVE Source D9:3 478F Mask D9:5 00FF Dest D9:12 008F	Masked Move MVM	When the input conditions are true, the processor passes the value the Source (D9:3) through the Mask (D9:5) and stores the result in t Destination (D9:12). This overwrites the original value in the Destination.

Status Bit	Description
C	always resets
V	always resets
Z	sets if the result is zero; otherwise resets
S	sets if most significant bit of resulting value is set; otherwise resets.

Instruction	Description	
BTD BIT FIELD DISTRIB Source N7:3 0 Source bit 3 Dest N7:4 0 Dest bit 10 Length 6	Bit Distribute BTD	When the input conditions are true, the processor copies the number of bits specified by Length, starting with the Source bit (3) of the Source (N7:3), and placing the values in the Destination (N7:4), starting with the Destination bit (10).

File Instructions

Instruction	Description	
FAL FILE ARITH/LOGICAL Control R6:1 Length 8 Position 0 Mode ALL Dest #N15:10 Expression #N14:0 - 256	File Arithmetic and Logic FAL Status Bits: EN - Enable DN - Done Bit ER - Error Bit	When the input conditions go from false-to-true, the processor reads 8 elements of N14:0, and subtracts 256 (a constant) from each element. This example shows the result being stored in the eight elements beginning with N15:10. The control element R6:1 controls the operation. The Mode determines whether the processor performs the expression on all elements in the files (ALL) per program scan, one element in the files (INC) per false-to-true transition, or a specific number of elements (NUM) per scan. The FAL instruction can perform these operations: add (+), subtract (-), multiply (*), divide (|), convert from BCD (FRD), convert to BCD (TOD), square root (SQR), logical and (AND), logical or (OR), logical not (NOT), exclusive or (XOR), negate (-), clear (0), move, and the new math instructions (see the CPT list).
FSC FILE SEARCH/COMPARE Control R9:0 Length 90 Position 0 Mode 10 Expression #B4:0 <> #B5:0	File Search and Compare FSC Status Bits: EN - Enable DN - Done Bit ER - Error Bit IN - Inhibit Bit FD - Found Bit	When the input conditions go from false-to-true, the processor performs the not-equal-to comparison on 10 elements between files B4:0 and B5:0. Mode determines whether the processor performs the expression on all elements in the files (ALL) per program scan, one element in the files (INC) per false-to-true transition, or a specific number of elements (NUM) per scan. Control element R9:0 controls the operation. When the corresponding source elements are not equal (element B4:4 and B5:4 in this example), the processor stops the search and sets the found .FD and inhibit .IN bits so your ladder program can take appropriate action. To continue the search comparison, you must reset the .IN bit. To see a list of the available comparisons, see the comparisons listed under the CMP instruction.

Instruction	Description	
COP COPY FILE Source #N7:0 Dest #N12:0 Length 5	File Copy COP	When the input conditions are true, the processor copies the contents of the Source file (N7) into the Destination file (N12). The source remains unchanged. The COP instruction copies the number of elements from the source as specified by the Length. As opposed to the MOV instruction, there is no data type conversion for this instruction.
FLL FILL FILE Source N10:6 Dest #N12:0 Length 5	File Fill FLL	When the input conditions are true, the processor copies the value in Source (N10:6) to the elements in the Destination (N12). The FLL instruction only fills as many elements in the destination as specified in the Length. As opposed to the MOV instruction, there is no data type conversion for this instruction.

Diagnostic Instructions

Instruction	Description	
FBC FILE BIT COMPARE Source #I:031 Reference #B3:1 Result #N7:0 Cmp Control R6:4 Length 48 Position 0 Result Control R6:5 Length 10 Position 0	File Bit Compare FBC Status Bits: EN - Enable DN - Done Bit ER - Error Bit IN - Inhibit Bit FD - Found Bit	When the input conditions go from false-to-true, the processor compares the number of bits specified in the CMP Control Length (48) of the Source file (#I:031) with the bits in the Reference file (#B3:1). The processor stores the results (mismatched bit numbers) in the Result file (#N7:0). File R6:4 controls the compare and file R6:5 controls the file that contains the results. The file containing the results can hold up to 10 (the number specified in the Length field) mismatches between the compared files. ***Note:*** To avoid encountering a possible run-time error when executing this instruction, add a ladder rung that clears S:24 (indexed addressing offset) immediately before a FBC instruction.
DDT DIAGNOSTIC DETECT Source #I:030 Reference #B3:1 Result #N10:0 Cmp Control R6:0 Length 20 Position 0 Result Control R6:1 Length 5 Position 0	Diagnostic Detect DDT Status Bits: EN - Enable DN - Done Bit ER - Error Bit IN - Inhibit Bit FD - Found Bit	When the input conditions go from false-to-true, the processor compares the number of bits specified in the CMP Control Length (20) of the Source file (# I:030) with the bits in the Reference file (#B3:1). The processor stores the results (mismatched bit numbers) in the Result file (#N10:0). Control element R6:0 controls the compare and the control element R6:1 controls the file that contains the results (#N10:0). The file containing the results can hold up to 5 (the number specified in the Length field) mismatches between the compared files. The processor copies the source bits to the reference file for the next comparison. The difference between the DDT and FBC instruction is that each time the DDT instruction finds a mismatch, the processor changes the reference bit to match the source bit. You can use the DDT instruction to update your reference file to reflect changing machine or process conditions. ***Note:*** To avoid encountering a possible run-time error when executing this instruction, add a ladder rung that clears S:24 (indexed addressing offset) immediately before a DDT instruction.

Instruction	Description	
DTR DATA TRANSITION Source I:002 Mask 0FFF Reference N63:11	Data Transition DTR	The DTR instruction compares the bits in the Source (I:002) through a Mask (0FFF) with the bits in the Reference (N63:11). When the masked source is different than the reference, the instruction is true for only 1 scan. The source bits are written into the reference address for the next comparison. When the masked source and the reference are the same, the instruction remains false.

Shift Register Instructions

Instruction	Description	
BSL BIT SHIFT LEFT File #B3:1 Control R6:53 Bit Address I:022/12 Length 5	Bit Shift Left BSL Status Bits: EN - Enable DN - Done Bit ER - Error Bit UL - Unload Bit	If the input conditions go from false-to-true, the BSL instruction shifts the number of bits specified by Length (5) in File (B3), starting at bit 16 (B3:1/0 = B3/16), to the left by one bit position. The source bit (I:022/12) shifts into the first bit position, B3:1/0 (B3/16). The fifth bit, B3:1/4 (B3/20), is shifted into the UL bit of the control structure (R6:53).
BSR BIT SHIFT RIGHT File #B3:2 Control R6:54 Bit Address I:023/06 Length 3	Bit Shift Right BSR Status Bits: EN - Enable DN - Done Bit ER - Error Bit UL - Unload Bit	If the input conditions go from false-to-true, the BSR instruction shifts the number of bits specified by Length (3) in File (B3), starting with B3:2/0 (=B3/32), to the right by one bit position. The source bit (I:023/06) shifts into the third bit position B3/34. The first bit (B3/32) is shifted into the UL bit of the control element (R6:54).
FFL FIFO LOAD Source N60:1 FIFO #N60:3 Control R6:51 Length 64 Position 0	FIFO Load FFL Status Bits: EN - Enable Load DN - Done Bit EM - Empty Bit	When the input conditions go from false-to-true, the processor loads N60:1 into the next available element in the FIFO file, #N60:3, as pointed to by R6:51. Each time the rung goes from false-to-true, the processor loads another element. When the FIFO file (stack) is full, (64 words loaded), the DN bit is set. See page 24-8 for a description of prescan activities for this instruction.
FFU FIFO UNLOAD FIFO #N60:3 Dest N60:2 Control R6:51 Length 64 Position 0	FIFO Unload FFU Status Bits: EU - Enable Unload DN - Done Bit EM - Empty Bit	When the input conditions go from false-to-true, the processor unloads an element from #N60:3 into N60:2. Each time the rung goes from false-to-true, the processor unloads another value. All the data in file #N60:3 is shifted one position toward N60:3. When the file is empty, the EM bit is set. See page 24-8 for a description of prescan activities for this instruction.

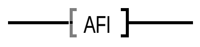
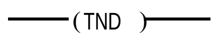
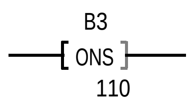
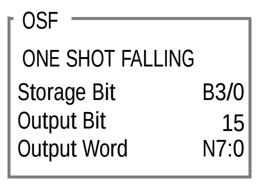
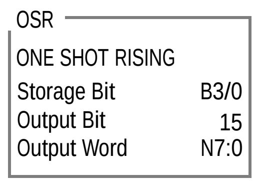
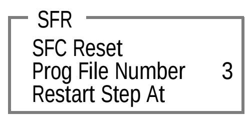
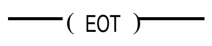
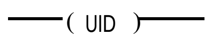
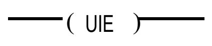
Instruction	Description	
LFL LIFO LOAD Source N70:1 LIFO #N70:3 Control R6:61 Length 64 Position 0	LIFO Load LFL Status Bits: EN - Enable Load DN - Done Bit EM - Empty Bit	When the input conditions go from false-to-true, the processor loads N70:1 into the next available element in the LIFO file #N70:3, as pointed to by R6:61. Each time the rung goes from false-to-true, the processor loads another element. When the LIFO file (stack) is full (64 words have been loaded), the DN bit is set. See page 24-8 for a description of prescan activities for this instruction.
LFU LIFO UNLOAD LIFO #N70:3 Dest N70:2 Control R6:61 Length 64 Position 0	LIFO Unload LFU Status Bits: EU - Enable Unload DN - Done Bit EM - Empty Bit	When the input conditions go from false-to-true, the processor unloads the last element from #N70:3 and puts it into N70:2. Each time the rung goes from false-to-true, the processor unloads another element. When the LIFO file is empty, the EM bit is set. See page 24-8 for a description of prescan activities for this instruction.

Sequencer Instructions

Instruction	Description	
SQI SEQUENCER INPUT File #N7:11 Mask FFF0 Source I:031 Control R6:21 Length 4 Position 0	Sequencer Input SQI	The SQI instruction filters the Source (I:031) input image data through a Mask (FFF0) and compare the result to Reference data (#N7:11) to see if the two values are equal. The operation is controlled by the information in the control file R6:21. When the status of all unmasked bits of the word pointed to by control element R6:21 matches the corresponding reference bits, the rung condition remains true if preceded by a true rung condition.
SQL SEQUENCER LOAD File #N7:20 Source I:002 Control R6:22 Length 5 Position 0	Sequencer Load SQL Status Bits: EN - Enable DN - Done Bit ER - Error Bit	The SQL instruction loads data into the sequencer File (#N7:20) from the source word (I:002) by stepping through the number of elements specified by Length (5) of the Source (I:002), starting at the Position (0). The operation is controlled by the information in the control file R6:22. When the rung goes from false-to-true, the SQL instruction increments the next step in the sequencer file and loads the data into it for every scan that the rung remains true. See page 24-8 for a description of prescan operation for this instruction.
SQO SEQUENCER OUTPUT File #N7:1 Mask 0F0F Dest O:014 Control R6:20 Length 4 Position 0	Sequencer Output SQO # Status Bits: EN - Enable DN - Done Bit ER - Error Bit	When the rung goes from false-to-true, the SQO instruction increments to the next step in the sequencer File (#N7:1). The data in the sequencer file is transferred through a Mask (0F0F) to the Destination (O:014) for every scan that the rung remains true. See page 24-8 for a description of prescan operation for this instruction.

Program Control Instructions

Instruction	Description	
—(MCR)—	Master Control Reset MCR	If the input conditions are true, the program scans the rungs between MCR instruction rungs and processes the outputs normally. If the input condition is false, rungs between the MCR-instruction rungs are executed as false.
10 —(JMP)—	Jump JMP	If the input conditions are true, the processor skips rungs by jumping to the rung identified by the label (10).
10 —[LBL]—	Label LBL	When the processor reads a JMP instruction that corresponds to label 10, the processor jumps to the rung containing the label and starts executing. **Important:** Must be the first instruction on a rung.
FOR FOR Label Number 0 Index N7:0 Initial Value 0 Terminal Value 10 Step Size 1	FOR Loop FOR	The processor executes the rungs between the FOR and the NXT instruction repeatedly in one program scan, until it reaches the terminal value (10) or until a BRK instruction aborts the operation. Step size is how the loop index is incremented. See page 24-8 for a description of prescan operation for this instruction.
NXT NEXT Label Number 0	Next NXT	The NXT instruction returns the processor to the corresponding FOR instruction, identified by the label number specified in the FOR instruction. NXT must be programmed on an unconditional rung that is the last rung to be repeated in a For-Next loop.
—[BRK]—	Break BRK	When the input conditions go true, the BRK instruction aborts a For-Next loop.
JSR JUMP TO SUBROUTINE Program File 90 Input par N16:23 Input par N16:24 Input par 231 Return par N19:11 Return par N19:12	Jump to Subroutine JSR	If the input conditions are true, the processor starts running a subroutine Program File (90). The processor passes the Input Parameters (N16:23, N16:24, 231) to the subroutine and the RET instruction passes Return Parameters (N19:11, N19:12) back to the main program, where the processor encountered the JSR instruction.
SBR SUBROUTINE Input par N43:0 Input par N43:1 Input par N43:2	Subroutine SBR	The SBR instruction is the first instruction in a subroutine file. This instruction identifies Input Parameters (N43:0, N43:1, N43:2) the processor receives from the corresponding JSR instruction. You do not need the SBR instruction if you do not pass input parameters to the subroutine.
RET RETURN ( ) Return par N43:3 Return par N43:4	Return RET	If the input conditions are true, the RET instruction ends the subroutine and stores the Return Parameters (N43:3, N43:4) to be returned to the JSR instruction in the main program.

Instruction	Description	
	Always False AFI	The AFI instruction disables the rung (i.e., the rung is always false).
	Temporary End TND	If the input conditions are true, the TND instruction stops the processor from scanning the rest of the program (i.e., this instruction temporarily ends the program).
	One Shot ONS	If the input conditions preceding the ONS instructions on the same rung go from false-to-true, the ONS instruction conditions the rung so that the output is true for one scan. The rung is false on successive scans. See page 24-8 for a description of prescan operation for this instruction.
 OSF ONE SHOT FALLING Storage Bit B3/0 Output Bit 15 Output Word N7:0	One Shot Falling OSF Status Bits: OB - Output Bit SB - Storage Bit	The OSF instruction triggers an event to occur one time. Use the OSF instruction whenever an event must start based on the change of state of a rung from true-to-false, not on the resulting rung status. The output bit (N7:0/15) is set (1) for one program scan when the rung goes from true-to-false. See page 24-8 for a description of prescan operation for this instruction.
 OSR ONE SHOT RISING Storage Bit B3/0 Output Bit 15 Output Word N7:0	One Shot Rising OSR Status Bits: OB - Output Bit SB - Storage Bit	The OSR instruction triggers an event to occur one time. Use the OSR instruction whenever an event must start based on the change of state of a rung from false-to-true, not on the resulting rung status. The output bit (N7:0/15) is set (1) for one program scan when the rung goes from false-to-true. See page 24-8 for a description of prescan operation for this instruction.
	SFC Reset SFR	The SFR instruction resets the logic in a sequential function chart. When the SFR instruction goes true, the processor performs a lastscan/postscan on all active steps and actions in the selected file, and then resets the logic in the SFC on the next program scan. The chart remains in this reset state until the SFR instruction goes false.
	End of Transition EOT	The EOT instruction should be the last instruction in a transition file. If you do not use an EOT instruction, the processor always evaluates the transition as true. See page 24-8 for a description of prescan operation for this instruction.
	User Interrupt Disable UID	The UID instruction temporarily disables an interrupt-driven ladder program (such as an STI or PII) from interrupting the currently executing program.
	User Interrupt Enable UIE	The UIE instruction re-enables the interrupt-driven ladder program to interrupt the currently executing ladder program.

Process Control, Message Instructions

Instruction	Description	
PID PID Control Block PD10:0 Proc Variable N15:13 Tieback N15:14 Control Output N20:21	Proportional, Integral, and Derivative PID Status Bits: EN - Enable DN - Done Bit (for N control blocks only)	The control block (PD10:0) contains the instruction information for the PID. The PID gets the process variable from N15:13 and sends the PID output to N20:21. The tieback stored in N15:14 handles the manual control station. If you use an N control block, the rung must transition from false to true for execution. If you use PD control block, then there is no done bit. Also, the rung input conditions need to be true. See page 24-8 for a description of prescan operation for this instruction.
MSG SEND/RECEIVE MESSAGE Control Block MG7:10 Bit # | Status Bits 15 | EN - Enable 14 | ST - Start Bit 13 | DN - Done Bit 12 | ER - Error Bit 11 | CO - Continuous 10 | EW - Enabled-Waiting 9 | NR - No Response 8 | TO - Time Out Bit		If the input conditions go from false to true, the data is transferred according to the instruction parameters you set when you entered the message instruction. The Control Block (MG7:10) contains status and instruction parameters. You can also use N control blocks. For continuous MSGs, condition the rung to be true for only one scan. See page 24-8 for a description of prescan operation for this instruction.

Block Transfer Instructions

Integer (N) control block

Word Offset	Description
0	status bits (see below)
1	requested word count
2	transmitted word count
3	file number
4	element number

Block Transfer (BT) control block

Word Mnemonic	Description
.EN through .RW	status bits
.RLEN	requested length
.DLEN	transmitted word length/error code
.FILE	file number
.ELEM	element number
.RGS	rack/group/slot

Word 0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
EN	ST	DN	ER	CO	EW	NR	TO	RW	**	rack **	**	group**		slot	

Instruction		Description
BTR BLOCK TRANSFER READ Rack 1 Group 0 Module 0 Control Block BT11:100 Data File N10:110 Length 40 Continuous Y	Block Transfer Read BTR	If the input conditions go from false to true, a block transfer read is initiated for the I/O module located at rack 1, group 0, module 0. The Control Block (BT11:100, 6-word file) contains status for the transfer. The Data File (N10:110) is where the data read from the module is stored. The BT Length (40) identifies the number of words in the transfer. A non-continuous block transfer is queued and run only once on a false-to-true rung transition; a continuous block transfer is repeatedly requeued. You can also use the N data type for the control blocks. See page 24-8 for a description of prescan operation for this instruction.

PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, -5/80E processors		PLC-5/40, -5/40L, -5/60, -5/60L, -5/80, -5/40E, -5/80E processors		PLC-5/60, -5/60L, -5/80, -5/80E processors	
S:7 bit #	BT queue full for rack	S:32 bit #	BT queue full for rack	S:34 bit #	BT queue full for rack
08[1]	0	08	10	08	20
09[1]	1	09	11	09	21
10[1]	2	10	12	10	22
11[1]	3	11	13	11	23
12	4	12	14	12	24
13	5	13	15	13	25
14	6	14	16	14	26
15	7	15	17	15	27

1. PLC-5/11, -5/20, and 5/20E processors also

Instruction		Description
BTW BLOCK TRANSFER WRITE Rack 1 Group 0 Module 0 Control Block BT11:0 Data File N10:10 Length 40 Continuous Y	Block Transfer Write BTW	If the input conditions go from false-to-true, the block transfer write is initiated for the I/O module located at rack 1, group 0, module 0. The Control Block (BT11:0, 6-word file) contains status for the transfer. The Data File contains the data to write to the module (N10:10). The BT Length (40) identifies the number of words in the transfer. A non-continuous block transfer is queued and run only once on a false-to-true rung transition; a continuous block transfer is repeatedly requeued. You can also use the N data type for the control block. See page 24-8 for a description of prescan operation for this instruction.

ASCII Instructions

Status Bits:

EN - Enable	EM - Empty Bit
DN - Done Bit	EU - Queue
ER - Error Bit	FD - Found Bit

Instruction		Description
ABL ASCII TEST FOR LINE Channel 0 Control R6:32 Characters	ASCII Test for Line ABL	If input conditions go from false-to-true, the processor reports the number of characters in the buffer, up to and including the end-of-line characters and puts this value into the position word of the control structure (R6:32.POS). The processor also displays this value in the characters field of the display. See page 24-8 for a description of prescan operation for this instruction.
ACB ASCII CHARS IN BUFFER Channel 0 Control R6:32 Characters	ASCII Characters in Buffer ACB	If input conditions go from false-to-true, the processor reports the total number of characters in the buffer and puts this value into the position word (.POS) of the control structure. The processor also displays this value in the characters field of the display. See page 24-8 for a description of prescan operation for this instruction.
ACI STRING TO INTEGER CONVERSION Source ST38:90 Dest N7:123 75	Convert ASCII String to Integer ACI	If input conditions are true, the processor converts the string in ST38:90 to an integer and stores the result in N7:123.

Status Bit	Description
C	set if a carry was generated during the conversion; otherwise resets
V	set if source is > 32,767 or < -32,768, otherwise resets
Z	set if source is zero; otherwise resets
S	set if destination is negative; otherwise resets

Instruction	Description	
ACN STRING CONCATENATE Source A ST38:90 Source B ST37:91 Dest ST52:76	ASCII String Concatenate ACN	If input conditions are true, the processor concatenates the string in ST38:90 with the string in ST37:91 and store the result in ST52:76.
AEX STRING EXTRACT Source ST38:40 Index 42 Number 10 Dest ST52:75	ASCII String Extract AEX	If input conditions are true, the processor extracts 10 characters starting at the 42nd character of ST38:40 and store the result in ST52:75.
AIC INTEGER TO STRING CONVERSION Source 876 Dest ST38:42	Convert Integer to ASCII String AIC	If input conditions are true, the processor converts the value 876 to a string and store the result in ST38:42.
AHL ASCII HANDSHAKE LINE Channel 0 AND Mask 0001 OR Mask 0003 Control R6:23 Channel Status	ASCII Handshake Lines AHL Status Bits: EN-Enable DN-Done Bit ER-Error Bit	If input conditions go from false-to-true, the processor uses the AND and OR masks to determine whether to set or reset the DTR (bit 0) and RTS (bit 1) lines, or leave them unchanged. Bit 0 and 1 of the AND mask cause the line(s) to reset if 1 and leave the line(s) unchanged if 0. BIt 0 and 1 of the OR mask cause the line(s) to set if 1 and leave the line(s) unchanged if 0. See page 24-8 for a description of prescan operation for this instruction.
ARD ASCII READ Channel 0 Dest ST52:76 Control R6:32 String Length 50 Characters Read	ASCII Read ARD Status Bits EN - Enable DN - Done Bit ER - Error Bit UL - Unload EM - Empty EU - Queue	If input conditions go from false-to-true, read 50 characters from the buffer and move them to ST52:76. The number of characters read is stored in R6:32.POS and displayed in the Characters Read Field of the instruction display. See page 24-8 for a description of prescan operation for this instruction.
ARL ASCII READ LINE Channel 0 Dest ST50:72 Control R6:30 String Length 18 Characters Read	ASCII Read Line ARL Status Bits EN - Enable DN - Done Bit ER - Error Bit UL - Unload EM - Empty EU - Queue	If input conditions go from false-to-true, read 18 characters (or until end-of-line) from the buffer and move them to ST50:72. The number of characters read is stored in R6:30.POS and displayed in the Characters Read Field of the instruction display. See page 24-8 for a description of prescan operation for this instruction.

Instruction	Description	
ASC STRING SEARCH Source ST38:40 Index 35 Search ST52:80 Result 42	ASCII String Search ASC	If input conditions are true, search ST52:80 starting at the 35th character, for the string found in ST38:40. In this example, the string was found at index 42. If the string is not found, the ASCII instruction minor fault bit S:17/8 is set and the result is zero.
ASR ASCII STRING COMPARE Source A ST37:42 Source B ST38:90	ASCII String Compare ASR	If the string in ST37:42 is identical to the string in ST38:90, the instruction is true. Note that this is an input instruction. An invalid string length causes the ASCII instruction error minor fault bit S:17/8 to be set, and the instruction is false.
AWA ASCII WRITE APPEND Channel 0 Source ST52:76 Control R6:32 String Length 50 Characters Sent	ASCII Write Append AWA Status Bits EN - Enable DN - Done Bit ER - Error Bit UL - Unload EM - Empty EU - Queue	If input conditions go from false-to-true, read 50 characters from ST52:76 and write it to channel 0 and append the two character configuration in the channel configuration (default CR/LF). The number of characters sent is stored in R6:32.POS and displayed in the characters sent field of the instruction display. See page 24-8 for a description of prescan operation for this instruction.
AWT ASCII WRITE Channel 0 Source ST37:40 Control R6:23 String Length 40 Characters Sent	ASCII Write AWT Status Bits EN - Enable DN - Done Bit ER - Error Bit UL - Unload EM - Empty EU - Queue	If input conditions go from false-to-true, write 40 characters from ST37:40 to channel 0. The number of characters sent is stored in R6:23.POS and displayed in the characters sent field of the instruction display. See page 24-8 for a description of prescan operation for this instruction.

Bit and Word Instructions

Category	Code	Title	Execution Time (µs) integer		Execution Time (µs) floating point		Words of Memory [1]
			True	False	True	False	
Relay	XIC	examine if closed	.32	.16			1[2]
	XIO	examine if open	.32	.16			1[2]
	OTL	output latch	.48	.16			1[2]
	OTU	output unlatch	.48	.16			1[2]
	OTE	output energize	.48	.48			1[2]
Branch		branch end	.16	.16			1
		next branch					1
		branch start					1
Timer and Counter	TON	timer on (0.01 base) (1.0 base)	3.8 4.1	2.6 2.5			2-3
	TOF	timer off (0.01 base) (1.0 base)	2.6 2.6	3.2 3.2			2-3
	RTO	retentive timer on (0.01 base) (1.0 base)	 3.8 4.1	 2.4 2.3			2-3
	CTU	count up	3.4	3.4			2-3
	CTD	count down	3.3	3.4			2-3
	RES	reset	2.2	1.0			2-3

1. Use the larger number for addresses beyond 2048 words in the processor's data table.
2. For every bit address above the first 256 words of memory in the data table, add 0.16 µs and 1 word of memory.

Category	Code	Title	Execution Time (µs) integer		Execution Time (µs) floating point		Words of Memory [1]
			True	False	True	False	
Arithmetic	ADD	add	6.1	1.4	14.9	1.4	4-7
	SUB	subtract	6.2	1.4	15.6	1.4	4-7
	MUL	multiply	9.9	1.4	18.2	1.4	4-7
	DIV	divides	12.2	1.4	23.4	1.4	4-7
	SQR	square root	9.9	1.3	35.6	1.3	3-5
	NEG	negate	4.8	1.3	6.0	1.3	3-5
	CLR	clear	3.4	1.1	3.9	1.1	2-3
	AVE	average file	152+E25.8	30	162+E22.9	36	4-7
	STD	standard deviation	262+E92.5	34	295+E85.5	34	4-7
	TOD	convert to BCD	7.8	1.3			3-5
	FRD	convert from BCD	8.1	1.3			3-5
	RAD	radian	57.4	1.4	50.1	1.4	3-5
	DEG	degree	55.9	1.4	50.7	1.4	3-5
	SIN	sine			414	1.4	3-5
	COS	cosine			404	1.4	3-5
	TAN	tangent			504	1.4	3-5
	ASN	inverse sine			426	1.4	3-5
	ACS	inverse cosine			436	1.4	3-5
	ATN	inverse tangent			375	1.4	3-5
	LN	natural log	409	1.4	403	1.4	3-5
	LOG	log	411	1.4	403	1.4	3-5
	XPY	X to the power of Y	897	1.5	897	1.5	4-7
	SRT	sort file (5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	 276 + 12[E**1.34] 224 + 25[E**1.34]	 227 189	 278 + 16[E**1.35] 230 + 33[E**1.35]	 227 189	3-5

[1] Use the larger number for addresses beyond 2048 words in the processor's data table.
E = number of elements acted on per scan
SRT true is only an approximation. Actual time depends on the randomness of the numbers.

Category	Code	Title	Execution Time (µs) integer		Execution Time (µs) floating point		Words of Memory [1]
			True	False	True	False	
Logic	AND	and	5.9	1.4			4-7
	OR	or	5.9	1.4			4-7
	XOR	exclusive or	5.9	1.4			4-7
	NOT	not	4.6	1.3			3-5
Move	MOV	move	4.5	1.3	5.6	1.3	3-5
	MVM	masked move	6.2	1.4			4-7
	BTD	bit distributor	10.0	1.7			6-9
Comparison	EQU	equal	3.8	1.0	4.6	1.0	3-5
	NEQ	not equal	3.8	1.0	4.5	1.0	3-5
	LES	less than	4.0	1.0	5.1	1.0	3-5
	LEQ	less than or equal	4.0	1.0	5.1	1.0	3-5
	GRT	greater than	4.0	1.0	5.1	1.0	3-5
	GEQ	greater than or equal	4.0	1.0	5.1	1.0	3-5
	LIM	limit test	6.1	1.1	8.4	1.1	4-7
	MEQ	mask compare if equal	5.1	1.1			4-7
Compare	CMP	all	2.48 + (Σ[0.8 + i])	2.16 + Wi[0.56]	2.48 + (Σ[0.8 + i])	2.16 + Wi[0.56]	2+Wi
Compute	CPT	all	2.48.+ (Σ[0.8 + i])	2.16 + Wi[0.56]	2.48.+ (Σ[0.8 + i])	2.16 + Wi[0.56]	2+Wi

[1]. Use the larger number for addresses beyond 2048 words in the processor's data table.
i = execution time of each instruction (e.g., ADD, SUB, etc.) used within the CMP or CPT expression
Wi = number of words used by the instruction (e.g., ADD, SUB, etc) within the CMP or CPT expression
CMP or CPT instructions are calculated with short direct addressing

File, Program Control, and ASCII Instructions

Category	Code	Title	Time (µs) integer		Time (µs) floating point		Words of Memory[1]
			True	False	True	False	
File Arithmetic and Logic	FAL	all	11 + (S[2.3 + i])E	6.16 + Wi[0.16]	11 + (Σ[2.3 + i])E	6.16 + Wi[0.16]	3-5 +Wi
File Search and Compare	FSC	all	11 + (S[2.3 + i])E	6.16 + Wi[0.16]	11 + (Σ[2.3 + i])E	6.16 + Wi[0.16]	3-5 +Wi
File	COP	copy	16.2+E[0.72]	1.4	17.8+E[1.44]	1.4	4-6
		counter, timer, and control	15.7+E[2.16]	1.4			
	FLL	fill	15.7+E[0.64]	1.5	18.1+E[0.80]	1.5	4-6
		counter, timer, and control	15.1+E[1.60]	1.5			
Shift Register	BSL	bit shift left	10.6+B[0.025]	5.2			4-7
	BSR	bit shift right	11.1 + B[0.025]	5.2			4-7
	FFL	FIFO load	8.9	3.8			4-7
	FFU	FIFO unload	10.0+E[0.43]	3.8			4-7
	LFL	LIFO load	9.1	3.7			4-7
	LFU	LIFO unload	10.6	3.8			4-7
Diagnostic	FBC	0 mismatch	15.4 + B[0.055]	2.9			6-11
		1 mismatch	22.4 + B[0.055]	2.9			
		2 mismatches	29.9+ B[0.055]	2.9			
	DDT	0 mismatch	15.4 + B[0.055]	2.9			6-11
		1 mismatch	24.5 + B[0.055]	2.9			
		2 mismatches	34.2 + B[0.055]	2.9			
	DTR	data transitional	5.3	5.3			4-7

[1] Use the larger number for addresses beyond 2048 words in the processor's data table.
i = execution time of each instruction (e.g., ADD, SUB, etc.) used within the FAL or the FSC expression
E = number of elements acted on per scan
B = number of bits acted on per scan
Wi = number of words used by the instruction (e.g., ADD, SUB, etc.) within the FAL or FSC expression
FAL or FSC instructions are calculated with short direct addressing

Category	Code	Title	Time (µs) integer True	Time (µs) integer False	Time (µs) floating point True	Time (µs) floating point False	Words of Memory[1]
Sequencer	SQI	sequencer input	7.9	1.3			5-9
	SQL	sequencer load	7.9	3.5			4-7
	SQO	sequencer output	9.7	3.7			5-9
Immediate I/O[2]	IIN	immediate input • PLC-5/11, -5/20, and -5/20E • PLC5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, and -5/80, -5/80E	 • 357 • 307	1.1			2
	IOT	immediate output • PLC-5/11, -5/20, and -5/20E • PLC5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	 • 361 • 301	1.1			2
Zone Control	MCR	master control	0.16	0.16			1
Program Control	JMP	jump	8.9+(file# - 2) * 0.96	1.4			2
	JSR[3] /RET	jump to subroutine /return — 0 parameters — 1 parameter — increase/ parameter	 12.3 16.1 3.8	 1.0 1.0 not applicable	 not applicable 17.3 5.0	 not applicable 1.0 not applicable	3+parameters/JSR 1+parameters/RET
	SBR						1+ parameters

[1] Use the larger number for addresses beyond 2048 words in the processor's data table.
[2] Timing for immediate I/O instructions is the time for the instruction to queue-up for processing.
[3] Calculate execution times as follows: (time) + (quantity of additional parameters)(time/parameter). For example: if you are passing 3 integer parameters in a JSR within a PLC-5/11 processor, the execution time = 16.1 + (2)(3.8) = 23.7 ms

Category	Code	Title	Time (μs) integer		Time (μs) floating point		Words of Memory[1]
			True	False	True	False	
Program Control	LBL	label	0.16	0.16			2
	END	end	negligible				1
	TND	temporary end					1
	EOT	end of transition					1
	AFI	always false	0.16	0.16			1
	ONS	one shot	3.0	3.0			2-3
	OSR	one shot rising	6.2	6.0			4-6
	OSF	one shot falling	6.2	5.8			4-6
	FOR/ NXT	for next loop	8.1+ L[15.9]+ (file# - 2) * 0.96	5.3 + N[0.75]			FOR 5-9 NXT 2
	BRK	break	11.3 + N[0.75]	0.9			1
	UID	user interrupt disable (PLC-5/11, -5/20, -5/30, -5/40, -5/60, and -5/80 processors)	175 119	1.0			1
	UIE	user interrupt enable (PLC-5/11, -5/20, -5/30, -5/40, -5/60, and -5/80 processors)	170 100	1.0			1

1. Use the larger number for addresses beyond 2048 words in the processor's data table.
L = number of FOR/NXT loops
N = number of words in memory between FOR/NXT or BRK/NXT

Category	Code	Title	Time (μs) integer		Time (μs) floating point		Words of Memory[1]
			True	False	True	False	
Process Control	PID	PID loop control					5-9
Gains		Independent • PLC-5/11, -5/20, -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L -5/80, -5/80E	• 462 • 655	3.0	1120	58	
		ISA • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 560 • 895		1180		
Modes		Manual • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 372 • 420		1150		
		Set Output • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 380 • 440		1130		
Cascade		Slave	▨	▨	1530	▨	
		Master	▨	▨	1080	▨	

[1] Use the larger number for addresses beyond 2048 words in the processor's data table.

Category	Code	Title	Time (μs) integer		Time (μs) floating point		Words of Memory[1]
			True	False	True	False	
ASCII [2]	ABL	test buffer for line • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 316 • 388	• 214 • 150			3-5
	ACB	no. of characters in buffer • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 316 • 389	• 214 • 150			3-5
	ACI	string to integer • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 220 + C[11] • 140 + C[21.4]	1.4			3-5
	ACN	string concatenate • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 237 + C[2.6] • 179 + C[5.5]	1.9			4-7
	AEX	string extract • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 226 + C[1.1] • 159 + C[2.2]	1.9			5-9
	AHL¡	set or reset lines • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 318 • 526	• 213 • 157			5-9

[1] Use the larger number for addresses beyond 2048 words in the processor's data table.
[2] Timing for ASCII instructions is the time for the instruction to queue-up for processing in channel 0.
C = number of ASCII characters

Category	Code	Title	Time (µs) integer		Time (µs) floating point		Words of Memory[1]
			True	False	True	False	
ASCII[2]	AIC	integer to string • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 260 • 270	1.4			3-5
	ARD	read characters • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 315 • 380	• 214 • 149			4-7
	ARL	read line • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 316 • 388	• 214 • 151			4-7
	ASC	string search • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 222 + C[1.7] • 151 + C[3.0]	1.9			5-9
	ASR	string compare • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 234 + C[1.3] • 169 + C[2.4]	• 202 • 119			3-5

1. Use the larger number for addresses beyond 2048 words in the processor's data table.
2. Timing for ASCII instructions is the time for the instruction to queue-up for processing in channel 0.
C = number of ASCII characters

Category	Code	Title	Time (μs) integer		Time (μs) floating point		Words of Memory[1]
			True	False	True	False	
ASCII[2]	AWA	write with append • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 319 • 345	• 215 • 154			4-7
	AWT	write • PLC-5/11, -5/20, and -5/20E • PLC-5/30, -5/40, -5/40E, -5/40L -5/60, -5/60L, -5/80, and -5/80E	• 318 • 344	• 215 • 151			4-7

1. Use the larger number for addresses beyond 2048 words in the processor's data table.
2. Timing for ASCII instructions is the time for the instruction to queue-up for processing in channel 0.
C = number of ASCII characters

Notes:

Switch Setting Reference

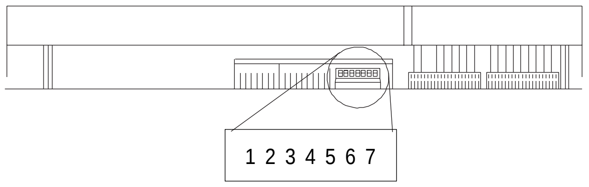
Using This Chapter

For this switch setting:	Go to page:
Enhanced and Ethernet PLC-5 switch 1 for defining the processor's DH+ address	23-2
Enhanced and Ethernet PLC-5 switch 2 for defining the processor's serial port electrical interface	23-3
I/O chassis containing a PLC-5 processor	23-4
I/O chassis containing a 1771-ASB, remote I/O adapter module	23-5
I/O chassis configuration plug for defining an external or slot power supply	23-6
1771-ASB not using complementary I/O	23-7
1771-ALX adapter module	23-9

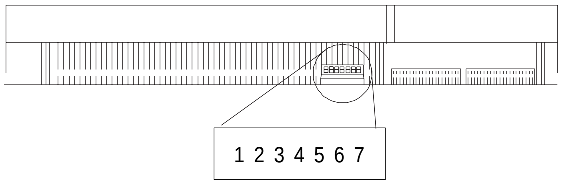
Processor Switches

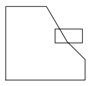
Switch 1

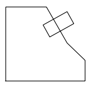
Side View of PLC-5/11, -5/20, -5/26, -5/20E processors Switch Assembly SW1



Side View of PLC-5/30, -5/40, -5/46, -5/40L, -5/60, -5/60L, -5/80, -5/86, -5/40E, and -5/80E processors Switch Assembly SW1





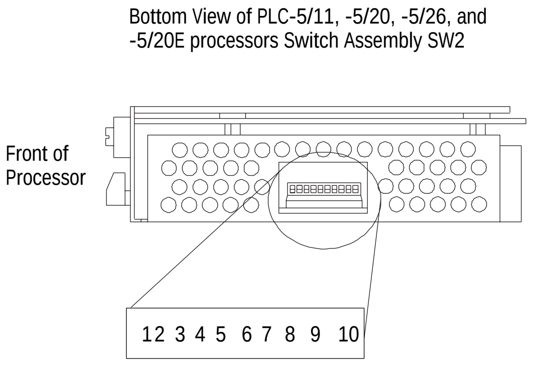


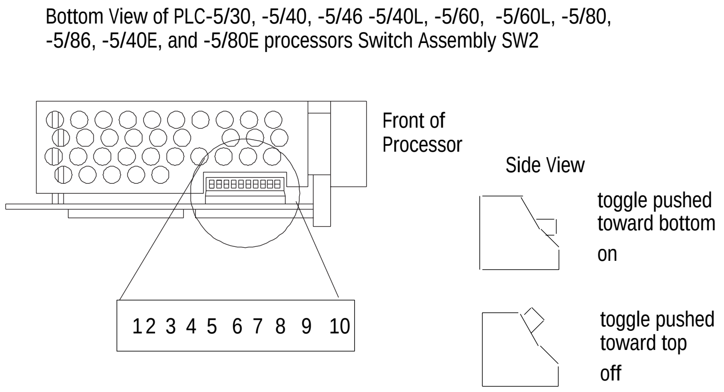
To select DH+ baud rate for channel 1A:	Set switch:	To:	
DH+ address	1 through 6	(See below)	
DH+ baud rate	7	on (down) off (up)	57.6 kbps 230.4 kbps

DH+ Station Number	Switch					
	1	2	3	4	5	6
0	on	on	on	on	on	on
1	off	on	on	on	on	on
2	on	off	on	on	on	on
3	off	off	on	on	on	on
4	on	on	off	on	on	on
5	off	on	off	on	on	on
6	on	off	off	on	on	on
7	off	off	off	on	on	on
10	on	on	on	off	on	on
11	off	on	on	off	on	on
12	on	off	on	off	on	on
13	off	off	on	off	on	on
14	on	on	off	off	on	on
15	off	on	off	off	on	on
16	on	off	off	off	on	on
17	off	off	off	off	on	on
20	on	on	on	on	off	on
21	off	on	on	on	off	on
22	on	off	on	on	off	on
23	off	off	on	on	off	on
24	on	on	off	on	off	on
25	off	on	off	on	off	on
26	on	off	off	on	off	on
27	off	off	off	on	off	on
30	on	on	on	off	off	on
31	off	on	on	off	off	on
32	on	off	on	off	off	on
33	off	off	on	off	off	on
34	on	on	off	off	off	on
35	off	on	off	off	off	on
36	on	off	off	off	off	on
37	off	off	off	off	off	on

DH+ Station Number	Switch					
	1	2	3	4	5	6
40	on	on	on	on	on	off
41	off	on	on	on	on	off
42	on	off	on	on	on	off
43	off	off	on	on	on	off
44	on	on	off	on	on	off
45	off	on	off	on	on	off
46	on	off	off	on	on	off
47	off	off	off	on	on	off
50	on	on	on	off	on	off
51	off	on	on	off	on	off
52	on	off	on	off	on	off
53	off	off	on	off	on	off
54	on	on	off	off	on	off
55	off	on	off	off	on	off
56	on	off	off	off	on	off
57	off	off	off	off	on	off
60	on	on	on	on	off	off
61	off	on	on	on	off	off
62	on	off	on	on	off	off
63	off	off	on	on	off	off
64	on	on	off	on	off	off
65	off	on	off	on	off	off
66	on	off	off	on	off	off
67	off	off	off	on	off	off
70	on	on	on	off	off	off
71	off	on	on	off	off	off
72	on	off	on	off	off	off
73	off	off	on	off	off	off
74	on	on	off	off	off	off
75	off	on	off	off	off	off
76	on	off	off	off	off	off
77	off	off	off	off	off	off

Switch 2

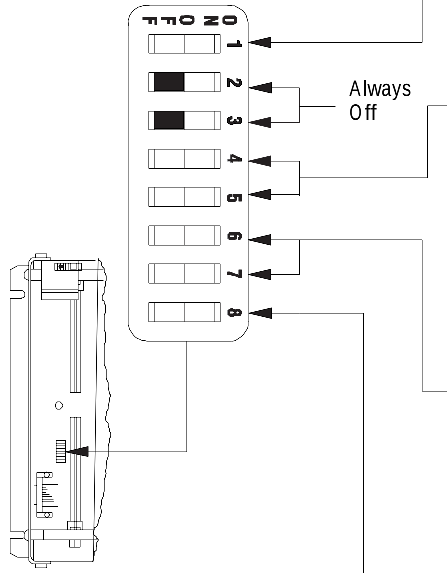

Bottom View of PLC-5/11, -5/20, -5/26, and -5/20E processors Switch Assembly SW2
Front of Processor
12 3 4 5 6 7 8 9 10


Bottom View of PLC-5/30, -5/40, -5/46 -5/40L, -5/60, -5/60L, -5/80, -5/86, -5/40E, and -5/80E processors Switch Assembly SW2
Front of Processor
12 3 4 5 6 7 8 9 10
Side View
toggle pushed toward bottom
on
toggle pushed toward top
off

To Specify:	Set Switches:									
	1	2	3	4	5	6	7	8	9	10
RS-232C	on	on	on	off	off	on	on	off	on	off
RS-422A	off	off	on	off	off	off	off	off	on	off
RS-423	on	on	on	off	off	on	off	off	on	off

I/O Chassis Backplane

PLC-5 Processor in the I/O Chassis



Switch 1	Last State
on	Outputs of this I/O chassis remain in their last state when a hardware failure occurs. ①
off	Outputs of this I/O chassis are turned off when a hardware failure occurs. ①

Switches		Addressing
4	5	
off	off	2 - slot
off	on	1 - slot
on	off	1/2 - slot
on	on	Not allowed

Switches		Memory Module Transfer
6	7	
off	off	memory module transfer to processor memory at powerup. ② ③
on	on	memory module transfers to processor memory if processor memory not valid.
on	off	memory module does not transfer to processor memory.

Switch 8	Processor Memory Protection
off	Processor memory protection disabled.
on	Processor memory protection enabled. ④

◻■	Pressed in at top ON (closed)
■◻	Pressed in at bottom OFF (open)

① Regardless of this switch setting, outputs are turned off when any of the following occurs:
- processor detects a major fault
- an I/O chassis backplane fault occurs
- you select program or test mode
- you set a status file bit to reset a local rack

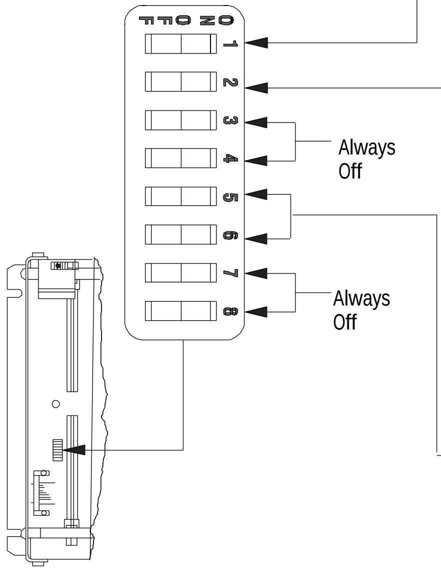
② If a memory module is not installed and processor memory is valid, the processor's PROC LED indicator blinks, and the processor sets S:11/9 in the major fault status word. Power down the processor chassis and either install the correct memory module or set switch 6 ON.

③ If the processor's keyswitch is set in REMote, the processor enters remote RUN after it powers up and has its memory updated by the memory module.

④ You cannot clear processor memory when this switch is on.

19309

1771-ASB Remote I/O Adapter or 1771-ALX Extended-Local I/O Adapter



Switch 1	Last State
on	Outputs of this I/O chassis remain in their last state when a communication fault is detected by this I/O adapter. [1]
off	Outputs of this I/O chassis are turned off when a communication fault is detected by this I/O adapter.

Switch 2	Processor Restart Lockout
on	Processor can restart the I/O chassis after a communication fault. [2]
off	You must manually restart the I/O chassis with a switch wired to the 1771-AS or -ASB.

Switches		Addressing
5	6	
off	off	2-slot
on	off	1-slot [3]
off	on	1/2-slot [3]
on	on	Not allowed

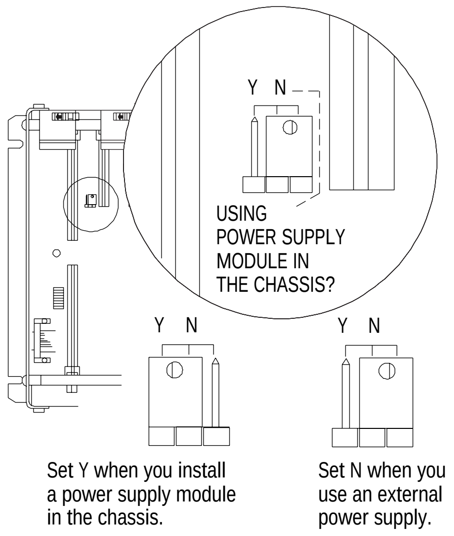
□■	Pressed in at top ON (closed)
■□	Pressed in at bottom OFF (open)

19308

1. **ATTENTION:** If you set this switch to the ON position, when a communication fault is detected, putputs connected to this chassis remain in their last state to allow machine motion to continue. We recommend that you set switch 1 to the OFF position to de-energize outputs wired to this chassis when a fault is detected.

   Also, if outputs are controlled by inputs in a different rack and a remote I/O rack fault occurs (in the inputs rack), the inputs are left in their last non-faulted state. The outputs may not be properly controlled and potential personnel and machine damage may result. If you want your inputs to be anything other than their last non-faulted state, then you need to program a fault routine.
2. Set this switch to ON if you plan to use I/O rack auto-configuration.
3. The 1771-ASB series A adapter does not support 1/2-slot addressing.

I/O Chassis Configuration Plug

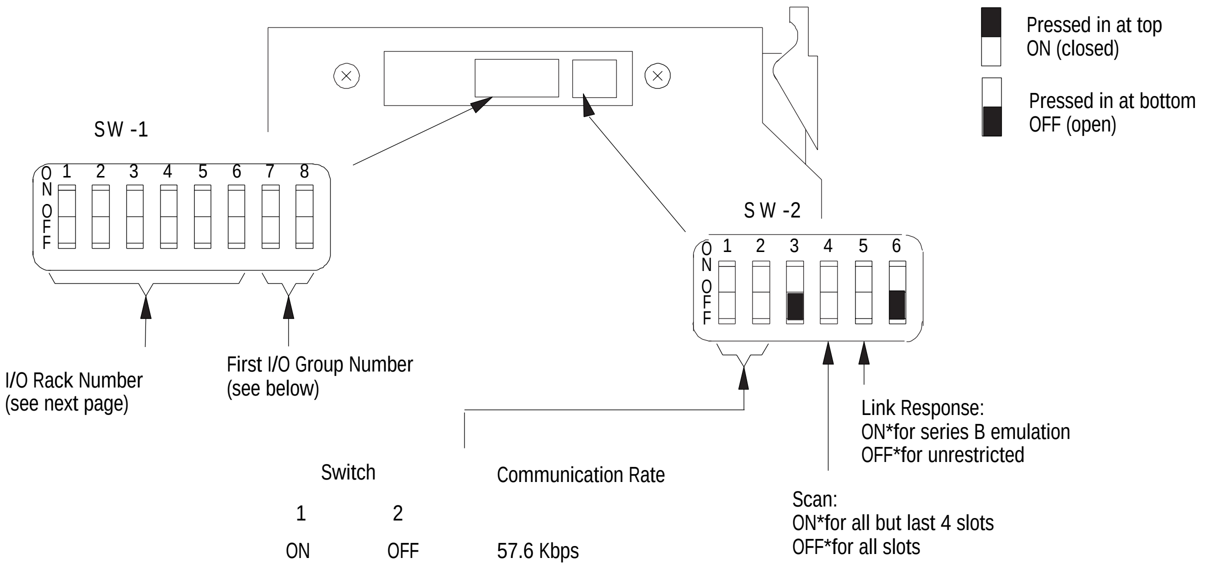


1. Locate the chassis configuration plug (between the first two left-most slots of the chassis).
2. Set the I/O chassis configuration plug.
   The default setting is N (not using a power supply module in the chassis).

Important: You cannot power a single I/O chassis with both a power supply module and an external power supply.

17075

Remote I/O Adapter Module (1771-ASB Series C and D) without Complementary I/O



Switch 1	Switch 2	Communication Rate
ON	OFF	57.6 Kbps
OFF	OFF	115.2 Kbps
OFF	ON	230.4 Kbps
ON	ON	Not used

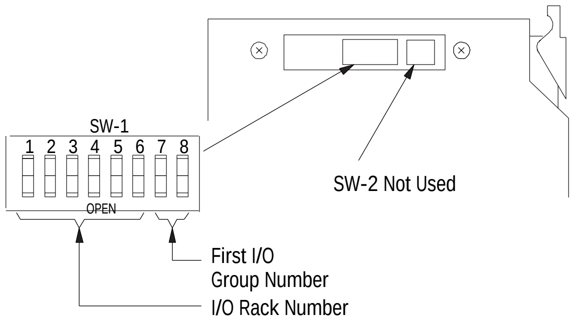
First I/O Group Number:	**7**	**8**
0	on	on
2	on	off
4	off	on
6	off	off

(1771-ASB Series C and D) I/O Rack Number—without Complementary I/O

Rack	1	2	3	4	5	6
01	on	on	on	on	on	off
02	on	on	on	on	off	on
03	on	on	on	on	off	off
04	on	on	on	off	on	on
05	on	on	on	off	on	off
06	on	on	on	off	off	on
07	on	on	on	off	off	off
10	on	on	off	on	on	on
11	on	on	off	on	on	off
12	on	on	off	on	off	on
13	on	on	off	on	off	off
14	on	on	off	off	on	on
15	on	on	off	off	on	off
16	on	on	off	off	off	on
17	on	on	off	off	off	off
20	on	off	on	on	on	on
21	on	off	on	on	on	off
22	on	off	on	on	off	on
23	on	off	on	on	off	off
24	on	off	on	off	on	on
25	on	off	on	off	on	off
26	on	off	on	off	off	on
27	on	off	on	off	off	off

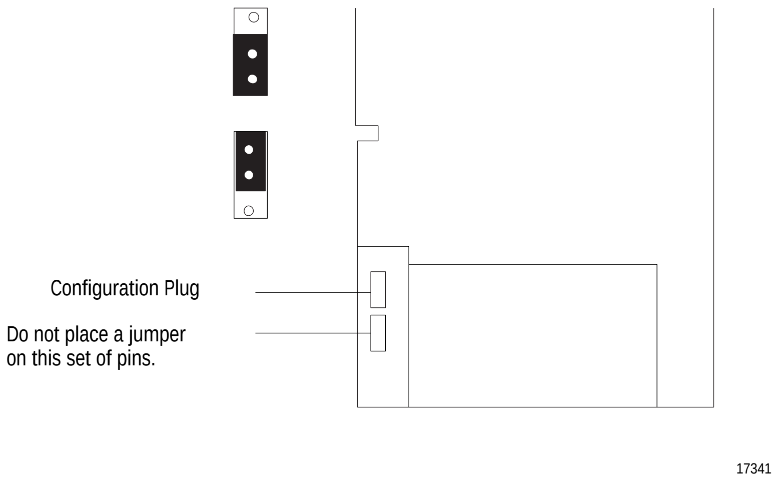
Extended-Local I/O Adapter Module

(1771-ALX) Switch SW1



Rack:	1	2	3	4	5	6
01	on	on	on	on	on	off
02	on	on	on	on	off	on
03	on	on	on	on	off	off
04	on	on	on	off	on	on
05	on	on	on	off	on	off
06	on	on	on	off	off	on
07	on	on	on	off	off	off
10	on	on	off	on	on	on
11	on	on	off	on	on	off
12	on	on	off	on	off	on
13	on	on	off	on	off	off
14	on	on	off	off	on	on
15	on	on	off	off	on	off
16	on	on	off	off	off	on
17	on	on	off	off	off	off
20	on	off	on	on	on	on
21	on	off	on	on	on	off
22	on	off	on	on	off	on
23	on	off	on	on	off	off
24	on	off	on	off	on	on
25	on	off	on	off	on	off
26	on	off	on	off	off	on
27	on	off	on	off	off	off

(1771-ALX) Configuration Plug



1. Lay the module on its right side.

   The configuration plugs are visible on the lower rear of the module.
2. Set the configuration plug as shown below according to your application.

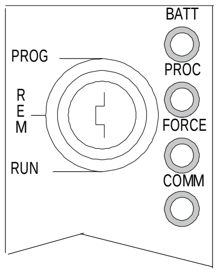
If you are using:	But Not:	Set Configuration Plug:
32-point I/O modules and any address method	1771-IX or 1771-IY	on the 2 lower pins
1771-IX and 1771-IY modules and any addressing method	32-point I/O modules	on the 2 upper pins

Troubleshooting

Using This Chapter

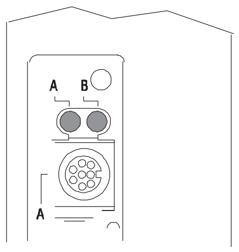
PLC-5 Processor

General Problems



Indicator	Color	Description	Probable Cause	Recommended Action
BATT	Red	Battery low	Battery low	Replace battery within 10 days
	Off	Battery is good	Normal operation	No action required
PROC	Green (steady)	Processor is in run mode and fully operational	Normal operation	No action required
	Green (blinking)	Processor memory is being transferred to memory module		
	Red (blinking)	Major fault	Major fault	Check major fault bit in status file (S:11) for error definition Clear fault bit, correct problem, and return to Run mode
	Red (steady)	Hardware fault	• Processor memory has checksum error • Memory module error • Internal diagnostics have failed	• Clear memory and reload program • Check backplane switch settings and/or insert correct memory module • Power down, reseat processor and power up; then, clear memory and reload your program. Replace memory module with new program; then, if necessary, replace the processor
	Off	Processor is in program load or test mode or is not receiving power	▒	Check power supply and connections
FORCE	Amber (steady)	SFC, I/O, and/or extended forces enabled	Normal operation	No action required
	Amber (blinking)	SFC, I/O, and/or extended forces present but not enabled		
	Off	SFC, I/O, and/or extended forces not present		
COMM	Off	No transmission on channel 0	Normal operation if channel is not being used	
	Green (blinking)	Transmission on channel 0	Normal operation if channel is being used	

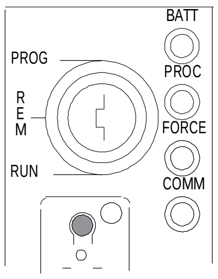
Processor Communication Channel Troubleshooting



Indicator	Color	Channel Mode	Description	Probable Cause	Recommended Action
A or B	Green (steady)	Remote I/O Scanner	Active Remote I/O link, all adapter modules are present and not faulted	Normal operation	No action required
		Remote I/O Adapter	Communicating with scanner		
		DH+	Processor is transmitting or receiving on DH+ link		
	Green (blinking rapidly or slowly)	Remote I/O Scanner	At least one adapter is faulted or has failed	• Power off at remote rack • Cable broken	• Restore power to the rack • Repair cable
		DH+	No other nodes on network		
	Red (steady)	Remote I/O Scanner Remote I/O Adapter DH+	Hardware fault	Hardware error	Turn power off, then on Check that the software configurations match the hardware set-up Replace the processor.
	Red (blinking rapidly or slowly)	Remote I/O Scanner	All adapters faulted	• Cable not connected or broken • Power off at remote racks	• Repair cable • Restore power to racks
		DH+	Bad communication on DH+	Duplicate node detected	Correct station address
	Off	Remote I/O Scanner Remote I/O Adapter DH+	Channel offline	Channel is not being used	Place channel online if needed

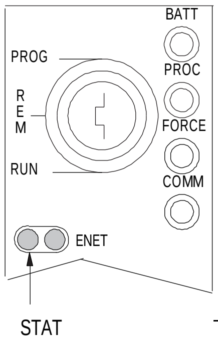
Extended-Local I/O Troubleshooting

PLC-5/40L and -5/60L processors only

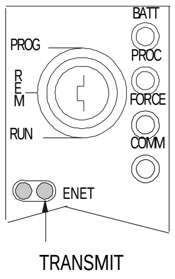


Indicator	Color	Channel Mode	Description	Probable Cause	Recommended Action
2	green (steady)	Extended local I/O Scanner	active extended-local I/O link, all adapter modules are present and not faulted	normal operation	no action required
	green (blinking rapidly or slowly)		at least one adapter is faulted or has failed	• power off at extended-local I/O rack • communication fault • cable broken	• restore power to the rack • restart adapters using the processor restart lockout pushbutton • repair cable
	red (steady)		hardware fault	hardware error	Turn power off, then on. Check that the software configurations match the hardware set-up. Replace the processor.
	red (blinking rapidly or slowly)	Extended local I/O Scanner	all adapters faulted	• cable disconnected or broken • terminator off • power off at extended-local racks	• repair cable • replace or repair terminator • restore power to racks
	off		channel offline	channel is not being used	Place channel online if needed

Ethernet Status Indicator



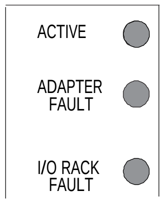
Indicator	Color	Description	Probable Cause	Recommended Action
STAT	Solid red	Critical hardware fault	Processor requires internal repair	Contact your local Allen-Bradley representative
	Blinking red	Hardware or software fault (detected and reported via a code)	Fault-code dependent	Contact Allen-Bradley's Global Technical Support (GTS)
	Off	Ethernet interface is functioning properly but it is not attached to an active Ethernet network	Normal operation	Attach the processor to an active Ethernet network
	Green	Ethernet channel 2 is functioning properly and has detected that it is connected to an active Ethernet network	Normal operation	No action required



Ethernet Transmit LED

The PLC-5 Ethernet interface contains an Ethernet Transmit LED that lights (green) briefly when the Ethernet port is transmitting a packet. It does not indicate whether or not the Ethernet port is receiving a packet.

Remote I/O System



Troubleshooting Guide for the 1771-ASB Series C and D Adapter Module

Indicators			Description	Probable Cause	Recommended Action
Active	Adapter Fault	I/O Rack			
On	Off	Off	Normal indication; remote adapter is fully operational		
Off	On	Off		RAM memory fault, watchdog timeout	Replace module.
On	Blink	Off	Module placement error	I/O module in incorrect slot.	Place module in correct slot in chassis.
Blink in unison		Off	Incorrect starting I/O group number	Error in starting I/O group number or I/O rack address	Check switch settings.
On	On	On	Module not communicating	Incorrect transmission rate setting	
Off	On	On	Module not communicating	Scan switch set for “all but last four slots” in 1/4 rack	Reset scan switch setting.
Blink	Off	Off	Remote adapter not actively controlling I/O (scanner to adapter communication link is normal)[1]	Processor is in program or test mode Scanner is holding adapter module in fault mode	Fault should be cleared by I/O scanner.
LEDs sequence on/off from top to bottom			Module not communicating	Another remote I/O adapter with the same address is on the link.	Correct the address.
Blink alternately		Off	Adapter module not actively controlling I/O[2] Adapter module in processor restart lockout mode (adapter to scanner link is normal)	Processor restart lockout switch on chassis backplane switch assembly on[3]	Press reset button to clear lockout feature or cycle power; if after repeated attempts indicators are still blinking, check: • push button not wired properly to field wiring arm • wiring arm not connected to adapter module • adapter module was reset by process or/ scanner, then immediately faulted

1.If a fault occurs and the processor is in the run mode but is actually operating in the dependent mode, the chassis fault response mode is selected by the last state switch on the chassis backplane.
2.¡The I/O chassis is in faulted mode as selected by the last state switch on the chassis backplane.
3.You must select the operating mode of the remote I/O adapter module as outlined in the publication furnished with the remote I/O scanner/distribution panel, remote I/O scanner-program interface module, or I/O scanner-message handling module. Pay close attention to the disable search mode in the 1771-SD, -SD2.

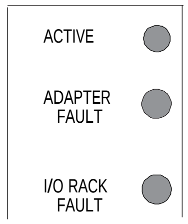
Troubleshooting Guide for the 1771-ASB Series C and D Adapter Module (continued)

Indicators			Description	Probable Cause	Recommended Action
Active	**Adapter Fault**	**I/O Rack**			
Off	Off	On	I/O chassis fault. [1] No communication on link.	Problem exists between: • adapter and module in chassis; the module will stay in fault mode until fault is corrected • shorted printed circuit board runs on backplane or I/O module	Cycle power to the chassis to clear a problem resulting from high noise.[2] • Remove and replace all I/O modules one at a time • If problem does not clear, something is wrong in chassis or I/O module
Blink	Off	On	Communication on link. Possible shorted backplane	• Noise on backplane • Shorted circuit board runs • Faulty card in chassis	• Eliminate noise • Isolate noise • Add surge suppression • Replace chassis • Replace defective card in chassis
Blink	On	Off	Module identification line fault	Excessive noise on backplane	Verify power supply and chassis grounding.
Off	Off	Off	Module not communicating	Power supply fault	Check power supply, cable connections, and make sure adapter module is fully seated in chassis.
				Wiring from scanner to adapter module disrupted	Correct cable and wiring defects
				Scanner not configured properly	See publication 1772-2.18 for scanner configuration.
				One faulted chassis within a rack group address causing scanner/distribution panel to fault all chassis in rack group address (when in disable search mode)	Check sequentially from the first module to the last module to pinpoint fault; correct any faults and proceed to the next chassis.

1.¡The I/O chassis is in faulted mode as selected by the last state switch on the chassis backplane.
2.Cycling power clears block-transfer request queue. All pending block transfers are lost. Your program must repeat the request for block transfers.

Extended-Local I/O System

Troubleshooting Guide for the 1771-ALX Adapter Module



Indicators			Description	Probable Cause	Recommended Action
Active	Adapter Fault	I/O Rack			
On	Off	Off	Normal indication; remote adapter is fully operational		
Off	On	Off	Local adapter fault [1]	Local adapter not operating; it will stay in fault mode until fault is corrected	Cycle power to the chassis to clear the adapter fault.[2] Replace adapter if fault does not clear.
Off	Off	On	I/O chassis fault [1]	Problem exists between: • adapter and module in chassis; the module will stay in fault mode until fault is corrected • shorted printed circuit board runs on backplane or I/O module	Cycle power to the chassis to clear a problem resulting from high noise.[2] • remove and replace all I/O modules one at a time • replace adapter • If problem does not clear, something is wrong in chassis or I/O module
Blinking	Off	Off	Outputs are reset	Processor is in program or test mode Local I/O Scanner is holding adapter module in fault mode	None Fault should be cleared by extended-local I/O scanner.
Blinking alternately		Off	Adapter module not actively controlling I/O [1] Adapter module in processor restart lockout mode (adapter to scanner link is normal)	Processor restart lockout switch on chassis backplane switch assembly on[3]	Press chassis reset button to clear lockout feature or cycle power; if after repeated attempts indicators are still blinking, check that adapter module was reset by processor/scanner, then immediately faulted.
Off	Off	Off	No power or no communication.	Power supply fault	Check power supply, I/O cable and power supply cable connections, and make sure adapter module is fully seated in chassis.
On	Blinking	Off	Module placement error in extended-local I/O chassis	Incorrect placement of high-density modules	Verify addressing modes and switch settings.

1.Cycling power clears the block-transfer request queue. All pending block transfers are lost. Your program must repeat the request for block transfers from the chassis.
2.If a fault occurs and the processor is in the run mode but is actually operating in the dependent mode, the chassis fault response mode is selected by switch 1 (the last state switch) on the chassis backplane.
3.The I/O chassis is in faulted mode as selected by switch 1 (the last stare switch) on the chassis backplane.

Unexpected Operation when Entering Run Mode

If unexpected operation occurs whenever your processor enters run mode, be sure to examine the prescan operation of the instructions in this section. These instructions execute differently during prescan than they do during a normal scan.

The ***prescan*** function is an intermediate scan between the transition from program to run modes, during which all rungs are scanned as false. The prescan examines all ladder program files and instructions and initializes the data table based on the results of the program.

For example, a subroutine that is called infrequently may contain a bad indirect address and generate a major fault. However, many normal program scans may occur before the major fault is actually generated. Prescan provides the opportunity for the processor to examine the program for errors such as this *before* transitioning to Run mode.

Instructions with Unique Prescan Operations

Use the table below to track prescan operations that deviate from normal instruction operation.

Table 24.A
Instruction Operation During Prescan

This instruction:	Executes these actions during prescan:
ARD ARL AWT AWA ACB ABL AHL	If the EN bit is set and the DN and ER bits are cleared, then the control word is cleared. If either the DN or ER bit is set, then the EN bit is cleared and the DN bit is set.
BTR BTW	All non-user configuration bits 15, 14, 13, 12, 10, and 9 are cleared (for both INT and BT file types).
CTU CTD	The CU/CD bit is set to prevent a false count when the first run-mode scan begins.
EOT	This instruction is skipped so all ladder instructions can be prescanned.
FFL LFL	The EL bit is set to prevent a false load when the first run-mode scan begins.
FFU LFU	The EU bit is set to prevent a false unload when the first run-mode scan begins.
FND	This instruction is skipped so all ladder instructions can be prescanned.
FOR	Ladder instructions within the FOR/NXT loop **are** prescanned.
MSG	If the SFC startover bit is cleared and the CO bit is cleared, then all non-user configuration bits 15, 14, 13, 12, 10, and 9 are cleared in both the INT and MG file types. The MG file type also clears bits 11, 7, 6, 5, 4, 2, 1, and 0.

This instruction:	Executes these actions during prescan:
ONS	The programmed bit address of the instruction is set to inhibit false triggering when the first run-mode scan begins.
OSF	The programmed bit address of the instruction is cleared to inhibit false triggering when the first run-mode scan begins. The output bit is also cleared.
OSR	
PID	For PD file type, the INI bit is cleared. INT file type clears status bits 8, 9, and 10 (deadband, upper, and lower output alarm). The error register from the previous scan is set to 32767, which indicates that the setpoint and ER bits from previous scans have not yet been initialized). The Integral Accumulator and Derivative Error bits are cleared.
SQL	The EN bit is set to prevent a false increment of the table pointer when the first run-mode scan occurs.
SQO	
TOF	The TT, TC, TE, and TO bits are cleared and the ACC = preset.
DTR [1]	The reference value is updated (regardless of the rung condition).

1.The DTR instruction operates in this manner during a normal scan as well.

Suggested Action

To avoid unexpected operation that may result from these prescan activities, follow these guidelines:

- Do not use indexed or indirect addressing with the instructions listed in Table 24.A.

  If you *must* use indexed or indirect addressing, use the first scan bit (S:1/15) to pre-initialize all of the other used variables.

- If using indirect addressing with any ladder instructions, do not use the data variable holding the indirect address for multiple functions.

Notes:

Chapter **25**

Cable Reference

Using This Chapter

For information about:	Go to page:
Pin assignments for the processor’s channel 0	25-1
Serial cable pin assignments	25-2
Cable connection diagrams	25-3
Programming cable specification	25-5
Ethernet cable connections	25-9

Channel 0 Pin Assignments

The side label of the processor shows a table listing channel 0 (RS-port) pin assignments. This table shows the same information:

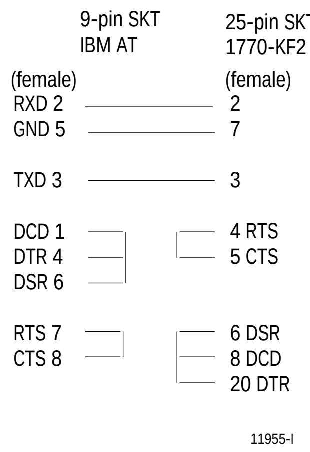
Pin	RS-232C	RS-422A	RS-423	Pin	RS-232C	RS-422A	RS-423
1	C.GND	C.GND	C.GND	14	NOT USED	TXD.OUT^{-}	SEND COM
2	TXD.OUT	TXD.OUT^{+}	TXD.OUT	15	▨	▨	▨
3	RXD.IN	RXD.IN^{+}	RXD.IN	16	NOT USED	RXD.IN^{-}	REC COM
4	RTS.OUT	RTS.OUT^{+}	RTS.OUT	17	▨	▨	▨
5	CTS.IN	CTS.IN^{+}	CTS.IN	18	▨	▨	▨
6	DSR.IN	DSR.IN	DSR.IN	19	NOT USED	RTS.OUT^{-}	NOT USED
7	SIG.GND	SIG.GND	SIG.GND	20	DTR.OUT	DTR.OUT	DTR.OUT
8	DCD.IN	DCD.IN	DCD.IN	21	▨	▨	▨
9	▨	▨	▨	22	NOT USED	DSR.IN	NOT USED
10	NOT USED	DCD.IN	NOT USED	23	NOT USED	DTR.OUT	NOT USED
11	▨	▨	▨	24	▨	▨	▨
12	▨	▨	▨	25	▨	▨	▨
13	NOT USED	CTS.IN^{-}	NOT USED				

The shading indicates that the pin is reserved.

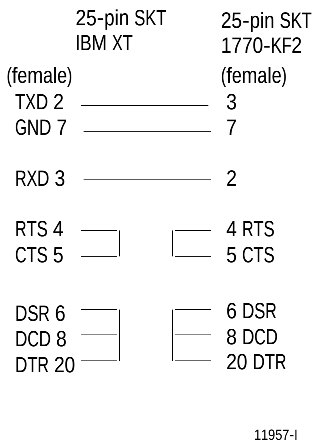
Serial Cable Pin Assignments

The following diagrams show the pin assignments for the cables you need for serial port communications.

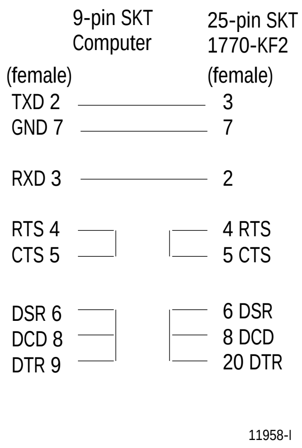
Cable #1



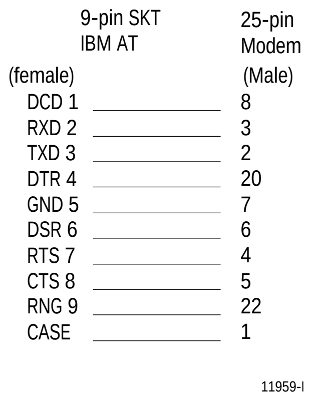
Cable #2



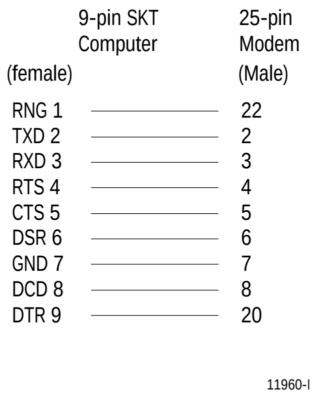
Cable #3



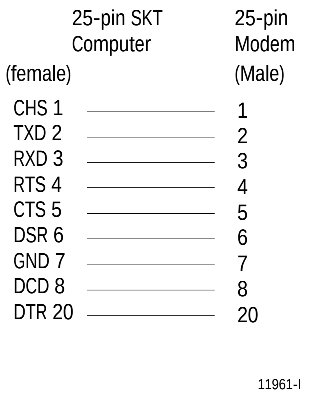
Cable #4



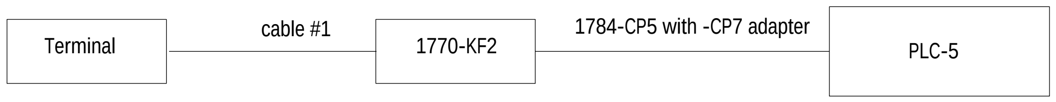
Cable #5



Cable #6

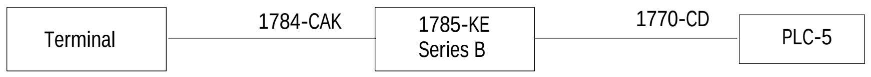


Connecting Diagrams

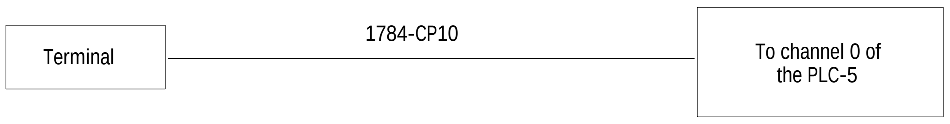


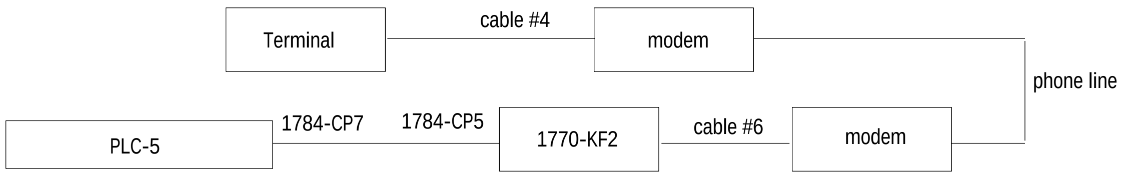
9-Pin Serial Port

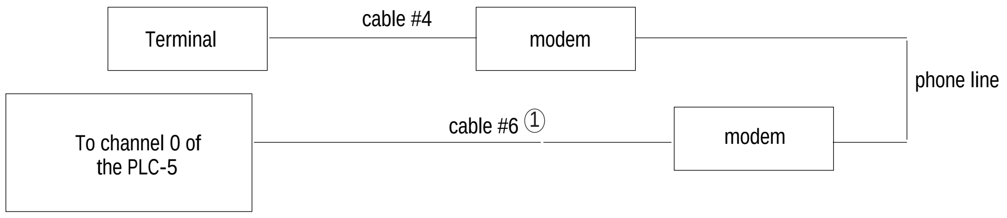
1784-T50
6160-T53
6160-T60
6160-T70
IBM PC/AT



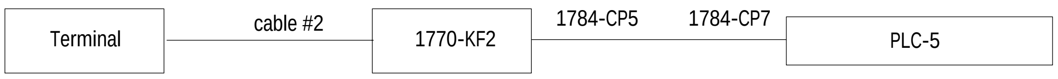
Note: 1785-KE Series A uses 1785-CP5 cable and 1785-CP7 adapter with the enhanced and Ethernet PLC-5 processors





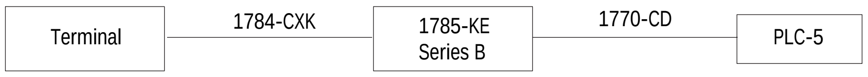


① Requires either a gender changer or one end of cable #2 fitted with a male 25-pin plug.

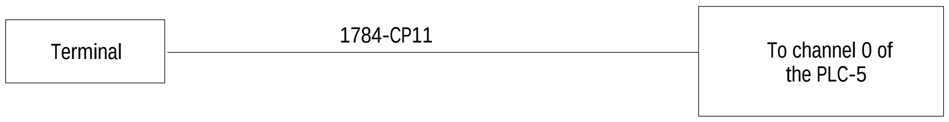


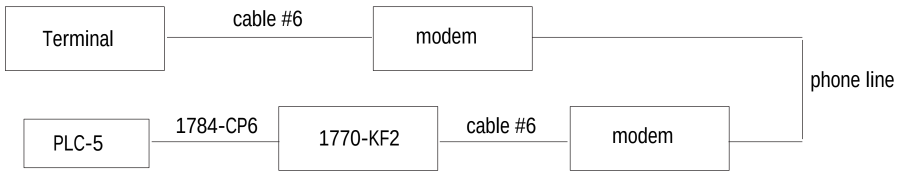
25-Pin Serial Port

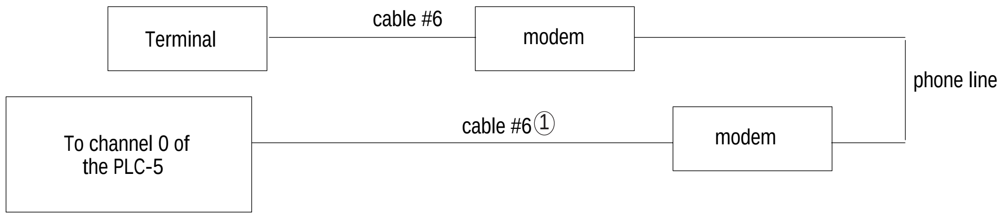
1784-T47
IBM XT
IBM PS/2 Model 30
IBM PS/2 Model 60



Note: 1785-KE Series A uses 1785-CP5 cable and 1785-CP7 adapter with the enhanced and Ethernet PLC-5 processors







① Requires either a gender changer or one end of cable #2 fitted with a male 25-pin plug.

Programming Cable Specifications

The specifications for each Allen-Bradley cable used for DH+ communications are shown on the following pages. See Table 25.A.

Table 25.A
Programming Cable Specifications

For:	To:	Use this cable:	See page:
6160-T53 6160-T60 6160-T70 IBM PC/AT	1785-KE	1784-CAK	25-5
enhanced or Ethernet PLC-5 processor	Terminal (using a 1784-KT, -KT2, -KL, or -KL/B)	1784-CP6 1784-CP with a 1784-CP7 adapter 1784-CP8 adapter	25-6 25-6 25-7
	Terminal (using a 1784-KTK1)	1784-CP5 with a 1785-CP7 adapter	25-6
	Terminal (using a 9-pin serial cable)	1784-CP10	25-7
	Terminal (using a serial 25-pin cable)	1784-CP11	25-8
	Terminal (using a 1784-PCMK)	1784-PCM5 with a 1784-CP7 adapter	25-8 and 25-6

Figure 25.1
Cable—1784-CAK
Connects 1785-KE to 6160-T53, 6160-T60, 6160-T70, or IBM PC/AT

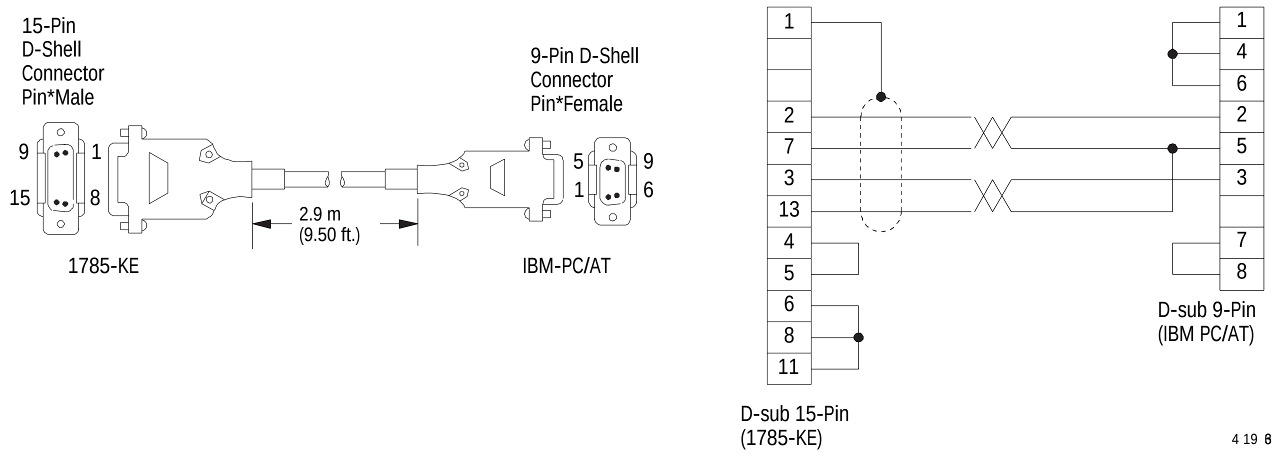


Figure 25.2
Cable—1784-CP6
Connects Terminal Using 1784-KT, -KT/2, -KL, or -KL/B to Processor

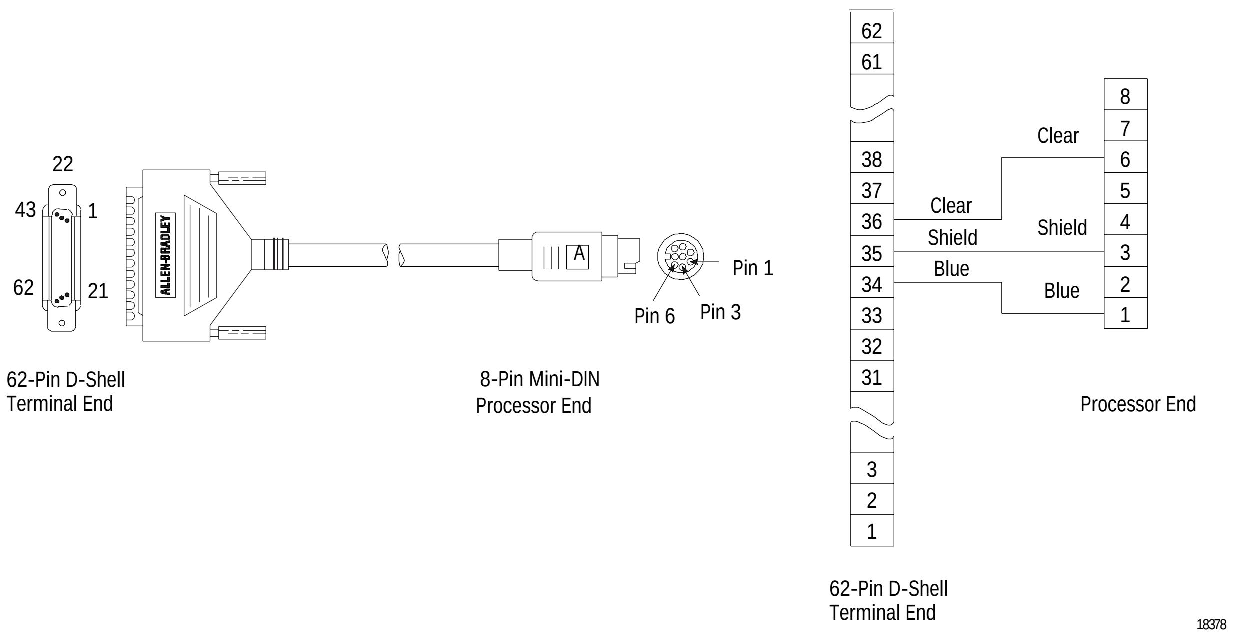


Figure 25.3
Cable and Adapter—1784-CP7 Connects to Processor via a a 9-pin D-Shell of a 1784-CP, -CP5, or -PCM5 cable

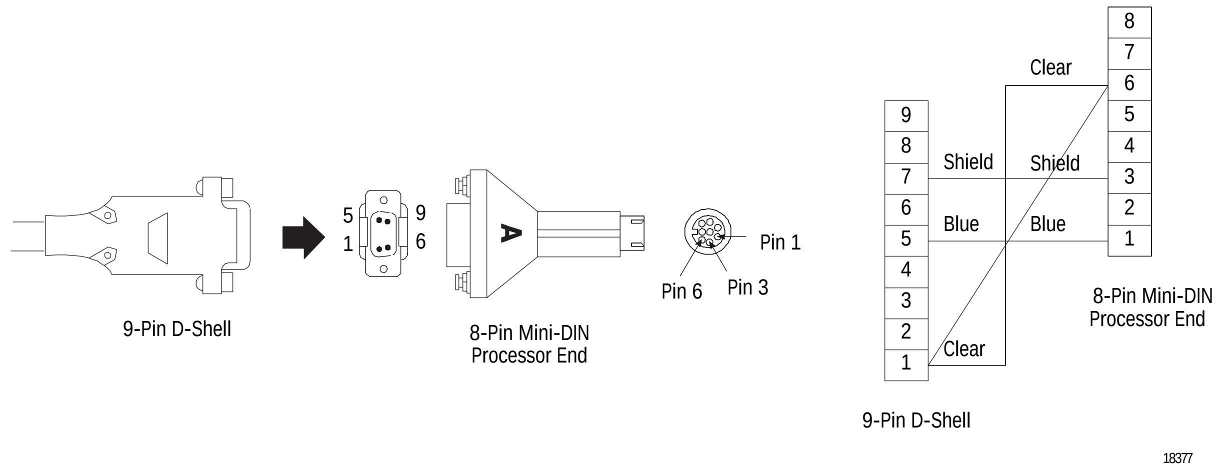


Figure 25.4
Cable Adapter—1784-CP8 Connects a Terminal Using a 1784-KT, -KT2, or -KL Card to a Permanent DH+ Network

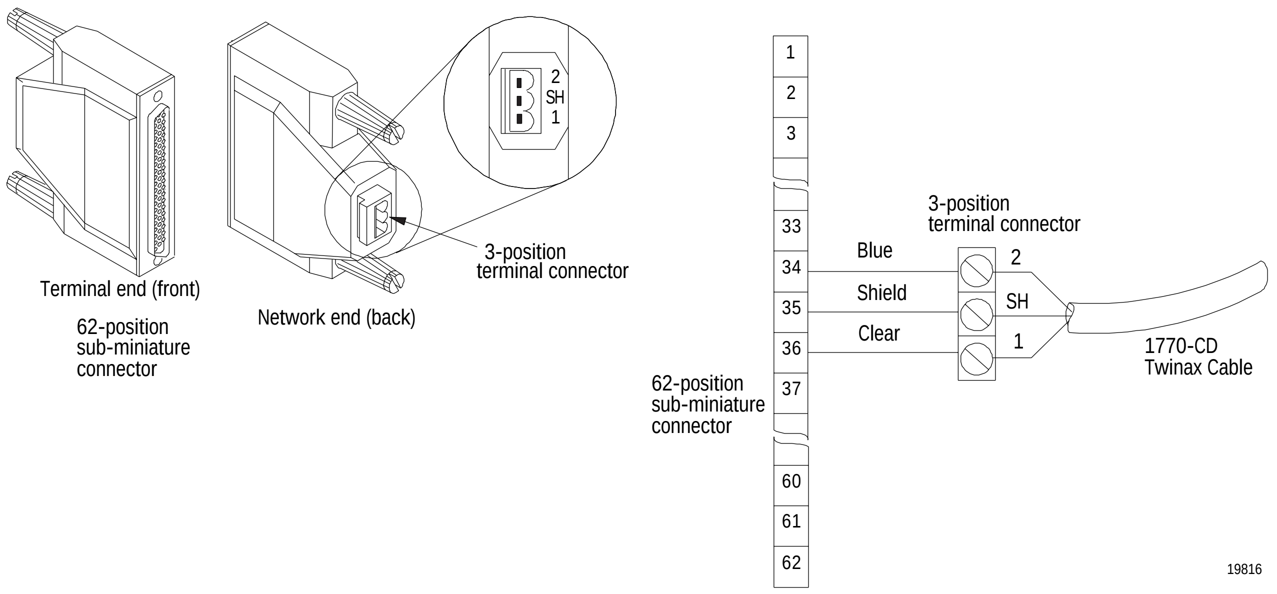

2
SH
1
3-position terminal connector
Terminal end (front)
Network end (back)
62-position sub-miniature connector
1
2
3
33
34
35
36
37
60
61
62
Blue
Shield
Clear
3-position terminal connector
2
SH
1
1770-CD Twinax Cable
62-position sub-miniature connector
19816

Figure 25.5
Cable—1784-CP10
Connects Terminal to Processor Using Serial Port

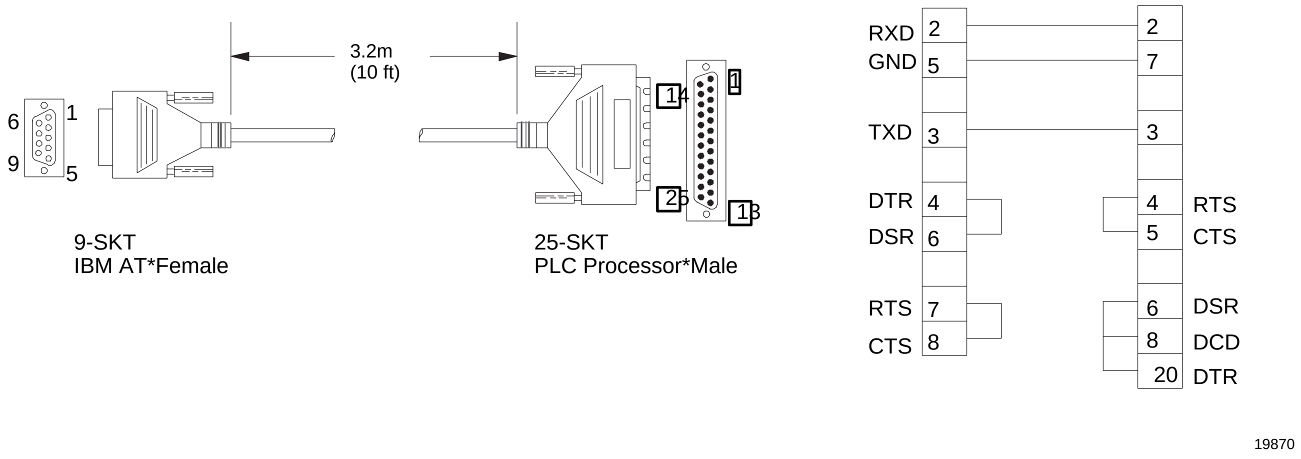


Figure 25.6
Cable—1784-CP11
Processor to Terminal Using a Serial Port

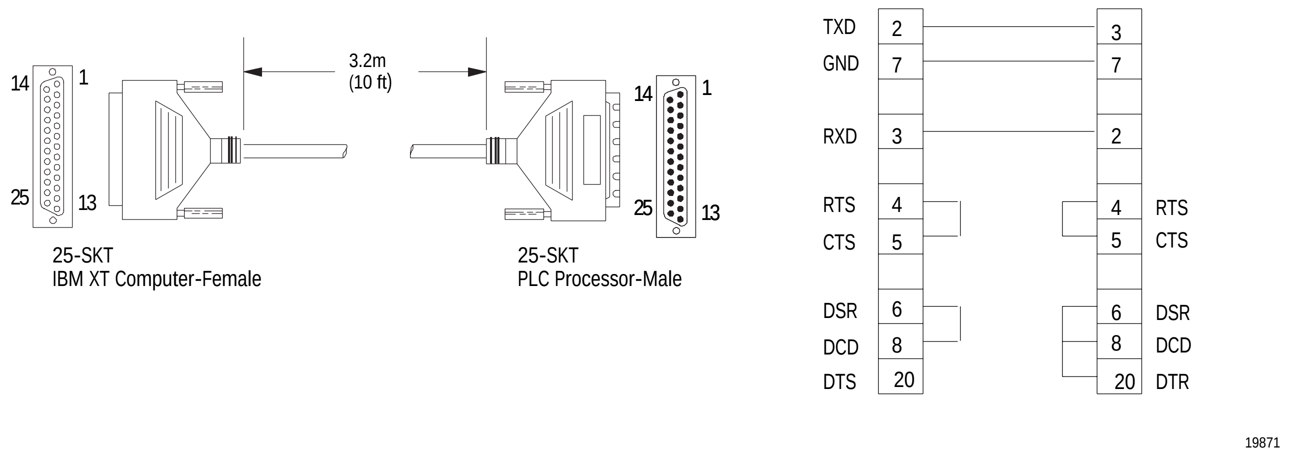
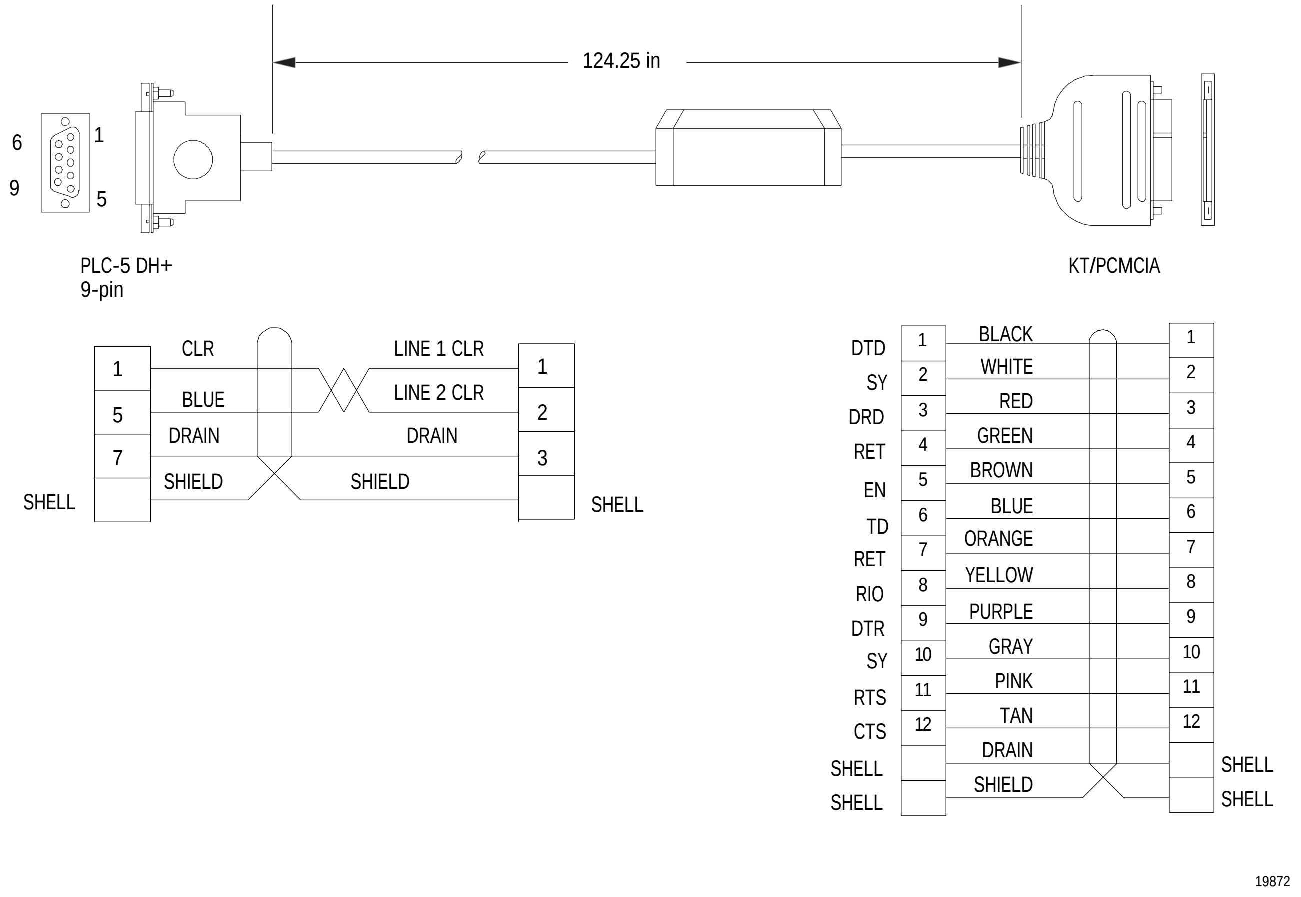
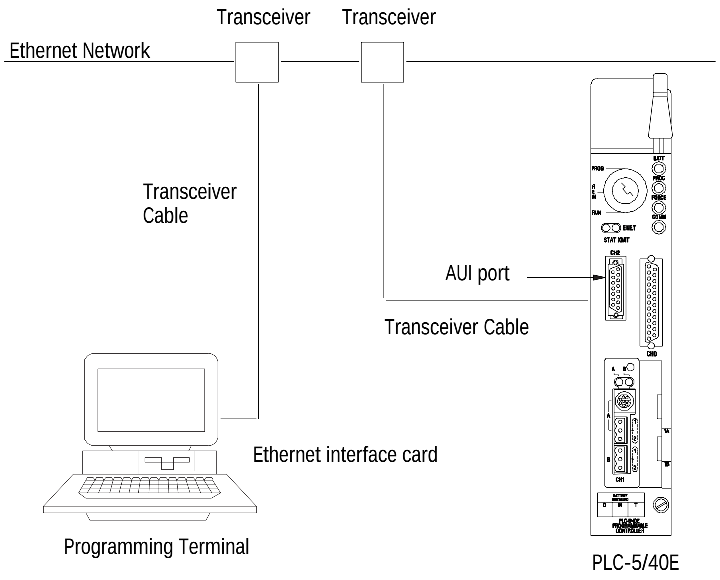


Figure 25.7
Cable - 1784-PCM5
Processor to Terminal (using a 1784-PCMK)



Ethernet Cable Connections

The Ethernet port connects to either a thin-wire or thick-wire network via a 15-pin transceiver or Medium Access Unit (MAU) connection.



To connect a programming terminal to a PLC-5/20E, -5/40E, or -5/80E processor through an Ethernet network, use the following:
- Ethernet PCMCIA or PC/AT-compatible (6628-A5) communication card
- Ethernet cable
- Transceivers and transceiver cables

Two types of Allen-Bradley transceivers are available.

Catalog Number:	Description:
5810-AXMT	Thin-wire Ethernet/802.3 transceiver
5810-AXMH	Thick-wire Ethernet/802.3 transceiver

The processor connects to the transceiver using a standard transceiver cable, which is also known as an Access Unit Interface (AUI) cable. Allen-Bradley has two lengths of transceiver cables and four kits consisting of transceivers and cables.

Catalog Number:	Description:
5810-TER	Thinwire Ethernet terminating resistors
5810-TC02/A	Thick-wire 2.0 m (6.5 ft) transceiver cable
5810-TC15/A	Thick-wire 15.0 m (49.2 ft) transceiver cable
5810-TAS/A (kit)	Thin-wire transceiver and 2.0 m (6.5 ft) cable
5810-TAM/A (kit)	Thin-wire transceiver and 15.0 m (49.2 ft) cable
5810-TBS/A (kit)	Thick-wire transceiver and 2.0 m (6.5 ft) cable
5810-TBM/A (kit)	Thick-wire transceiver and 15.0 m (49.2 ft) cable

Connection to “10baseT” (fiber-optic) and broadband networks is also supported if you purchase the appropriate transceivers and cables from a third-party source.

Notes:

Allen-Bradley Publication Problem Report

If you find a problem with our documentation, please complete and return this form

Pub. Name **Enhanced and Ethernet PLC-5 Programmable Controllers User Manual**

Cat. No. **1785 series** Pub. No. **1785-6.5.12** Pub. Date **November 1998** Part No. **955133-82**

Check Problem(s) Type:	Describe Problem(s):			Internal Use Only
☐ Technical Accuracy	☐ text		☐ illustration	
☐ Completeness What information is missing?	☐ procedure/step ☐ example ☐ explanation	☐ illustration ☐ guideline ☐ other	☐ definition ☐ feature	☐ info in manual (accessibility) ☐ info not in manual
☐ Clarity What is unclear?				
☐ Sequence What is not in the right order?				
☐ Other Comments Use back for more comments.				

Your Name ______________________ Location/Phone ______________________

Return to: Marketing Communications, Allen-Bradley Co., 1 Allen-Bradley Drive, Mayfield Hts., OH 44124-6118 Phone: (440)646-3166
FAX: (440)646-4320

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE REMOVE

PLEASE FOLD HERE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Rockwell *Automation*

Allen-Bradley

1 ALLEN BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

1785-6.5.12 November 1998
Supersedes 1785-6.5.12 January 1997

PN 955133-82