GE
Intelligent Platforms

Programmable Control Products

Series 90-70*
Genius Bus Controller

User's Manual

GFK-2017 *March 2010*

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note: Notes merely call attention to information that is especially significant to understanding and operating the equipment.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Intelligent Platforms assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Intelligent Platforms makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

Contact Information

If you purchased this product through an Authorized Channel Partner, please contact the seller directly.

General Contact Information

Online technical support and GlobalCare	http://www.ge-ip.com/support
Additional information	http://www.ge-ip.com/
Solution Provider	solutionprovider.ip@ge.com

Technical Support

If you have technical problems that cannot be resolved with the information in this guide, please contact us by telephone or email, or on the web at www.ge-ip.com/support

Americas

Online Technical Support	www.ge-ip.com/support
Phone	1-800-433-2682
International Americas Direct Dial	1-780-420-2010 (if toll free 800 option is unavailable)
Technical Support Email	support.ip@ge.com
Customer Care Email	customercare.ip@ge.com
Primary language of support	English

Europe, the Middle East, and Africa

Online Technical Support	www.ge-ip.com/support
Phone	+800-1-433-2682
EMEA Direct Dial	+352-26-722-780 (if toll free 800 option is unavailable or if dialing from a mobile telephone)
Technical Support Email	support.emea.ip@ge.com
Customer Care Email	customercare.emea.ip@ge.com
Primary languages of support	English, French, German, Italian, Czech, Spanish

Asia Pacific

Online Technical Support	www.ge-ip.com/support
Phone	+86-400-820-8208
	+86-21-3217-4826 (India, Indonesia, and Pakistan)
Technical Support Email	support.cn.ip@ge.com (China)
	support.jp.ip@ge.com (Japan)
	support.in.ip@ge.com (remaining Asia customers)
Customer Care Email	customercare.apo.ip@ge.com
	customercare.cn.ip@ge.com (China)

Contents

Chapter 1 Introduction 1-1

Product Documentation 1-1
Content of this Manual 1-1
Related Publications 1-2
System Overview 1-3
Number of Bus Controllers in a PLC 1-3
Bus Controller Description 1-4
Status LEDs 1-4
Hand-held Monitor Connector 1-4
Terminal Assembly 1-4
The Genius Bus 1-5
Bus Controller Operation 1-6
The Genius Bus Scan 1-6
Input Data from Devices on the Bus 1-7
Output Data from the CPU 1-7
Outputs for 4 Input/2 Output Analog Blocks 1-7
Diagnostics 1-9
Input Defaults if the Bus Controller is Lost 1-9
Datagrams 1-10
Global Data 1-11
Sending Global Data 1-11
Receiving Global Data 1-12

Chapter 2 Installation 2-1

Installing the Bus Controller 2-2
Removing the Bus Controller 2-2
Connecting the Serial Bus 2-3
Replacing an Older Bus Controller 2-3
Shield In and Shield Out Connections in an Existing Installation 2-3
Terminating the Bus 2-4
Wiring for Bus Continuity 2-4

Chapter 3 Bus Controller Configuration 3-1

Adding a Bus Controller to the Configuration 3-2
Configuring the Bus Controller Settings 3-3
Configuring Global Data 3-5
Configuring Genius Redundancy on the Redundancy Tab 3-6
Selecting the Redundancy Mode 3-6
Configuring Redundancy Parameters 3-9

Using the Redundancy Wizards 3-10
Redundancy Wizards 3-11
Redundancy Wizard: Set up a Primary Rack System for CPU Redundancy 3-12
Redundancy Wizard: Add GBCs for Genius Redundancy 3-14
Add GBCs for Genius Redundancy: Redundant Controllers, One PLC 3-17
Add GBCs for Genius Redundancy: Redundant Controllers, Two PLCs 3-19
Add GBCs for Genius Redundancy: Dual Bus and Redundant Controllers with Two PLCs 3-21
Redundancy Wizard: Generate Secondary Hardware Configuration from the Current Configuration 3-23
Adding Devices to the Bus Configuration 3-24
Configuring the Serial Bus Address of a Device 3-24
Configuring the Device Reference Address 3-25
Disabling Outputs 3-29
Bus Device Redundancy Configuration 3-31
Changing a Device's Serial Bus Address 3-32
Configuring a Remote Drop 3-33
Configuring a Remote Bus Controller, Genius Communications Module, PCIM, QBIM, or GENI-based Device on the Bus 3-34
Configuring a PowerTRAC Block or High-speed Counter Block 3-36
Configuring a Generic Device on the Bus 3-38
Copying a Bus Configuration 3-39
Copying a Bus Configuration without Copying the Bus Controller Parameters 3-39
Copying a Bus Configuration and the Bus Controller Configuration 3-39
Setting All Bus Devices to Redundant or Non-Redundant 3-40
Programmer Configuration of Bus Devices 3-41
Devices that Support Programmer Configuration Features 3-41
Programmer Instructions for Genius Devices 3-41
Viewing the Bus Controller Power Consumption 3-42

Chapter 4 Diagnostics 4-1

System Status References 4-2
Fault and No Fault Contacts 4-3
Fault Locating References 4-3
How FAULT and NO FAULT Contacts Handle Bus Controller Faults in an RX7i System 4-3
High Alarm and Low Alarm Contacts 4-4
Fault Table Definitions for Genius Devices 4-5
Clearing Faults in the Fault Table 4-5
Removing I/O Force Messages from the I/O Fault Table 4-5
Loss of Device Faults Caused by High Bus Error Rate 4-5

Fault Table Definitions Associated with Genius Devices 4-6

Chapter 5 Communication Requests 5-1

Programming for a Communication Request 5-2
COMMREQ Command Block Format 5-3
Command Block Contents 5-3
Command Block Quick Reference 5-5
The COMMREQ Instruction 5-6
COMMREQ Inputs and Outputs 5-6
COMMREQ Inputs 5-7
COMMREQ Outputs 5-7
COMMREQ Status Block 5-8
Clearing the Status Block 5-8
Contents of the Status Block 5-9
Programming Examples 5-10
COMMREQs and Datagrams 5-12
COMMREQ #1: Pulse Test Command 5-13
Command Block Format for the Pulse Test Command 5-13
COMMREQ #2: Read Configuration Command 5-14
Command Block for the Read Configuration Command 5-14
COMMREQ #3: Write Configuration Command 5-15
Command Block for the Write Configuration Command 5-15
COMMREQ #4: Read Diagnostics Command 5-16
Command Block for the Read Diagnostics Command 5-16
COMMREQ #5: Clear Circuit Faults Command 5-17
Command Block for Clear Circuit Faults 5-17
COMMREQ #6: Clear All Circuit Faults Command 5-17
Command Block for Clear All Circuit Faults 5-17
COMMREQ #7: Assign Monitor Command 5-18
Command Block for the Assign Monitor Command 5-19
COMMREQ #8: Enable/Disable Outputs Command 5-20
Command Block for the Enable/Disable Outputs Command 5-20
COMMREQ #9: Enable/Disable Global Data 5-21
Command Block for the Enable/Disable Global Data Command 5-21
COMMREQ #10: Switch BSM Command 5-22
Command Block for the Switch BSM Command 5-22
COMMREQ #11: Read Device Command 5-23
Command Block for the Read Device Command 5-23
Memory Specification for Read Device and Write Device 5-24

COMMREQ #12: Write Device Command 5-28
Using Write Device Messages Instead of Global Data 5-28
Command Block for the Write Device Command 5-28
COMMREQ #13: Dequeue Datagram Command 5-29
Command Block for the Dequeue Datagram Command 5-29
Number of Dequeue Datagram Commands Needed 5-30
Additional Logic for Incoming Datagrams 5-30
Format of Returned Data 5-31
COMMREQ #14: Send Datagram Command 5-32
Command Block for the Send Datagram Command 5-33
Datagram Priority 5-34
Datagrams and I/O Blocks 5-34
Number of Datagrams per CPU Sweep 5-34
COMMREQ #15: Request Datagram Reply Command 5-36
Command Block for the Request Datagram Reply Command 5-36
Format of Returned Data 5-36
COMMREQ #16: Enable/Disable I/O Fault Categories 5-37
Command Block for the Enable/Disable I/O Fault Categories Command 5-37
COMMREQ #17: Do Output Command 5-38
Command Block for the Do Output Command 5-39
Example 5-40
Using the Do Output Command for Global Data 5-41
Command Block for the Do Output Command for Global Data 5-41
COMMREQ #18: Read Serial Bus Address of the Bus Controller 5-42
Command Block for the Read SBA Command 5-42

Chapter 6 Reading Bus Status Information 6-1

Reading the Serial Bus Addresses of Active Devices 6-2
Format of the Bus Status Data that is Returned 6-2
RX7i CPU: Reading Serial Bus Addresses 6-3
Series 90-70 CPU: Reading Serial Bus Addresses 6-4
Reading the Status of the Datagram Queue 6-5
Format of the Datagram Queue Status Data that is Returned 6-5
RX7i CPU: Reading Datagram Queue Status 6-5
BUS READ Parameters for Reading the Datagram Queue Status 6-5
Series 90-70 CPU: Reading Datagram Queue Status 6-6
VME READ Parameters for Reading Datagram Queue Status 6-6

Chapter 7 Data Monitoring, Distributed Control, and Redundancy 7-1

Data Monitoring 7-2

Monitoring Inputs 7-2
Monitoring Diagnostics and Configuration Change Messages 7-2
Distributed Control 7-3
Redundancy 7-4
Dual Bus Redundancy 7-5
Redundant Controllers 7-10
Dual Bus and Redundant Controllers 7-13
Bus and Controller Redundancy for Hot Standby Devices 7-14
Bus and Controller Redundancy for Duplex Devices 7-15
Operation Remains with Primary PLC, Dual Bus and Dual Controllers 7-16
Genius Redundancy for Series 90-70 Rev. 3 7-17
Dual Bus and Dual Controllers 7-17

Appendix A ASCII Code List A-1

Chapter 1 Introduction

This manual describes the features and operation of a Series 90™-70 Genius® Bus Controller when it is used in a PACSystems RX7i or Series 90-70 PLC system. This manual also provides the configuration and programming information needed to complete the interface between the system CPU and a Genius bus.

Product Documentation

Content of this Manual

Chapter 1. Introduction: Chapter 1 describes the Bus Controller and explains how it operates.

Chapter 2. Installation: Chapter 2 explains how to install or remove a Bus Controller, and how to connect it to a Genius serial bus.

Chapter 3. Bus Controller Configuration: Chapter 3 explains how to complete the software configuration steps for a Bus Controller and its bus.

Chapter 4. Diagnostics: Chapter 4 describes diagnostics capabilities provided by the Bus Controller.

Chapter 5. Communication Request: Chapter 5 describes the use of the COMMREQ program instruction with a Bus Controller.

Chapter 6. Reading Bus Status Information: Chapter 6 explains how the application program can read the Serial Bus Addresses of active devices or read the status of the Datagram queue.

Chapter 7. Data Monitoring, Distributed Control, and Redundancy: Chapter 7 describes basic types of data monitoring, distributed control and redundancy systems that are supported by the Bus Controller.

Appendix A. ASCII Code List: Lists ASCII characters and their decimal and hexadecimal equivalents.

Related Publications

For more information, refer to these publications:

Genius I/O System User's Manual (GEK-90486-1). Reference manual for system designers, programmers, and others involved in integrating Genius I/O products in a PLC or host computer environment. This book provides a system overview, and describes the types of systems that can be created using Genius products. Datagrams, Global Data, and data formats are defined.

Genius Discrete and Analog Blocks User's Manual (GEK-90486-2). Reference manual for system designers, operators, maintenance personnel, and others using Genius discrete and analog I/O blocks. This book contains a detailed description, specifications, installation instructions, and configuration instructions for all currently-available discrete and analog blocks.

Series 90-70 PLC Genius Bus Controller Manual (GFK-0398). This book describes the use of the Series 90-70 Genius Bus Controller in a Series 90-70 PLC system, and explains its configuration and programming using the Logicmaster software.

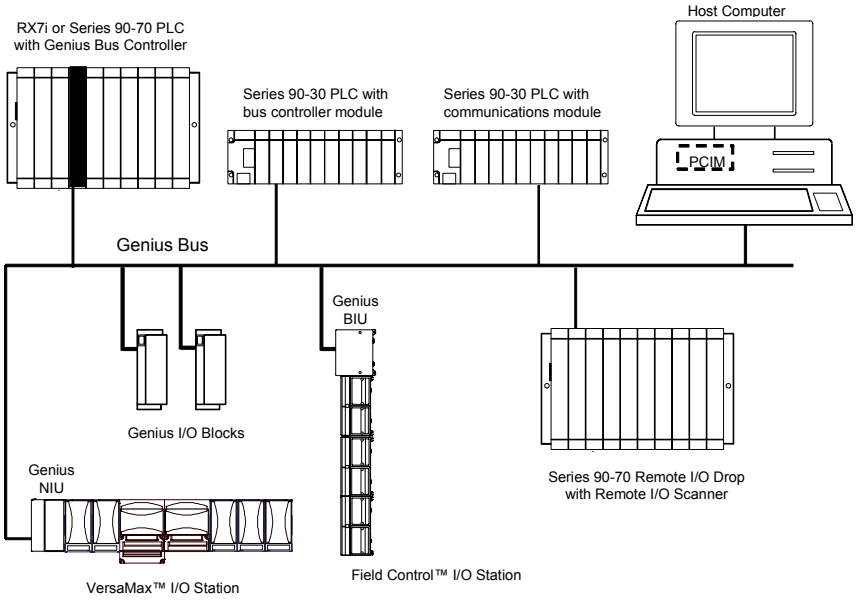
Series 90-70 Remote I/O Scanner User's Manual (GFK-0579). Reference manual for the Remote I/O Scanner, which interfaces a drop containing Series 90-70 modules to a Genius bus. Any CPU capable of controlling the bus can be used as the host. This book describes the Remote I/O Scanner features, configuration, and operation.

Series Six™ Bus Controller User's Manual (GFK-0171). Reference manual for the Bus Controller, which interfaces a Genius bus to a Series Six PLC. This book describes the installation and operation of the Bus Controller. It also contains the programming information needed to interface Genius I/O devices to a Series Six PLC.

Series Five™ Bus Controller User's Manual (GFK-0248). Reference manual for the Bus Controller, which interfaces a Genius bus to a Series Five PLC. This book describes the installation and operation of the Bus Controller. It also contains the programming information needed to interface Genius I/O devices to a Series Five PLC.

System Overview

The Genius Bus Controller (catalog number IC697BEM731) can be used to interface a PACSystems RX7i controller or Series 90-70 PLC to a broad range of other devices on a Genius I/O serial bus.



A Genius bus may serve:

- **Individual Genius I/O Blocks.** Genius blocks are self-contained discrete, analog, and special-purpose modules with advanced diagnostics capabilities and many software-configurable features.
- **Other PLCs**: RX7i, Series 90-70, Series 90-30. Series Six and Series Five PLCs can also be connected to the Genius bus.
- **Series 90-70 Remote Drops**. A remote drop is a Series 90-70 rack that is interfaced to the bus by a Remote I/O Scanner module. Each remote drop can exchange up to 128 bytes of input data and 128 bytes of output data.
- **VersaMax and Field Control I/O Stations**. An I/O Station consists of a group of I/O and special-purpose modules connected to the bus via an interface module. Each remote drop can exchange up to 128 bytes of input data and 128 bytes of output data.
- **Multiple hosts**, for communications using datagrams and Global Data.

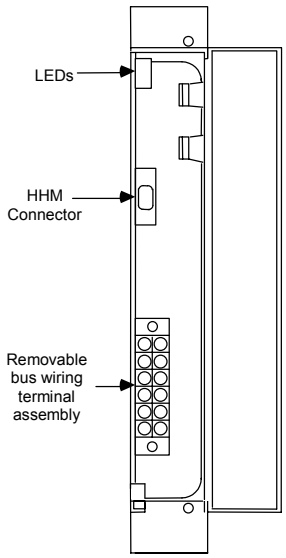
A bus may be used entirely for I/O control, or it may feature I/O control enhanced by communications commands in the program. A bus may also be dedicated to CPU communications, with multiple CPUs and no I/O devices. More complex systems can also be developed, with dual CPUs and one or more additional CPUs for data monitoring.

Number of Bus Controllers in a PLC

Up to 31 Bus Controllers can be included in the RX71 or Series 90-70 rack system. In some redundant systems, fewer Bus Controllers are possible. See chapter 7 for details.

Bus Controller Description

The Genius Bus Controller is a standard, rack-mounted Series 90-70 PLC module.



Status LEDs

The LEDs on the front of the Bus Controller indicate its operating status. The top two LEDs should be on during normal operation. The bottom LED is not used.

Module OK Shows the status of the Bus Controller. This LED blinks during powerup diagnostics.

Channel OK Shows the status of the bus. This LED is on steadily when the bus is operating properly. It blinks for intermittent bus errors and is off for a failed bus. It is also off when no configuration has been received from the PLC CPU.

Hand-held Monitor Connector

The Hand-held Monitor connector on the Bus Controller faceplate provides attachment for a Hand-held Monitor. All Hand-held Monitor functions except I/O block Device Number assignment can be performed with the HHM connected to the Bus Controller. Bus and block operation can be monitored, circuits forced or unforced, outputs Pulse Tested, diagnostic messages displayed, and faults cleared, from this convenient central location. Hand-held Monitor version IC660HHM501C (or later), permitting selection of a “host CPU” is recommended.

Terminal Assembly

Serial bus and shield wiring connections are made to the removable terminal strip on the front of the Bus Controller. Only the upper three terminals are used. To remove the Terminal Assembly without disturbing the continuity of the bus, jumpers are used. See chapter 2.

The Genius Bus

The Genius bus is a shielded twisted-pair wire, daisy-chained between devices, and terminated at both ends. Proper cable selection is critical to successful operation of the system. Suitable cable types are listed in the *Genius I/O System User's Manual*.

Conservative wiring practices, as well as national and local codes, require physical separation between control circuits and power distribution or motor power. Refer to sections 430 and 725 of the National Electric Code.

Bus Type	Daisy-chained bus cable; single twisted pair plus shield or Twinax. Fiber optics cable and modems can also be used.
Bus Termination	75, 100, 120, or 150 ohm resistor at both ends of electrical bus cable.
Baud Rate	Configurable. 153.6 Kbaud standard, 153.6 Kbaud extended, 76.8 Kbaud, or 38.4 Kbaud.
Maximum Bus Length	7500 feet at 38.4 Kbaud, 4500 feet at 76.8 Kbaud, 3500 feet at 153.6 Kbaud extended, 2000 feet at 153.6 Kbaud, standard. Maximum length at each baud rate also depends on cable type. Chapter 2 provides a complete list of cable types, showing corresponding bus lengths and baud rates. Greater bus lengths are possible using sections of fiber optics cable with modems.
Maximum Number of Devices	32 devices at 153.6 Kbaud standard, 153.6 Kbaud extended, or 76.8 Kbaud. 16 devices at 38.4 Kbaud. Includes bus controller and typically a Hand-held Monitor.
Data Encoding	Each bit is encoded into three dipulses, majority voted at the receiver to correct any single dipulse errors. A dipulse is an AC code consisting of a positive then negative excursion of voltage. Dipulses are individually sampled to reject low and high frequency interference.
Modulation Technique	Frequency Shift Keying (FSK) 0 to 460.8 KHz max. (153.6 Kilobaud)
Isolation	2000 volts Hi-Pot, 1500 volts transient common mode rejection.
Signal/noise Ratio	60 db

Bus Controller Operation

The Bus Controller handles all data transfer between the PLC and the devices on its bus. In order to do this, the Bus Controller must interface two completely separate and asynchronous activities:

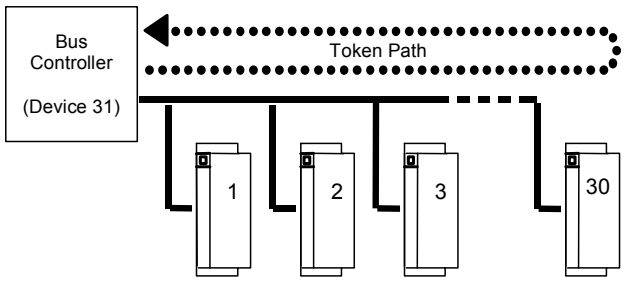
A. The Genius bus scan, a cycle of communications between the devices on a bus (including the Bus Controller itself). The cycle follows the order of Bus Addresses (0-31).

B. The CPU sweep, the cycle of actions that includes communications between the CPU and the Bus Controller.

The Bus Controller manages data transfer between the bus and the CPU by maintaining two separate on-board RAM memories. One interfaces with the bus and the other interfaces with the CPU. The Bus Controller automatically transfers data between these two memories, making data available to the bus or to the CPU when it is needed.

The Genius Bus Scan

A bus scan consists of one complete rotation of a "token" among the devices on the bus.



As mentioned earlier, these devices may include other Bus Controllers, or Remote I/O Scanners, in addition to (or instead of) the Genius blocks illustrated above.

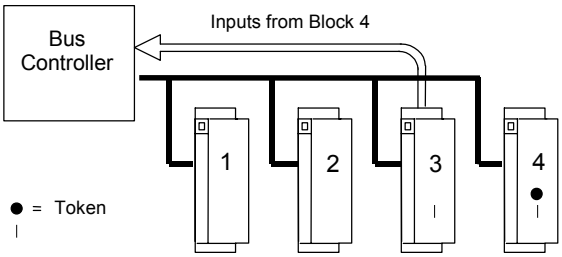
During a bus scan, the Bus Controller automatically:

- Receives all input data that has been sent by devices on the bus.
- Broadcasts Global Data.
- Updates outputs, as permitted, to the devices on the bus. Transmission of outputs from the Bus Controller can be disabled for one or more devices on the bus.
- Receives any fault messages issued by devices on the bus and sets diagnostic status references for use by the CPU.
- Sends a single command received from the CPU (for example, Clear Circuit Faults) to the appropriate devices.

The amount of time it takes for the communications token to pass to all devices depends on the baud rate, the number and types of devices on the bus, and the use of Global Data and datagram communications.

Input Data from Devices on the Bus

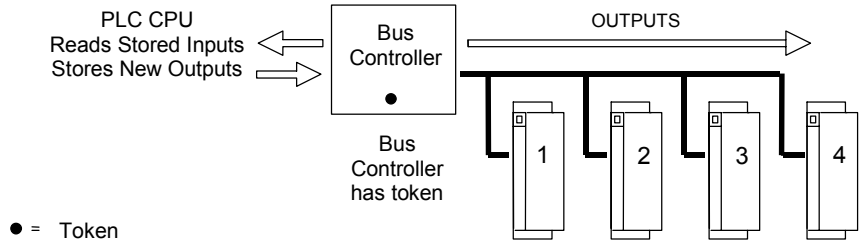
The Bus Controller receives input data from each input block, I/O block, and remote drop each time the block or Remote I/O Scanner has the communications token. (Because this data is broadcast, it may be received by any other bus interface module operating on the bus).



The Bus Controller stores all the input data it receives. Once per CPU sweep, the CPU reads all discrete and analog inputs from the Bus Controller. (Analog data is not multiplexed).

Output Data from the CPU

As the application program executes, the CPU sends outputs and any commands to the Bus Controller. The Bus Controller stores this data, transmitting it on the bus each time it has the communications token. Unlike inputs, which are broadcast, outputs are directed to the specific device that should receive them.



Outputs for 4 Input/2 Output Analog Blocks

Four words of %AQ memory are assigned to a 4 Input/2 Output block by the configuration software. The CPU stores the output data as shown below. Locations "n+2" and "n+3" are not used by the block.

n+3	n+2	n+1	n	**%AQ**
not used	not used	channel 2	channel 1	

Amount of I/O Data on the Bus

The amount of I/O data exchanged during one Genius bus cycle depends on the types of devices on the bus.

Data Lengths for Genius Blocks

Data lengths for Genius I/O blocks are shown below. For %I and %Q memory, the sizes shown are in bits. For %AI and %AQ memory, the sizes shown are in words.

Block Type	Module Data Lengths			
	%I (bits)	%Q (bits)	%AI (words)	%AQ (words)
115 VAC Grouped I/O blocks	8	8		
115 VAC Isolated I/O blocks	8	8		
16 Ckt AC Input Block	16			
16 Ckt DC Sink/source blocks	16	16		
32 Ckt DC Sink/source blocks	32	32		
Relay Output blocks		16		
4 Input/2 Output Analog Blocks			4	2
Current-source Analog I/O Blocks			4	2
Current-source Analog Output Blocks				6
RTD Input blocks			6	
Thermocouple blocks			6	
High-speed Counter	16	16	15	
PowerTRAC Module	16	16	18	

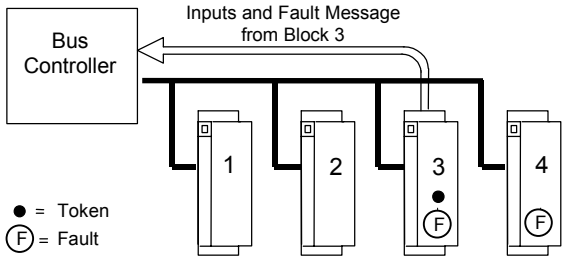
Many Genius I/O blocks have both inputs and outputs on the same block. Blocks configured in the software as having both inputs and outputs will occupy identical references in both %I and %Q memory. Unused references cannot be assigned to other inputs or outputs, *and should not be used in the application program*.

Data Lengths for an I/O Station or Series 90-70 Remote Drop

The Bus Controller sees each VersaMax I/O Station, Field Control I/O Station, or Series 90-70 Remote Drop on the bus as a single I/O device. Each I/O Station or Remote Drop can exchange up to a total of 128 bytes of inputs and 128 bytes of outputs (8 discrete points represent one byte and 1 analog channel uses 2 bytes) on the Genius bus.

Diagnostics

Genius blocks and other devices on the bus will automatically report faults, alarms and certain other predefined conditions to the CPU.



Only one diagnostic message can be sent during any bus scan. If a fault message has already been sent (by another device) during that scan, a device saves its own diagnostic message until the next available bus scan. For example, if the communications token is currently at device 2, and faults occur at devices 3 and 4 at the same time, device 3 can send its diagnostic message if another message has not already been sent. Device 4 must wait at least one more bus scan to send its diagnostic message.

The Bus Controller stores any diagnostic messages it receives. They are read automatically by the CPU. Faults may then be displayed in the fault table. Details are in chapter 4.

In addition the built-in diagnostics capabilities of Genius devices, the application program can make use of additional diagnostics mechanisms provided by the CPU:

- System Status References that have been defined for Genius use.
- Fault and No Fault contacts that can be used to detect fault and lack of fault conditions.
- Alarm contacts that can be used to indicate when an analog value has reached an assigned alarm limit.

See chapter 4 for details.

Input Defaults if the Bus Controller is Lost

In an RX7i system, if a Bus Controller is missing, mismatched, or failed, the RX7i CPU sets the input data from each of that Bus Controller's bus devices to its configured default states (either Hold Last State or Off (0).

In a Series 90-70 system, if a Bus Controller fails, the Series 90-70 CPU holds all input data from that Bus Controller at its last valid states, regardless of each block's configured input default parameter.

Datagrams

The Bus Controller supports all Genius datagrams:

Datagram Type Type	*Description*
Read ID	Requests identifying information from a device on the bus.
Read ID Reply	The automatic response to a Read ID datagram.
Read Configuration	Requests configuration data from a device on the bus.
Read Configuration Reply	The automatic response to a Read Configuration datagram.
Write Configuration	Sends configuration data to a device on the bus.
Assign Monitor	Commands a device on the bus to direct an extra copy of each Fault Report to another device on the bus.
Read Diagnostics	Requests diagnostics data from a device on the bus.
Read Diagnostics Reply	The automatic response to a Read Diagnostics datagram.
Write Point	Sends up to 1 word of bit data to a Series Six or Series Five PLC or to a host computer.
Read Block I/O	Requests I/O data from some types of Genius blocks.
Read Block I/O Reply	The automatic response to a Read Block I/O datagram.
Report Fault	An automatic diagnostic message received from a device on the bus.
Pulse Test	Commands a discrete block to pulse its outputs.
Pulse Test Complete	Automatic indication that outputs have been pulsed.
Clear Circuit Fault	Clears one specific circuit fault.
Clear All Circuit Faults	Clears all circuit faults on bus devices.
Switch BSM	Causes a Bus Switching Module to switch to alternate bus, if operational.
Read Device	Reads up to 128 bytes of CPU data via another Bus Controller.
Read Device Reply	The response to a Read Device datagram.
Write Device	Sends up to 128 bytes of data to a CPU, via its Bus Controller.
Read Data	Requests temporary data from a High-speed Counter block.
Read Data Reply	The automatic reply to a Read Data datagram.
Write Data	Sends temporary data to a High-speed Counter block.
Read Map	Requests the I/O map configuration of a Remote I/O Scanner.
Read Map Reply	Automatic response to a Read Map datagram.
Write Map	Sends I/O map configuration to a Remote I/O Scanner.

Additional datagrams, not listed above, are sent as system messages; they do not involve any application programming. The *Genius I/O System User's Manual* explains datagrams in detail. It also shows the formats of the data that is transferred by datagrams.

In the application program, COMMREQ instructions are used to send datagrams and to read any unsolicited datagrams that have been received. See chapter 5 for information.

Global Data

Global Data is data that is automatically and repeatedly broadcast by a Bus Controller. The Genius Bus Controller can send up to 128 bytes of Global Data each bus scan. It can receive up to 128 bytes of Global Data each bus scan from each Bus Controller on its bus.

Sending Global Data

Once set up by configuration (see chapter 3), Global Data is broadcast automatically. Other Bus Controllers that receive the Global Data will place it in these memory locations:

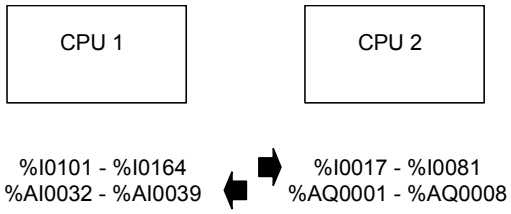
Bus Controller Sends Global Data To:	***Other CPU Places Global Data in this Memory Location:***
PACSystems RX7i CPU	%I, %Q, %G, %R, %AI, %AQ memory if manually-configured, or %G memory if automatically-configured. Memory type and beginning address are chosen during configuration of the *receiving* bus controller.
Series 90-70 CPU	%I, %Q, %G, %R, %AI, %AQ memory if manually-configured, or %G memory if automatically-configured. Memory type and beginning address are chosen during configuration of the *receiving* bus controller.
Series 90-30 CPU	%G memory location corresponding to Device Number (16-23) of the Bus Controller that sent the data.
Series Six CPU	Register memory. Beginning address selected during configuration of the Series 90-70 Bus Controller that sent the data.
Series Five CPU	Register memory. Beginning address selected during configuration of the Series 90-70 Bus Controller that sent the data.
Computer Interface Module	PCIM or QBIM Input Table Segment corresponding to Device Number of the Series 90-70 Bus Controller that sent the data.

Receiving Global Data

The Bus Controller can be configured to receive or ignore Global Data from any other Bus Controller. The memory type and length for incoming Global Data are also selected during configuration, as described in chapter 3. The RX7i or Series 70-70 CPU can place incoming Global Data in %I, %Q, %G, %R, %AI, or %AQ memory.

Example

In the following example, CPU 1 sends 64 bits of Global Data beginning at %I0101 to CPU 2. CPU 2 places this data into its own memory beginning at %I0017. CPU 2 sends 8 words of %AQ data beginning at %AQ0001 to CPU 1. CPU 1 places this data into its own memory beginning at %AI0032.



Chapter 2 Installation

This chapter explains:

- How to install and remove a Bus Controller.
- How to connect a Genius serial bus.
- How to terminate a bus if a Bus Controller is physically at either end.

For Additional Information, Also See:

Chapter 1 for a description and illustration of the Bus Controller, explanation of its LEDs, and specifications for the Genius bus.

Chapter 3 for configuration instructions.

Chapter 6 for information about dual bus and dual controller systems.

Installing the Bus Controller

1. Be sure the rack is powered down.
2. Position the Bus Controller at its intended location.
3. Push the Bus Controller into the card guide until it is aligned with the connector on the rack backplane.
4. Pressing the upper and lower flanges on the left of the module, push it into the connector until it clicks onto the rack rails.

   Look to see that the board has seated properly in the connector.
5. Complete the bus connections to the front of the board as described on the next page.

Removing the Bus Controller

1. Power down the rack in which the Bus Controller is located. Before removing power, it is important to consider the impact on the controlled process.
2. If the PLC is not part of a redundant system, the bus wiring can be removed from the Bus Controller.

   If the PLC is part of a redundant system and another CPU on the bus is now functioning as the controller, the Bus Controller can be removed without powering down the bus, provided the Bus Controller's Serial 1 terminals and Serial 2 terminals have been jumpered as described in this chapter. If this has been done, do not disconnect the bus cable or any terminating resistor. Remove the terminal assembly from the Bus Controller carefully. Avoid contact with exposed cable wiring. Place the terminal assembly with the bus wiring still attached, in a protected location.

   Caution

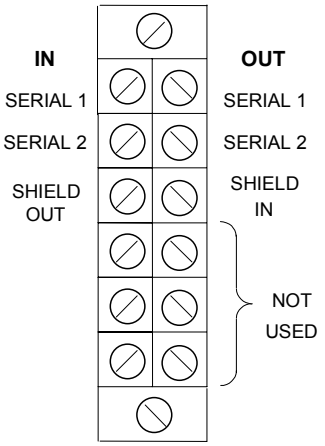
   If exposed wiring comes in contact with conductive material, data on the bus may be corrupted, possibly causing the system to shut down.
3. Squeeze the retaining clips at the top and bottom of the cover to disengage them from the rack rails.
4. Pull the board firmly to remove it from the backplane connector.
5. Slide the board out of the card guide to remove it from the rack.

Connecting the Serial Bus

For information about bus selection and installation, you should refer to the *Genius I/O System User's Manual*.

Connect the bus cable to the terminal assembly on the front of the Bus Controller. The tie–down screws can be removed to accommodate ring-type connectors. Terminal designations, illustrated below, are also shown on the module faceplate.

The maximum exposed length of bare wires should be two inches. For added protection, each shield drain wire should be insulated with spaghetti tubing to prevent the Shield In and Shield Out wires from touching each other or the signal wires.



Replacing an Older Bus Controller

If this hardware (GIOC1) is being used to replace older hardware (GIOA1 or GIOB1: see markscreen on the edge of the board), the GENIUS bus connections to the Bus Controller must be rewired. Refer to the wiring label inside the module cover for detals concerning the proper wiring of the connector. Note that GIOC1 hardware was also used with Genius Bus Controller versions IC697BEM731B and C.

Shield In and Shield Out Connections in an Existing Installation

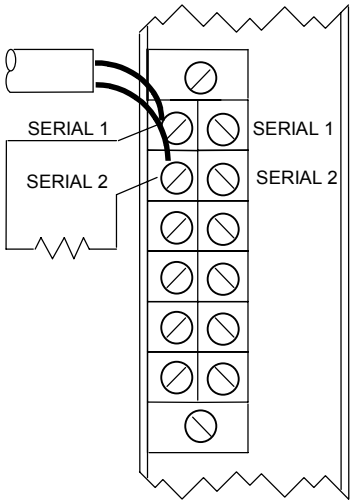
The actual positions of the Bus Controller's Shield In and Shield Out terminals are correctly shown above. On the faceplates of older Bus Controllers and in earlier revisions of the documentation, these terminals are shown reversed. *Regardless of the markings on the faceplate, all Series 90-70 Bus Controllers have their Shield In and Shield Out terminals in the positions shown above.*

Because of this inconsistency, Bus Controllers in an existing installation may have their Shield In and Shield Out terminals incorrectly connected (that is, not as illustrated above). For most applications, this should not be a problem, and rewiring is not necessary. If noise immunity is a particular concern, however, rewiring of the Shield In and Shield Out terminals on these older Bus Controllers is recommended.

Terminating the Bus

Each Genius communications bus must be terminated at both ends by its characteristic impedance, as explained in the *Genius I/O System User's Manual*.

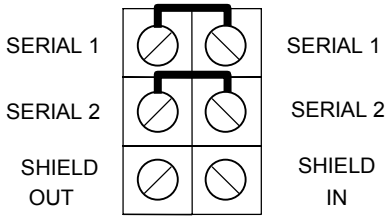
If the Bus Controller is located at the end of a bus, install the appropriate resistor across its Serial 1 and Serial 2 terminals.



Wiring for Bus Continuity

For a redundancy system, where another CPU on the bus will be capable of acting as a controller, jumpers should be installed on the Bus Controller's terminal assembly as shown at right. This will allow possible removal of the terminal assembly in the future without breaking the continuity of the bus.

For bus continuity, jumper the Serial 1 terminals together and jumper the Serial 2 terminals together (even if the Bus Controller is at the end of the bus). Alternatively, use only one terminal of each pair, and wire both cable ends to the selected terminals.



Chapter 3 Bus Controller Configuration

A Bus Controller and the devices on its bus must be configured in two basic, different procedures.

1. The Bus Controller and the devices on its bus must be configured as part of the PLC system using the programming software.
2. The devices on the bus must also be configured separately. This includes:

   A. Configuring I/O blocks with a Hand-held Monitor and/or Write Configuration COMMREQs. For a Series 90-70 PLC, some devices can be configured directly from the programmer on an operating Genius bus.

   B. Configuring Remote Drops with the programming software.

   C. Configuring redundant Bus Controllers with the programming software

This book only covers configuration of Bus Controllers with the programming software.

For Additional Information, Also See:

Chapter 5, which describes Read Configuration and Write Configuration COMMREQs.

Chapter 7, which describes data monitoring, distributed control, and redundant control systems.

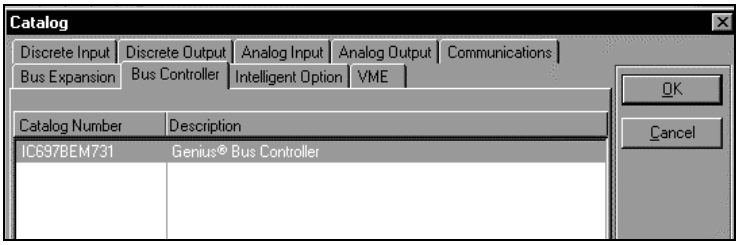
The *Genius Analog and Discrete Blocks Manual*, which includes instructions for configuring I/O blocks.

The *Genius I/O System and Communications Manual*, which details the data that can be transferred using Read Configuration and Write Configuration COMMREQs.

The *Series 90-70 Remote I/O Scanner User's Manual*, which covers configuration of Remote Drops.

Adding a Bus Controller to the Configuration

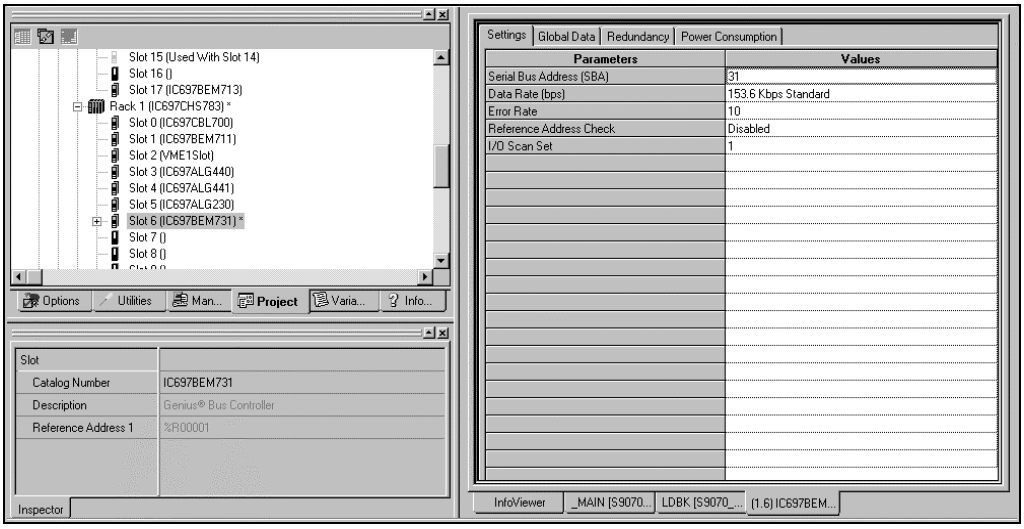
Select the slot location for the Bus Controller. Right click. Select Add Module from the menu. Select the Bus Controller tab.



Click OK.

> To edit a previously-selected module's operating characteristics, double-click the module in the Project tab or right-click on it and choose Configure.

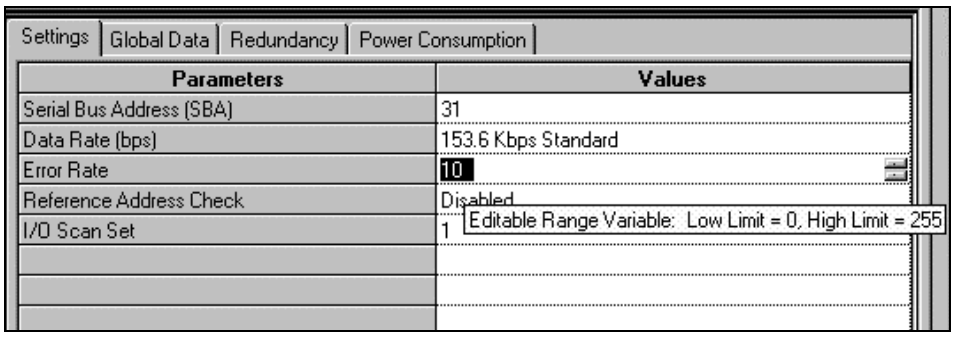
The Parameter Editor window appears.



The default entries can be used as is, or changed. *Until a valid configuration is stored to the CPU, the Bus Controller will not operate on the Bus, and its Channel OK LED will not light*.

Configuring the Bus Controller Settings

Select the Settings Tab.



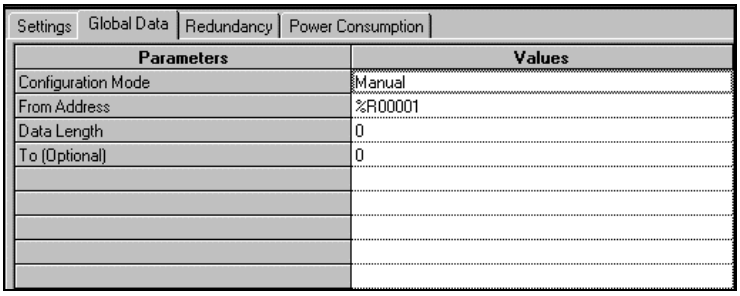
Configure the following parameters as needed for the system:

Serial Bus Address	Ordinarily, the Serial Bus Address assigned to a Bus Controller is 31. Any number from 0 to 31 can be used; each must be unique on that bus. For redundancy applications, specific Serial Bus Addresses are required, as described in chapter 7. Note: If Serial Bus Address and Redundancy Mode (on the Redundancy tab) are valid, two redundant Bus Controllers (source and paired) are moved or inserted to the new bus locations. However, if the newly-selected Serial Bus Address has already been assigned to another device on the bus, the two Bus Controllers are not moved. If you make the Serial Bus Address available, you must return to this tab and re-enter the Serial Bus Address to update the paired Bus Controller.
Data Rate (bps)	All devices on a bus *must* use the same data rate: 153.6 Kbaud standard, 153.6 Kbaud extended, 76.8 Kbaud, or 38.4 Kbaud. Selection of a data rate depends on the application, as explained in the *Genius I/O System User's Manual*. Usually, the bus length determines the data rate. The entry made here establishes the data rate for the Bus Controller *only*. If the default data rate (153.6 Kbaud standard) will not be used, the data rate of other devices on the bus must also be changed. Typically, this is done using a Hand-held Monitor.

Error Rate	This entry determines how the Bus Controller will respond to errors on the bus. If the Bus Controller should drop off the bus when a specified number of errors occurs within a 10-second period, enter that number of errors here. If the Bus Controller should remain on the bus when errors occur and try to maintain communications, enter 0 here. **Caution** If the bus includes a Bus Switching Module (BSM) or another device that controls bus switching, the Error Rate MUST be set to 0. Otherwise, the Bus Controller may drop off the bus when the BSM is switching a device to the bus. If the error rate is set to 0, the Bus Controller will use a rate of 10 errors in 10 seconds. If the detected error rate exceeds 10, the Bus Controller will remain on the bus and flash the Communications LED. The LED will continue to flash until the number of detected errors is less than 10 errors in 10 seconds. If the error rate is set to a value greater than 0, the Bus Controller will monitor the bus for errors. If the number of errors exceeds the configured error rate, the Bus Controller will turn off the Communications LED and drop off the bus. The Bus Controller will remain off the bus until the number of detected errors is less than the configured error rate.
Reference Address Check	This entry can be used to verify that references already configured for devices on the Genius bus match the status references assigned to the same devices with the programmer software. If ENABLED, references are checked for all configured devices except PowerTRAC blocks or “GENA”-based bus devices. This feature will *not detect or configure* an unconfigured device, *or correct references that do not match.*
I/O Scan Set	The Scan Set (as defined in the CPU's Scan Sets tab) assigned to this module. This can be used only on RX7i and on Release 7 or later 900-series 90-70 CPUs. For all other CPUs, leave this field at its default setting (1).

Configuring Global Data

Select the Global Data tab to configure the Bus Controller to send Global Data.



Configuration Mode	(Read-only) the Configuration Mode is always Manual.
From Address	Specify the beginning PLC address from which data will be transmitted on the bus. It can be from %I, %Q, %G, %R, %AI, or %AQ memory.
Data Length	This entry specifies the amount of Global Data to be sent each bus scan. If Global Data will not be used, set Data Length to zero. If bit-oriented memory (%I, %Q, or %G) is selected, this may be 0 to 1024 bits. It must be a multiple of 8. If you enter a number that is not a multiple of 8, the software will automatically adjust it upward to the next highest multiple of 8. For example, any number between 9 and 15 would be automatically adjusted upward to 16. If word-oriented memory (%AI, %AQ, or %R) is selected, this may be 0 to 64 words. If more than 64 words are selected, the software automatically adjusts the length to 64 words. The total amount of memory specified must not exceed the configured memory size for that memory type. For example, if %R00001 is selected for From Address and the Data Length is set to 4, the block of data would consist of %R00001 through %R00004.
To (Optional)	If the data is going to be sent to another Series 90-70 Bus Controller, this entry is not needed. The destination memory address is specified as part of the target device's configuration. If the data is going to be transmitted to a Series Six PLC or a Series Five PLC, enter the beginning register address where the data will be stored in the other PLC. Only one destination address can be specified for Global Data sent by each Bus Controller. If there is more than one Series Six and/or Series Five PLC on the bus, they must all use the same register address for Global Data received from this Bus Controller. For information about selecting and entering a register address for one of these PLCs, refer to the *Genius I/O System and Communications User's Manual*, GEK-90486-1.

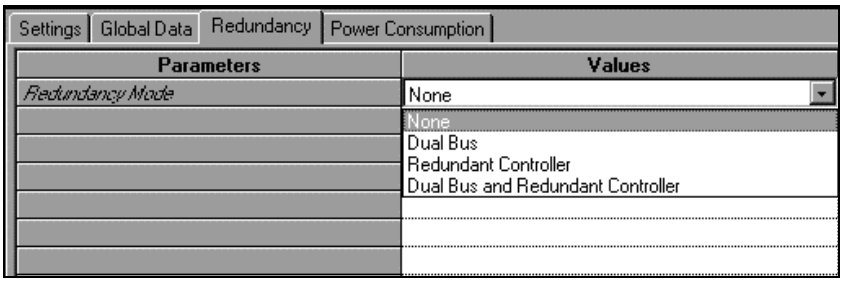
Configuring Genius Redundancy on the Redundancy Tab

Genius redundancy for the Bus Controller can be configured on the Redundancy tab as described below. Genius redundancy for the system can be configured using built-in wizards as explained next in this chapter. The wizards can also be used to configure Series 90-70 Hot Standby CPU redundancy for specific Series 90-70 CPU models.

When configuring a redundant Series 90-70 PLC system, remember to change the **Loss of IOC** fault from fatal to diagnostic. Otherwise, loss of a Bus Controller will cause the CPU to shut down. (This change can be made in the CPU configuration.) Also, when using Genius redundancy, do not set the CPU's Backplane Communications window (for Series 90-70 CPUs, this is the System Communications window) to 0. When using Constant Sweep mode, be sure to allow enough time for the Backplane Communications window to run. There must be enough time for the Bus Controllers to exchange information about Genius devices that are lost or added.

Selecting the Redundancy Mode

Select the Bus Controller's Redundancy Mode first. Changing the Redundancy Mode selection can reset some or all of the parameters that you have already entered.



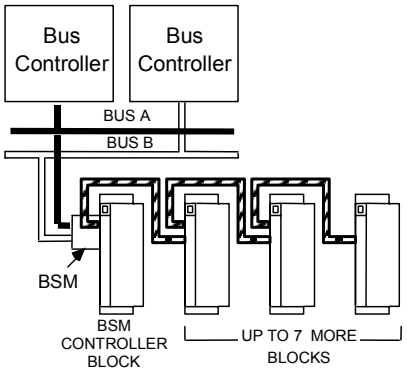
Redundancy Mode	The type of redundancy, if any. See chapter 7 for more detailed descriptions of redundancy modes. Configuration examples for redundancy are also shown on the following pages. *Note: if you set up a Bus Controller for redundancy, then either COPY or UNDELETE the Bus Controller's configuration, the **Redundancy Mode** of the copy or restored version is reset to NONE and the **Redundancy** of blocks on the bus is reset to NO.* **None:** This is the default. **None** means the Bus Controller communicates with a single bus, it is the only controller on the bus sending outputs, and no I/O devices on the bus are set up for any type of redundancy. *If, during subsequent configuration of devices on the bus, any device is set up for redundancy, **Redundancy Mode** is automatically changed to **Dual Bus.*** *Similarly, if **Redundancy Mode** is set to anything except NONE, any devices on the bus that have already been configured automatically have their **Redundancy** parameter set to YES.*

Dual Bus:

Settings	Global Data	Redundancy	Power Consumption

Parameters	Values
Redundancy Mode	**Dual Bus**
Paired GBC	External
Switch Time (Sec)	2.50

Select Dual Bus if there are two busses, each of which has its own Bus Controller. The Bus Controllers can be in the same PLC or separate PLCs. Switching devices, usually Genius Bus Switching Modules (BSMs), each link up to seven additional devices to the dual busses.

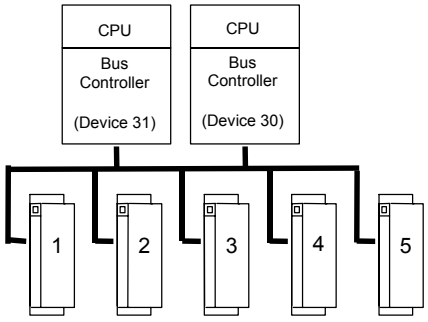


Redundant Controller:

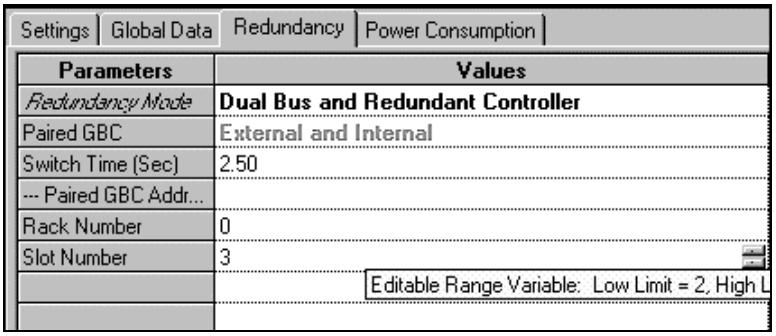
Settings	Global Data	Redundancy	Power Consumption

Parameters	Values
Redundancy Mode	**Redundant Controller**
Paired GBC	External
Switch Time (Sec)	2.50

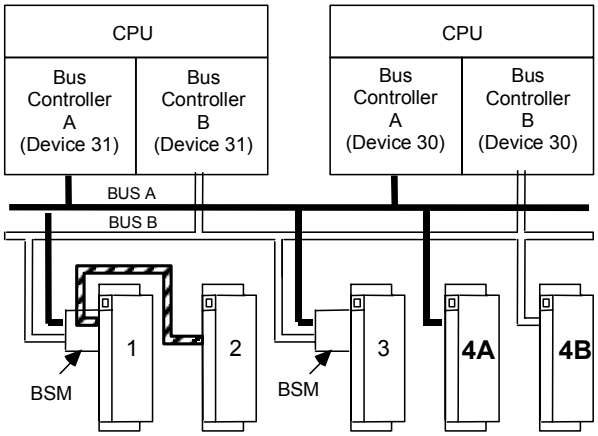
Select this for redundant Bus Controllers, either in the same PLC or separate PLCs:



Dual Bus and Redundant Controller:



Select Dual Bus and Redundant Controller for a system that combines redundant Bus Controllers with a dual bus. It requires two PLCs and four Bus Controllers:



Configuring Redundancy Parameters

When you select the Redundancy Mode, the additional parameters described below can be configured.

Paired GBC: Both dual bus and dual controller redundancy use pairs of Bus Controllers. This selection specifies the location of the other Bus Controller of the pair. The three choices are:

Internal: If the **Redundancy Mode** is either dual bus or redundant control and both Bus Controllers are located in the same PLC (not necessarily in the same rack), select **Internal**. You must also enter a **Paired GBC Addr** (see below).

External: If the **Redundancy Mode** is either dual bus or redundant control and the other Bus Controller is in another PLC, select **External**.

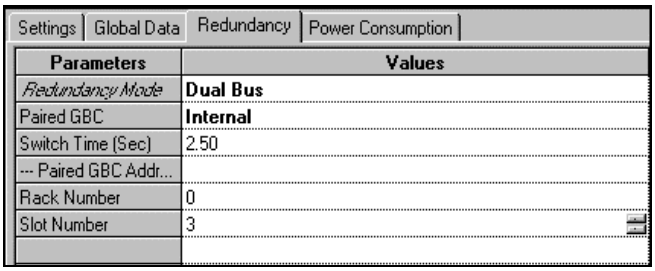
Int/Ext: Automatically selected if **Redund Mode** is set to **Dual Bus / Redundant Controllers**. You must also enter a **Paired GBC Addr** (see below).

*If, during subsequent configuration of devices on the bus, any is set up for redundancy, **Paired GBC** is automatically changed to **External.***

Switch Time: This is the amount of time that will be allowed for switching on a dual bus. The choices are 2.5 seconds and 10 seconds. If the **Redundancy Mode** is either **Dual Bus** or **Dual Bus / Redundant Controllers** and the total bus scan time on either bus is expected to exceed 100ms, change the **Switch Time** selection to 10 seconds. If the Bus Controller stops receiving input data from a device or devices on the bus, it will wait this specified time period before defaulting inputs or generating fault reports.

Be sure to select the same time period when configuring the devices on the bus with a Hand-held Monitor or Write Configuration COMMREQs. This determines the length of time I/O devices on the bus allow for bus switching, before defaulting their outputs.

Paired GBC Addr If you selected **Internal** for **Paired GBC**, enter the location of the other Bus Controller. For example:

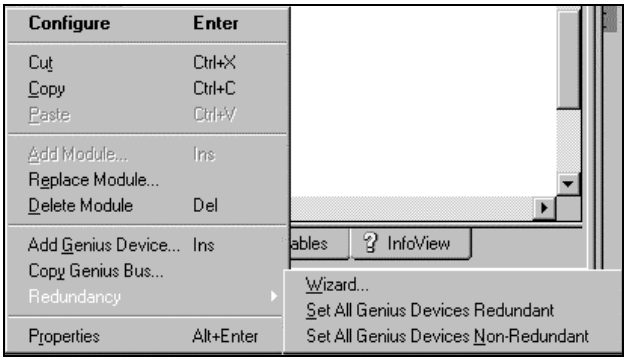


For **rack #** and **slot #**, enter the rack and slot number where the other Bus Controller is located. The **bus #** entry should be left as 1.

Using the Redundancy Wizards

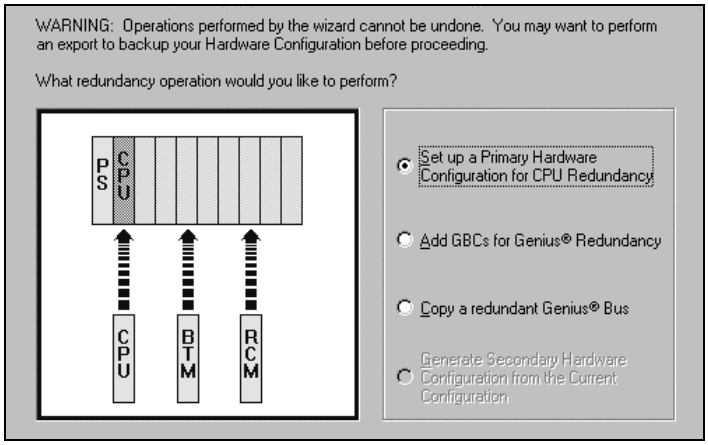
The Redundancy wizards automatically configure different redundancy options. They will add the required modules, with the correct parameter settings, to the rack system. For a Series 90-70 system only, this wizard can be called multiple times to create additional secondary hardware configurations. Operations performed by the wizard cannot be undone. Back up the Hardware Configuration before using the wizard. When configuring a redundant Series 90-70 PLC system, remember to change the **Loss of IOC** fault from fatal to diagnostic. Otherwise, loss of a Bus Controller will cause the CPU to shut down. (This change can be made in the CPU configuration.)

To use the Redundancy wizards to configure a system, right-click in the Hardware Configuration and choose Redundancy. Select Wizard.



Redundancy Wizards

The Redundancy wizards that are selectable will depend on whether you are configuring a Series 90-70 or RX7i system. For example:

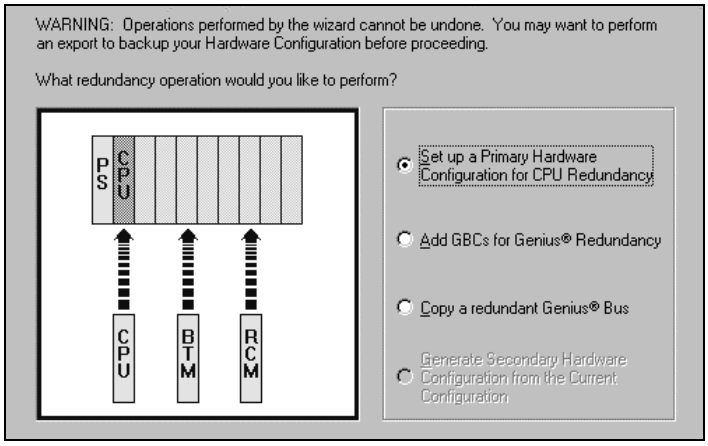


- **Set up a Primary Rack System for CPU Redundancy**. This wizard configures Hot Standby CPU Redundancy for a Series 90-70 CPU IC697CPU780, IC697CGR772, or IC697CGR935 only.
- **Add GBCs for Genius Redundancy**, selects a Genius redundancy scheme and the location of the Bus Controllers.
- **Copy a Redundant Genius Bus**, synchronizes two redundant busses. You can choose whether the destination bus is cleared before the copy and whether all devices or only devices marked redundant are copied. The source Bus Controller defaults to the slot that is currently selected. If the source and destination are not properly paired, including Redundancy Mode and Serial Bus Address, the copy operation is not allowed.
- **Generate Secondary Hardware Configuration from the Current Configuration**, creates a secondary hardware configuration from an existing primary rack system when configuring Hot Standby CPU Redundancy for a Series 90-70 CPU IC697CPU780, IC697CGR772, or IC697CGR935 only. This wizard is available only if the current rack system contains a CPU780, CGR935, or CGR772, and the Redundancy Mode (on the CPU configuration Redundancy tab) is set to Primary.

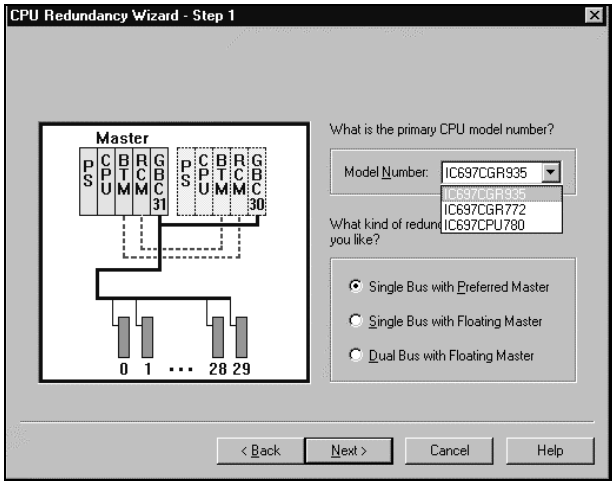
Redundancy Wizard: Set up a Primary Rack System for CPU Redundancy

For a Series 90-70 PLC, this wizard sets up Hot Standby CPU redundancy for three CPU models: IC697CPU780, IC697CGR772, or IC697CGR935. For more information about Series 90-70 CPU redundancy for CPU models IC697CGR772 and IC697CGR935, please see the *Series 90-70 Enhanced Hot Standby CPU Redundancy User's Guide*, GFK-1527A. For information about redundancy for CPU model IC697CPU780, please see the *Series 90-70 Hot Standby CPU Redundancy User's Guide*, GFK-0827. Both documents are available online at GE.com.

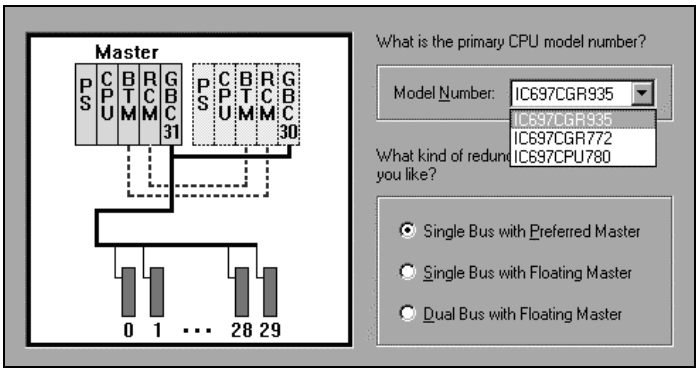
1. Select Set up a Primary Hardware Configuration for CPU Redundancy:



2. Select Next.

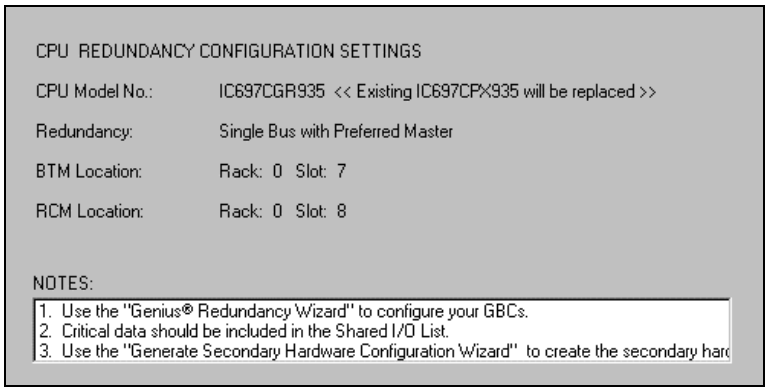


3. Choose the CPU type and redundancy type to configure. For example, “Single Bus with Preferred Master”:



4. Select Next.

5. Review the configuration parameters:



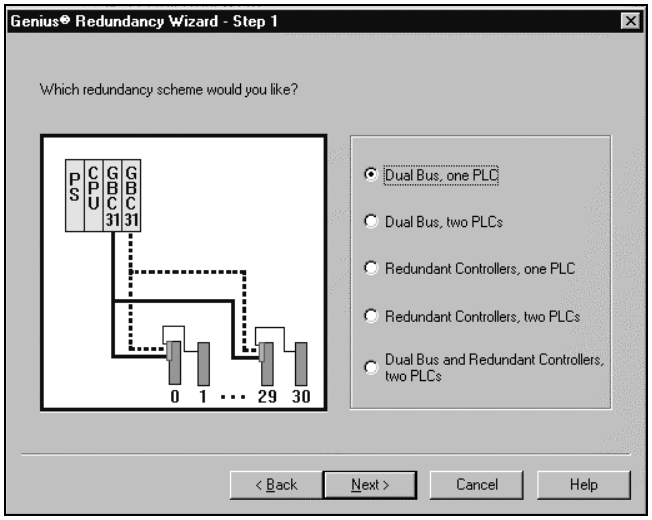
6. If the CPU, redundancy type, Bus Transmitter location and Redundancy module location are correct, select Finish to complete the redundancy configuration.

7. Another wizard, also accessed from the Redundancy: Wizards menu can be used to generate the secondary hardware configuration, as described later in this chapter.

Redundancy Wizard: Add GBCs for Genius Redundancy

Use this wizard to select a Genius Redundancy type. You can call this wizard multiple times to configure additional redundant busses in the same system.

1. In the Redundancy wizard, select: Add GBCs for Genius Redundancy. Select Next.

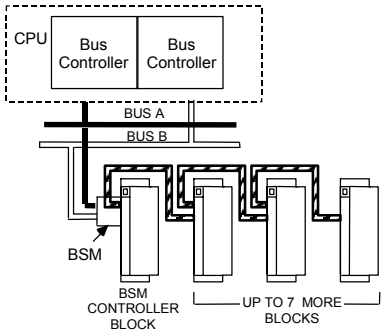


2. Select a redundancy scheme:
   - Dual bus, one PLC
   - Dual bus, two PLCs
   - Redundant controllers, one PLC
   - Redundant controllers, two PLCs
   - Dual bus and redundant controllers, two PLCs

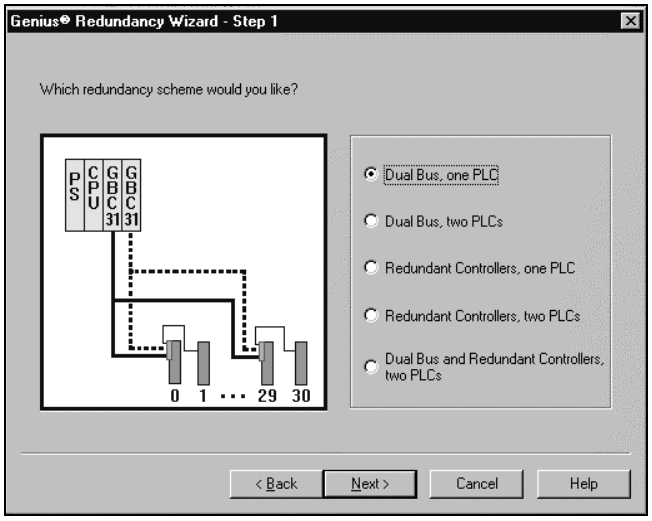
   All of these options are described on the following pages.

Add GBCs for Genius Redundancy: Dual Bus One PLC

Select Dual Bus One PLC if there are two busses, each of which has its own Bus Controller. The Bus Controllers are in the same PLC, but can be located in different racks. Switching devices, usually Genius Bus Switching Modules (BSMs), each link up to seven additional devices to the dual busses. All Genius devices downstream of a bus switching device must be configured for BSM Present.

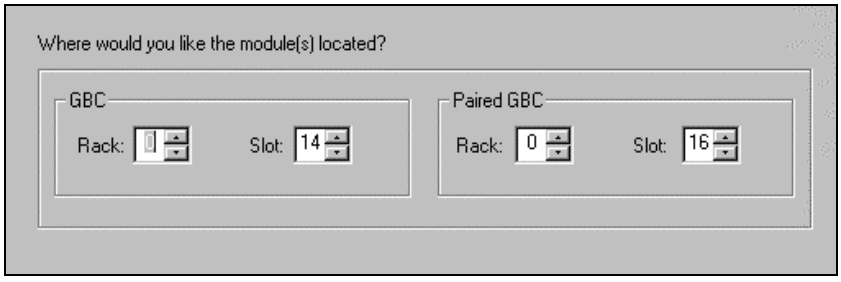


1. In the Redundancy wizard, select: Add GBCs for Genius Redundancy, then Dual Bus One PLC:



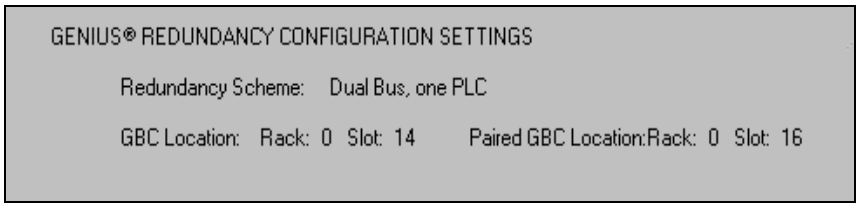
2. Click Next.

3. Select rack and slot locations for the primary Bus Controller and the paired Bus Controller. On a PACSystems RX7i, if you cannot select the expansion rack you want, you need to add it to the rack system.



4. Click Next.

5. Review the settings for Redundancy Scheme and Bus Controller locations.

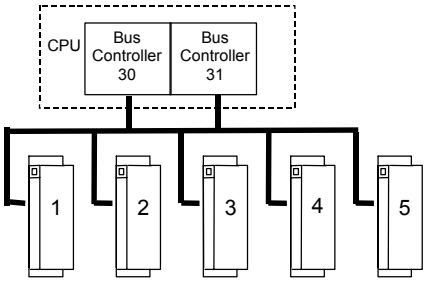


6. If settings are correct, click Finish.

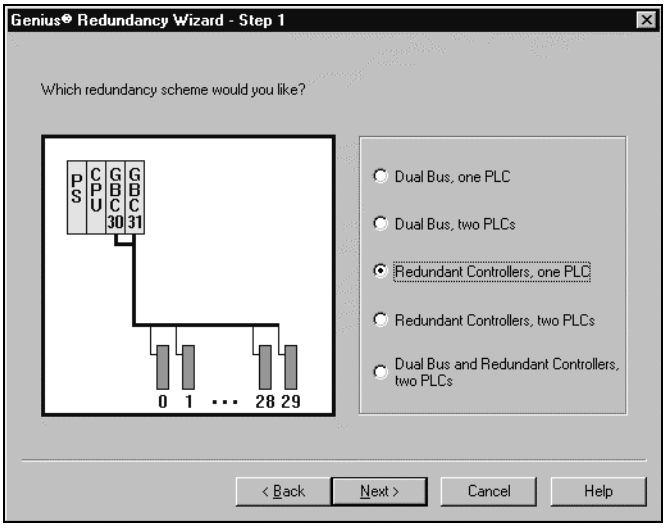
When you click Finish, the software creates the two Genius busses and adds two Bus Controllers to the rack system. The parameter settings of the second Bus Controller (except for the rack and slot numbers, and Serial Bus Address) are the same as those of the first Bus Controller. The two Bus Controllers are, by default, assigned Serial Bus Addresses 30 and 31. Each bus now contains an image of its Bus Controller.

Add GBCs for Genius Redundancy: Redundant Controllers, One PLC

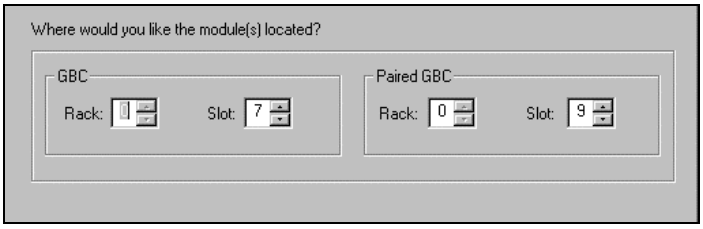
This type of Genius Redundancy uses two Genius Bus Controllers (GBCs) in the same CPU system, controlling one Genius bus.



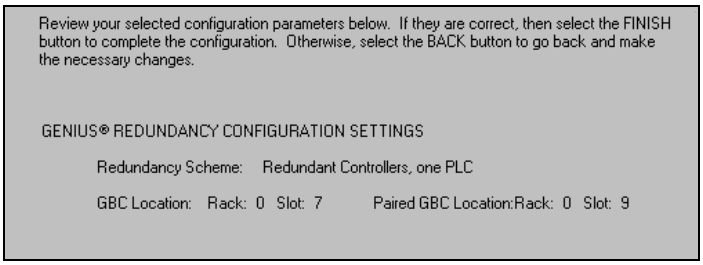
1. In the Redundancy wizard, select: Add GBCs for Genius Redundancy.
2. Click Next.



3. Select Redundant Controllers, One PLC:
4. Click Next.
5. Select rack and slot locations for the primary and paired Bus Controllers. On a PACSystems RX7i, if you cannot select the expansion rack you want, you need to add it to the rack system.



6. Click Next.



7. Review the settings for Redundancy Scheme and Bus Controller locations. If settings are correct, click Finish.

When you click Finish, the software adds two Bus Controllers to the PLC. The parameter settings of the second Bus Controller (except for the rack and slot numbers, and Serial bus Address) are a copy of those of the first Bus Controller. The two Bus Controllers are, by default, assigned Serial Bus Addresses 30 and 31. Those Serial Bus Addresses should not be changed.

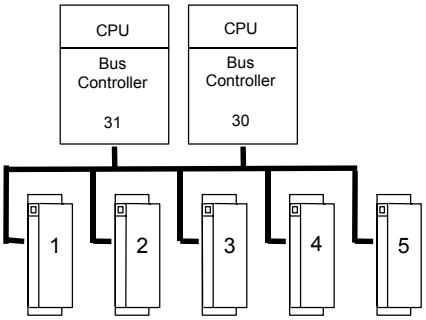
Configuring the Devices on the Bus

Genius devices used in this system may be redundant or non-redundant. To configure devices on a Genius bus, see Configuring a Genius Bus.

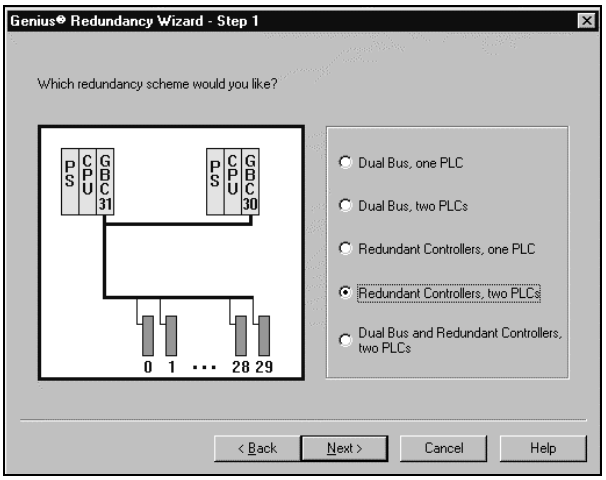
Although there is only one Genius bus in this system, each Bus Controller has its own Genius bus structure. If all devices on the bus are configured as redundant, the bus structures for the two bus controllers will be identical. However, if there are non-redundant devices on the bus, they will only appear in the Genius Bus window of the Bus Controller in which they were configured.

Add GBCs for Genius Redundancy: Redundant Controllers, Two PLCs

This type of Genius Redundancy uses redundant controllers in two PLCs, controlling the same bus.

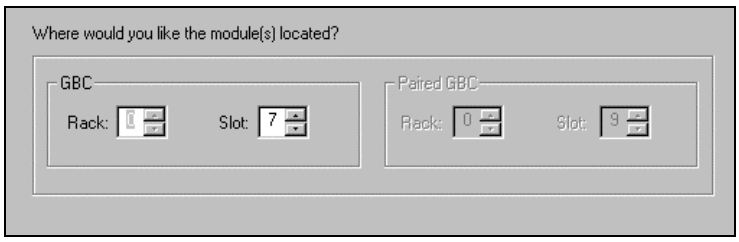


1. In the Redundancy, wizard, select Add GBCs for Genius Redundancy.
2. Click Next.

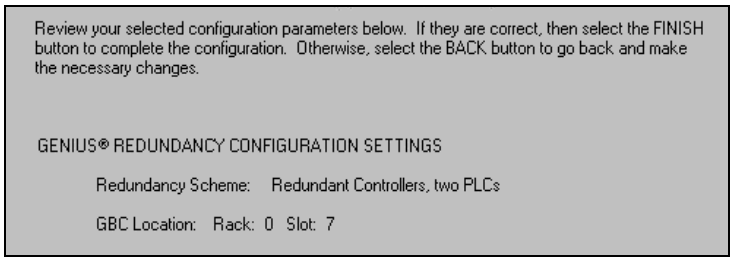


3. Select Redundant Controllers, Two PLCs:

4. Click Next. Select the rack and slot location for the primary Bus Controller. On a PACSystems RX7i, if you cannot select the expansion rack you want, you need to add it to the rack system.



5. Click Next.



6. Review the settings for Redundancy Scheme and Bus Controller locations. If settings are correct, click Finish. If you need to make changes, click Back.

When you click Finish, a second Genius Bus Controller is added to the PLC. The parameter settings of this second Bus Controller (except the rack and slot numbers, and Serial Bus Address) are the same as the first Bus Controller. The Serial Bus Address for the primary Bus Controller is, by default, 31. The Serial Bus Address of the other Bus Controller is 30. You should leave the Serial Bus Addresses at their default settings.

Configuring the Devices on the Bus

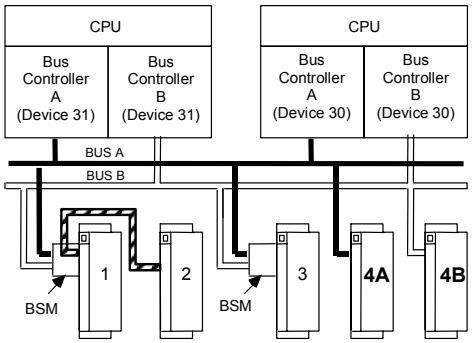
Genius Devices used on this system may be redundant or non-redundant. To configure devices on a Genius bus, see Configuring a Genius Bus.

Although there is only one Genius bus in this system, each Bus Controller has its own Genius Bus window. If all devices on the bus are configured as redundant, the Genius Bus windows for both Bus Controllers are the same. If there are non-redundant devices on the bus, they only appear in the Genius Bus window of one Bus Controller.

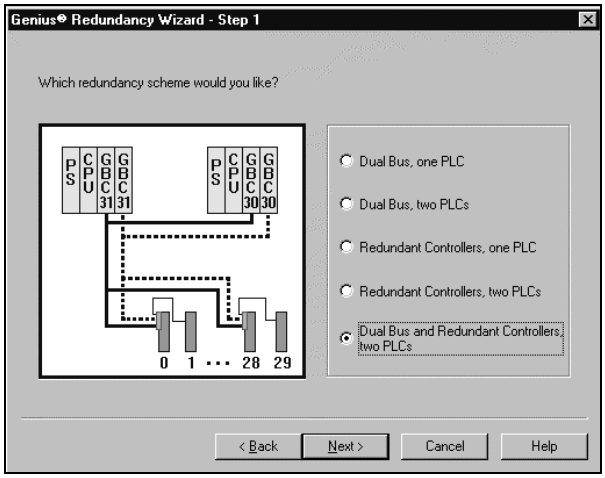
Add GBCs for Genius Redundancy: Dual Bus and Redundant Controllers with Two PLCs

This Genius Redundancy option consists of two PLCs, each with two Bus Controllers, controlling dual busses. Two open rack slots must be available in this PLC for the Bus Controllers. The paired Bus Controller may be in any rack.

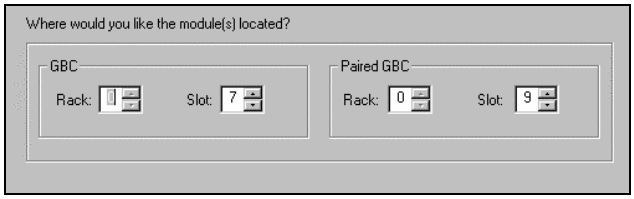
Select Dual Bus and Redundant Controller for a system that combines redundant Bus Controllers with a dual bus. It requires two PLCs and four Bus Controllers:



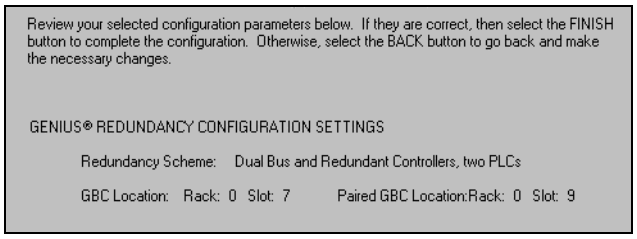
1. In Redundancy wizard, select Add GBCs for Genius Redundancy and click the Next button.
2. Select Dual Bus and Redundant Controllers, Two PLCs:



3. Click Next.
4. Select the rack and slot locations for the two Bus Controllers. On a PACSystems RX7i, if you cannot select the expansion rack you want, you need to add it to the rack system.



5. Click Next.



6. Review the settings for Redundancy Scheme and Bus Controller locations. If settings are correct, click Finish. If you need to make changes, click Back.

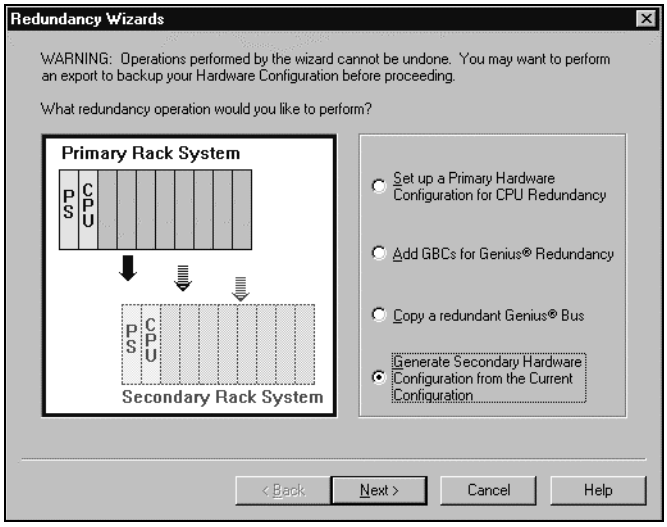
When you click Finish, two Genius busses are created and two Bus Controllers are added to the PLC. The parameter settings of the second Bus Controller (except for the rack and slot numbers and Serial Bus Address) are the same as for the first Bus Controller. The two Bus Controllers are, by default, assigned Serial Bus Addresses 30 and 31. Each bus contains an image of both Bus Controllers.

Redundancy Wizard: Generate Secondary Hardware Configuration from the Current Configuration

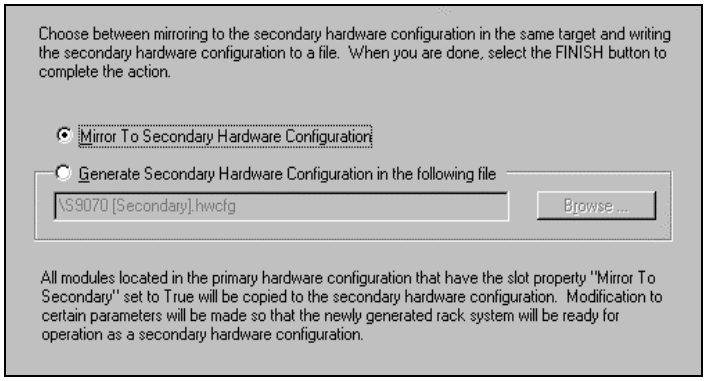
For a Series 90-70 PLC, this wizard generates a secondary hardware configuration for three CPU models: IC697CPU780/CGR772/CGR935. For more information about Series 90-70 CPU redundancy for CPU models IC697CGR772 and IC697CGR935, please see the *Series 90-70 Enhanced Hot Standby CPU Redundancy User's Guide*, GFK-1527A. For information about redundancy for CPU model IC697CPU780, please see the *Series 90-70 Hot Standby CPU Redundancy User's Guide*, GFK-0827. Both documents are available online at GE.com.

This wizard is available only if the current rack system contains a CPU780, CGR935, or CGR772, and the Redundancy Mode (on the CPU configuration Redundancy tab) is set to Primary.

1. In the Redundancy wizard, select: Generate Secondary Hardware Configuration from the Current Configuration.



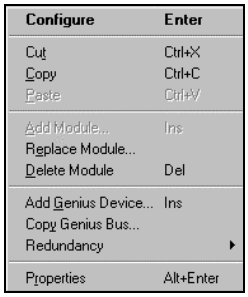
2. Click Next.



3. Select Mirror to Secondary Hardware Configuration or Generate Secondary Hardware Configuration in the following file.

Adding Devices to the Bus Configuration

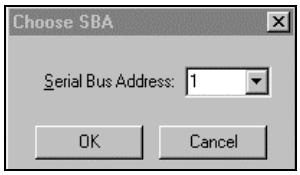
1. In the Project tab of the Navigator, expand the Hardware configuration folder and then the rack that contains the Bus Controller.
2. Right-click the slot containing the Bus Controller to display this menu.



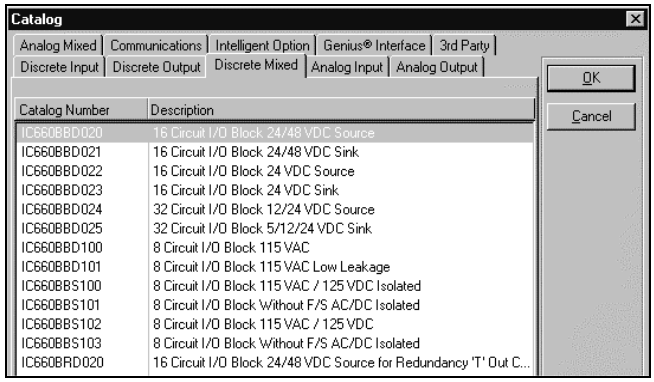
From here you can add, copy, cut, replace, paste, and delete modules on the bus.

Configuring the Serial Bus Address of a Device

1. When you add a device, select a Serial Bus Address for the Genius device.



2. After selecting the Serial Bus Address, the Device Catalog appears. Use the tabs to display lists of modules. For example:



3. Select the device to add at the selected Serial Bus Address and click OK. A configuration screen for that device will appear.

You must enter a configuration screen for each device. This assigns program references to the device's data and establishes certain system parameters. It does NOT configure the characteristics of the bus devices themselves. That separate configuration is normally done using a Hand-held Monitor, but may also be done using Communication Request instructions in the application program. For information about Communication Request instructions, see chapter 5. For a Series 90-70 PLC CPU, it may be possible to directly configure device parameters as described later in this chapter.

Configuring the Device Reference Address

A device's Reference Address is the beginning reference for its input and output data.

The software automatically assigns the next available reference address within a memory type. If the address displayed is not appropriate, a different address can be entered. Discrete references must begin on a byte boundary (a byte boundary is a number which is one greater than a multiple of 8, for example: 9, 17, or 25). If you assign a reference address out of sequence, the software will then continue to increment that number for additional modules. For example, if you assigned the reference %I0401 to the first input module and it had 16 circuits, the software would next assign %I0417 or %QI0417 to an input or combination block. You could change this to a different address. A message appears when the highest available address has been assigned, although you may have skipped lower addresses.

References for Devices having both Discrete and Word Inputs: For some bus devices (an example is the High-speed Counter block), the input data that is routinely broadcast by the block consists of BOTH discrete and word-type data. For such a block, the configured Reference Address represents three memory locations (in %I, %Q, and %AI memories) instead of the two (%I and %Q) assigned to other types of blocks. This is shown in the Settings tab illustrated below.

Settings

Parameters	Values
Reference Address	**%I00017**
Length	16
Reference Address	%Q00001
Length	16
Reference Address	**%AI00041**
Length	18
Input Default	Off
Output Enable	Yes
Redundancy	**Yes**

References for Inputs-only or Outputs-only Devices: An Inputs-only block uses one reference in %I or %AI memory for each circuit on the block. Similarly, a block with outputs only requires one reference in %Q or %AQ memory only.

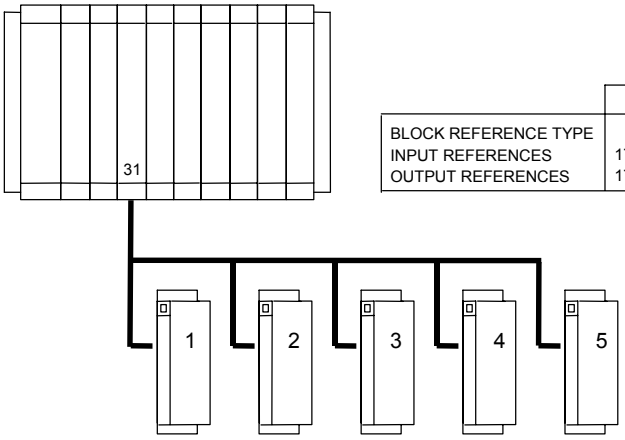
References for Mixed I/O Blocks with both Inputs and Outputs: A Genius I/O block that has both inputs and outputs uses the same number of input and output references, regardless of the block's actual I/O mix.

An analog block with 4 inputs and 2 outputs requires four words of analog input memory and four words of analog output memory. The block only uses the first two output words, but the second two output words cannot be used for outputs because they cannot be assigned by the configuration software. However, they can be used for internal registers in the application program.

References for Redundancy: During operation, the CPU handles I/O data the same way for redundant and non-redundant systems. For any redundant Bus Controller pair in the PLC, each CPU sweep the CPU receives a one set of bus inputs and sends one set of bus outputs. The CPU does *not* maintain two sets of references for devices that are set up for redundancy.

Reference Address Configuration Example

As the following examples show, it is not necessary to configure blocks in Serial Bus Address sequence. Serial Bus Addresses are unrelated to the assignment of Reference Addresses.



	1	2	3	4	5
BLOCK REFERENCE TYPE	QI	AQI	Q	AI	I
INPUT REFERENCES	17-24	1-4	–	5-10	1-16
OUTPUT REFERENCES	17-24	1-4	1-16	–	–

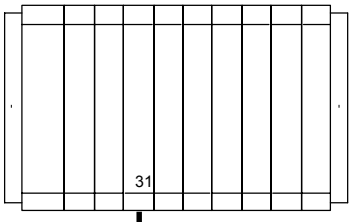
The bus has five blocks:

Serial Bus Address 1:	8-circuit Isolated I/O block
Serial Bus Address 2:	4 Input/2 Output analog block
Serial Bus Address 3:	16-circuit discrete Relay Output block
Serial Bus Address 4:	RTD analog block (6 inputs, no outputs)
Serial Bus Address 5:	16-circuit discrete Inputs-only block

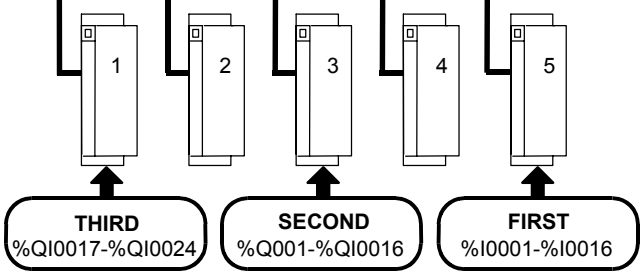
The order in which the blocks are configured determines their reference assignments.

Reference Address Example Configuration 1:

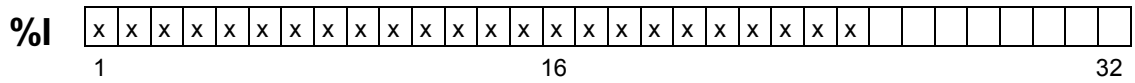
Here, the first block to be configured is Serial Bus Address 5, the 16-circuit Inputs-only block. The configuration software assigns to it %I0001. The second block configured is Serial Bus Address 3, the Relay Output block. The software assigns to it %Q0001. If Serial Bus Address 1, the 8-circuit Isolated I/O block, were configured next, the software would automatically assign it %QI0017.

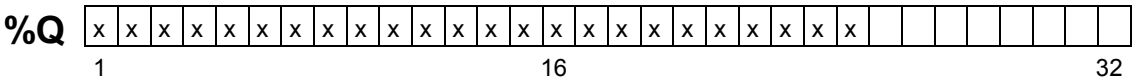


	1	2	3	4	5
BLOCK REFERENCE TYPE	QI	AQI	Q	AI	I
INPUT REFERENCES	17-24	1-4	–	5-10	1-16
OUTPUT REFERENCES	17-24	1-4	1-16	–	–



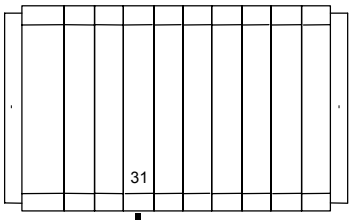
The assignment of %I and %Q memory would then be:



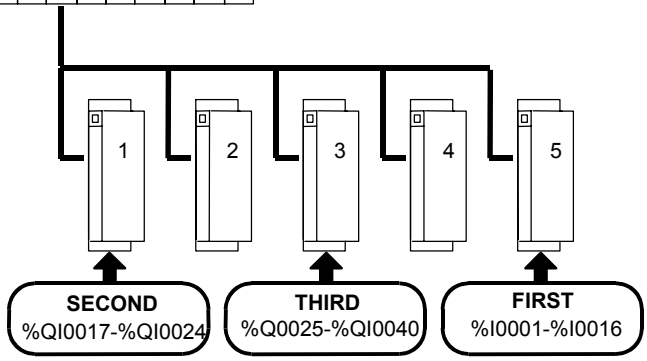


Reference Address Example Configuration 2:

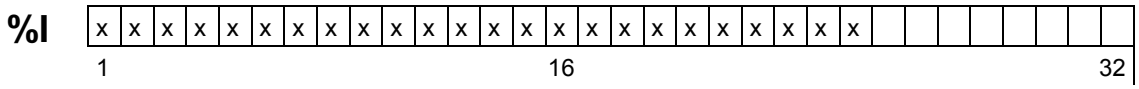
If the 16-circuit Input block were configured first, the 8-circuit Isolated block second, and the 16-circuit Relay block third, the software would not go back and assign reference address %Q0001 to the Relay block.

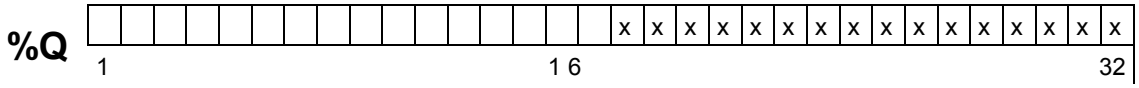


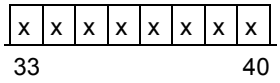
	1	2	3	4	5
BLOCK REFERENCE TYPE	QI	AQI	Q	AI	I
INPUT REFERENCES	17-24	1-4	–	5-10	1-16
OUTPUT REFERENCES	17-24	1-4	25-40	–	–



The automatic memory assignments in %I and %Q would be like this instead:







In this case, the Reference Address %Q0001 could be entered from the keyboard, resulting in the memory usage shown first.

Disabling Outputs

If outputs are disabled, the Bus Controller will not send output data from the CPU to the designated device(s). *Output Disable is not selectable for inputs-only devices. Inputs-only blocks are ALWAYS sent a dummy message to turn on their I/O Enabled LEDs.*

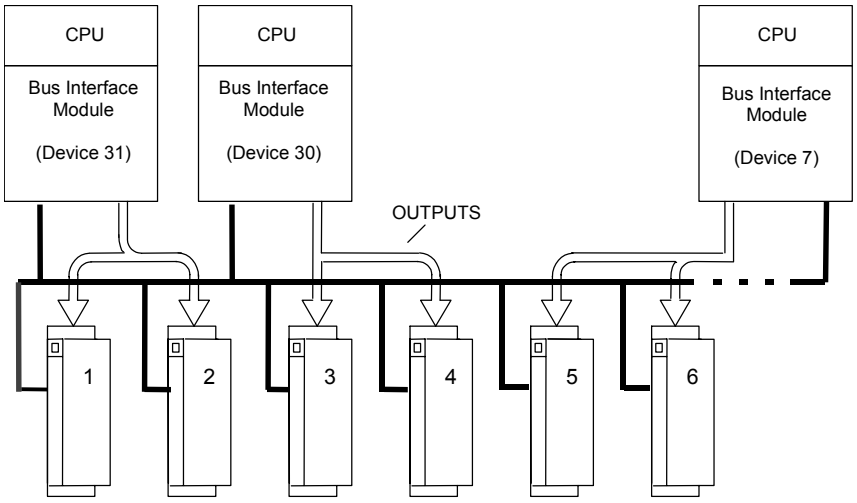
It is possible for outputs to be disabled or re-enabled using Communication Request instructions in the application program. If this capability will be needed, the outputs should be *enabled* during I/O configuration.

Ordinarily, the configuration software would be used to disable outputs that should remain disabled. To re-enable such inputs, it would be necessary to change the configuration and re-store the new configuration to the PLC.

Outputs might be disabled in a system where multiple CPUs are used for distributed control, or a system using the CPU as an assigned monitoring device. Examples are shown below.

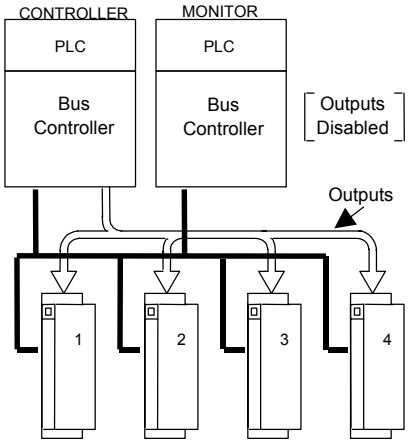
Example

Selectively Disabling Outputs for Distributed Control of I/O Blocks: Some systems use two or more CPUs on the same bus for distributed control of I/O blocks. In a distributed control system, each CPU sends outputs to (and receives fault reports from) certain blocks on the bus and not others. This is accomplished by selectively enabling or disabling outputs to the blocks.

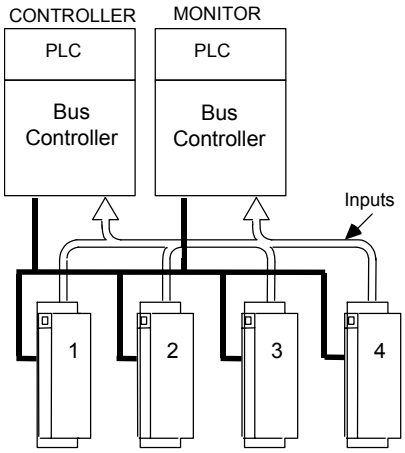


Example

Disabling Outputs for an Assigned Monitor: If the CPU will be used to monitor inputs from certain blocks on the bus, outputs to those blocks should be disabled. When being used as a monitor, the CPU will also receive fault reports and configuration change messages if the blocks have been sent Assign Monitor datagrams.



Output data for these blocks will be supplied by one or more other CPUs on the *same* bus.



If a CPU is used as a monitor, it may NOT have two of its Bus Controllers located on the same bus. Otherwise, the CPU would receive input data from both Bus Controllers for the same references, and internal system errors will result.

Bus Device Redundancy Configuration

If a bus device will be used in dual bus or dual controller mode, or both, set the entry for **Redundancy** to YES.

If **Redundancy** is set to YES for any device on a bus, the Bus Controller must also be configured for a form of redundancy: dual bus, redundant control, or dual bus/redundant control.

The configuration software will automatically attempt to supply a correct configuration when you set device **Redundancy** to YES:

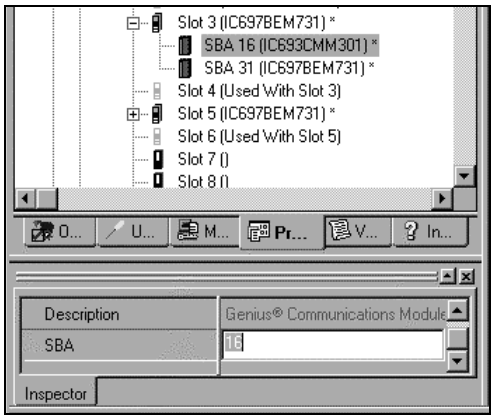
- If the Bus Controller is configured for a **Redundancy Mode** of NONE, and you set the **Redundancy** of any device on the bus to YES, the Bus Controller's configuration is automatically changed to **Redundancy**: DUAL BUS and **Paired GBC**: EXTERNAL.
- If the Bus Controller is configured for a **Redundancy Mode** of either DUAL BUS or Redundant Control (Redundant Controller), and **Paired GBC** is INTERNAL, each device on the bus is automatically configured at the same bus address (Serial Bus Address) on the redundant bus, and given the same reference addresses.

  If **Paired GBC** is set to EXTERNAL, the block is not automatically configured on the other bus of the pair.
- If the Bus Controller is configured for a **Redundancy Mode** of dual bus/redundant controllers each device on the bus is automatically configured at the same bus address (Serial Bus Address) on the redundant bus, and given the same reference address.

Changing a Device's Serial Bus Address

If you want to change the Serial Bus Address of any device that has already been configured on the bus (including the Bus Controller:

1. In the Project tab of the Navigator, expand the Hardware configuration folder.
2. Expand the Rack that contains the Bus Controller and expand the slot containing the Bus Controller.
3. Select the device you want to configure. Its Serial Bus Address is displayed in the Inspector.



Edit the Serial Bus Address as required. It can be any value from 0 through 31. You cannot change a bus device's Serial Bus Address to that used by the Bus Controller. If you change the selected device's Serial Bus Address to one already that is assigned to a device on the bus, the existing device will be replaced.

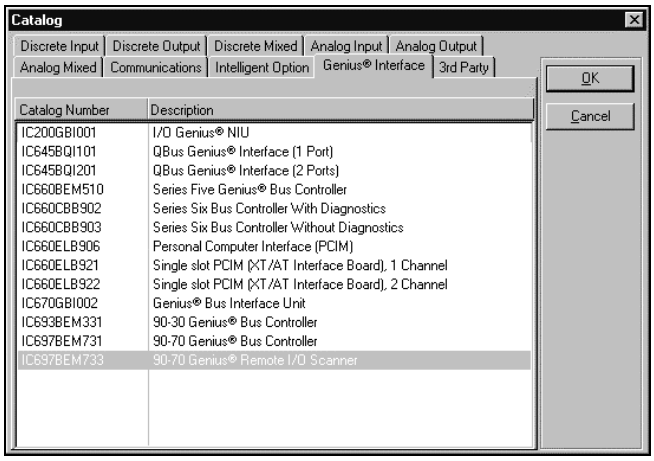
Configuring a Remote Drop

Enter or select the Serial Bus Address of the Remote I/O Scanner on the bus.

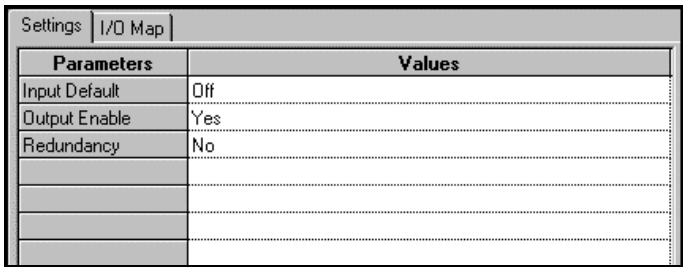
Note

If a Remote I/O Scanner has an SBA conflict on an operating bus, it will not scan the modules in the remote drop until the fault is cleared.

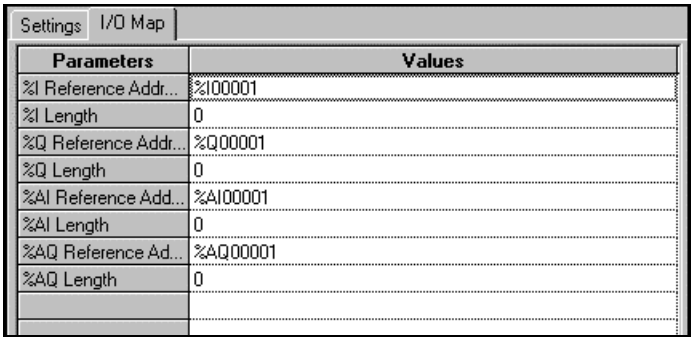
Select the Remote I/O Scanner from the Genius Interface tab:



The redundancy mode of the Remote I/O Scanner is automatically matched to the **Redundancy Mode** configuration of the Bus Controller.



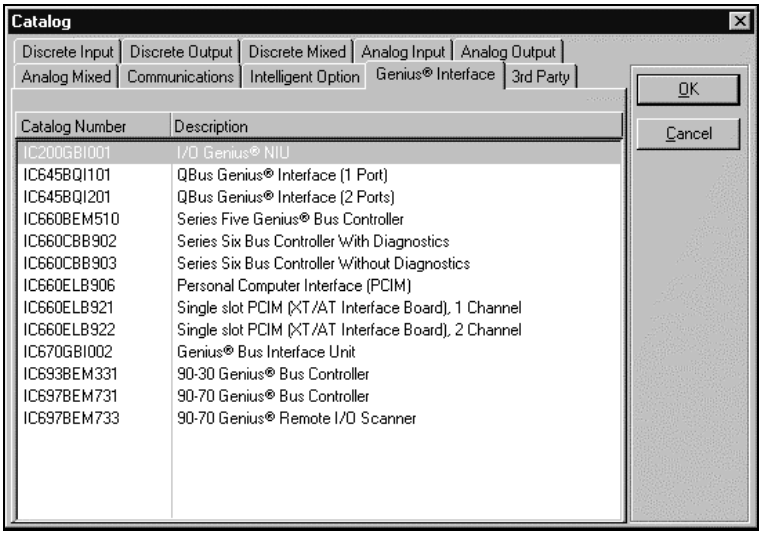
Please refer to the *Series 90-70 Remote I/O Scanner User's Manual* for information about selecting input defaults, enabling outputs, and defining the I/O mapping for the Remote I/O Scanner.



.

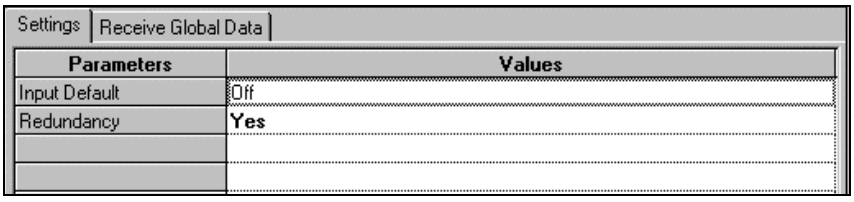
Configuring a Remote Bus Controller, Genius Communications Module, PCIM, QBIM, or GENI-based Device on the Bus

Select the Bus Controller or other device from the appropriate list.



Select the Settings

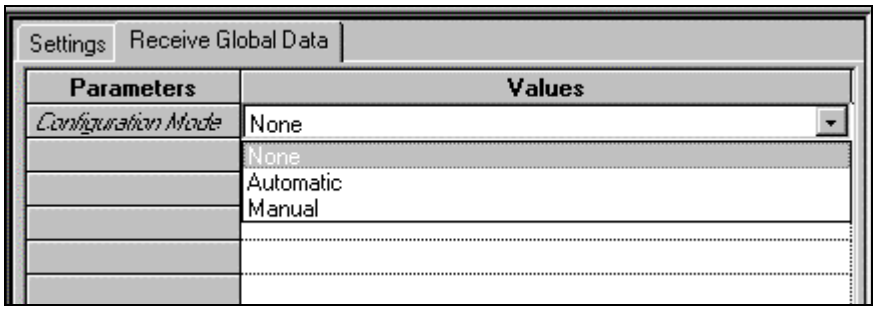
Select the device's communications properties.



If **Redundancy** is set to YES, the local Bus Controller must also be configured for a form of redundancy. The configuration software will automatically attempt to supply a correct configuration when you set device **Redundancy** to YES.

Set Up Global Data

If a Bus Controller on the bus will not accept Global Data from the Bus Controller being configured, its entry for **Config Mode** should be NONE.



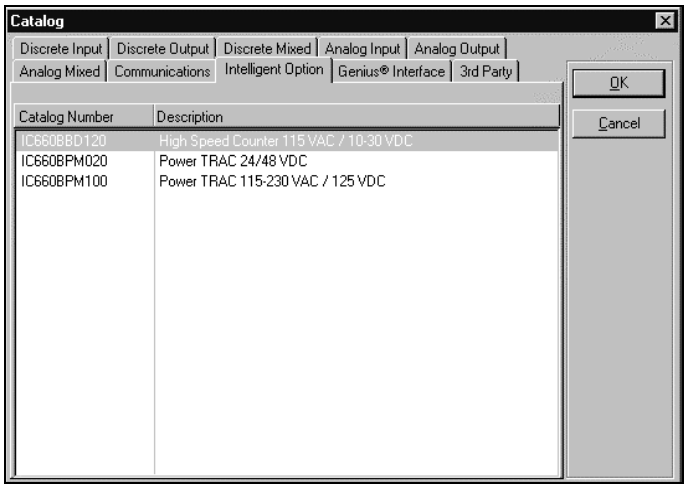
If it will accept the Global Data, select the CPU reference address for the data to be placed after it is received.

To: The beginning address in %I, %Q, %G, %AI, %AQ, or %R memory.

Input Length: The amount of Global Data expected to be received. For bit-oriented data, this is the number of bits. For word-oriented data, it is the number of words. If the expected data length (defined by configuration) and the actual data length (defined by the content of the Read ID Reply message from the module) don't agree, a System Configuration Mismatch fault is placed in the PLC Fault Table.

Configuring a PowerTRAC Block or High-speed Counter Block

Select the Intelligent Option tab. Select the High-speed Counter or correct PowerTRAC block type.



Configuring Parameters for a PowerTRAC Block

A PowerTRAC Block has both bit-type data and word-type data. You can select the beginning references for both. The required lengths are shown. You can also select the default state for the block's input data, and enable or disable CPU outputs to the block.

Settings

Parameters	Values
Reference Address	**%I00017**
Length	16
Reference Address	%Q00001
Length	16
Reference Address	**%AI00041**
Length	18
Input Default	Off
Output Enable	Yes
Redundancy	**Yes**

If **Redundancy** is set to YES, the Bus Controller must also be configured for a form of redundancy. The configuration software will automatically attempt to supply a correct configuration when you set device **Redundancy** to YES.

For more information about the content and format of PowerTRAC block data, please see the *PowerTRAC Block User's Manual*.

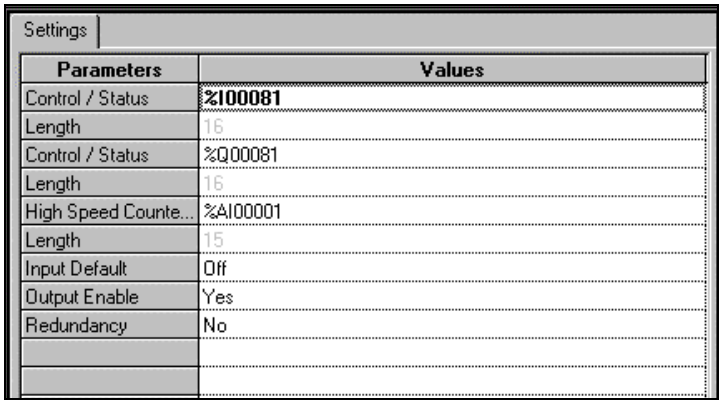
Configuring Parameters for a High-speed Counter Block

A High-speed Counter Block has both bit-type data and word-type data. You can select the beginning references for both.

Control/Status	The %QI reference location for the block's discrete I/O data. The length is fixed at 16 bits.
High Speed Counter Data	The %AI reference for the block's word data. The length is fixed at 15 words.

For more information about the content and format of this data, please see the *High-speed Counter User's Manual*.

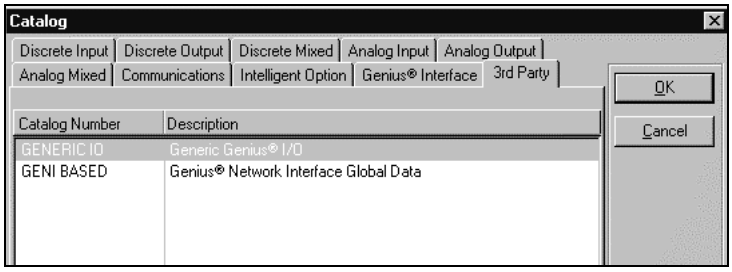
If **Redundancy** is set to YES, the Bus Controller must also be configured for a form of redundancy. The configuration software will automatically attempt to supply a correct configuration when you set device **Redundancy** to YES.



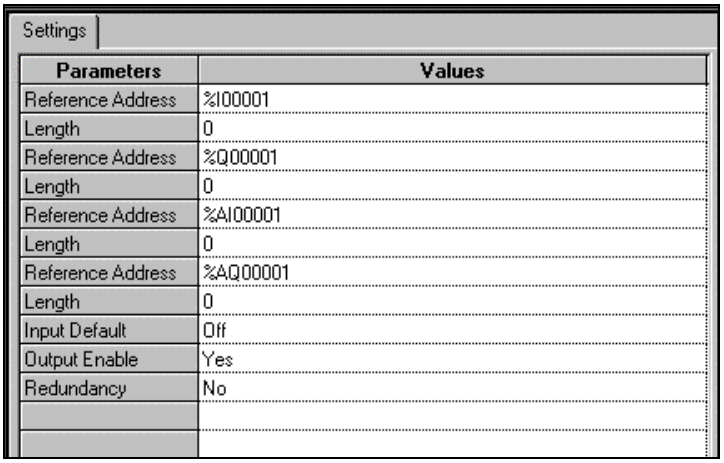
Configuring a Generic Device on the Bus

A device on the bus can be configured as a “generic” I/O device. This might be done to provide selections for “input defaults” and “outputs enabled” that are not otherwise available for a given Genius product, or to configure a device that is not included in the other menus.

To configure a generic device, select Generic Genius I/O from the menu of “3rd Party” modules.



Press the Enter key. A configuration screen like this will appear:



Select the beginning references and lengths for the module’s bit and word data. The combined lengths of bit and word inputs (%I and %AI) must exactly match the amount of data that will be sent by the device. The combined lengths of bit and word outputs (%Q and %AQ) must exactly match the amount of data that will be sent by the Bus Controller to the device. *If the generic device being configured is a Bus Controller, assign it INPUTS ONLY*.

You can also select the default state for the device’s input data, and enable or disable CPU outputs to the device.

If the generic module's **Redundancy** is set to YES, the local Bus Controller must also be configured for a form of redundancy. The configuration software will automatically attempt to supply a correct configuration when you set device **Redundancy** to YES.

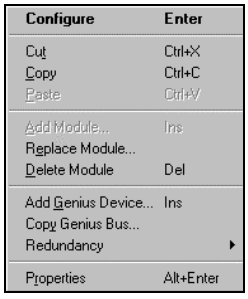
Copying a Bus Configuration

To copy a Genius bus configuration, the destination Bus Controller must already be configured. You can optionally choose whether the destination bus is cleared before the copy and whether all devices or only devices marked redundant are copied. The source Bus Controller defaults to the slot that is currently selected. If the source and destination are not properly paired, including Redundancy Mode and Serial Bus Address, the copy operation is not allowed.

Copying a Bus Configuration without Copying the Bus Controller Parameters

To copy only the bus configuration without copying the parameter settings of the Bus Controller:

1. In the Project tab of the Navigator, expand the Hardware configuration folder and then the Rack that contains the Bus Controller.
2. Right-click the slot containing the source Bus Controller and choose Copy Genius Bus.



3. The Copy Bus dialog box appears. Select the rack and slot location of the destination Bus Controller and click OK.

Copying a Bus Configuration and the Bus Controller Configuration

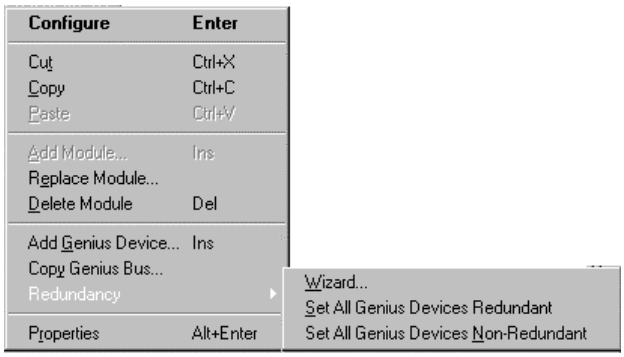
To copy a bus configuration with the same Bus Controller parameters:

1. Right-click the slot containing the Bus Controller and choose Copy.
2. Then, right-click the destination slot and choose Paste.

The Bus Controller parameter settings and the bus settings are copied.

Setting All Bus Devices to Redundant or Non-Redundant

1. In the Project tab of the Navigator, expand the Hardware configuration folder and then the Rack that contains the Bus Controller.
2. Right-click the slot containing the Bus Controller and choose either Set All Devices Redundant or Set All Genius Devices Non-Redundant:



3. Devices that have a Redundancy parameter will be set accordingly.

Programmer Configuration of Bus Devices

For a Series 90-70 PLC only, it is possible to download parameters and data points to some types of Genius devices over a functioning bus. You can also upload block parameters and points data from the device, verify the equality of the device, and protect or unprotect the device configuration.

Devices that Support Programmer Configuration Features

The Genius devices that support these functions with a Series 90-70 PLC are:

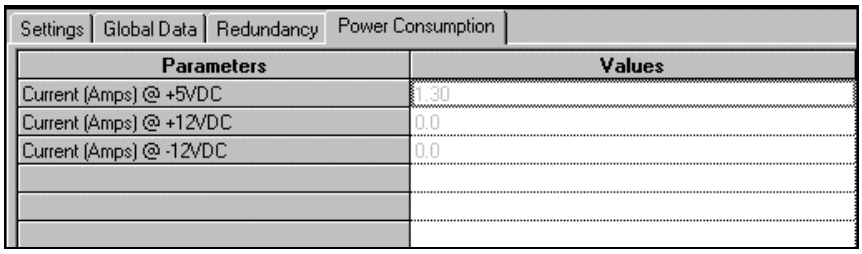
Discrete Input Modules	IC660BBD110
Discrete Output Modules	IC660BBR100, IC660BBR101
Discrete Mixed Modules	IC660BBD020, IC660BBD021, IC660BBD022, IC660BBD023, IC660BBD024, IC660BBD025, IC660BBD100, IC660BBD101, IC660BBS100, IC660BBS101, IC660BBS102, IC660BBS103
Analog Input Modules	IC660BBA101, IC660BBA103, IC660BBA106, IC660BBA021, IC660BBA023, IC660BBA026
Analog Output Modules	IC660BBA105, IC660BBA025
Analog Mixed Modules	IC660BBA100, IC660BBA104, IC660BBA020, IC660BBA024
Intelligent Option Modules	IC660BBD120, IC660BPM020, IC660BPM100
Genius Bus Controller	IC697BEM731

Programmer Instructions for Genius Devices

1. If the Hardware Configuration is unequal, download the Hardware Configuration to the PLC.
2. In the Project tab of the Navigator, expand the Hardware configuration folder.
3. Expand the Rack that contains the GBC and expand the slot containing the GBC.
4. Right-click the Genius device and choose:
   - ***Download Genius Configuration***.
   - ***Upload Genius Configuration***
   - ***Verify Genius Configuration***
   - ***Protect Genius Device***. If this operation is successful, neither a Hand-held Monitor nor programmer can change any parameter on the Genius device until you unprotect it
   - ***Unprotect Genius Device***. If this operation is successful, any parameter on the Genius device can be changed with a Hand-held Monitor or programmer.
5. For some operations, a confirmation dialog box appears. Click Yes.
6. The results of the operation are displayed in the Feedback Zone.

Viewing the Bus Controller Power Consumption

The Power Consumption tab shows the power consumed by the Genius Bus Controller. This information is a read-only.



Chapter 4 Diagnostics

This chapter describes the following diagnostics capabilities in a system that uses Genius I/O and communications:

- Relevant system status references.
- Fault and No Fault Contacts, which can be used with program references or with the built-in fault-locating references.
- High Alarm and Low Alarm Contacts, which will indicate when an analog reference has reached one of its alarm limits.
- Fault Table faults related to the Bus Controller and Genius devices.

For Additional Information, Also See:

Chapter 1 for an overview of Genius fault reporting.

Chapter 5, which describes the use of Read Diagnostics, Clear Circuit Faults, and Clear All Circuit Faults COMMREQs (Communication Requests).

Chapter 7 to learn how the Bus Controller's redundancy capabilities can optionally be utilized to check its I/O and diagnostics data memory when it is not used with a dual bus or as a dual controller.

System Status References

System status references are pre-defined locations and nicknames. They can be included in an application program to check for fault-related conditions. The following system status references are of special interest for a system with a Bus Controller:

Reference	*Nickname*	*Conditions Indicated When Set*
%SA0009	CFG_MM	System Configuration Mismatch
%SA0012	LOS_RCK	Loss of Rack
%SA0013	LOS_IOC	Loss of Bus Controller
%SA0014	LOS_IOM	Loss of I/O module
%SA0017	ADD_RCK	Addition of Rack
%SA0018	ADD_IOC	Addition of Bus Controller
%SA0019	ADD_IOM	Addition of I/O module
%SA0022	IOC_FLT	Bus fault or Bus Controller fault
%SA0023	IOM_FLT	I/O module fault
%SA0029	SFT_IOC	Bus Controller software failure
%SB0016	MAX_IOC	Too many Bus Controllers (maximum is 31)
%SC0011	IO_FLT	I/O fault occurred
%SC0013	IO_PRES	Fault logged into I/O Fault Table
%S00010	IO_FULL	I/O Fault Table is full

These references and their Nicknames can be used like any other type of reference.

Example:

A PLC system includes one Bus Controller. During CPU configuration, the system status fault **LOS_IOC** has been designated a diagnostic (rather than fatal) fault. **LOS_IOC** represents loss of the Bus Controller; if this occurs, the Loss of IOC fault will be placed in the I/O Fault Table. In this example, the application program also monitors the **LOS_IOC** reference. If this reference is set, the contact passes power flow to an output coil, which energizes a warning light on an operator panel.

Fault and No Fault Contacts

FAULT and NO FAULT contacts can be used to detect fault or lack of fault conditions on a discrete (%I or %Q) or analog (%AI or %AQ) reference, or they can be programmed with fault locating references (see below). Unless they are used ONLY with fault-locating references, fault memory for their use must be set up as part of the CPU configuration.

A FAULT contact will detect a fault in a discrete or analog input or output, or a hardware component of the system. The contact passes power flow if the reference has a fault.

Example:

When used with a %I, %Q, %AI, or %AQ reference, a fault associated with the FAULT contact must be cleared to remove it from the fault table and stop the contact passing power flow. *Clearing such a fault with a Hand-held Monitor does not remove it from the fault table or stop the contact passing power flow.*

NOFAULT contacts will also detect faults in discrete or analog inputs and outputs. A NO FAULT contact passes power flow if its associated reference does not have a circuit fault.

Example:

Fault Locating References

Both FAULT and NO FAULT contacts can be programmed with fault-locating references to identify faults associated with system hardware. These fault references are for informational purposes only. The PLC does not halt execution if one of these reference faults occurs. For a Genius device in a Series 90-70 system, the format of the fault-locating reference is **M_rsbmm** In an RX7i system, the format is **M_rssbmmm**. In these formats, **r** is the rack number 0 to 7, and **s** is the slot number of the Bus Controller; **b** is the bus number, and **mm** is the Device Number (serial bus address) of the affected Genius device (00 to 31). For example, **M_46128** represents rack 4, slot 6, bus 1, module 28.

How FAULT and NO FAULT Contacts Handle Bus Controller Faults in an RX7i System

For an RX7i system (only), if a Bus Controller failure fault is reported to the fault tables, all bus and module fault-locating references associated with that Bus Controller are set. The FAULT contact passes power flow and the NO FAULT contact does not pass power flow. When a Bus Controller reset fault is reported to the fault tables, all of the bus and module fault-locating references are cleared. The FAULT contact does not pass power flow and the NO FAULT contact passes power flow.

High Alarm and Low Alarm Contacts

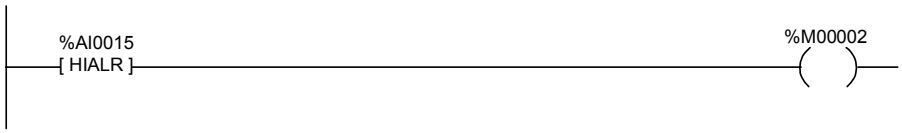
The HI ALARM and LO ALARM contacts will indicate that an analog reference has reached one of its alarm limits. These alarm limits are established when a device is configured. If an alarm limit is reached, a block or Remote I/O Scanner sends the high alarm or low alarm message to the Bus Controller. Analog alarms are not considered fault conditions. This information is ignored by the FAULT and NO FAULT contacts, as explained on the previous page.

Example

The analog input assigned to reference %AI00015 has been configured to have the following Alarm Limits:

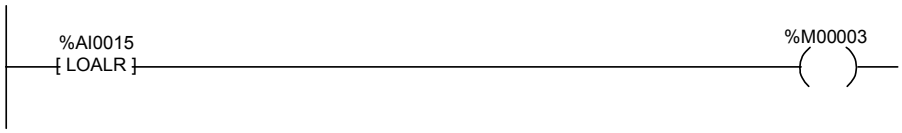
150 ft/sec	High Alarm
25 ft/sec	Low Alarm

If the input exceeds a rate of 150 feet per second, a HI ALARM contact energizes internal coil %M00002.



Example

If the same analog input slows to a rate of 22 feet per second, its LO ALARM contact energizes internal coil %M00003.



Fault Table Definitions for Genius Devices

Faults and alarms from I/O devices, Bus Controller faults, and bus faults are automatically logged into the I/O Fault Table.

Clearing Faults in the Fault Table

You must clear the I/O Fault Table from the programmer for the fault to be cleared in the PLC CPU and for the associated fault contact to be cleared. *Clearing faults with a Hand-held Monitor alone does not remove them from the Fault Table, or cause any associated* **-[FAULT]-** *contacts to stop passing power flow.*

Clearing the Fault Table causes the Bus Controller to send a Clear All Circuit Faults background message to all blocks on the bus. Faults can be cleared from the Fault Table either from the programmer screen or by the application program.

Clearing the Fault Table removes the faults it contains; it does not clear fault conditions in the system. If the condition that caused a fault still exists and is detected, the fault will be reported again.

Removing I/O Force Messages from the I/O Fault Table

When a point is forced on a Genius block with a Hand-held Monitor, a fault is registered in the I/O Fault Table. Subsequent forces on the same block do not generate additional messages. Only when *all* forces are removed from the block does the Bus Controller log an Unforce message in the I/O Fault Table.

Loss of Device Faults Caused by High Bus Error Rate

If the bus is experiencing a high error rate (possibly due to electrical interference or damaged cable), Loss of Device faults may be logged into the Fault Table. Loss of Device faults that are logged in conjunction with I/O Bus Faults can be usually be attributed to the poor quality of the bus installation. The condition causing the bus errors should be corrected as soon as possible.

Fault Table Definitions Associated with Genius Devices

Fault Category	*Diag. or Fatal*	*Indicates*	*Fault Type*	*Indicates*	*Fault Description*	*Indicates*
CIRCUIT FAULT	D	Short circuit, open wire, etc.	DISCRETE FAULT	Circuit fault on discrete I/O point	LOSS POWER	Loss of user side power
					SHORT CIRCUIT	Short in user wiring
					OVERLOAD	Sustained overcurrent
					NO LOAD	Very low or no current flow
					OVER TEMP	Switch temperature too high
					SWITCH FAIL	Genius “smart switch” failure
					POINT FAULT	Integral individual point fault
					FUSE BLOWN	Integral output fuse blown.
			ANALOG FAULT	Fault on analog I/O channel	AI LOW ALARM	Input channel low alarm
					AI HI ALARM	Input channel high alarm
					AI UNDER RANGE	Input channel under range
					AI OVER RANGE	Input channel over range
					OPEN WIRE	Open wire detected on input channel
					AQ UNDER RANGE	Output channel under range
					AQ OVER RANGE	Output channel over range
					CS FEEDBACK ERR	Feedback error from Current-source Analog block
			GENA FAULT	Fault on a GENA	GENA CKT FLT	Fault on a GENA analog or discrete point
			LL ANALOG FAULT	Fault on a low-level analog channel	AI LOW ALARM	Input channel low alarm
					AI HI ALARM	Input channel high alarm
					AI UNDER RANGE	Input channel under range
					AI OVER RANGE	Input channel over range
					OPEN WIRE	Open wire detected on input channel
					WIRING ERROR	Improper RTD connection or thermocouple reverse junction fault
					INTERNAL FAULT	Cold junction sensor fault on thermocouple block, or internal error in RTD block.
					INPUT SHORT	Input channel shorted
			REMOTE FAULT	Fault on a Remote I/O Scanner	n/a	Any fault detected by a Remote I/O Scanner and sent to the PLC.

Fault Table Definitions Associated with Genius Devices (continued)

Fault Category	*Diag or Fatal*	*Indicates*	*Fault Type*	*Indicates*	*Fault Description*	*Indicates*
LOSS OF DEVICE	D	Block no longer responding.	NOT SPEC FAULT	No reason specified		
			AD COMM FAULT	Loss of A/D communications.		
ADDITION OF DEVICE	D	New block appeared				
GENIUS BUS SWITCH	D	Redundant bus switched				
I/O BUS FAULT	D	Genius bus fault	BUS FAULT	Genius bus fault		
			BUS OUT DISABLE	Bus Controller disabled all outputs on the bus because communications timed out between the PLC CPU and the Bus Controller.		
			SBA CONFLICT	Bus Controller's Device Number duplicated elsewhere on bus.		
I/O MODULE FAULT	D	EEPROM fault, watch dog timeout	HEADEND FAULT	Block Fault (EEPROM, Watchdog, etc..)	CONFIG MEM FAIL	Genius EEPROM or NVRAM failure
					CAL MEM FAIL	Genius calibration memory failure
					SHARE RAM FAIL	Genius Shared RAM fault
					INTRNAL CKT FLT	Genius internal circuit fault
					WD TIMEOUT	Watchdog Timeout (discrete I/O modules only)
					POINT FAULT	Point fault (also indicated for CIRCUIT FAULT category)
					FUSE BLOWN	Integral output fuse blown (also indicated for CIRCUIT FAULT category)
			A TO D COMM FAULT	Analog to digital communications fault or calibration error		
			USER SCALING ERROR	Scaling error cause out of range values		

Fault Table Definitions Associated with Genius Devices (continued)

Fault Category	*Diag. or Fatal*	*Indicates*	*Fault Type*	*Indicates*	*Fault Description*
ADDITION OF IOC	D	Addition of Bus Controller			
LOSS OF IOC	F*	Loss of or missing Bus Controller			
BUS CONTROLLER SOFTWARE FAULT	F *	Bus Controller software fault			
FORCED CIRCUIT	D	Genius I/O point forced (eg: from Hand-held Monitor)			
UNFORCED CIRCUIT	D	Last forced circuit released (eg: from Hand-held Monitor)			
EXTRA DEVICE	D	Found extra device on Genius bus			
EXCESSIVE FAULTS	D	Bus Controller has stopped reporting faults because too many have occurred	HIGH ERROR RATE	Bus Controller has dropped off the bus for at least 1.5 seconds.	
GBC SOFTWARE EXCEPTION	D	Bus Controller software exception	DG QUEUE FULL	Incoming datagram queue is full	
			RW QUEUE FULL	The queue for Read/Write requests in the Bus Controller is full. The requests may be from the Genius bus or from COMREQs.	
			LP MAIL REJECTED	The low-priority mail queue from the Bus Controller to the PLC is full. The response to the PLC was lost.	

* May be configured as (D)iagnostic, *particularly in a redundant system.*

Chapter 5 Communication Requests

This chapter explains how to use Communication Requests to:

- Pulse Test outputs on Genius blocks
- Read the configuration of a device on the bus, or the Bus Controller
- Write configuration data to a device on the bus or a Bus Controller
- Assign a device on the bus to monitor fault reports from Genius blocks
- Clear a circuit fault on the bus
- Clear all faults on the bus
- Switch a Bus Switching Module
- Read diagnostics (faults) from a device on the bus or the Bus Controller
- Read up to 64 words of data from a device on the bus
- Write up to 64 words of data from the CPU to a device on the bus
- Enable/disable all outputs from the Bus Controller to devices on the bus
- Enable/disable the Bus Controller's ability to receive or transmit Global Data
- Enable/disable I/O fault categories
- Send a datagram to a device on the bus.
- Send a datagram to a device which then sends a reply datagram.
- Transfer an unsolicited incoming datagram from the Bus Controller to the CPU.
- Write up to 2 words of output data to each device on the bus.
- Change the first two words of Global Data sent by the Bus Controller.
- Read the Serial Bus Address of the Bus Controller.

For Additional Information, Also See:

The *Genius I/O System and Communications Manual* (GEK-90486-1), which describes Genius Datagrams in detail.

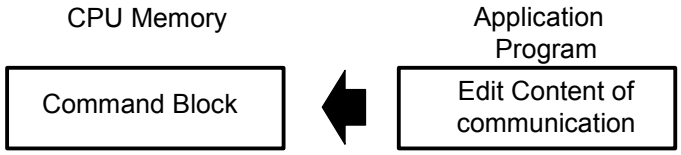
COMMREQs and Passwords

Level 1 and 2 passwords, which prevent write access, cannot be used in applications that include COMMREQs. COMMREQs require write access to return their completion status.

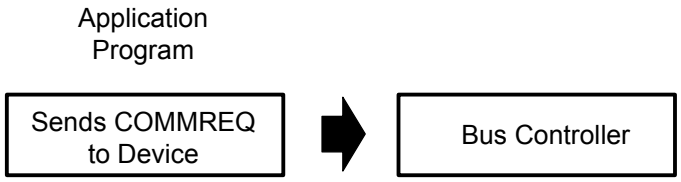
Programming for a Communication Request

In order to communicate with an intelligent module (such as a Bus Controller), the application program should perform the following three actions.

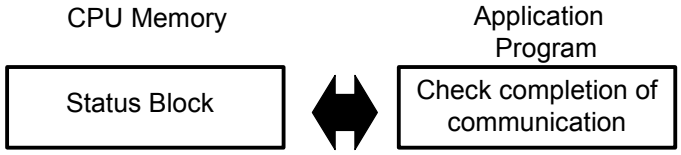
First, the program must supply the content of the communication. Block Moves or similar program instructions can be used to place the information into CPU memory. This content is called the Command Block.



Second, the program must use a COMMREQ instruction to perform the intended function.



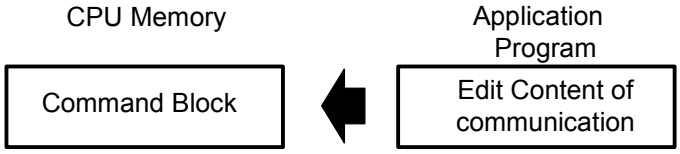
Third, the program should check the status of the requested task by looking at an area of CPU memory that is referred to as the Status Block.



COMMREQs should be executed sequentially. The application program should check the status of the previous COMMREQ to a Bus Controller before sending it another one. Failure to do this may result in improper operation of the Bus Controller.

COMMREQ Command Block Format

The first step in programming communications requests is to set up the contents of the communication. This can be done using Block Moves or similar program instructions, as shown later in this chapter.



Data is placed together in adjacent locations in CPU memory to form a Command Block.

Location	*Data*
address	"Data Block" Length
address + 1	Wait/No Wait Flag
address + 2	Status Pointer Memory Type
address + 3	Status Pointer Offset
address + 4	Idle Timeout Value
address + 5	Max. Communication Time
address + 6 to address + 70	Data Block

The length of the Command Block depends on the type of COMMREQ being sent. 70 words is the maximum, for a COMMREQ that transfers a 128-byte datagram; most Command Blocks are much shorter. A table on page 5-5 gives an overview of the contents of each type of COMMREQ that may be sent to a Series 90-70 Genius Bus Controller.

Command Block Contents

Command Block contents are described below:

Length: The first word of the Command Block indicates the "data block" length. This is the amount of data from **[address + 6]** to the end of the Command Block Each type of COMMREQ command has a unique Data Block, as shown in this chapter.

Wait/No Wait Flag: This must be set to 0 for No Wait.

Status Pointer Memory Type: The Status Pointer Memory Type and Offset (see below) identify the location of the function's associated Status Block. The Status Block is where the COMMREQ will return its status. If one of the bit-oriented memories (%I or %Q) is used as the status location, its bits can be monitored (see page 5-9).

Location	*Data*
address + 1	Status Pointer Memory
address + 2	Status Pointer offset

The high byte of address + 2 of the pointer is not used; it must be zero. The low byte of address + 2 specifies the type of memory where the Status Pointer will be located.

For This Memory Type:		*Enter This Number:*
%I	discrete input table	70
%Q	discrete output table	72
%R	register memory	8
%AI	analog input table	10
%AQ	analog output table	12
%W	Bulk memory (RX7i only)	196

Status Pointer Offset:

Address + 3 of the Command Block contains the address within the memory type selected. The offset of the status location is 0-based. For example, if the Status Block were located at %R099, memory type would be specified as 08 (for %R memory) and the offset would be 98. If %W memory is used, the address must be in the range %W00001 to %W65536.

Idle Timeout Value:

This field is not used for the No Wait mode of communication.

Maximum Communication Time:

This field is not used for the No Wait mode of communication.

Data Block:

The Data Block contains the parameters of the command. Complete descriptions of all commands appear later in this chapter. The Data Block begins with a Command Number in Address +6. The Command Number identifies the type of communications function to be performed. The following Command Numbers are used for the Genius Bus Controller:

Command	*Function*
1	Pulse Test Outputs
2	Read Configuration
3	Write Configuration
4	Read Diagnostics
5	Clear Circuit Fault
6	Clear All Circuit Faults
7	Assign Monitor
8	Outputs enable/disable
9	Global Data enable/disable
10	Switch BSM
11	Read Device
12	Write Device
13	Dequeue Datagram
14	Send Datagram
15	Request Datagram Reply
16	I/O Fault Category enable/disable

Commands 14 and 15 are used to send Datagrams. Most of the other commands listed above can also be sent as datagrams. For more information, see page 5-12.

Command Block Quick Reference

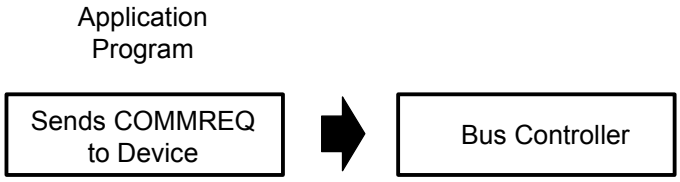
This table summarizes the content of the COMMREQ commands for a Bus Controller.

COMMREQ Description	***Command Block Content***									
	Add.	***Add. +1***	***Add.+2***	***Add.+3***	***Add.+4***	***Add.+5***	***Add.+6***	***Add.+7***	***Add.+8***	***to Add.+n***
	Length	***Wait/ No Wait***	***Status Pointer Memory Type***	***Status Pointer Offset***	***Idle Timeout Value***	***Max. Comms. Time***	***COMMR EQ #***	***Additional Content***		
Pulse Test	2	0	”	”	0	0	1	Device SBA*		
Read Configuration	5	”	”	”	”	”	2	Add.+7 to Add.+10: see page 5-14		
Write Configuration	data length +3	”	”	”	”	”	3	Add.+7 to Add.+n: see page 5-15		
Read Diagnostics	5	”	”	”	”	”	4	Add.+7 to Add.+n: see page 5-16		
Clear Circuit Fault	3	”	”	”	”	”	5	Device SBA*	circuit number	
Clear All Circuit Faults	2	”	”	”	”	”	6	”		
Assign Monitor	3	”	”	”	”	”	7	”	Monitor SBA* (0-13)	
Outputs Enable, Disable	0	”	”	”	”	”	8	”	1 (enable) or 0 (disable)	
Global Data Enable, Disable	3	”	”	”	”	”	9	”	”	
Switch BSM	3	”	”	”	”	”	10	”	0 (bus A) or 1 (bus B)	
Read Device	16	”	”	”	”	”	11	Add.+7 to Add.+21: see page 5-23		
Write Device	13 to 77 words	”	”	”	”	”	12	Add.+7 to Add.+n: see page 5-28		
Dequeue Datagram	7	”	”	”	”	”	13	Add.+7 to Add.+12: see page 5-29		
Send Datagram	6 to 70 words	”	”	”	”	”	14	Add.+7 to Add.+n: see page 5-32		
Request Datagram Reply	10 to 78 words	”	”	”	”	”	15	Add.+7 to Add.+n: see page 5-35		
I/O Faults Enable, Disable	3	”	”	”	”	”	16	Device SBA*		

* Serial Bus Address (Device Number)

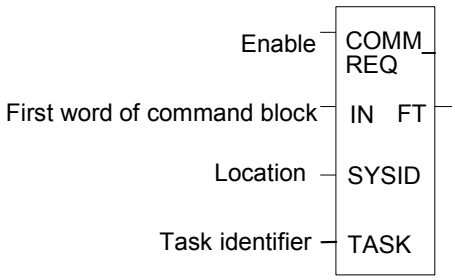
The COMMREQ Instruction

After supplying the content of the communication in the Command Block, the application program uses a COMMREQ instruction to request communications with the Bus Controller.



COMMREQ Inputs and Outputs

The COMMREQ instruction has four inputs and two outputs:



COMMREQ Inputs

(enable) Permissive logic that controls power flow to the COMMREQ function block.

IN: The memory location of the Command Block, which contains the specific command information. The Command Block may be located in any word-oriented area of memory (%P, %L, %R, %AI, or %AQ).

SYSID: A hex value that gives the rack and slot location of the Bus Controller. Use this format:

Examples

Rack	*Slot*	*Hex word value*
0	4	0004h
7	2	0702h

TASK: For Bus Controller version IC697BEM731, the task is always “1”.

COMMREQ Outputs

The function’s OK and FT outputs can provide power flow to optional logic which can verify successful completion of the COMMREQ. The OK and FT outputs may have these states:

ENable	*Error?*	*OK output*	*FT output*
active	no	true	false
active	yes	false	true
not active	no execution	false	false

The OK and FT outputs are never both true at the same time; OK indicates correct execution while FT indicates a fault condition. The COMMREQ passes power flow to OK unless:

- The specified Device Number (serial bus address) is not present.
- The specified task is not valid for the device. This is not checked if the specified device is a Genius Bus Controller.
- The data length is zero.

If any fault above occurs, the function passes power flow to FT instead.

If there are errors in the portion of the Command Block used specifically by the Bus Controller (for example, the Device Number entered is incorrect), these errors are reflected in the value returned in the status location, not in the FT output.

COMMREQ Status Block

When the Bus Controller receives the communication from the CPU, it returns its current status to the CPU at the memory location reserved for that purpose. This memory location is referred to as the "Status Block". Possible status values that may be returned are listed on the next page.

When a command is complete, the Bus Controller writes any resulting data into the area designated in the command, and sets the status to Complete (4).

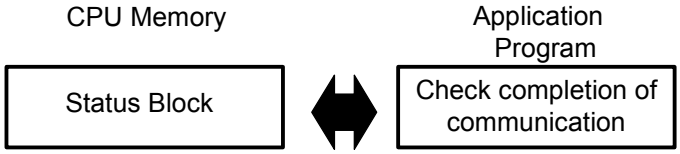
Note

Because COMMREQs require write access to return their status, level 1 and 2 passwords, which prevent write access, cannot be used with COMMREQs.

If one of the bit-oriented memories (%I or %Q) is used as the status location, its bits can be monitored. These bits correspond to the binary values listed below. For example, if %I048 were selected as the beginning location, reference %I050 would be set to 1 each time the COMMREQ completed successfully.

Clearing the Status Block

COMMREQs to the Bus Controller should be executed sequentially. Before sending a COMMREQ to the Bus Controller, the application program should check the status of any previous COMMREQ to that Bus Controller.



When the previous COMMREQ has completed, the program should set the Status Block to a value not in the list on the next page. Establishing this initial condition allows the program to differentiate between the result of an earlier command and the currently-executing command.

Contents of the Status Block

The Status Block is two words of memory to which the Bus Controller returns the status of the COMMREQ.

The lower word is used for general information about the execution of the COMMREQ:

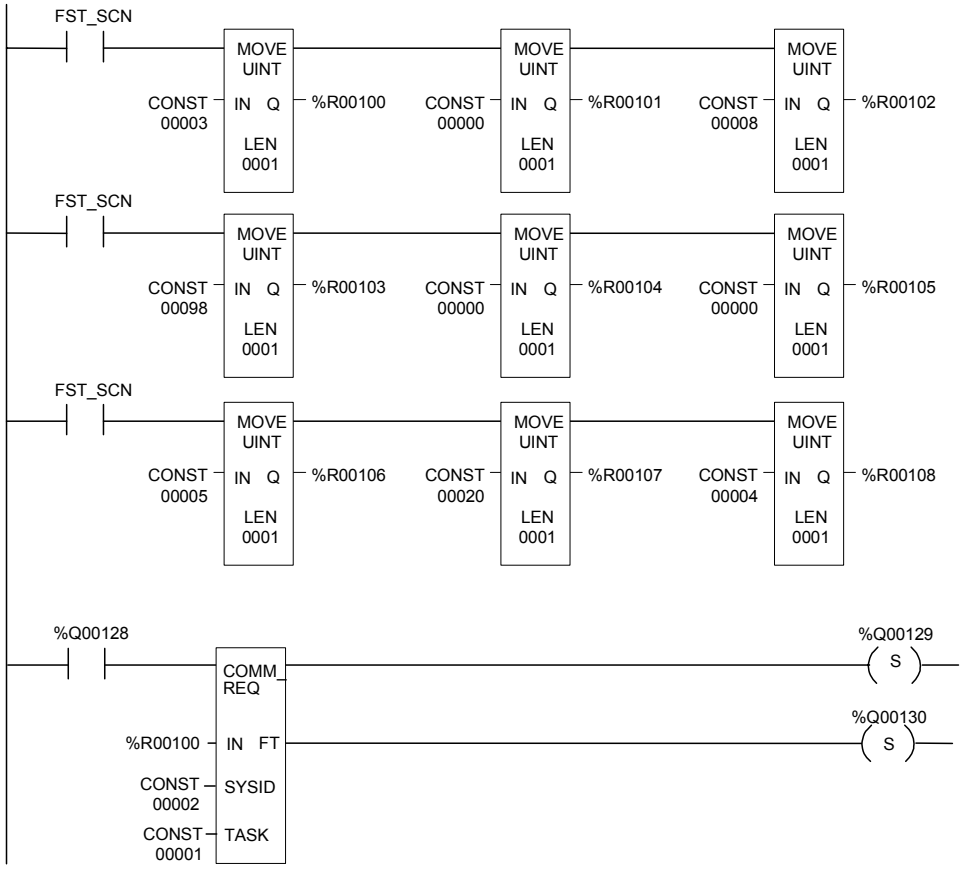
VALUE		DESCRIPTION
decimal (word)	binary (bit)	
	MSB	
0	000000000000	Bus Controller busy
1	000000000001	Command not accepted, Bus Controller busy with previous request
4	000000000100	Command completed successfully
8	000000001000	Command terminated due to syntax error
16	000000010000	Command terminated due to data error
32	000000100000	Command terminated due to suspended activity on bus
64	000001000000	No data to transfer
128	000010000000	Command not supported by target device
256	000100000000	Only No Wait commands may be sent to the target device
512	001000000000	Maximum Comms. Time must be greater than or equal to 5ms
1024	010000000000	Text buffer invalid in wait mode
2048	100000000000	Device did not accept the message, or timed out.

The upper word of the status location provides additional status information.

VALUE decimal (word)	DESCRIPTION
11	Non-discrete block specified for Pulse Test
21	Non-I/O device specified for Read Configuration
51	Invalid circuit number
71	Non-controller device specified for Assign Monitor
101	Switch BSM - device not BSM
102	Switch BSM - bus position greater than 1
121	P and L access not available
141	Function code greater than 111
142	Sub function code greater than 255
143	Priority greater than 1
144	Datagram length greater than 134
201	Invalid Device Number (greater than 31, but not 255)
202	Incorrect length for the command type
203	Device Number not configured or not active
204	Previous No Wait command in progress; current No Wait command not accepted
205	Invalid status pointer location specified
206	Command number is out of range
207	Subcommand code is out of range
208	Only partial data transferred
209	Device Number 255 not allowed for this command
210	Command specified is not valid for Genius Bus Controller
211	Command specified is only valid for controller devices
212	Command specified is not supported by the device to which it was sent
213	Invalid Alarm Enable/Disable mask

Programming Examples

The following example shows how a Communication Request can be used to clear a circuit fault on point 4 of a Genius I/O block whose Device Number is 20.

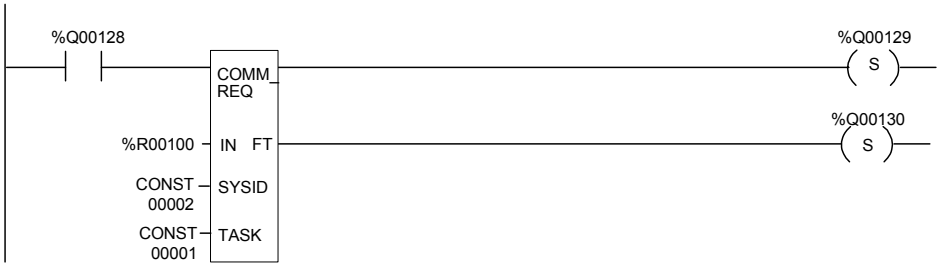


This example logic uses a series of Move instructions to assemble the data that will be used as inputs for the Communication Request instruction, and for its associated Command Block.

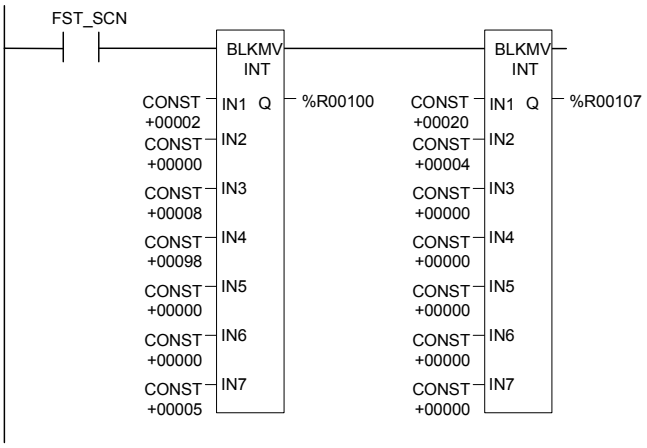
Address	*Contents*	*Value*	*Description*
%R100	Command length	3	No wait
%R101	Wait/No Wait Flag	0	Selects %R memory type
%R102	Status Pointer Memory	08	Address in %R memory (%R099)
%R103	Status Pointer Offset	98	Unused (No Wait selected)
%R104	Idle Timeout Value	0	Unused (No Wait selected)
%R105	Max.Communication Time	0	Clear Circuit Fault
%R106	Command Number	5	Device Number of the block
%R107	Device Number	20	Clear 4th point on block (For a COMMREQ, points
%R108	Point to be cleared	4	are numbered starting at 1 (not 0). If this were a datagram message instead of a COMMREQ command, points would begin at 0).

The Move instructions are executed during the first CPU sweep, when the special reference **FST_SCN** is true. This assures that the Communication Request will never be executed with incomplete or incorrect parameters.

The example uses the reference %Q128 as a permissive to the Communication Request. Output %Q129 is set if the Communication Request executes successfully. If it does not, output %Q130 is set instead. For the Communication Request, failure might occur if the Communication Request has been set up incorrectly, or for any of the other errors specified in the beginning of this chapter. A fault output is NOT caused by failure to receive a reply. This must be detected from the contents of the status location.



Another way to assemble the data for the example Command Block would be to use a Block Move instruction:



COMMREQs and Datagrams

The table below lists datagrams with their Subfunction Codes, shows the best ways to send datagrams, and explains what happens to datagrams from other devices.

Datagram (hex code)	Ways to Send It	How Incoming Datagram is Handled
Read ID (00)	COMMREQ 15 (Request Datagram Reply) *	Bus Controller replies automatically to Read ID datagram received from bus device.
Read ID Reply (01)	(Sent automatically)	Handled automatically if COMMREQ 15 was used to send Read ID datagram. *
Read Configuration (02)	COMMREQ 2 (Read Configuration) COMMREQ 15 (Request Datagram Reply) *	Bus Controller ignores it.
Read Configuration Reply (03)	(Sent automatically)	Handled automatically if COMMREQ 2 or 15 was used to send Read Configuration datagram. *
Write Configuration (04)	COMMREQ 3 (Write Configuration) COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
Assign Monitor (05)	COMMREQ 7 (Assign Monitor) COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
Begin Packet Sequence(06)	COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
End Packet Sequence (07)	COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
Read Diagnostics (08)	COMMREQ 4 (Read Diagnostics) COMMREQ 15 (Request Datagram Reply) *	Bus Controller replies automatically.
Read Diagnostics Reply (09)	(Sent automatically)	Handled automatically if COMMREQ 4 or 15 was used to send Read Diagnostics datagram. *
Write Point (0B)	COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
Read Block I/O (0C)	COMMREQ 15 (Request Datagram Reply) *	Bus Controller ignores it.
Read Block I/O Reply (0D)	(Sent automatically)	Handled automatically if COMMREQ 15 was used to send Read Block I/O datagram. *
Report Fault (0F)	(Sent automatically)	Received from bus devices; Bus Controller automatically places the fault in the Fault Table.
Pulse Test (10)	COMMREQ 1 (Pulse Test) *	Bus Controller ignores it.
Pulse Test Complete (11)	(Sent automatically)	Handled automatically if COMMREQ 1 was used to send Pulse Test datagram. *
Clear Circuit Faults (12)	COMMREQ 5 (Clear Circuit Fault) COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
Clear All Circuit Faults (13)	COMMREQ 6 (Clear All Ckt Faults) COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
Switch BSM (1C)	COMMREQ 10 (Switch BSM) COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
Read Device (1E)	COMMREQ 11 (Read Device) COMMREQ 15 (Request Datagram Reply)	Bus Controller automatically sends reply.
Read Device Reply (1F)	(Sent automatically)	Handled automatically if COMMREQ 11 or 15 was used to send Read Device datagram. *
Write Device (20)	COMMREQ 12 (Write Device) COMMREQ 14 (Send Datagram)	Bus Controller processes automatically.
Read Data (27)	COMMREQ 15 (Request Datagram Reply)	Bus Controller ignores it.
Read Data Reply (28)	(Sent automatically)	Handled automatically if COMMREQ 15 was used to send Read Data datagram. *
Write Data (29)	COMMREQ 14 (Send Datagram)	Bus Controller ignores it.
Read Map (2A)	COMMREQ 15 (Request Datagram Reply)	Bus Controller ignores it.
Read Map Reply (2B)	(Sent automatically)	Handled automatically if COMMREQ 15 was used to send Read Map datagram.
Write Map (2C)	COMMREQ 14 (Send Datagram)	Bus Controller ignores it.

* All datagrams can be sent using COMMREQ 14 (Send Datagram). If COMMREQ 14 is used to send a datagram that has a reply, COMMREQ 13 (Dequeue Datagram) must also be used, to obtain the reply from the Bus Controller's queue of unsolicited incoming datagrams

COMMREQ #1: Pulse Test Command

The Pulse Test command causes the Bus Controller to send a normal-priority Pulse Test datagram.

Pulse testing is used to verify the operation of outputs on discrete Genius I/O blocks. It checks whether the outputs will change state, and whether output circuits (wires, power sources, loads) will start or stop current flow. Any circuit faults generated by pulse tests are reported through the normal Report Fault message. Pulse testing is recommended for blocks that seldom change state. It is typically done once per hour, or once per shift; it should not be done more often than once per minute. Pulse testing provides assurance that when needed, an output will operate correctly. Blocks that control outputs that change state frequently do not need to be pulse-tested. Pulse testing does not provide enough energy to activate mechanical devices such as motor starters, relays, or solenoid valves, but may change the state of a very small load. If appropriate, blocks can be configured (with the Hand-held Monitor or via a Write Configuration command) to ignore a Pulse Test datagram. Pulse testing can also be done using a Hand-held Monitor.

Command Block Format for the Pulse Test Command

Address:	Command Length	2
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	1
Address +7:	Device Number	0 - 31. Or enter 255 to Pulse Test all discrete blocks.

COMMREQ #2: Read Configuration Command

The Read Configuration command is used to request configuration data from any block on the bus. It causes the Bus Controller to send a normal-priority Read Configuration datagram to the indicated block. After receiving the request, the block returns its configuration data to the Bus Controller in 16-byte increments. When the Bus Controller has received all the configuration data, it transfers the data to the memory location specified in the Command Block. Because configuration data consists of both bit-type and byte-type portions, it is best to place it in word memory, then move the bit-oriented data to bit memory (%M or %T is recommended). Contents of Read Configuration Reply messages for I/O blocks are shown in the *Genius I/O System User's Manual*.

Before a block can be sent this command, its Device Number (serial bus address) must be set up by software configuration. In addition, the block must have had its Device Number entered using a Hand-held Monitor.

Command Block for the Read Configuration Command

Address:	Command Length	5
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	2
Address +7:	Device Number	0 - 31.
Address +8:	Maximum data memory length. May represent either bits or words (depends on the memory type selected below).	18 words (288 bits): any discrete block 13 words (208 bits): 16 Circuit AC Input block 42 words (672 bits): Analog blocks (4 inputs/2 outputs) 42 words (672 bits): RTD or Thermocouple blocks 42 words (672 bits): 6-input Analog blocks 35 words (560 bits): High-speed Counter 13 words (208 bits): PowerTRAC Block If the length of data returned by the device exceeds this length, the Bus Controller writes as much data as possible to the PLC CPU and returns a data error to the COMMREQ status location. If the same COMMREQ will be used to read configuration data from more than one type of block (for example, in a subroutine), be sure to allow enough length to accommodate the largest amount of data that might be returned.
Address +9:	Memory type of the location where the Bus Controller will place the data in the CPU.	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W). If %W is used, the offset selected (below) must be in the range 00001 – 65536.
Address +10:	Memory offset	Beginning address for the data.

COMMREQ #3: Write Configuration Command

The Write Configuration command is used to send configuration data from the CPU to a block on the bus. (The Bus Controller cannot write configuration data to another bus interface module or to a Hand-held Monitor). A Write Configuration command to the Bus Controller itself would be rejected with status 128 (command not supported by target device).

Before a block can be sent this command, its Device Number (serial bus address) must be set up by software configuration. In addition, the block must have had its Device Number entered using a Hand-held Monitor.

The PLC sends the intended configuration data from CPU memory to the Bus Controller. The Bus Controller schedules background Write Configuration messages to the block. Once message transmission begins, the Bus Controller sends the configuration data to the block, up to 16 bytes per bus scan. The block does not use any of the new configuration data until it all has been received. No new commands can be sent to the block until the operation has been completed. When all the data has been sent, the Bus Controller changes the status to 4 (Done).

The length of the data sent with this command must exactly match the length required by the device. If the lengths are not equal the Bus Controller returns a Syntax Error to the COMMREQ status location.

Command Block for the Write Configuration Command

Address:	Command Length	This number equals the amount of configuration data to be sent, plus 3. For example, for an RTD block, which has 42 words of configuration data, you would enter 45 here. Configuration data formats for all Genius I/O blocks are shown in the *Genius I/O System User's Manual*.
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	3
Address +7:	Device Number	0 - 31. (SBA of the block to which configuration data will be written).
Address +8:	Length of configuration data in bytes.	Up to 248 bytes (128 words) of configuration data may be written to a device. See "COMMREQ #2: Read Configuration" for data lengths.
Address +9 to Address +n:	Configuration Data	Configuration data formats are given in the *Genius I/O System User's Manual*.

COMMREQ #4: Read Diagnostics Command

Use this command to request diagnostic information from a block or a bus interface module. Diagnostics can be requested from any block, even those configured not to issue Report Fault messages. The diagnostic data returned by a block indicates faults that have occurred since powerup or since the last Clear Faults datagram. Current diagnostic state can be found by issuing a Clear Faults command to the circuit(s) or channel(s) to clear the fault history, then issuing a Read Diagnostics command.

This command causes the Bus Controller to send a Read Diagnostics datagram to the specified device. When the device receives this datagram, it returns a Read Diagnostics Reply datagram. I/O blocks return data in message segments of up to 16 bytes per bus scan. The content of the Read Diagnostics Reply message depends on the device being queried. The first word of the reply must contain the length of the data that follows. Data is packed two bytes per word. Message formats are shown in the *Genius I/O System User's Manual*. When all the data has been received, the Bus Controller transfers it to the CPU and sets the COMMREQ status to 4 (Done).

Command Block for the Read Diagnostics Command

Address:	Command Length	5
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	4
Address +7:	Device Number	0 - 31 (of the device whose diagnostics are to be read).
Address +8:	Maximum data memory length, in bits or words (depends on the memory type selected below).	This entry tells the CPU how much memory will be needed to store the data returned by the block. The number of bits or words needed depends on the number of circuits on the block, and the block type: 10 words (160 bits): Discrete I/O blocks, 8-ckt 18 words (288 bits): Discrete I/O blocks, 16-ckt 34 words (544 bits): Discrete I/O blocks, 32-ckt 8 words (128 bits): Analog, 4 input/2 output blocks 8 words (128 bits): RTD Input Blocks 8 words (128 bits): Thermocouple Input Blocks 6 words (96 bits): High-speed Counter (For a PowerTRAC Block, status information is automatically provided as "input" data. If the data returned by the designated device exceeds this length, the Bus Controller will write as much as possible to the PLC CPU and return a data error to the COMMREQ status location.
Address +9:	Memory type of the location where the Bus Controller will place the data in the CPU.	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). Since diagnostic data is both bit-type and byte-type, use of word memory is recommended. Bit-type data can then be moved to a bit memory such as %T or %M.
Address +10:	Memory offset	Beginning address for the data. For %W, this must be in the range 00001 – 65536.

COMMREQ #5: Clear Circuit Faults Command

The Clear Circuit Faults command is used to clear any faults on a specified circuit of a Genius I/O block. The Clear Circuit Fault command causes the Bus Controller to issue a normal-priority Clear Circuit Fault datagram.

Command Block for Clear Circuit Faults

Address:	Command Length	3
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	5
Address +7:	Device Number	0 - 31 (.of the Genius block on which the circuit to be cleared is located).
Address +8:	Circuit Number	This is the relative number of the circuit, not its reference number. The first circuit on the block is considered to be number 1. For example, to clear faults on circuit 5, you would enter 5 here. For a 4 Input/2 Output analog block, circuit numbers 1 to 4 are for inputs, 5 and 6 are for outputs.

COMMREQ #6: Clear All Circuit Faults Command

The Clear All Circuit Faults command is used to clear all faults on a Genius I/O block. It causes the Bus Controller to issue a normal-priority Clear All Circuit Faults datagram.

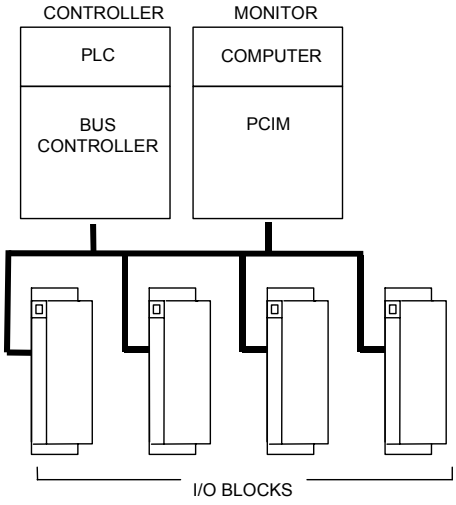
Command Block for Clear All Circuit Faults

Address:	Command Length	2
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	6
Address +7:	Device Number	0 - 31 (of the Genius block on which the circuit to be cleared is located). To send this datagram to all blocks on the bus, enter the number 255.

COMMREQ #7: Assign Monitor Command

An Assigned Monitor is an additional bus interface module (usually in another CPU) that monitors Genius I/O devices on the bus. Remote I/O Scanners and I/O blocks broadcast their inputs to all devices on the bus. Therefore, any interface module on the bus will receive all inputs sent by the blocks. However, blocks direct fault reports and configuration change messages only to the bus interface module that sends them outputs. Blocks configured for CPU Redundancy will automatically transmit two copies of any fault report or configuration change message, directing them to Device Numbers 30 and 31.

The Assign Monitor command can be used to have Genius I/O devices send extra fault report and configuration change messages to a monitoring bus interface module. Blocks would send two copies of each fault report or configuration message in a non-redundant system. Blocks in a redundant system would send three (two to the redundant bus interface modules, and the third to the Assigned Monitor).



Multiple CPUs might be used to monitor different blocks on the same bus. However, only one device can be assigned to monitor any given block.

When the Bus Controller receives the Assign Monitor COMMREQ command from the CPU, it issues a normal-priority Assign Monitor Datagram to one block or to all blocks on the bus. If sent to bus interface modules, it has no effect.

Command Block for the Assign Monitor Command

Address:	Command Length	3
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	7
Address +7:	Device Number of the block(s) that should send extra fault reports	0 - 31
Address +8:	Device Number of the bus interface module that will RECEIVE the extra fault reports.	0 - 31 to send this command to one block. To send this command to ALL blocks, enter the number 255. If only some blocks should report to the faults to the assigned monitor (for example, to minimize bus scan time), program separate Assign Monitor commands to each.

COMMREQ #8: Enable/Disable Outputs Command

The Enable/Disable Outputs command can be sent to the Bus Controller to disable sending outputs *to any blocks whose outputs were enabled during I/O (software) configuration.* Outputs that were configured as *disabled* are NOT affected by this COMMREQ command. The effect of disabling outputs is the same as running the control in Run/Disable mode.

Command Block for the Enable/Disable Outputs Command

Address:	Command Length	3
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	8
Address +7:	Device Number	Enter 0-31 to enable or disable outputs to one block. To enable or disable outputs to ALL devices on the bus, enter the number 255.
Address +8:	Enable/Disable command	To disable outputs to the device(s) specified in address +7, enter 0. To enable outputs, enter 1.

COMMREQ #9: Enable/Disable Global Data

If Global Data has been enabled, this COMMREQ command can be used after powerup to disable or re-enable the sending of Global Data from the Bus Controller, or receiving it from one or more devices on the bus.

If this COMMREQ attempts to enable Global Data when it is already enabled, or to disable Global Data when it is already disabled, the Bus Controller ignores the request and returns status 4 (successful completion) to the Status Block.

Command Block for the Enable/Disable Global Data Command

Address:	Command Length	3
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	9
Address +7:	Device Number	Enter 0-31 to specify the bus interface module to/from which Global Data is being enabled or disabled. This may be the Device Number of the Bus Controller itself, or of any other bus interface module on the bus. To enable or disable the Bus Controller *sending* Global Data, enter the Bus Controller's Device Number. To enable or disable the Bus Controller *receiving* Global Data from another bus interface module, enter the Device Number of that bus interface module. To enable or disable the Bus Controller *sending or receiving* ANY Global Data, enter the number 255. If 255 is entered, the command will complete successfully if there are any controller devices on the bus.
Address +8:	Enable/disable command	To disable Global Data to or from the device specified in address +7, enter 0. To enable Global Data, enter 1.

COMMREQ #10: Switch BSM Command

In a dual bus system, the Switch BSM command can be used to cause a Bus Switching Module to select a bus. This command causes the Bus Controller to issue a normal-priority Switch BSM datagram.

The program must already know which bus is *currently* selected. The CPU may issue the Switch BSM command at intervals to ensure continued proper bus switching capability. If the command is successful, the CPU will report a Loss of Block diagnostic for the BSM Controller block and for any other block on the same bus stub. If the dual bus system includes a second Bus Controller controlling the other bus, that Bus Controller should report an Addition of Block diagnostic for each of those blocks. If the BSM position is currently forced by the Hand-held Monitor, the command will have no effect. A data error is returned to the status reference if the block does not control a BSM.

Command Block for the Switch BSM Command

Address:	Command Length	3
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	10
Address +7:	Device Number of the Genius block to which the Bus Switching Module is attached	0-31
Address +8:	Desired bus position	Bus A (0) or Bus B (1). If not 0 or 1, syntax error is returned.

COMMREQ #11: Read Device Command

To read up to 128 bytes of data from another CPU and place it in PLC CPU memory, use the Read Device command. This causes the Bus Controller to issue a normal-priority Read Device datagram. When the data is received, it will automatically be placed in the CPU memory location specified in the Command Block.

Command Block for the Read Device Command

Address:	Command Length	16
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, the range is 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	11
Address +7:	Device Number	0 - 31, for the device which is the source of the data.
Address +8: Address +9:	Memory address, bytes 1, 2. ”	Specify the location where data will be read FROM. See the instructions on the following pages. (It is not necessary to specify a memory address when sending a Read Device COMMREQ to a computer).
Address +10: Address +11: Address +12: Address +13:	Program name, characters 1, 2 ” characters 3, 4 ” characters 5, 6 ” characters 7, 8	Required to read %P or %L memory in an RX7i or Series 90-70 PLC CPU. See the instructions on the following pages. If the target of the command is another type of device, Address +10 through Address +17 are ignored; they may contain any value. Program names are limited to 7 characters. Character 8 and all other trailing characters MUST be entered as nulls.
Address +14: Address +15: Address +16: Address +17:	Block name, characters 1, 2 ” characters 3, 4 ” characters 5, 6 ” characters 7, 8	Required to read %L memory in an RX7i or Series 90-70 PLC CPU. For %P, Address +14 through Address +17 are ignored. Block names are limited to 7 characters. Character 8 and all other trailing characters MUST be entered as nulls.
Address +18:	Data length, in words, bytes or bits. This is the amount of data to be read.	For RX7i or Series 90-70, data length is bits or words, depending on the memory type being read. For other types of devices, the length is given as expected by the device. The maximum length is equal to 128 bytes.
Address +19:	Maximum memory length needed for the returned data	Value in bits or words (depends on memory type selected below).
Address +20:	Memory type to receive the returned data	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +21:	Memory offset	Beginning address for the data. For %W, the range is 00001 – 65536.

Memory Specification for Read Device and Write Device

The following pages explain how to specify the target memory type when sending a Read Device or Write Device datagram to an RX7i or Series 90-70 PLC CPU, or to a Series Six PLC, Series Five PLC, or host computer.

Memory Specification: RX7i, Series 90-70 PLC, or Series 90-30 PLC CPU

In address + 8 enter the memory type, using one of the numbers listed in the table below.

Target Memory Type	Value (decimal)	Description	Bits per Reference
%L	0	Local register memory (each subroutine)	16
%P	4	Program register memory	16
%R	8	Register memory	16
%AI	10	Analog input memory	16
%AQ	12	Analog output memory	16
%I	16 70	Discrete input memory (byte mode) Discrete input memory (bit mode)	8 1
%Q	18 72	Discrete output memory (byte mode) Discrete output memory (bit mode)	8 1
%T	20 74	Discrete temporary memory (byte mode) Discrete temporary memory (bit mode)	8 1
%M	22 76	Discrete momentary internal memory (byte mode) Discrete momentary internal memory (bit mode)	8 1
%SA	24 78	Discrete system memory group A (byte mode) Discrete system memory group A (bit mode)	8 1
%SB	26 80	Discrete system memory group B (byte mode) Discrete system memory group B (bit mode)	8 1
%SC	28 82	Discrete system memory group C (byte mode) Discrete system memory group C (bit mode)	8 1
%S	30 84	Discrete system memory (byte mode) Discrete system memory (bit mode)	8 1
%G	56 86	Discrete Genius automatic global data table (byte mode) Discrete Genius automatic global data table (bit mode)	8 1
%W	196	Bulk memory (for RX7i only)	16

Memory Offset

In address +9, enter a numerical offset within this memory type, for the beginning of the data. Memory offsets start at 0; thus %R1 and %I1 are both accessed using a Memory Offset of 0.

Example A: For 3 bits starting at %I0014, you would enter the offset 13, and a data length of 3 bits.

Example B: To write data to an RX7i, Series 90-70 PLC, or Series 90-30 PLC CPU beginning at %R100, you would enter the Memory Type 8 (decimal) and the Memory Offset 99 (decimal).

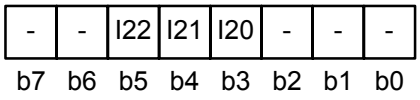
Bit Mode or Byte Mode for an RX7i, Series 90-70 PLC, or Series 90-30 PLC CPU

Bit-oriented memories (%I and %Q) can be accessed either on byte boundaries (byte mode) or as a string of bits (bit mode). Bit mode is used to access a single point within a discrete memory, or a collection of points within a discrete memory which need not start or end on a byte boundary. Byte mode is used to access one or more groups of 8 contiguous points within a discrete memory, and must start on a byte boundary.

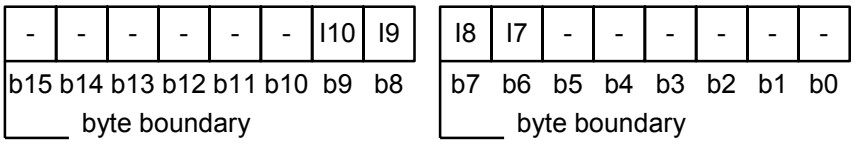
In byte mode, the Message Offset reflects the byte being read or written. Offset 0 corresponds to bits 1-8, offset 1 to bits 9-16, and so on.

In bit mode, the Message Offset reflects the bit being read or written, offset 0 corresponds to bit 1, offset 1 to bit 2, and so on.

In bit mode, one or more bytes of data are read or written, even though some of the bits within the bytes might be ignored. The bit or bits will be in the correct offset position within the byte. For example, if three bits starting a %I0020 are requested, they will appear in the middle of the returned data byte. The "-" indicates unused bits. On READ, they are guaranteed to be 0. On WRITE, they are ignored.

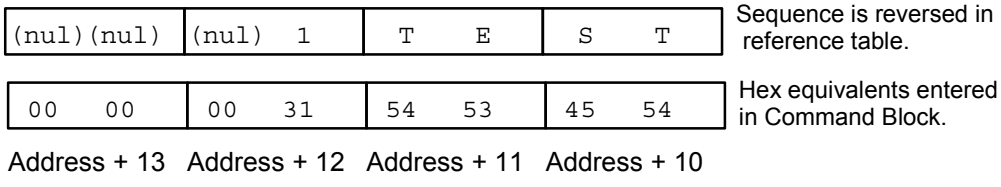


If four bits starting at %I00007 are requested, two bytes are transferred.



Entering a Program or Block Name

If the target of the command is an RX7i, Series 90-70 PLC, or Series 90-30 PLC CPU, and the memory type to be read is either %P or %L, a program name and, possibly a block name, must be entered. Names are limited to 7 characters. Character 8 and any other trailing characters must be nulls. Names are entered in ASCII hex format, as indicated by the following example:



Hex ASCII equivalents are listed in appendix A. Lowercase letters are not valid in names.

Memory Specification: Series Six PLC

For a Series Six PLC, Read Device and Write Device include an absolute memory location in either Register memory or I/O Status Table memory. Byte 4 of the address must be 80 hex.

Series Six Memory Type		*Absolute Address*	
		Decimal	*Hexadecimal*
I/O Status Table	Outputs Inputs	08192 - 08319 08320 - 08447	2000 - 207F 2080 - 20FF
Register Memory	R00001-R16384	16384 - 32767	4000 - 7FFF

Caution

When sending a Write Device COMMREQ to a Series Six PLC, be sure the CPU address specified is for the register table (first hex digit is 4-7) or the I/O Status Table (first hex digit is 2). Writing CPU data to any other absolute memory location may cause potentially hazardous control conditions.

Memory Specification: Series Five PLC

For a Series Five PLC, **[Address+8]** of the Read Device or Write Device COMMREQ contain a memory offset, which is the beginning location for the data:

Series Five Memory Type		*Offset (hex)*
Register Memory	R00001 to R16384	0000 - 7FFF
I/O Memory	I1+0001 to I1+1024 I2+0001 to I2+1024 O1+0001 to O1+1024 O2+0001 to O2+1024 I0001 to I1024 O0001 to O1024 O1-0001 to O1-1024 O2-0001 to O2-1024 I1-0001 to I1-0512	8000 - 807F 8080 - 80FF 8100 - 817F 8180 - 81FF 8200 - 827F 8280 - 82FF 8300 - 837F 8380 - 83FF 8500 - 853F

To find the exact offset in the register table, follow these steps:

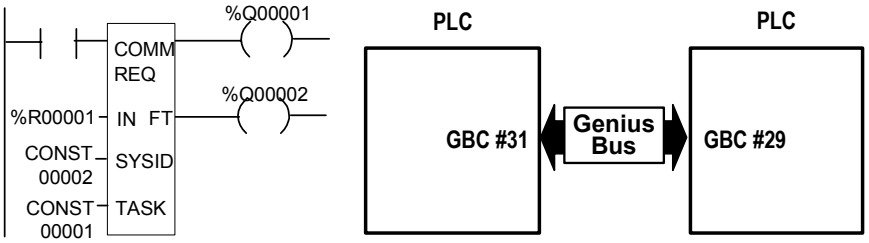
1. Subtract 1 from the register number.
2. Multiply the result by 2 to find the decimal byte offset.
3. Continue as described below.

For a decimal offset in the register or I/O tables:

1. Convert the decimal number to hex.
2. Add the hex number to the beginning offset for that memory type.

Example of Read Device

In the following example, an RX7i CPU reads ten words of %P memory starting at location %P0050 from program “TEST1” in a Series 90-70 PLC.



Program Name	ABC	TEST1
Initiating Device	X	
Target Device		X
Command Block Location (COMMREQ parameters)	%R001-%R022	
Status Block Location (COMMREQ Status)	%R023-%R024	
COMMREQ Output References	%Q001, %Q002	
Memory location to read data from		%P050-%P060
Memory location to place data	%AQ050-%AQ060	

When the data is received from the target PLC, the requesting PLC will store it beginning at %AQ0050 in its own memory.

Example Command Block

Command Block Register	*Description*	*Value*	*Comment*
Address +1	Wait/No Wait	0	No Wait
Address +2	Memory type for Status Pointer	8	%R memory
Address +3	Starting address for Status Pointer	22	%R023 (1 + 22 offset)
Address +4	Timeout value	0	Not used for No Wait
Address +5	Max Communication Time	0	Not used for No Wait
Address +6	Command Code	11	Read Device
Address +7	Device Number	29	Target Bus Controller
Address +8	Memory Address bytes 1 & 2	2	%P memory
Address +9	Memory Address bytes 3 & 4	49	%P0050 (1 + 49 offset)
Address +10	Program Name characters 1 & 2	’ET’	TEST1
Address +11	Program Name characters 3 & 4	’TS’	
Address +12	Program Name characters 5 & 6	’1’	character 6 is null *
Address +13	Program Name character 7 & 8	xx	Ignored (don’t care)
Address +14	Block Name characters 1 & 2		Ignored
Address +15	Block Name characters 3 & 4		Ignored
Address +16	Block Name characters 5 & 6		Ignored
Address +17	Block Name character 7 & 8		Ignored
Address +18	Data length (word) to read	10	10 words
Address +19	Destination length (words)	10	10 words
Address +20	Memory type for received data	12	%AQ memory
Address +21	Memory offset for data	49	%AQ0050 (1+49 offset)

* In hex, the two values in Address +12 are 31h (ASCII 1) and 00h (ASCII nul).

COMMREQ #12: Write Device Command

To send up to 128 bytes of data to another CPU on the bus, use the Write Device command. Any type of data that can be addressed by its memory type and offset can be sent. This command causes the Bus Controller to issue a normal-priority Write Device datagram to the specified device. To send a Write Device datagram with high priority, see *COMMREQ #14: Send Datagram*.

Using Write Device Messages Instead of Global Data

Write Device datagrams can be used together with Global Data, or can replace Global Data. Consider using Write Device datagrams instead of Global Data if Global Data takes up too much bus scan time for the application, data does not need to be sent every bus scan, or the CPU sweep time becomes too long for the application.

If this datagram will be broadcast, and there is another Series 90-70 Bus Controller on the bus that should NOT receive it, send the datagram using COMMREQ #14 instead.

Command Block for the Write Device Command

Address:	Command Length	13 - 77. Enter the number of words from Address +6 to the end of the data.
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	12
Address +7:	Device Number	0 - 31, for the device to be written to.
Address +8: Address +9:	Memory address, bytes 1, 2. ”	Enter the location for data to be written to. See the instructions for “Read Device”. (It is not necessary to specify a memory address when sending a Write Device COMMREQ to a computer).
Address +10: Address +11: Address +12: Address +13:	Program name, characters 1, 2. ” characters 3, 4. ” characters 5, 6. ” characters 7, 8.	Required to write %P or %L memory in an RX7i, Series 90-70, or Series 90-30 CPU. See the instructions for “Read Device”. If the target of the command is another type of device, Address +10 through Address +17 are ignored; they may contain any value.
Address +14: Address +15: Address +16: Address +17:	Block name, characters 1, 2. ” characters 3, 4. ” characters 5, 6. ” characters 7, 8.	Required to write %L memory in an RX7i, Series 90-70, or Series 90-30 CPU. For %P, Address +14 through Address +17 are ignored. Block names are limited to 7 characters. Character 8 and all other trailing characters MUST be entered as nulls.
Address +18:	Data length, in words, bytes or bits. This is the amount of data to be read.	Writing an RX7i, Series 90-70, or Series 90-30 CPU, data length is bits or words, depending on the memory type being read. For other types of devices, the length is given as expected by the device. The maximum length is 128 bytes.
Address +19 to Address +n:	Data to be written to the other device.	

COMMREQ #13: Dequeue Datagram Command

The Bus Controller handles most incoming datagrams automatically, with no additional programming required. Under certain circumstances, however, the Dequeue Datagram command must be used to transfer incoming datagrams to the CPU. Program the Dequeue Datagram command for the following:

- Replies that are received after sending Reply-type datagrams with the Send Datagram command. (If Send Datagram with Reply is used instead, it automatically handles replies).
- Unsolicited datagrams that are not recognized by the Bus Controller (Function Code not 20).

Command Block for the Dequeue Datagram Command

Address:	Command Length	7
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, the range is 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	13
Address +7:	Maximum data memory length	Enter bit or word value (depends on the memory type selected below). This entry tells the CPU how much memory will be needed to store all the data. If the length of data returned by the device exceeds this length, the Bus Controller writes as much data as possible to the PLC CPU and returns a data error to the COMMREQ status location.
Address +8:	Memory type	Enter the number that represents the location where the Bus Controller will place the data in the CPU: 70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ).). For RX71 only, also 196 (%W).
Address +9	Starting address	Beginning address for the data. For %W, this must be in the range 00001 – 65536.
Address +10	Function code of the datagram.	Enter a function code. or enter FF hex to match any function code.
Address +11:	Subfunction code of the datagram.	Enter a subfunction code, or FF hex to match any subfunction code.
Address +12:	Device Number (sender)	Enter 0 - 31, or FF hex to match any Device Number.

Number of Dequeue Datagram Commands Needed

One Dequeue Datagram command is needed for each incoming datagram. If multiple incoming Datagrams are expected during one CPU sweep, it will be necessary to place multiple Dequeue Datagram commands in the program to assure their efficient transfer to the CPU.

The number of Dequeue Datagram commands needed depends on whether the Datagrams have been sent using Normal or High Priority, and the relative lengths of the CPU sweep time and the scan time of the bus, as explained below.

If the Bus Scan Time is Greater than the CPU Sweep Time

If all Datagrams on the bus are sent with Normal Priority, there is a limit of one incoming Datagram per CPU sweep. Therefore, only one Dequeue Datagram command per sweep will be needed to handle incoming Datagrams.

If all Datagrams on the bus are sent with High Priority, the Bus Controller can potentially receive one Datagram from each transmitting device during a scan. The program should include the same number of Dequeue Datagram commands as incoming Datagrams.

If the Bus Scan Time is Less than the CPU Sweep Time

If the bus scan time is significantly shorter than the CPU sweep time, you can estimate the number of Dequeue Datagram commands that must be sent to the Bus Controller to accommodate incoming Datagrams on that bus.

First, determine how many scans can occur in one CPU sweep. For example, if the bus scan were 20ms and the CPU sweep were 90ms, the ratio between them would be 4.5 to 1. This should be rounded upward to 5.

This is the maximum number of Normal Priority Datagrams that might be received in a single CPU sweep. Plan to have the same number of Dequeue Datagram commands to that Bus Controller in the program to handle the incoming Datagrams.

For High Priority Datagrams, multiply the number found above by the total number of devices on the bus that might send a High Priority Datagram to the Bus Controller in one bus scan. This is the total number of incoming Datagrams from that bus that the program might have to handle in a single CPU sweep. Plan on this number of Dequeue Datagram commands to the Bus Controller.

Additional Logic for Incoming Datagrams

Up to 16 datagrams are enqueued by the Bus Controller in an internal queue. These include any unsolicited reply-type datagrams. This permits the program to, for example, send a Read ID Send Datagram and dequeue the Read ID Reply with the Dequeue Datagram COMMREQ.

If the 16-item queue fills, an informational fault GBC_SOFTWR_EXCPTN is logged (Fault Type is DQ_QUEUE_FULL) in the I/O Fault Table. If the Dequeue Datagram is issued and there are no datagrams in the queue, the Status Pointer is set to NO DATA TO TRANSFER.

Program logic should be used to assure that no datagrams are lost by being accidentally written over. This might be done by copying each datagram to another memory location, or by changing the data memory location specified in the Command Block after each incoming datagram is received.

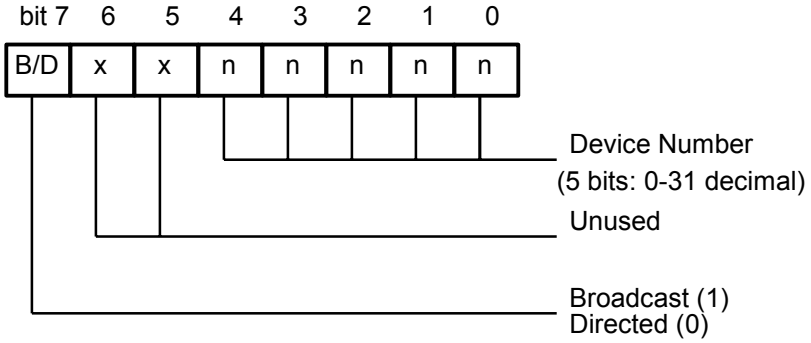
Format of Returned Data

The Dequeue Datagram returns data in the following format.

Location	*High Byte*	*Low Byte*
Memory Address	Data Length	Status byte
Memory address +1	Subfunction code	Function code
Memory address +2	Data byte 2	Data byte 1
▼	▼	▼
▼	▼	▼
▼	▼	▼
Memory address +69	Data byte 134	Data byte 133

Items are explained below.

Status Byte: The status byte reports the Device Number of the device that sent the datagram. It also indicates whether the message was broadcast or directed by the other device.



Data Length: The number (0 to 134) of data bytes after the subfunction code.

Function Code: The function code of the received message: 0 to 111 decimal or 0 to 6F hex.

Subfunction Code: The subfunction code of the received message: 0 to 255 decimal or 0 to FF hex.

COMMREQ #14: Send Datagram Command

Most datagrams are normally programmed using their assigned COMMREQ command numbers. However, datagrams can also be sent using the Send Datagram command and the Request Datagram Reply command. The Send Datagram command might be used to send:

- Datagrams for which no COMMREQ command number is defined, such as Begin Packet Sequence, End Packet Sequence, and Write Point.
- Read Device and Write Device datagrams that are broadcast, but which should be ignored by another Series 90-70 Bus Controller.
- Datagrams that must be guaranteed transmission during the next bus scan. This should be done with restraint, for the reasons explained on the following pages.
- Datagrams that do not cause another device to send back a reply, such as Pulse Test, or Write Configuration.

  Datagrams that DO cause another device to send back a reply, such as Read Diagnostics or Read Configuration, are usually programmed using their assigned COMMREQ numbers or the Request Datagram Reply command (COMMREQ #15). However, if Send Datagram is used to send datagrams that cause replies, the Dequeue Datagram command must be used to transfer the replies to the CPU.

Before using Send Datagram, refer to the table on page 5-12 for more information about COMMREQs and datagrams.

Command Block for the Send Datagram Command

Address:	Command Length	6 – [6+ datagram length in words]. Enter the number of words from Address +6 to Address +n. The length should be able to support the maximum data length in words that might be sent with the command.
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	14
Address +7:	Device Number of the device to receive the message.	0 - 31, or 255 to broadcast the message.
Address +8:	Function code	For any datagram listed below, 32 decimal (20 hex).
Address +9	Subfunction code (hex)	See the list on page 5-12.
Address +10	Priority	Enter 0 for normal priority, or 1 for high priority.
Address +11:	Datagram length (in bytes)	Enter the actual length of the Datagram, beginning at [address +12].
Address +12: to Address+n:	Datagram content	Enter the entire datagram as part of the Command Block. The *Genius I/O System User's Manual* shows datagram structures. *If the Send Datagram command is used to broadcast a Read or Write Device datagram, and that datagram should be IGNORED by another Series 90-70 Bus Controller, set the first byte of the datagram as shown in the System User's Manual (this byte is normally 0), to FE hex.*

Datagram Priority

A Bus Controller can send one datagram per bus scan. That datagram may be assigned either normal priority or high priority. Therefore, during one bus scan, there may be one normal priority datagram followed by up to 31 high priority datagrams, *or* up to 32 high priority datagrams sent by the devices on the bus.

In one bus scan (one complete rotation of the bus token among all devices on the bus), there can be just one normal priority datagram sent by any device. If a normal priority datagram or similar system message (such as a fault report) has already been sent by any device (including itself), a device must wait until its next turn on the bus before it can send a normal priority datagram.

Datagrams and I/O Blocks

If the bus will also be used for I/O block control, normal priority datagrams are recommended to allow other messages such as fault reports (which the system handles as normal priority datagrams) to get through. In addition, normal priority datagrams ensure that bus scan time is only modestly delayed for communications. Bus scan time affects the response time of any I/O data on the bus. If there are I/O blocks on the bus, use high priority only if the datagram transmission cannot be delayed. Normal priority will work satisfactorily except when there are many devices attempting to send datagrams simultaneously.

Number of Datagrams per CPU Sweep

The application program should include logic that verifies successful completion of earlier datagrams before requesting new ones. Because a Bus Controller can only send one datagram per bus scan, the number of datagrams that can be executed during the same CPU sweep of program logic depends on the relative lengths of the CPU sweep and the bus scan.

If the Bus Scan Time is Greater than the CPU Sweep Time: If the bus scan time is greater than the CPU sweep time, the Bus Controller will be able to send no more than one datagram during one execution of the application program. Successful transmission of a normal priority datagram will depend on the absence of datagram and system message traffic on the bus.

If the Bus Scan Time is Less than the CPU Sweep Time: If the bus scan time is significantly shorter than the CPU sweep time, the bus may be able to transmit multiple datagrams during one execution of the application program.

Effect of Datagrams on the Genius I/O Bus: Normal Priority Datagrams allow fault reports and Hand-held Monitor communications on a bus to continue undisturbed. Only one Normal Priority Datagram is allowed each bus scan, so the scan time stays relatively constant, and I/O update timing varies only by small increments.

If High Priority Datagrams are being transmitted constantly, the Hand-held Monitor will not function properly; fault reports from blocks will be prevented from being transmitted on the bus, and regular Communication Request commands (such as Write Configuration commands) to that Bus Controller will fail with a transmission error. For these reasons, use of High Priority Datagrams on a bus with I/O blocks should be avoided if possible.

If High Priority Datagrams are transmitted infrequently, they will cause some delay in the Hand-held Monitor communications and other normal system messages, but the delay should not be noticeable.

High Priority Datagrams will typically put more pressure on the Bus Controller to transfer multiple Datagrams per CPU sweep. However, this can also occur with Normal Priority Datagrams if the bus scan time is much shorter than the CPU sweep time.

Maximum CPU Sweep Time Increase for Datagrams: To estimate the impact of Datagrams on CPU sweep time, add together the times required for all Datagrams that might be sent between the Bus Controller and the CPU during one sweep if No Wait mode is selected. Repeat this for each Bus Controller in the PLC that sends or receives Datagrams.

	Total Datagram Bytes Sent (may be none)	x	.031ms	=	
+	LARGEST incoming Normal Priority Datagram Received, bytes	x	.031ms	=	
OR					
+	Total incoming High Priority Datagrams Bytes Received	x	.031ms	=	
+			1.200ms	=	_____ms

Additional Information about Timing

If you need more information about timing for datagrams, Global Data, I/O devices, and remote drops, please refer to the *Genius I/O System User's Manual*.

COMMREQ #15: Request Datagram Reply Command

The Request Datagram Reply command can be used to send any datagram that causes the target device to return a reply, such as: Read Configuration or Read Diagnostics. With this command, the Bus Controller automatically transfers replies to the CPU; no separate Dequeue Datagram command is needed to handle them. These datagrams are normally programmed using their assigned COMMREQ command numbers. The primary reason for sending any of these datagrams using COMMREQ #15 would be to assign it high priority, guaranteeing that it would be sent on the next bus scan. Before doing this, see *COMMREQ #14: Send Datagram* for important information about datagram priority.

Command Block for the Request Datagram Reply Command

Address:	Command Length	10 - 78. Enter the number of words from Address +6 to Address +n.
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	15
Address +7:	Device Number of the device to receive the message.	0 - 31
Address +8:	Function code	For any datagram listed below, 32 decimal (20 hex).
Address +9	Subfunction code (hex) of the datagram to be sent.	00 Read ID 02 Read Configuration 08 Read Diagnostics 0C Read Block I/O 1E Read Device 27 Read Data
Address +10	Priority	Enter 0 for normal priority, or 1 for high priority.
Address +11:	Datagram length (in bytes)	Enter the actual length of the Datagram, beginning at [address +16].
Address +12:	Subfunction code (hex) of the reply	01 Read ID Reply 03 Read Configuration Reply 09 Read Diagnostics Reply 0D Read Block I/O Reply 1F Read Device Reply 28 Read Data Reply
Address +13:	Memory type for the reply	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +14:	Memory offset	Beginning address for the data. For %W, this must be in the range 00001 – 65536.
Address +15:	Maximum data memory length needed If the length of the memory is smaller than the amount of reply data received, the extra portion of the data will be lost, and a data error (16) will be returned to the status location.	Enter a value in bits or words, depending on the memory type selected. This entry tells the CPU how much memory will be needed to store all the reply data. The length depends on the message and device type. - for Read Configuration Reply, see COMMREQ #2. - for Read Diagnostics Reply, see COMMREQ #4. - for Read Device Reply, message length depends on device type. May be up to 64 words. - for Read Data Reply, message length is 5 words. - for Read ID Reply, message length depends on device type. See the *Genius I/O System User's Manual*.
Address +16 to Address +n:	Datagram Content	Enter the entire datagram as shown in the *Genius I/O System User's Manual.*

Format of Returned Data

Returned data format is the same as for Dequeue Datagram. See page 5-31.

COMMREQ #16: Enable/Disable I/O Fault Categories

The Enable/Disable I/O Fault Categories command can be sent to the Bus Controller to disable or re-enable the reporting of all I/O faults, or Addition/Loss of Block faults.

If all I/O faults are disabled, the Bus Controller will not forward to the CPU any fault reports it receives from devices on its bus. This includes all I/O faults, as well as Loss of Block and Addition of Block messages.

It is also possible to disable only reports of Addition or Loss of Block conditions, while still forwarding other faults from the devices on the bus. This can be useful in a system where blocks are intentionally switched on and off the bus, or in other applications where these messages are not wanted.

If the passing of some or all fault reports is disabled for non-redundant devices, the corresponding point-specific fault contacts *will* operate. They are not affected by the use of this COMMREQ. Some system-level and block-level fault contacts are affected by the loss of the inhibited information. For redundant devices, the CPU won't set the point faults for redundant devices if fault reporting is turned off. In addition, when fault reporting is turned off, the associated fault locating references are not updated.

Command Block for the Enable/Disable I/O Fault Categories Command

Address:	Command Length	3
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	16
Address +7:	I/O Fault Category to be enabled or disabled.	It may be: 0 0 0 0 all I/O fault categories enabled. F F F F all I/O fault categories disabled. 0 0 0 6 Addition/Loss of Block fault categories disabled.
Address +8:		Must be: 0 0 0 0 if Address +7 is 0000 or 0006. Must be: F F F F if Address +7 is FFFF.

The default is for all fault categories to be enabled.

COMMREQ #17: Do Output Command

To immediately send a total of up to 128 bytes (64 words) of output data to selected discrete devices on the bus (2 words per device), use the Do Output command. The output data is transferred immediately to the Bus Controller, which transmits it to the specified devices during its next turn on the bus. The user should also place the same data in the output table, to insure that the same data will be transferred to the Bus Controller as part of the normal output update.

To use the command, set flags of all bus addresses that are to receive output data, then enter the data into the COMMREQ addresses that correspond to those devices (for example, you would enter output data into locations [address+11] and [address+12] to update a block at serial bus address #1). To minimize the impact on bus scan time. Devices to be updated using this method should be configured to have the lowest possible serial bus addresses, and the overall message length should be kept as short as possible.

Caution: none of the outputs on a targeted device should be updated in any other way during the PLC sweep.

Command Block for the Do Output Command

<table>
<tr><td>Address:</td><td>Command Length</td><td>5 to 64. Enter the number of words from Address +6 to the end of the output data to be transferred to the Bus Controller.</td></tr>
<tr><td>Address +1:</td><td>No Wait</td><td>0</td></tr>
<tr><td>Address +2:</td><td>Status Block memory type</td><td>70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).</td></tr>
<tr><td>Address +3:</td><td>Status Block offset</td><td>Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.</td></tr>
<tr><td>Address +4:</td><td>Idle timeout value</td><td>0</td></tr>
<tr><td>Address +5:</td><td>Max. communications time</td><td>0</td></tr>
<tr><td>Address +6:</td><td>Command number</td><td>17</td></tr>
<tr><td>Address +7,
Address +8:</td><td>Flags for Device Numbers 0-15
Flags for Device Numbers 16-31</td><td>Each bit represents a potential target bus address. In Address +7, LSB is Device Number 0. MSB is Device Number 15. In Address +8, LSB is Device Number 16. MSB is Device Number 31.
LSB
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
MSB
Device Numbers (SBAs)
Set to 1 any bit that corresponds to a block that will receive output data from the Do Output command.</td></tr>
<tr><td>Address +9,
Address +10</td><td>Outputs for Device Number 0</td><td>Each bit represents an output, with output 1 in the LSB.
LSB
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17
MSB
Circuit Numbers
Bits that correspond to circuits that are used as inputs, or are unused, or are not present on the block are ignored.</td></tr>
<tr><td>Address +11
Address +12</td><td>Outputs for Device Number 1</td><td rowspan="2">Use command length to determine the last serial bus address serviced. Registers past that point will not be used in the command.</td></tr>
<tr><td>to Address +72</td><td>Outputs for Device Number 31</td></tr>
</table>

Example

In this example, there are two blocks on the bus that should be updated using Do Outputs. The first is a 16-circuit block located at Device Number 1. All 16 of its circuits are used as outputs. The second device is a 32-circuit block located at Device Number 2. On this block, circuits 17 to 32 are used as outputs. (In this application, Device Number 0 is used by the Hand-held Monitor).

Address +0:	Command Length (number of words from Address +6 to the end of the data.)	For this example, the length is 9 (the number of words from Address +6 to Address +14). This limits the number of registers used, and only serial bus addresses 0, 1 and 2 will be affected.
Address +1:	No Wait	0
Address +2:	Status Block memory type	8 (%R)
Address +3:	Status Block offset	Beginning address for the COMMREQ status.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	17
Address +7:	Flags for Device Numbers 0-15	This marks serial bus addresses 1 and 2 as the targets. LSB 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
Address +8	Flags for Device Numbers 16-31	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 MSB
Address +9, Address +10	Outputs for Device Number 0	32 bits: “don’t care” in this example, because the flag for serial bus address 0 is set to 0.
Address +11 Address +12	Outputs for Device Number 1	32 bits. In this example, Device Number 1 is a 16-circuit block. All 16 circuits (shown below as 1s for clarity) are outputs. LSB 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 MSB The result will be to turn all 16 outputs on. The remaining 16 bits are ignored, because the block has only 16 points.
Address +13 Address +14	Outputs for Device Number 2	32 bits. In this example, Device Number 2 is a 32-circuit block. Circuits 17-32 (shown below as 1s) are outputs. LSB 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 MSB The result will be to turn points 17 - 32 on. To turn any of these points off, place a zero in the corresponding bit (for example, to turn output #32 off, put a zero in the MSB).

Using the Do Output Command for Global Data

The Do Output command can also be used to immediately change the content of the first 2 words of Global Data being sent by a Bus Controller in the same PLC. Program COMMREQ #17 with the contents listed below. The two data words are transferred immediately to the Bus Controller, overwriting the first two words of that Bus Controller's usual Global Data. During that Bus Controller's next turn on the bus, the Global Data it transmits will begin with these two words, followed by its remaining Global Data. The changed data is sent only once following receipt of the COMMREQ. During its next turn on the bus, the Bus Controller will again send the entire contents of its assigned Global Data memory references. If the new data should be sent more than once, the application program should also copy the data into those memory references.

To use the command, in [address+7] and [address+8] set the flag corresponding to the serial bus address of the Bus Controller that will send the Global Data. Enter the data into the COMMREQ addresses that correspond to those devices (for example, you would enter output data into locations [address+71] and [address+72] to update a Bus Controller at serial bus address #31).

Command Block for the Do Output Command for Global Data

Address:	Command Length	5 to 64. Enter the number of words from Address +6 to the end of the Global Data to be transferred to the Bus Controller.
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	17
Address +7, Address +8:	Flags for Device Numbers 0-15 Flags for Device Numbers 16-31	Each bit represents a potential target bus address. In Address +7, LSB is Device Number 0. MSB is Device Number 15. In Address +8, LSB is Device Number 16. MSB is Device Number 31. LSB 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 MSB **Device Numbers (SBAs)** Set to 1 the bit that corresponds to the serial bus address of the Bus Controller that will be sending the Global Data.
Address +9 up to Address +72	Global Data	Each serial bus address is assigned 2 words in the command. Enter the data into the 2 words that correspond to the device you are updating.

COMMREQ #18: Read Serial Bus Address of the Bus Controller

To read the serial bus address of a Bus Controller in the same system, send it a Read Serial Bus Address command.

Command Block for the Read SBA Command

Address:	Command Length	3
Address +1:	No Wait	0
Address +2:	Status Block memory type	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +3:	Status Block offset	Beginning address for the COMMREQ status. For %W, this must be in the range 00001 – 65536.
Address +4:	Idle timeout value	0
Address +5:	Max. communications time	0
Address +6:	Command number	18
Address +7:	Memory type to receive data returned	70 (%I), 72 (%Q), 8 (%R), 10 (%AI), or 12 (%AQ). For RX71 only, also 196 (%W).
Address +8	Offset for memory type specified in Address +7.	If a bit memory type (%I or %Q) is specified in Address +7, the offset must be on a byte boundary (multiple of 8, +1). For %W, this must be in the range 00001 – 65536.

Chapter 6

Reading Bus Status Information

This chapter explains how the application program in a PACSystems or Series 90-70 CPU can read information about the status of the Genius bus:

- Reading the Serial Bus Addresses of Active Devices
  - Format of the Bus Status Data that is Returned
  - RX7i CPU: Reading Serial Bus Addresses
  - Series 90-70 CPU: Reading Serial Bus Addresses
- Reading the Status of the Datagram Queue
  - Format of the Datagram Queue Status Data that is Returned
  - RX7i CPU: Reading Datagram Queue Status
  - Series 90-70 CPU: Reading Datagram Queue Status

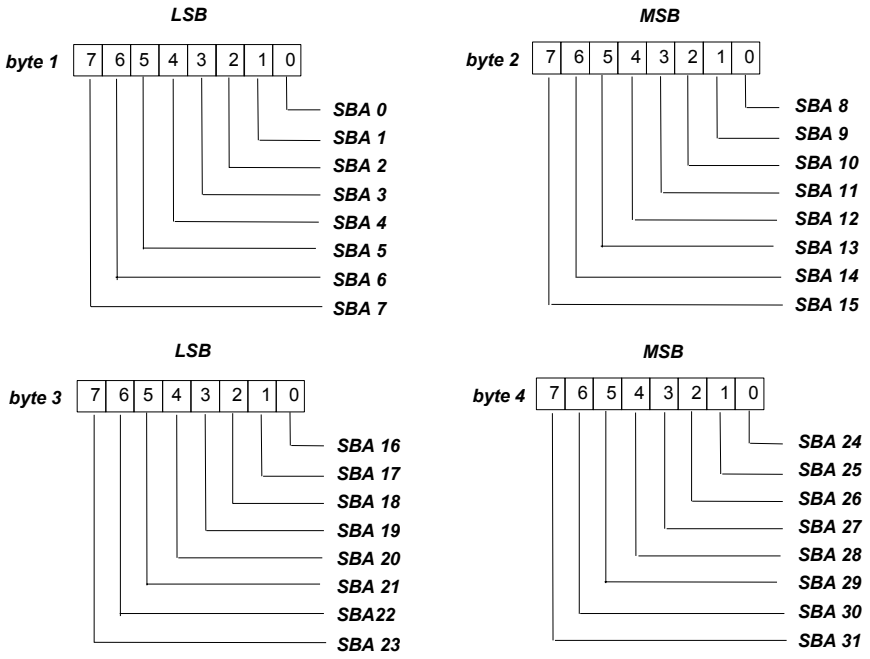
Reading the Serial Bus Addresses of Active Devices

Chapter 5 explains how the application program can read the Serial Bus Address of a "local" Genius Bus Controller in the system using COMMREQ #18.

The application program can also read the status of each Serial Bus Address on the bus directly from the Bus Controller's 32k shared ram memory. Similar program functions are available for an RX7i CPU and for a Series 90-70 CPU. Both are described on the following pages.

Format of the Bus Status Data that is Returned

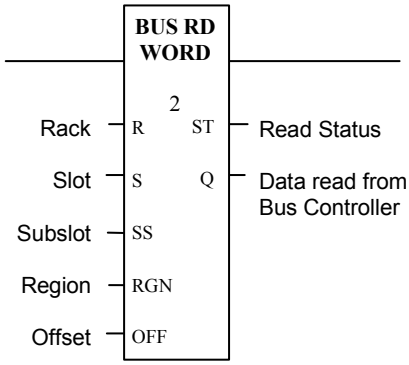
The function returns 32 bits; one for each potential Serial Bus Address. The bit corresponding to the bus address of the Bus Controller itself is always 0.



Bits are set to 1 for devices that are on the network and properly configured with the correct data length. Bits for devices that are configured as outputs-only are off if the CPU is not controlling the device.

RX7i CPU: Reading Serial Bus Addresses

In an RX7i system, the application program can determine which devices are currently active on a Genius bus by sending a BUSRD instruction to the Bus Controller.



No special configuration is required to use this function. The CPU automatically defines region 1 for the Bus Controller to be the Bus Controller's 32k dual port memory. Program the Bus Read parameters as shown below.

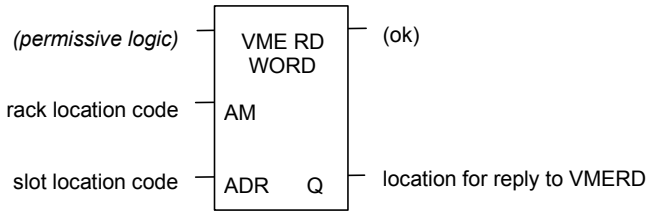
BUS READ Parameters for Reading Serial Bus Addresses

Program the BUSRD_WORD with the following parameters:

Parameter	*Description*	*Enter*
LEN	The number of words to read.	2 for BUSRD_WORD, or 4 for BUSRD_BYTE
R	Rack number	The rack number of the Bus Controller
S	Slot number	The slot number of the Bus Controller
SS	Subslot number (optional, defaults to 0)	0
RGN	Region. (Optional, defaults to 1)	1
OFF	Offset	07E8h
ST	(Optional.) Reference for the BUSRD status information.	
Q	Reference for bus status data that is read from the Bus Controller	

Series 90-70 CPU: Reading Serial Bus Addresses

In a Series 90-70 system, the application program can read determine which devices are currently active on a bus by sending a VMERD instruction to the Bus Controller. The format is:



VME READ Parameters for Reading Serial Bus Addresses

For a Series 90-70 CPU, program a VMERD_WORD with the following parameters:

Parameter		Enter	
AM	Bus Controller Rack Location Code	Rack 0 = 29h Rack 1 = 1Eh Rack 2 = 1Dh Rack 3 = 1Ch Rack 4 = 1Bh Rack 5 = 1Ah Rack 6 = 19h Rack 7 = 18h	
ADR	Bus Controller Slot Location Code	The sum of: 07E8h + (800h x slot number of Bus Controller) Example: Slot 2 = 07E8h + 2 x 800h = 17E8h	
		Slot	***Code***
		2	17E8h
		3	1FE8h
		4	27E8h
		5	2FE8h
		6	38E8h
		7	3FE8h
		8	47E8h
		9	4FE8h
Q	Location for Reply to VMERD	The reply will be 32 bits of data. Specify %I, %Q, %M, %T, or %G for byte data, or %R, %P, %L, %AI, or %AQ for word data. The output Q address should be a bit memory type starting on a byte boundary.	
LEN	Length of data	Must be 2 words or 4 bytes to read the 32 address bits.	

Reading the Status of the Datagram Queue

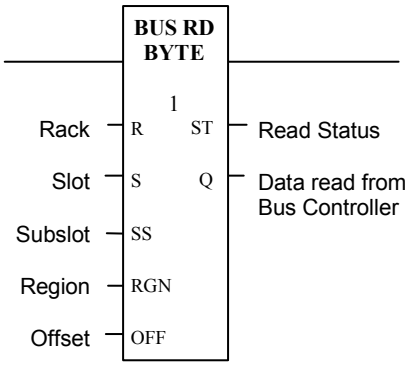
The application program can read the status of a Bus Controller's internal datagram queue directly from the Bus Controller's 32k shared ram memory. Similar program functions are available for an RX7i CPU and for a Series 90-70 CPU. Both are described here.

Format of the Datagram Queue Status Data that is Returned

The function returns one byte of data. If 0 is returned, there are no datagrams in the queue. If 1 is returned, there are one or more datagrams in the queue.

RX7i CPU: Reading Datagram Queue Status

In an RX7i system, the application program can read the datagram queue status by sending a BUSRD_BYTE instruction to the Bus Controller.



No special configuration is required to use this function. The CPU automatically defines region 1 for the Bus Controller to be the Bus Controller's 32k dual port memory. Program the Bus Read parameters as shown below.

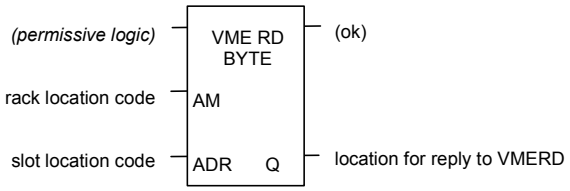
BUS READ Parameters for Reading the Datagram Queue Status

Program the BUSRD_BYTE with the following parameters:

Parameter	Description	Enter
LEN	The number of bytes to read.	1
R	Rack number	The rack number of the Bus Controller
S	Slot number	The slot number of the Bus Controller
SS	Subslot number (optional, defaults to 0)	0
RGN	Region. (Optional, defaults to 1)	1
OFF	Offset	07F2h
ST	(Optional.) Reference for the BUSRD status information.	
Q	Reference for bus status data that is read from the Bus Controller	

Series 90-70 CPU: Reading Datagram Queue Status

In a Series 90-70 system, the application program can read the datagram queue status by sending a VMERD_BYTE instruction to the Bus Controller.



VME READ Parameters for Reading Datagram Queue Status

Parameter		Enter	
AM	Bus Controller Rack Location Code	Rack 0 = 29h Rack 1 = 1Eh Rack 2 = 1Dh Rack 3 = 1Ch Rack 4 = 1Bh Rack 5 = 1Ah Rack 6 = 19h Rack 7 = 18h	
ADR	Bus Controller Slot Location Code	The sum of: 07E8h + (800h x slot number of Bus Controller) Example: Slot 2 = 07E8h + 2 x 800h = 17E8h	
		Slot	**Code**
		2	17E8h
		3	1FE8h
		4	27E8h
		5	2FE8h
		6	38E8h
		7	3FE8h
		8	47E8h
		9	4FE8h
Q	Location for Reply to VMERD	The reply will be 1 byte of data. Specify %I, %Q, %M, %T, or %G for byte data.	
LEN	Length of data	1 byte	

Chapter 7 Data Monitoring, Distributed Control, and Redundancy

This chapter describes the following types of I/O control systems:

- **Data Monitoring:** Where an additional CPU (either a PLC or a computer) monitors inputs and diagnostics from some or all of the blocks on a bus.
- **Distributed Control**: Where two or more CPUs control different I/O blocks on the same bus.
- **Redundancy:** The use of dual busses, dual controllers, or both.

These types of systems are possible because of the unique operation and communications capabilities of Genius I/O devices on a bus. Each Genius I/O block broadcasts its input messages to all other devices on the bus. Therefore, more than one CPU can receive inputs from the same blocks.

The CPUs can communicate on the same bus, allowing formation of a common database. In addition, any CPU can send datagram messages to any other device on the bus.

For Additional Information, Also See:

Chapter 1, which describes non-redundant types of systems.

Chapter 4 for configuration details.

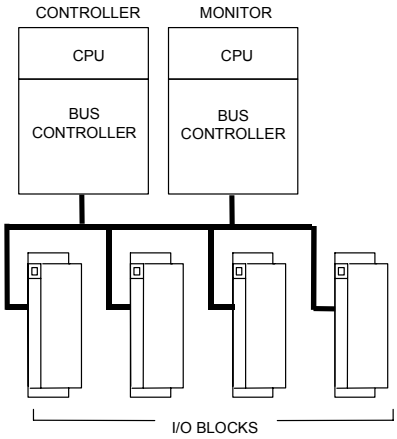
Chapter 5 for descriptions of COMMREQs that might be used in a monitoring or redundancy system.

For information about GMR (Genius Modular Redundancy) for CPU models IC697CPM790, CPU789, and CPU788, please see the *GMR User's Manual*, GFK-1277, which is online at www.ge.com.

For information about Enhanced Hot Standby CPU Redundancy Series 90-70 PLC CPU models IC697CGR772 and CGR935, please see the *Series 90™-70 Enhanced Hot Standby CPU Redundancy User's Guide*, GFK-1527.

Data Monitoring

In addition to the PLC or computer running the application program, other CPUs on the bus can monitor inputs, diagnostics, and Configuration Change messages sent by Genius I/O devices.



Monitoring Inputs

Genius I/O devices broadcast their inputs once per bus scan. These inputs may be accessed by any PLC or computer on the bus.

If the RX7i or Series 90-70 CPU will be used to monitor inputs from I/O devices not being controlled by its application program, the devices will be configured in the same manner as other I/O devices on the bus. The PLC will use the Reference Number assigned to each I/O device to store its inputs. Even though the monitoring PLC would not ordinarily be expected to send outputs to devices being monitored, outputs to those devices should be disabled when the PLC's I/O configuration is done. This will prevent any unwanted outputs being sent to the I/O devices from the monitoring PLC.

If a computer is used to monitor I/O data on the bus, it is important to consider data type, message length, and message format when programming the computer. For example, a High-speed Counter block sends its word-type data first, followed by discrete data. Other devices have different data formats.

Monitoring Diagnostics and Configuration Change Messages

In addition to receiving the broadcast input data, one PLC or computer on the bus may also receive extra copies of any fault reports and configuration change messages that may be sent by the bus devices. This PLC or computer, referred to as the Assigned Monitor, may not send control outputs to an I/O device. If the monitor is capable of sending outputs to a I/O device it is monitoring, those outputs must be disabled.

The monitoring device can communicate with other devices on the bus through Global Data or datagram messages. For example, it would be possible for a monitoring device to send a Read Diagnostics datagram to an I/O device that was not configured to send it fault reports automatically.

This feature is compatible with both redundant and non-redundant Genius configurations.

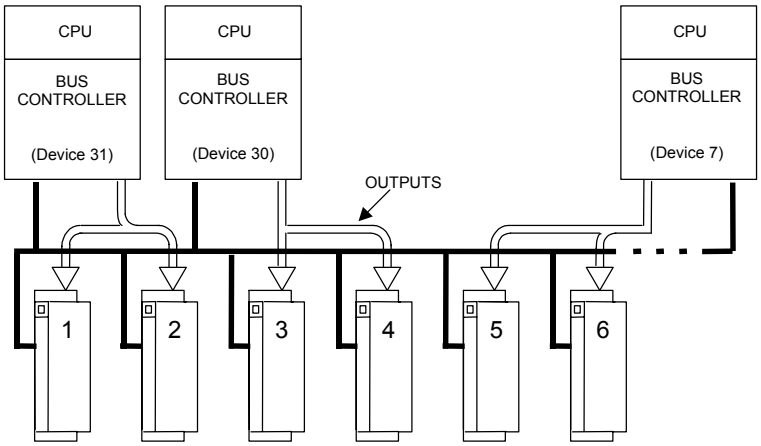
Distributed Control

Distributed control means that two or more Bus Controllers are sending control outputs to different I/O devices on the same bus. Ordinarily, these Bus Controllers would be in different PLCs, but with the Series 90-70 Bus Controller, they can be in the same PLC.

Diagnostics are only automatically sent from the block to the Bus Controller that is controlling its outputs. The Assign Monitor datagram can be used to command blocks on the bus to also direct fault reports to a second Bus Controller.

This is a form of split control, not a type of redundancy. Bus Controllers and devices on the bus are set up for CPU Redundancy Mode = None, since each I/O device is receiving outputs from only one Bus Controller. Remember that all I/O devices on the bus broadcast inputs to all bus interface modules automatically.

For example, a Bus Controller is configured at Bus Address 31, and the I/O devices it will control are configured at Bus Addresses 1 and 2. A Bus Controller in another PLC is located at Bus Address 30. The I/O blocks it will control are located ate Bus Addresses 3 and 4. A third Bus Controller is at Bus Address 7. Two I/O blocks on its bus are located at Bus Addresses 5 and 6. All devices are connected by the same bus.



When setting up an RX7i or Series 90-70 system for distributed control, there are two different ways to assign references to I/O devices:

A. Each Bus Controller can be assigned just those I/O devices whose outputs it controls. If this is done, devices that are not configured for a Bus Controller (but which are actually present on the bus) will generate Extra Device faults in that PLC at startup. Once these faults are cleared, they will not reappear unless power is cycled to the Bus Controller or I/O device.

B. Each Bus Controller can be assigned all of the I/O devices actually present on the bus. Outputs are disabled to I/O devices controlled by another Bus Controller. This means that each I/O device must be assigned a Reference Number in each CPU. References assigned to devices controlled by another CPU are unavailable for further use.

Redundancy

Redundancy provides extra protection for critical processes through duplication of system components. For an RX7i or Series 90-70 (rev 4.0 or later) PLC CPU and Bus Controller (rev. 4.0 or later), the following can be configured:

- Dual bus, one PLC
- Dual bus, two PLCs
- Redundant controllers, one PLC
- Redundant controllers, two PLCs
- Dual bus and redundant controllers, two PLCs

These redundant systems are described on the following pages. If the Series 90-70 PLC is rev. 3, see page 7-17 instead.

Important Considerations

The suitability of a redundancy scheme depends on the requirements of the application. Some important factors to be considered are described below.

1. CPU synchronization is not supported.
   - Using a Genius bus, transferring data from the master CPU to the backup CPU can take 10 to 20 CPU sweeps, depending on the quantity of data.
   - The RX7i or Series 90-70 CPU has transitional bits, but does not have a table that can be transferred from one CPU to another for synchronization. One-shots, counters, and transitional contacts cannot be guaranteed to be the same in both CPUs.
   - The timebase is not transferrable, so timers (real time and running time) cannot be guaranteed to be the same. Timers and counters in program blocks that are not called every sweep may produce different results.
   - For PID function blocks, elapsed time may be different in the two PLCs, because it represents the total time since PLC powerup.
2. Hot Standby redundancy should only be used for systems that do not require bumpless transfer of control from one CPU to the other.

Specific Series 90-70 PLC CPUs provide advanced redundancy features that are not described here.

- Three CPU models support GMR (Genius Modular Redundancy): IC697CPM790, CPU789, and CPU788. A GMR system normally consists of one to three identical CPUs running identical application programs. Control and diagnostic functions are provided by special GMR software. For information about GMR, please see the *GMR User's Manual*, GFK-1277, which is online at www.GE.com.
- Series 90-70 PLC CPU models IC697CGR772 and CGR935, used with a Redundancy Communications Module, provide Enhanced Hot Standby CPU Redundancy for one or more Genius I/O networks. For information about Enhanced Hot Standby CPU Redundancy, please see *the Series 90™-70 Enhanced Hot Standby CPU Redundancy User's Guide*, GFK-1527, also available at www.GE.com.

Dual Bus Redundancy

Dual busses can be used to provide backup protection against cable break or loss or removal of a Bus Controller. Each bus of the dual bus pair has its own Bus Controller. The two Bus Controllers can be located in the same PLC or in two PLCs.

If the Bus Controllers are in the same PLC, they can be placed in the same rack, or they can be placed in different racks to protect against rack failure.

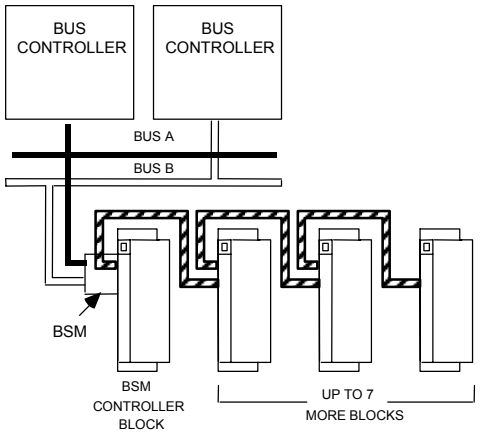
For applications that do not require bumpless transfer of control, the Bus Controllers can be located in different PLCs.

Note

If bumpless transfer is a requirement, bus redundancy with two PLCs is not recommended since the second PLC is essentially off-line before the switch.

Dual Bus Operation

In dual bus redundancy, bus selection is controlled by a switching device (either a Bus Switching Module, or a Remote I/O Scanner module with built-in bus switching capability).



Clusters of up to eight devices each can be connected to a dual bus by a switching device. The maximum number of devices that can be located on both busses is 30, which requires at least 4 bus switching devices.

If the bus switching device stops receiving outputs from the active bus, it automatically switches to the other bus. If the bus it switches to is operational, the regular I/O updates will resume with the Bus Controller on the new bus. An “Output Default Timeout” of 2.5 or 10 seconds must be selected for each bus device, so its outputs do not default during this switchover/login process. Bus switching and block login requires finite periods of time. This varies from system to system, depending on the Genius bus scan time, the CPU sweep time, and the number of devices switching. Generally, switchovers are completed before the 2.5 second timeout expires. The 10-second option is available for systems requiring a longer switchover period. During the timeout period, outputs hold their last valid output state.

If, after switching due to loss of communications on the original bus, no outputs are received on the new bus, the bus switching device does not switch back. It waits until communications are restored on the newly-connected bus, or until power is cycled. This prevents unnecessary switching when no communications are available.

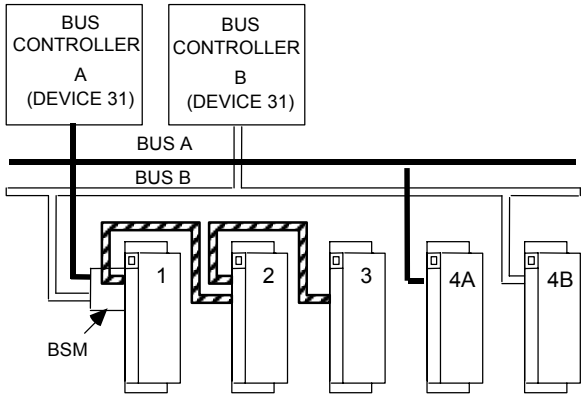
Data Transfer on a Dual Bus

In dual bus redundancy, both Bus Controllers are capable of sending outputs to the devices on the bus. However, the devices in a bus cluster will only receive outputs from the bus that is currently selected by their switching device.

Similarly, although the devices in a cluster continually broadcast input data and diagnostic messages, they are only received by the Bus Controller on the bus that is currently selected by their switching device. The Bus Controller on the inactive bus cannot receive inputs, fault reports or Configuration Change messages.

Non-redundant Devices on a Dual Bus

Although most devices in a dual bus system will probably be connected to both bus cables via a switching device, it is possible to have non-redundant devices connected directly to one bus of the pair. The following illustration represents a dual bus with some non-redundant I/O blocks.

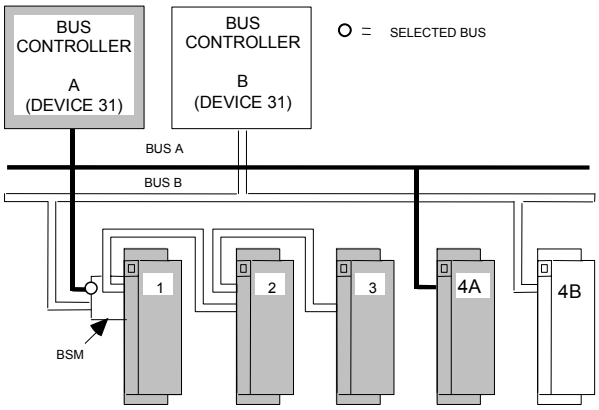


Both of the Bus Controllers are configured at Bus Address 31 on their respective busses. A Bus Switching Module interfaces three redundant I/O blocks to the dual bus. The redundant blocks are configured at Bus Addresses 1, 2, and 3 on both busses.

There are also two non-redundant I/O blocks. Each of them is configured at Bus Address 4 on its bus.

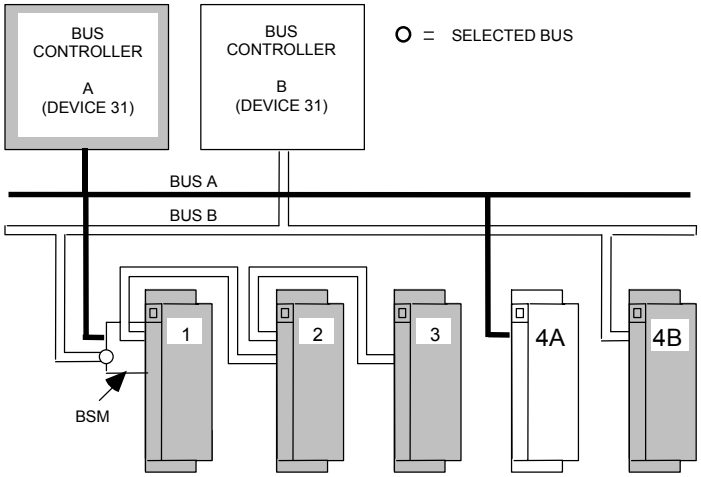
During normal operation, both bus A and bus B operate in the same way as a single bus:

- Blocks 1, 2, and 3 interface to either bus A or bus B, as selected by the BSM.
- Block 4A communicates with the Bus Controller on bus A only.
- Block 4B communicates with the Bus Controller on bus B only.



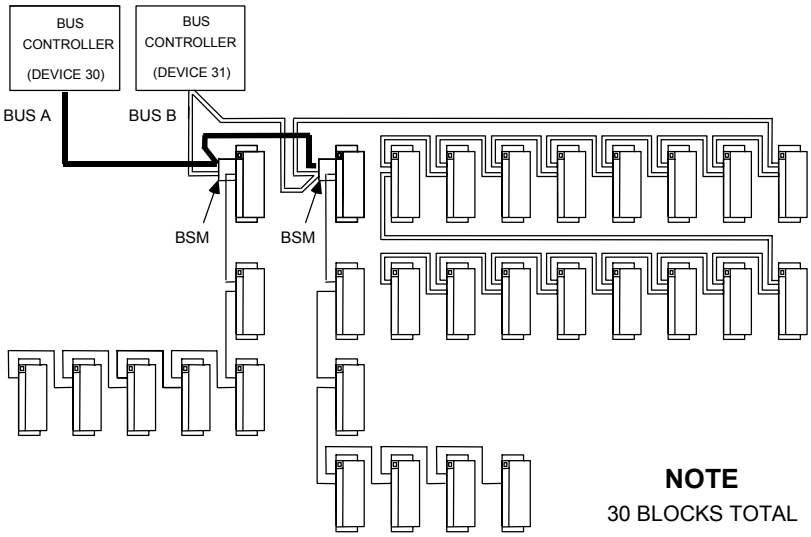
If Bus Controller A stops communicating with the redundant blocks (due to program action, a Bus Controller fault, a cable break, or loss of power), then:

- The BSM will switch the cluster of blocks 1, 2, and 3 to bus B.
- Block 4A, which is a non-redundant block, will no longer receive outputs from its Bus Controller, and will no longer be able to send inputs or diagnostics to the PLC. If there are outputs on block 4A, they will either hold their last state or default, depending on the block's configuration. Although communications have been interrupted, the block is still receiving power, so any outputs that were ON or that default to ON will continue to operate.



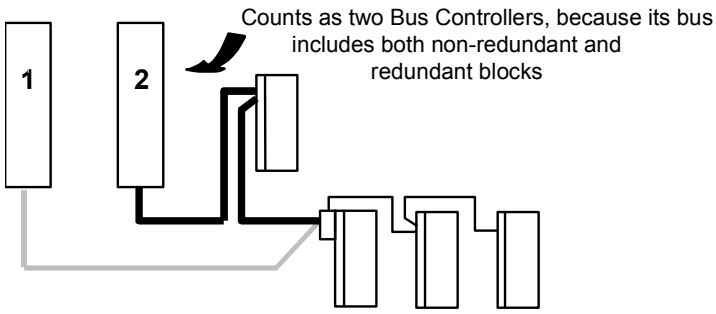
Number of Bus Devices on a Dual Bus

Up to 30 bus devices can be connected, either directly or as part of a cluster, to a dual bus. Redundant devices count toward both busses' totals. Non-redundant devices only count in the total of the bus to which they are directly connected. That means more devices can be used on a dual bus if some are not redundant.



Number of Bus Controllers in a PLC with Dual Busses

Although using non-redundant devices on a dual bus increases the total number of bus devices that can be used on a dual bus, *it decreases the number of Bus Controllers that can be used in the PLC.* That is because any Bus Controller that has both redundant devices and non-redundant devices on its bus *counts as 2 Bus Controllers* against the total of 31 permitted in a system.

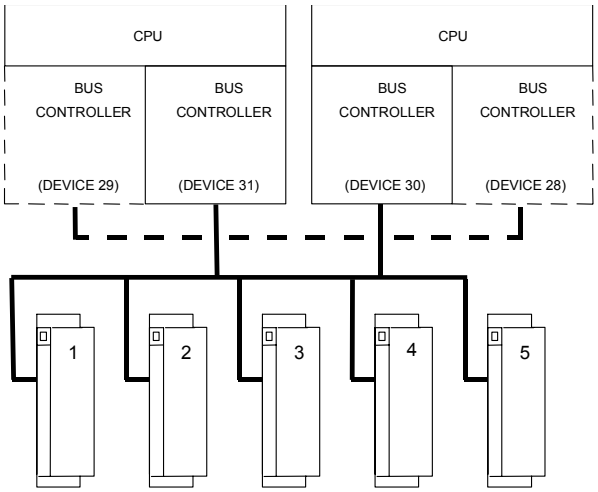


In a large system, grouping non-redundant devices on the same bus or busses will permit the greatest number of Bus Controllers to be used in the PLC. For example, a PLC could have 30 Bus Controllers with all redundant devices, and 1 Bus Controller with all non-redundant devices.

Dual Bus with the Bus Controllers in Two PLCs

In a dual bus system where the Bus Controllers are in the same PLC, the same application program automatically acts on inputs received from the devices and creates outputs for them, regardless of which bus is active at any given time.

But if the Bus Controllers are NOT in the same PLC, the application program must monitor the busses dynamically to determine the correct reference to use at any given time. Because the CPUs cannot communicate with each other on the dual bus, another Bus Controller is needed in each CPU, on each bus, to transmit synchronization data between the two CPUs. Global Data or datagrams can be used.



The illustration shows an optional Bus Controller in each CPU, connected via an additional Genius bus, for data sharing between the CPUs. In this example, the additional Bus Controllers are for communications only; they do not control I/O. In that case, it is not necessary to give them the same Bus Addresses.

Disabling Outputs from the Backup Bus Controller

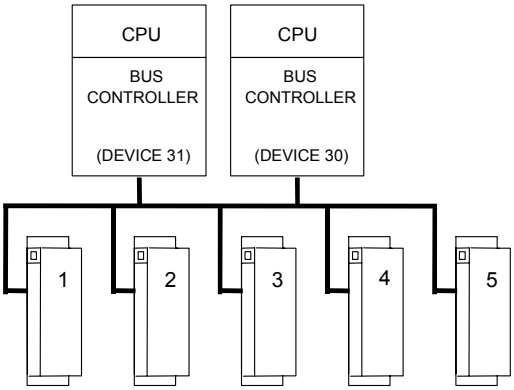
When using bus redundancy with two PLCs, it may be necessary to disable the outputs sent by the backup Bus Controller until the application program has logged in all the devices, then enable outputs under program control. If this is done, the additional time without outputs must not cause the total time without outputs to exceed the 2.5 or 10 second timeout selected for the block.

Redundant Controllers

Redundant controllers provide backup controller protection for devices on a bus. The redundant controllers can be in the same PLC (in the same rack or in different racks), or in two PLCs.

Bus Controller redundancy with one PLC provides protection against failure of the Bus Controller.

Bus Controller redundancy with two PLCs, represented below, provides protection against failure in the Bus Controller or elsewhere in the primary PLC.



Synchronizing Dual CPUs

Since bus devices broadcast their inputs to all CPUs on a bus, redundant Bus Controllers in separate PLCs need to maintain synchronization of their output data. Datagrams and Global Data can be used to synchronize the PLCs.

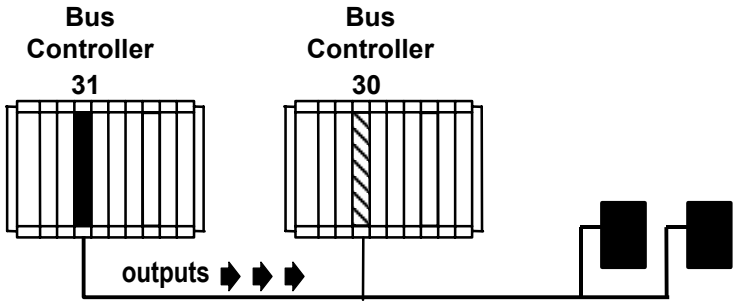
Either PLC can monitor the outputs of the other using the Outputs with Feedback feature of discrete Genius I/O blocks. Since I/O blocks can monitor the actual state of the load and feed this state back to the PLC as input data, both PLCs can monitor the actual state of all outputs.

Bus Device Configuration: Hot Standby or Duplex Mode

For a redundant controller system, devices on the bus can be individually configured (using a Hand-held Monitor or Write Configuration datagrams) for Hot Standby or Duplex CPU Redundancy mode, or none.

Hot Standby Mode

If the system does NOT require bumpless transfer of control from one PLC to the other, devices on the bus can be configured for Hot Standby CPU redundancy. Here, Hot Standby mode is shown using two PLCs. However, it can also be done with one PLC; with one rack or separate racks.

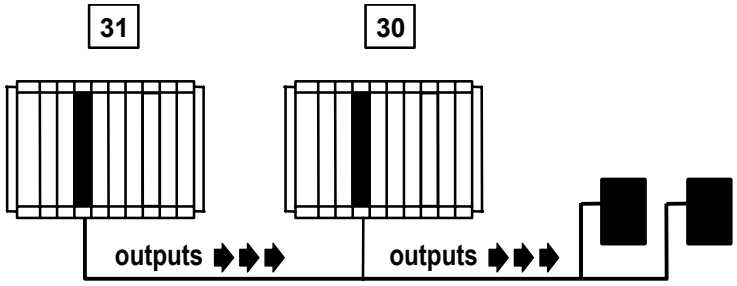


In Hot Standby mode, blocks receive outputs from both Bus Controllers, but they are normally controlled directly by the Bus Controller at Bus Address (Device Number) 31. If no output data is available from Bus Address 31 for a period of three bus scans, the outputs are immediately controlled by the Bus Controller at Bus Address 30. If output data is not available from either 30 or 31, outputs go to their configured default or hold their last state. The Bus Controller at Bus Address 31 always has priority, so that when 31 is on-line, it always has control of the outputs.

Analog blocks, when configured for CPU redundancy, must be operated in Hot Standby redundancy mode.

Duplex Redundancy Mode

If a bus device is configured for Duplex mode, it receives outputs from BOTH Bus Address 30 and 31 and compares them. Here, Duplex mode is shown using two PLCs. It can also be done with one PLC, with one rack or separate racks.



If both outputs are the same, the device sets the output to that state. If both outputs are not the same, the device sets the output to its preselected Duplex Default State. The following table shows how outputs operate in Duplex redundancy.

Commanded State, from Bus Address 31	*Commanded State, from Bus Address 30*	*Configured Duplex Default State*	*Actual Output State*
On	On	Don‘t Care	On
Off	On	Off	Off *
Off	Off	Don’t Care	Off
On	Off	On	On *
* Decided by “Duplex Default State” selection.			

If either 30 or 31 stops sending outputs to a device, the outputs are directly controlled by the remaining device.

Only discrete blocks can be configured for Duplex redundancy mode. If there are analog blocks on the same bus, they can be configured in Hot Standby mode or no CPU redundancy.

Note

In both Hot Standby and Duplex modes, both CPUs get the inputs from the blocks automatically. In addition, the blocks automatically send fault reports to both Bus Controllers.

Dual Bus and Redundant Controllers

The two methods just described can be combined for dual bus and controller redundancy. A dual bus/dual controller system provides protection against failure in the bus trunk cable, the Bus Controller and the PLC. Through application programming, dual bus/dual controller redundancy can be implemented in two different ways:

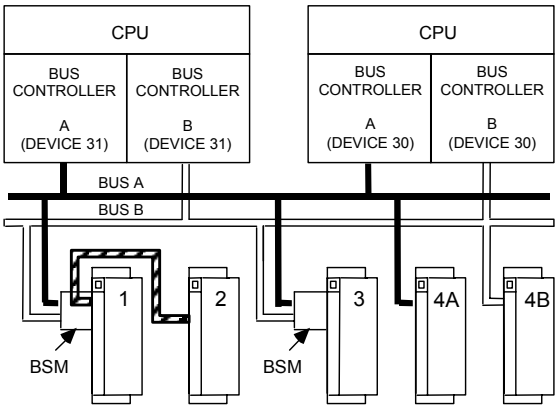
A. For operation with both Hot Standby and Duplex devices on the bus. This application does NOT provide bumpless transfer of control.

B. For bumpless transfer of control as long as both Bus Controllers in the primary PLC are available (transfer is not bumpless between PLCs, however). This application is not suitable for devices that must operate in Duplex CPU Redundancy mode.

Details of both types of application are given on the following pages.

Basic Operation of a Dual Bus/Dual Controller System

Both Bus Controllers in a given PLC must use the same Bus Address (either 30 or 31).



In the example system represented above, both bus A and bus B operate in the same way as a single bus, dual CPU system. Blocks 1, 2, and 3 interface to both PLCs via Bus Controllers 31(A) and 30(A) whenever the active bus is bus A, or via Bus Controllers 31(B) and 30(B) whenever the active bus is bus B. Block 4(A) interfaces to both PLCs via Bus Controllers 31(A) and 30(A). Block 4(B) interfaces to both PLCs via Bus Controllers 31(B) and 30(B).

Outputs

All four Bus Controllers are capable of sending outputs, although only outputs from the Bus Controllers on the active bus are actually received. It may be necessary to disable the outputs sent by the backup Bus Controller until the application program has logged in all the devices, then enable outputs under program control. If this is done, the additional time without outputs must not cause the total time without outputs to exceed the 2.5 or 10 second timeout selected for the block.

Inputs and Diagnostics

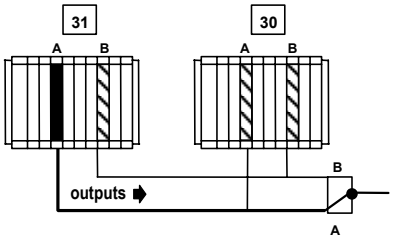
Both Bus Controllers on the selected bus automatically receive all inputs and fault reports from any device on the bus that has been configured as being in “CPU Redundancy” mode.

Bus and Controller Redundancy for Hot Standby Devices

When both PLCs provide outputs, and devices are configured for Hot Standby CPU Redundancy, the sequence of control in case of bus controller, bus, or PLC failure is:

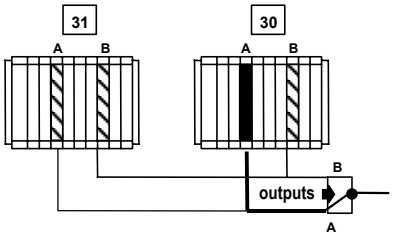
primary PLC, bus A
► ***backup PLC, bus A***
► ***primary PLC, bus B***
► ***backup PLC, bus B***

Because the PLCs are operating independently, each time the control switches from one PLC to the other there may be a “bump” in the process. This may be of no consequence in some applications, and of significant consequence in others.



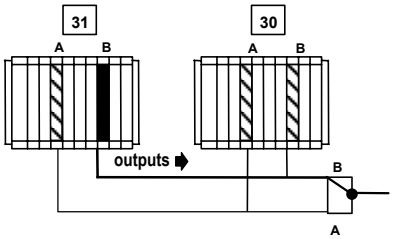
Normal Operation

In the default setup shown at left, during normal operation Bus Controller 31A in the primary PLC controls all devices set up for Hot Standby CPU redundancy.



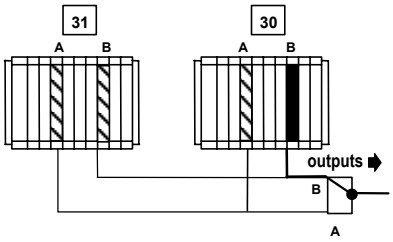
Control Passes to Backup PLC on Bus A

If the device fails to receive valid output data from Bus Controller 31 for three bus scans, it will permit Bus Controller 30A to control its outputs.



Control Passes to Bus B

If the device that controls bus switching stops receiving outputs from bus A for a period of three bus scans, it switches to bus B. Normal operation then resumes on bus B. Bus Controller 31B controls (in the primary PLC) controls all devices set up for Hot Standby CPU redundancy.



Control Passes to Backup PLC on Bus B

If the device fails to receive valid output data from Bus Controller 31 for three bus scans,it will permit Bus Controller 30B to control its outputs.

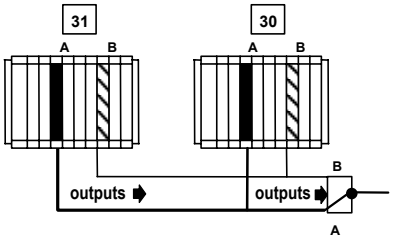
Bus and Controller Redundancy for Duplex Devices

If the application requires that bus devices operate in Duplex CPU Redundancy mode, outputs must be enabled to both the primary and the backup PLC. The sequence of control in case of Bus Controller, bus, or PLC failure is:

control shared by Bus Controllers 31 and 30 on bus A

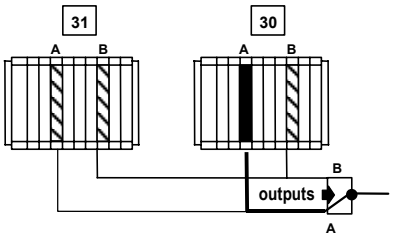
- ***Bus Controller 30, bus A***
  - ***control shared by Bus Controllers 31 and 30 on bus B***
    - ***Bus Controller 30, bus B***

Because the PLCs are operating independently, each time the control switches from one PLC to the other there may be a “bump” in the process. This may be of no consequence in some applications, and of significant consequence in others.



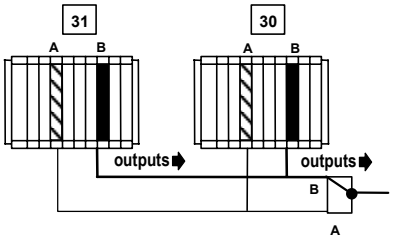
Normal Operation

In the default setup shown at left, during normal operation Bus Controllers 31A and 30A jointly control any devices set up for Duplex CPU redundancy.



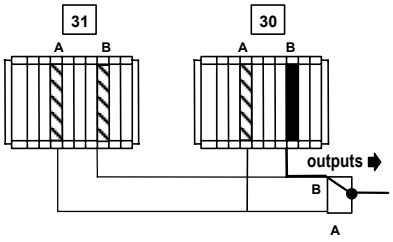
Devices Controlled by Bus Controller 30 on Bus A

If a duplex device fails to receive output data from Bus Controller 31A for three bus scans, it will permit Bus Controller 30A to control its outputs.



Control Passes to Bus B

If the device that controls bus switching stops receiving outputs from bus A for a period of three bus scans, it switches to bus B. Normal operation then resumes on bus B. Bus Controllers 31B and 30B jointly control any devices set up for Duplex CPU redundancy.



Control Passes to Backup PLC on Bus B

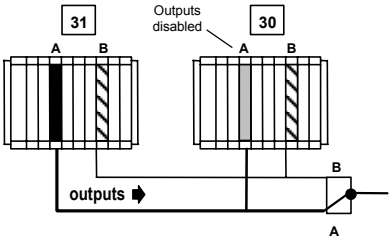
If a duplex device fails to receive output data from Bus Controller 31B for three bus scans, it will permit Bus Controller 30B to control its outputs.

Operation Remains with Primary PLC, Dual Bus and Dual Controllers

The following application is for devices configured in Hot Standby CPU Redundancy mode. It is not suitable for devices that operate in Duplex CPU redundancy mode. The order of control is:

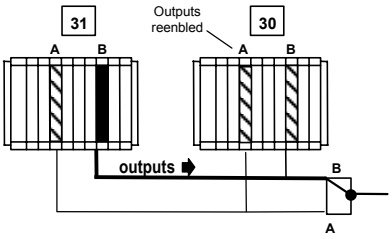
primary PLC, bus A
► primary PLC, bus B
► backup PLC, bus B
► backup PLC, bus A

This provides "bumpless" transfer of control within the primary PLC, and within the backup PLC, although there will still be a bump in the process when control is transferred from PLC to PLC.



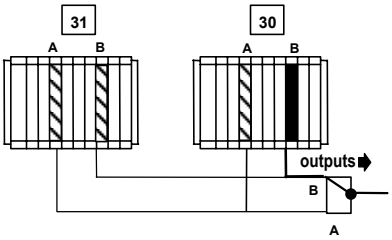
Normal Operation

During normal operation, Bus Controller 31A in the primary PLC controls all devices. At powerup, the application program in the backup PLC sends a Disable Outputs COMREQ to Bus Controller 30A. That prevents the backup PLC from assuming control following a bus switch from bus A to bus B (see below).



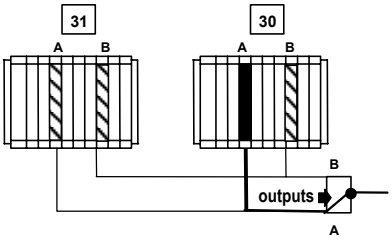
Control Passes to Bus B

Because Bus Controller 30A in the backup PLC is not sending outputs, if the bus switching device stops receiving outputs from Bus Controller 31A for a period of three bus scans, it switches to bus B. Normal operation then resumes on bus B. Bus Controller 31B in the primary PLC controls all devices set up for Hot Standby CPU redundancy. After the bus switch is completed, the application program in the backup PLC should re-enable outputs from Bus Controller 30A, so it will be ready to resume control if needed.



Control Passes to Backup PLC on Bus B

If Bus Controller 31B stops sending outputs, or if the primary PLC is not available when the bus switches, Bus Controller 30B in the backup PLC controls all devices.



Control Passes to Backup PLC on Bus A

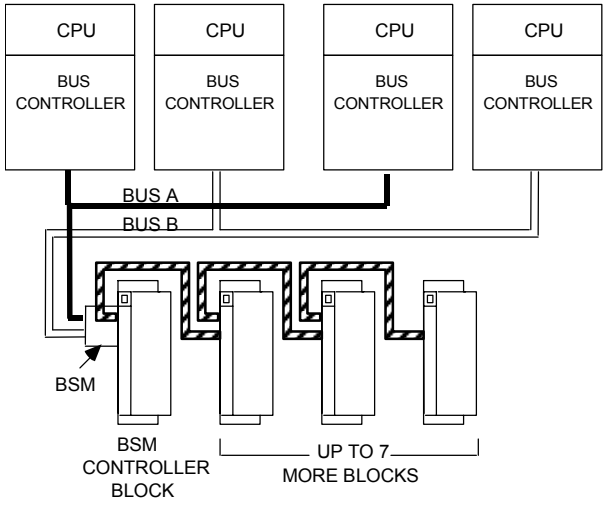
If a device stops receiving outputs from Bus Controller 30B for three bus scans, the bus switches to A again. If outputs from Bus Controller 31A have not been restored, Bus Controller 30A in the backup PLC assumes control. If outputs from Bus Controller 31A have been restored, normal operation resumes. The application program in the backup PLC should once again disable outputs from Bus Controller 30A to bring the system back to its original operating mode.

Genius Redundancy for Series 90-70 Rev. 3

Bus redundancy for a rev. 3 CPU and Bus Controller requires the written approval of GE application engineering. Operation of bus redundancy is as described earlier in this chapter.

Dual Bus and Dual Controllers

To provide CPU redundancy, Bus Controller redundancy, and bus redundancy, a Series 90-70 PLC with version 3 CPU and Bus Controller must include two or four CPUs and four Bus Controllers. Clusters of up to eight devices each can be connected to both busses by bus switching devices.



A system like the one depicted above provides protection against single point failure in a CPU or Bus Controller or on a bus trunk cable. It does not protect against failure of a bus switching device, a BSM controller block, or a bus stub connecting blocks in a cluster.

The devices in each cluster communicate only with the bus that is currently selected. The other CPU does not receive inputs, Report Fault datagrams, or Configuration Change datagrams from the blocks. Therefore, if the switching device switches busses, the newly-selected CPUs will not have the most current inputs or diagnostics from the devices in the cluster.

In a version 3 system, bus devices MUST be configured with one set of references for operations on bus A, and a separate set of references for operations on bus B. The application program must decide which bus is operational, and use the appropriate set of references for the devices' I/O data.

Appendix
A

ASCII Code List

In Read Device and Write Device datagrams, either uppercase or lowercase letters can be used for program and task names *if the Bus Controller is version 3.0 or later*. For earlier versions of the Series 90-70 Bus Controller, program and task names must be all uppercase.

Char.	Dec.	Hex.	Char.	Dec.	Hex.	Char.	Dec.	Hex.
NUL	0	00	0	48	30	[	91	5B
SOH	1	01	1	49	31	\	92	5C
STX	2	02	2	50	32	]	93	5D
ETX	3	03	3	51	33	^	94	5E
EOT	4	04	4	52	34	_	95	5F
ENQ	5	05	5	53	35	`	96	60
ACK	6	06	6	54	36	a	97	61
BEL	7	07	7	55	37	b	98	62
BS	8	08	8	56	38	c	99	63
HT	9	09	9	57	39	d	100	64
LF	10	0A	:	58	3A	e	101	65
VT	11	0B	;	59	3B	f	102	66
FF	12	0C	<	60	3C	g	103	67
CR	13	0D	=	61	3D	h	104	68
SO	14	0E	>	62	3E	i	105	69
SI	15	0F	?	63	3F	j	106	6A
DLE	16	10	@	64	40	k	107	6B
DC1	17	11	A	65	41	l	108	6C
DC2	18	12	B	66	42	m	109	6D
DC3	19	13	C	67	43	n	110	6E
DC4	20	14	D	68	44	o	111	6F
NAK	21	15	E	69	45	p	112	70
SYN	22	16	F	70	46	q	113	71
ETB	23	17	G	71	47	r	114	72
CAN	24	18	H	72	48	s	115	73
EM	25	19	I	73	49	t	116	74
SUB	26	1A	J	74	4A	u	117	75
ESC	27	1B	K	75	4B	v	118	76
FS	28	1C	L	76	4C	w	119	77
GS	29	1D	M	77	4D	x	120	78
RS	30	1E	N	78	4E	y	121	79
US	31	1F	O	79	4F	z	122	7A
SP	32	20	P	80	50	{	123	7B
!	33	21	Q	81	51	|	124	7C
"	34	22	R	82	52	}	125	7D
#	35	23	S	83	53	~	126	7E
$	36	24	T	84	54	“	127	7F
%	37	25	U	85	55			
&	38	26	V	86	56			
'	39	27	W	87	57			
(	40	28	X	88	58			
)	41	29	Y	89	59			
*	42	2A	Z	90	5A			
+	43	2B						
,	44	2C						
-	45	2D						
.	46	2E						
/	47	2F						

A

AD Comm fault, 4-7
Add. of Bus Controller fault, 4-7
Addition of Block fault, 4-7
Addition of Bus Controller status reference, 4-2
Addition of I/O Module status reference, 4-2
Addition of Rack status reference, 4-2
AI Hi/Low Alarm fault, 4-6
AI Under/Overrange fault, 4-6
Alarm Contacts, 4-4
Analog Data Format, 1-7, 3-25
Analog faults, 4-6
AQ Under/Overrange fault, 4-6
Assign Monitor COMREQ, 5-1, 5-5, 5-18, 6-1
Assigned Monitor, 3-30

B

Baud Rate, 1-5, 3-3
Bit or Byte mode for memory-access datagrams, 5-25
Bus
- Configuration, 3-24
- Connection to, 2-3
- Continuity, 2-4
- Length, 1-5
- Terminating, 2-4
- Type, 1-5

Bus Address, 3-2
Bus Controller
- Configuration, 3-2, 3-34
- Description, 1-4
- Installation, 2-2
- Number in System, 1-3
- Operation, 1-6
- Removal, 2-2

Bus Controller Fault status reference, 4-2
BUS CONTROLLER software fault, 4-8
Bus fault, 4-7
Bus Fault status reference, 4-2
Bus Out Disable fault, 4-7
Bus Scan, 1-6, 5-30
Bus Scan Time, 5-34

C

Cal Mem fault, 4-7
Catalog Number, 1-3
Circuit Faults, 4-6
Clear All Circuit Faults COMREQ, 5-1, 5-5, 5-17, 6-1
Clear Circuit Fault COMREQ, 5-1, 5-5, 5-17, 6-1
Command Block, 5-11, 5-27
Communication Request Commands, 5-1, 6-1
COMREQ, 5-1, 6-1
- Command Block, 5-3
- Command Numbers, 5-4
- Commands
  - Assign Monitor, 5-5, 5-18
  - Clear All Circuit Faults, 5-5, 5-17
  - Clear Circuit Fault, 5-5, 5-17
  - Dequeue Datagram, 5-5, 5-29
  - Global Data Enable, 5-5, 5-21
  - I/O Fault Categories Enable, 5-37
  - I/O Faults Enable, 5-5
  - Outputs Enabled, 5-5, 5-20
  - Pulse Test, 5-5, 5-13
  - Read Configuration, 5-5, 5-14
  - Read Device, 5-5, 5-23
  - Read Diagnostics, 5-5, 5-16
  - Request Datagram Reply, 5-5, 5-36
  - Send Datagram, 5-5, 5-32
  - Switch BSM, 5-5, 5-22
  - Write Configuration, 5-5, 5-15
  - Write Device, 5-5, 5-28
- Examples, 5-10, 5-11
- Outputs, 5-7
- Program Instruction, 5-6
- Quick Reference, 5-5
- Status, 5-9
- Status Block, 5-8
- Status Pointer, 5-3

COMREQs and Passwords
- Allowable password levels, 5-1

Config Mem fault, 4-7
Configuration
- Bus Controller, 3-2, 3-34
- Devices on bus, 3-24
- Global Data, 3-11, 3-34
- High-speed Counter, 3-36
- PCIM, QBIM, 3-34
- PowerTRAC Block, 3-36
- Remote I/O Scanner, 3-33

Configuration mismatch reference, 4-2
Connection to the Bus, 2-3
CPU Redundancy, 7-10
CPU Sweep, 5-30, 5-34
CS Feedback Error fault, 4-6

D

Data Monitoring, 3-30, 7-2
Data quantities, 1-8
Datagram Priority, 5-34
Datagrams, 1-10, 5-5
- Incoming, 5-12, 5-30

Number per CPU Sweep, 5-34
Priority, 5-32
Ways to Send, 5-12, 5-32
Dequeue Datagram Command, 5-29
Dequeue Datagram COMREQ, 5-5
DG Queue fault, 4-8
Diagnostics, 1-9, 7-2
Disable Outputs, 3-29, 3-30
Discrete faults, 4-6
Distributed Control, 3-29
Dual Bus
configuration, 3-15

E

Enable/Disable Global Data COMREQ, 5-1, 5-21, 6-1
Enable/Disable Outputs COMREQ, 5-1, 5-20, 6-1
Errors, number on bus, 3-2
Excessive faults, 4-8
External GBC, redundant
configuration, 3-9
Extra Block fault, 4-8

F

Fault
Contacts, 4-3
Fault Clearing, 4-5
Fault Identification Reference, 4-6
Fault Locating References, 4-3
Fault Logged, status reference, 4-2
Fault Reports, disable sending to CPU, 5-37
Fault Table, 4-5
Forced/Unforced Circuit fault, 4-8
Fuse Blown fault, 4-6, 4-7

G

GBC Software Exception fault, 4-8
GENA faults, 4-6
Genius blocks, 1-3, 1-8
Genius Bus Scan, 1-6
Global Data, 5-28
%G memory for, 3-11
Automatic configuration, 3-11
Configuration, 3-11, 3-34
Length, 3-11
Receiving, 3-35
Reference for another host type, 3-11
Series 90-70 PLC, 1-11
Global Data Enable COMREQ, 5-5, 5-21

H

Hand-held Monitor
Compatibility, 1-4
Connector on Bus Controller, 1-4
Headend fault, 4-7
HIgh Error Rate fault, 4-8
High-speed Counter, 3-25
High-speed Counter, configuration, 3-36

I

I/O Blocks, 1-3, 1-8
I/O bus fault, 4-7
I/O data, 1-8
I/O Fault Categories Enable COMREQ, 5-5, 5-37
I/O Module Fault status reference, 4-2
I/O Table Full, status reference, 4-2
Input Short fault, 4-6
Inputs
Monitoring, 7-2
Inputs and Outputs, 1-6, 1-7, 3-25
Install the Bus Controller, 2-2
Internal fault, 4-6
Internal GBC, redundant
configuration, 3-9
Intrnal Ckt fault, 4-7
Isolation, 1-5

J

Jumpers on Terminal Assembly, 2-4

L

LEDs
Channel OK, 1-4, 3-2
I/O Enabled, blocks, 3-29
Module OK, 1-4
LL Analog Faults, 4-6
Loss of Block fault, 4-7
Loss of BUS CONTROLLER fault, 4-8
Loss of Bus Controller status reference, 4-2
Loss of I/O Module status reference, 4-2
Loss of or Missing IOC fault, 3-6, 3-10
Loss of Rack status reference, 4-2
Loss Power fault, 4-6
LP Mail Rejected fault, 4-8

M

Memory for Global Data, 3-11
Memory for I/O Blocks, 1-8, 3-25

Memory for Read/Write Device, 5-24
Modulation technique, 1-5
Module Description, 1-4
Monitoring, 7-2

N

No Fault contacts, 4-3
No Load fault, 4-6
Not Spec fault, 4-7

O

Open Wire fault, 4-6
Outputs
  Disable, 3-29, 3-30
Outputs Enable COMREQ, 5-5, 5-20
Over Temp fault, 4-6
Overload fault, 4-6

P

Paired GBC
  configuration, 3-9
Passwords
  Levels suitable for COMREQs, 5-1
PCIM
  Configuration, 3-34
  Receiving Global Data, 1-11
Point fault, 4-6, 4-7
PowerTRAC Block, configuration, 3-36
Priority, 5-34
Programming for a COMREQ, 5-2
Pulse Test COMREQ, 5-1, 5-5, 5-13, 6-1

Q

QBIM
  Configuration, 3-34
  Receiving Global Data, 1-11

R

Read Configuration COMREQ, 5-1, 5-5, 5-14, 6-1
Read Configuration Datagram, 5-36
Read Data Datagram, 5-36
Read Device COMREQ, 5-1, 5-5, 5-23, 6-1
Read Device Datagram, 5-36
Read Diagnostics COMREQ, 5-1, 5-5, 5-16, 6-1
Read Diagnostics Datagram, 5-36
Redundancy, 3-31, 7-4
Redundancy Mode
  configuration, 3-6, 3-15
Redundancy, references for, 3-25
Redundant control
  configuration, 3-7
Reference Address, 3-25
References, not used, 1-8
References, status, 4-2
Remote Drop, 1-3, 3-33
Remote I/O Scanner, 3-33
Remove the Bus Controller, 2-2
Request Datagram Reply COMREQ, 5-5, 5-36
RW Queue Full fault, 4-8

S

SBA conflict fault, 4-7
Send Datagram COMREQ, 5-5, 5-32
Series 90-70 PLC Global Data Operation, 1-11
Share RAM fault, 4-7
Short Circuit fault, 4-6
Signal/noise ratio, 1-5
Software Failure status reference, 4-2
Status LEDs, 1-4
Status of COMREQ, 5-9
Subfunction Code, 5-12, 5-36
Switch BSM COMREQ, 5-1, 5-5, 5-22, 6-1
Switch Fault fault, 4-6
Switching time, of redundant bus
  configuration, 3-9
System Status references, 4-2

T

Terminating the Bus, 2-4
Timing
  Bus scan and CPU sweep, 5-34
Token passing, 1-6
Too Many Bus Controllers, status reference, 4-2

U

User Scaling Error fault, 4-7

W

WD Timeout fault, 4-7
Wiring Error fault, 4-6
Write Configuration COMREQ, 5-1, 5-5, 5-15, 6-1
Write Device COMREQ, 5-1, 5-5, 5-28, 6-1