GE Fanuc Automation

Series Five®
Programmable Controllers
Data Communications

User's Manual

GFK-0244B

June 1990

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 Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation 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.

Preface

This manual describes the installation, operation, and programming of the Series Five® PLC Communications Control Module (CCM). The CCM provides a direct serial communication link to the data communication network.

You should become familiar with the operation of the Series One®, Series Three®, Series Five®, or Series Six® PLC (depending on your application) before reading this manual. Refer to the appropriate Series type of Programmable Logic Controller (PLC) communication manual for complete information.

Chapter 1. Introduction: Describes the capabilities of the CCM and possible system configurations of Series One, Series Three, Series Five and Series Six PLCs with the Series Five PLC or host computer.

Chapter 2. Installation: Describes the physical layout, configuration, and Series Five PLC installation of the CCM. Provides information need to construct cables to connect the CCM to other devices on the network.

Chapter 3. Memory Map: Identifies the areas in the Series Five PLC internal memory which provide either special functions, system status information, or error reporting information. These items in memory are defined in the input, output and register tables, and the CPU's scratch pad memory.

Chapter 4. Communication Examples: Explains the Series Five PLC ladder logic programming which initiates communications between the CCM and other nodes on the network. Example programs have been included.

Chapter 5. CCM Protocol: Provides reference information on the CCM serial interface protocol and timing to allow the user to write a serial communications driver for a host computer or microprocessor.

Chapter 6. RTU Communications Protocol: Describes in detail the protocol used when configured in Remote Terminal Unit (RTU) mode.

Chapter 7. Additional Protocols: Describes additional modes of operation which provide communication with non-CCM devices such as printers, bar code readers, and personal computers.

Appendix A. CCM Memory Types: Provides an expanded listing of the memory mapping of input and outputs.

Related Publications

- *GEK-90842 Series One/Series One Plus PLC User's Manual*
- *GEK-25376 Series Three PLC User's Manual*
- *GEK-90507 Series One/Series Three Remote I/O User's Manual*
- *GEK-90477 Series One/Series Three PLC Data Communications Manual*
- *GFK-0122 Series Five PLC User's Manual*
- *GFK-0023 Logicmaster® Five Programming User's Manual*
- *GEK-25364 Series Six Data Communications Manual*
- *GEK-96602 Series Six Plus PLC User's Manual*

Janet Swisher
Technical Writer

CUT ON DOTTED LINE

READER'S COMMENTS

We invite your comments and welcome suggestions to make this manual more useful.

Publication No. _____ Date of Publication _____ Today's Date _____

GENERAL COMMENTS:

	Improve	Acceptable	Good	Excellent
Contents	☐	☐	☐	☐
Organization	☐	☐	☐	☐
Accuracy	☐	☐	☐	☐
Clarity	☐	☐	☐	☐
Completeness	☐	☐	☐	☐
Examples/Illustrations	☐	☐	☐	☐
Referencing/Indexing	☐	☐	☐	☐
Readability	☐	☐	☐	☐

DETAILED COMMENTS: (Correct, expand, etc. – Please be specific.)

Page No. **Comment**

_____ _____

_____ _____

_____ _____

_____ _____

Other suggestions for improving this document: _____

_____

As compared to other manufacturers of a similar product, how would you rate this document ?

Superior ☐ Comparable ☐ Inferior ☐ Don't Know ☐

Commments: _____

_____

Are you interested in subscribing to a documentation update plan? Yes ☐ No ☐

APPLICATION INFORMATION:

Indicate the type of user/reader function that you most nearly represent:

☐ System Designer ☐ Programmer
☐ Distributor ☐ Maintenance
☐ OEM ☐ Operator
☐ Installation ☐ Other (Please Specify) _____

Type of Equipment: ☐ Series 90-70 ☐ Series 90-30 ☐ Series 90-20 ☐ Series Six
☐ Series Five ☐ Series One ☐ Genius I/O ☐ Other _____

Comments concerning your specific application: _____

_____

_____

_____

From:

Name: _____

Title: _____

Company: _____

Address: _____

City/State/Zip: _____

Telephone: _____

Fold Here

No Postage Necessary If Mailed In The United States

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 995 CHARLOTTESVILLE, VA

POSTAGE WILL BE PAID BY ADDRESSEE:

ATTENTION MANAGER TECHNICAL PUBLICATIONS
GE Fanuc Automation North America Inc
P O BOX 8106
CHARLOTTESVILLE VA 22907-6063

Fold Here

GFK-0244

Table

Chapter 1 Introduction

GFK-0244

This chapter describes the capabilities and system configurations for the Series Five® PLC Communications Control Module (CCM) for serial communications with the Series One®, Series Three®, Series Five or Series Six® family of Programmable Logic Controllers (PLCs).

Communication Capabilities

Communication Control Modules (CCM) can be installed only in the Series Five CPU rack to provide serial data communications between the Series Five PLC and other PLCs, or host computers on the network. Operational characteristics of the CCM are as follows:

- Master-to-Slave, or Peer-to-Peer* communication
- High execution rate through Random Access Memory (RAM), shared with CPU RS-232C, or RS-422
- Data transmission error checking
- Built in RS-232C to RS-422 conversion
- Remote Terminal Unit (RTU) protocol capability

*IC655CCM500E (Rev.E) module or later

System Configuration

The CCM plugs directly into the Series Five PLC rack and provides a serial data communication link to other nodes on the multidrop network. Figure 1-1 identifies the major components of the Multidrop network and how they relate to the Series Five CCM.

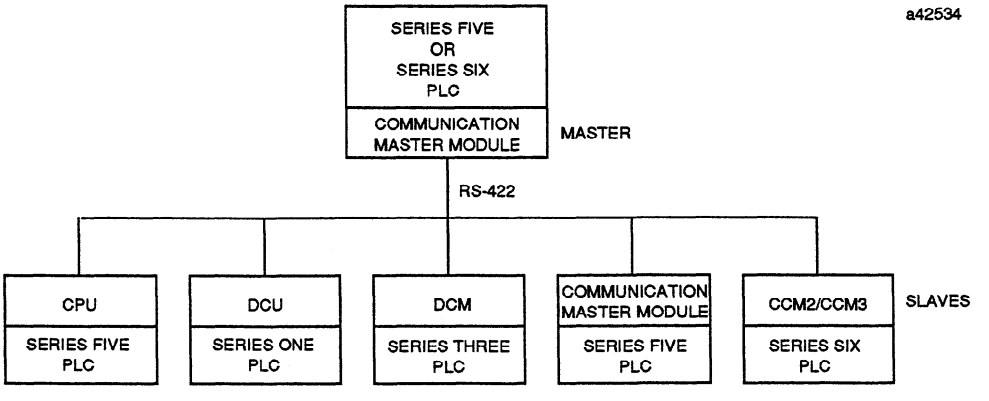


Figure 1-1. Master/Slave Multidrop Serial Data Communication Link

Communications Control Module (CCM) Specifications

The hardware, electrical and specifications for the CCM communications module (IC655CCM500) are defined in this section.

Table 1-1. General Specifications

Dimensions:	41W X 250H X 129D (mm) 1.6W X 9.8H X 5.9D (inches)
Space Requirements:	One Series Five PLC I/O slot -- any I/O slot (0-7) in the CPU base (eight CCMs maximum per Series Five CPU)
Power Requirements:	5V dc ±5%, 1.0A Max. (Supplied by rack power supply)
Ambient Temperature:	0° to 60°C
Storage Temperature:	-20° to +70°C
Humidity:	5% to 95% RH. (non-condensing)
Dielectric Voltage:	Withstand more than 1500 V ac, 1-minute (between Logic Circuit--Communication line--Earth Ground--Circuit Common)
Noise Immunity:	Meets NEMA ICS3-304 and impulse 1000V, 1 msec. pulse

Table 1-2. Transmission Specifications

Linkable Units:	90 Units Max. -- using IC630CCM390 or IC655CCM590, RS-232C/RS-422 converter/repeater
System:	Half-Duplex, 8-bit Asynchronous transmission Parity: Odd, None Serial Data Format: 1 Start bit, 8 Data bits, 1 Parity bit or none, 1 Stop bit. (Except "Additional Protocol Mode 2" which requires even parity)
Standard:	RS-422 and RS-232C
Distance:	1.2Km (4000 feet) Max. -- RS-422 15 meters (50 feet) Max. -- RS-232C
Data Rate:	Selectable 300, 600, 1200, 2400, 4800, 9600 or 19200 bits/sec.
Protocol:	CCM2 or CCM3/RTU
Error Checking:	Parity check, LRC check, CRC-16 (RTU mode)

Interface Compatibility

Memory Mode: Some software packages that interface to the "Series Six" PLC use the "absolute addressing" mode which is available with Series Six CCM2 communications. This mode is not available with the Series Five PLC because the CCM protocol only allows a memory space of up to 64K to be addressed, and the Series Five CPU uses additional memory above 64K. The Series Five PLC retains all the memory types of the original Series Six PLC except absolute addressing (memory type 0).

Memory Addressing: Except for the RTU mode, the Series Five CCM uses "byte" oriented addressing for I/O and override references. This method of addressing is similar to the CCM implementation for the Series One and Series Three PLC, rather than the "point" oriented addressing used in the Series Six PLC. The "start address" in the CCM header to the Series Five PLC is the byte number which includes the desired data. Software packages that use this addressing method to interface the Series Six PLC may need to be re-written for the Series Five PLC. If the software interfaces only with registers, the application will probably be compatible.

Memory Mapping: The mapping of I/O tables within the I/O memory types should be given particular attention. These tables are the I/O 1+, I/O 2+ and I/O in ascending order of the starting address. The override tables for these status tables are also in the same order. Refer to Table 3.1, CCM/CPU Memory Mapping in Chapter 3, and Table 6.3, RTU Table Addressing in Chapter 6.

Scratch Pad Modification: The Series Five PLC contains a great deal of setup and status information in its scratch pad (memory type 6). Before writing to this memory, for any reason, it is **STRONGLY** recommended that you consult GE Fanuc Automation or your nearest field service office.

Target Address: If a Series Six PLC is being used as a master CCM device with the Series Five PLC as a CCM slave, it should be noted that the "target starting addresses" do not need to be on byte boundaries. Each address is a byte address from the Series Five PLC perspective. Also, be sure to note the locations of the I/O tables as mentioned in "Memory Mapping" comments above.

Chapter 2
Installing the Series Five CCM

GFK-0244

This chapter describes the Communications Control Module (CCM) physical layout, module features, configuration, and installation in the Series Five CPU rack.

- CCM Physical Layout
- CCM Internal Functions
- Module Front Panel
- Communication Ports
- Cable Configuration
- Module Installation
- Network Configurations

CCM Physical Layout

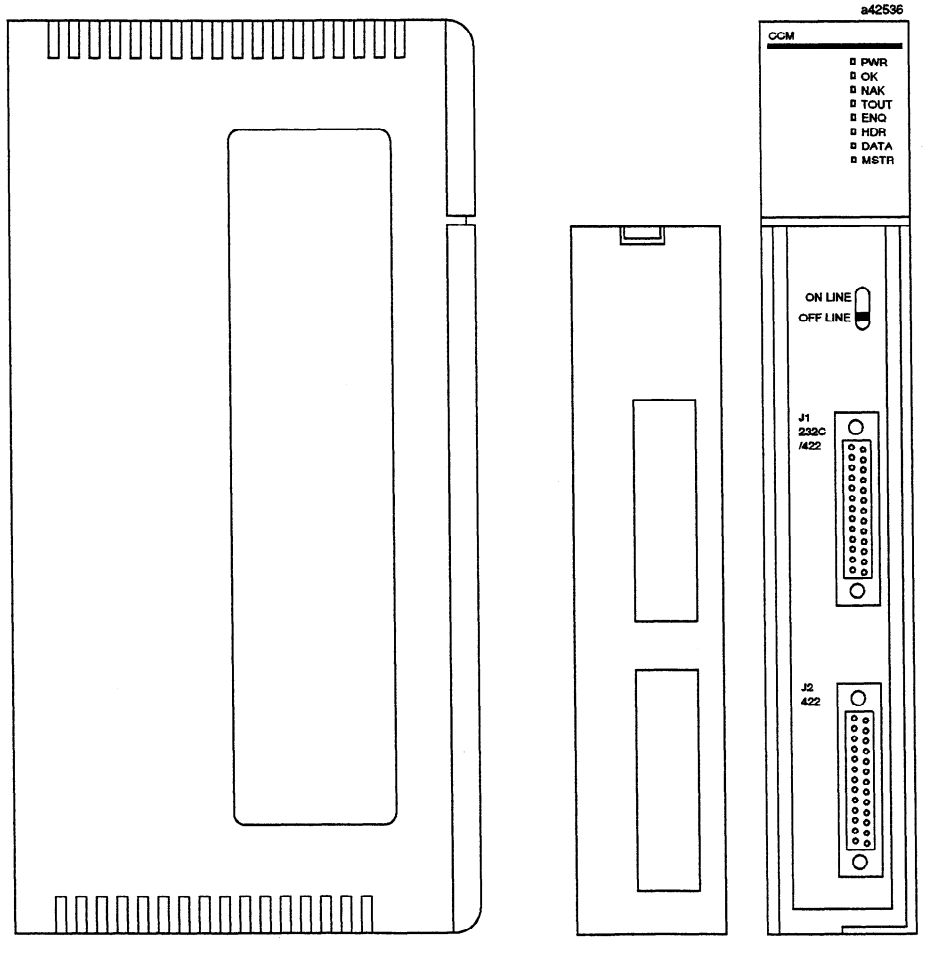


Figure 2-1. Communication Control Module (CCM) Physical Layout

CCM Module Internal Functions

The CCM module contains an internal Printed Wired Board (PWB) which is visible through the cutout on the left side of the module housing. The following user items are accessible through this opening:

- Two Dual-In-Line (DIP) switch packages (SW1, SW2)
- One jumper connection location (F-G, G-G)

Set the DIP switches (SW1 and SW2) and position the Jumper before installing the CCM module into the Series Five PLC rack. Refer to Tables 2-1, 2-2 for the switch settings and Figure 2-2 below for the PWB component location.

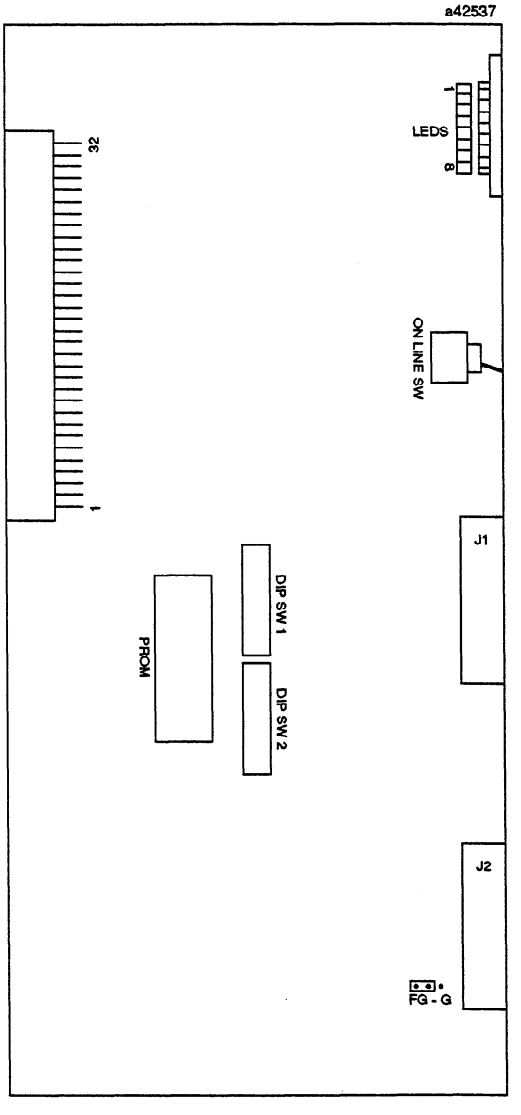


Figure 2-2. Communication Control Module (CCM) Internal PWB

Jumper Connection (Shorting Plug)

By changing the position of the jumper connection (Shorting Plug) located on the PWB, it is possible to connect or disconnect between Earth-Ground (FG) and 0V of the CCM communication circuit. Refer to Figure 2-2 for the Jumper location on the PWB.

- Jumper Position (F-G) Earth Ground is connected to internal 0V. This is the default position.
- Jumper Position (G-G) Earth Ground is NOT connected to internal 0V. Consult GE Fanuc Automation before moving the jumper to this position.

DIP Switch Package

The two switch packages (Sw1, Sw2) located on the PWB are used to set the CCM communication protocol parameters.

Switch Package 1: (SW1) selects the network, station address, and master/slave communication protocol.

Switches 1-7,	Sets the slave station address	(Sw No. 1 = LSB, Sw No. 7 = MSB)
Switch 8,	Switch should be ON	for RTU mode, or Peer mode
	Switch should be OFF	for Master, or Slave station selection.
Switch 9,	Switch should be ON	for RTU, or Master mode
	Switch should be OFF	for Peer, or Slave mode.

Switch Package 2: (SW2) selects the data type, delay time and data transmission rate.

Switches 1-3,	Select the data rate	300 - 19200 bits/sec
Switch 4,	Parity select	ON = odd, OFF= none
Switch 5,	Factory Diagnostic mode	OFF
Switch 6,	Turnaround delay	ON = 10 msec. delay, OFF = no delay
Switch 7 and 8,	Response delay time	0 - 500 msec.
Switch 9	Should be in the OFF position.	

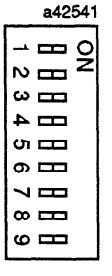
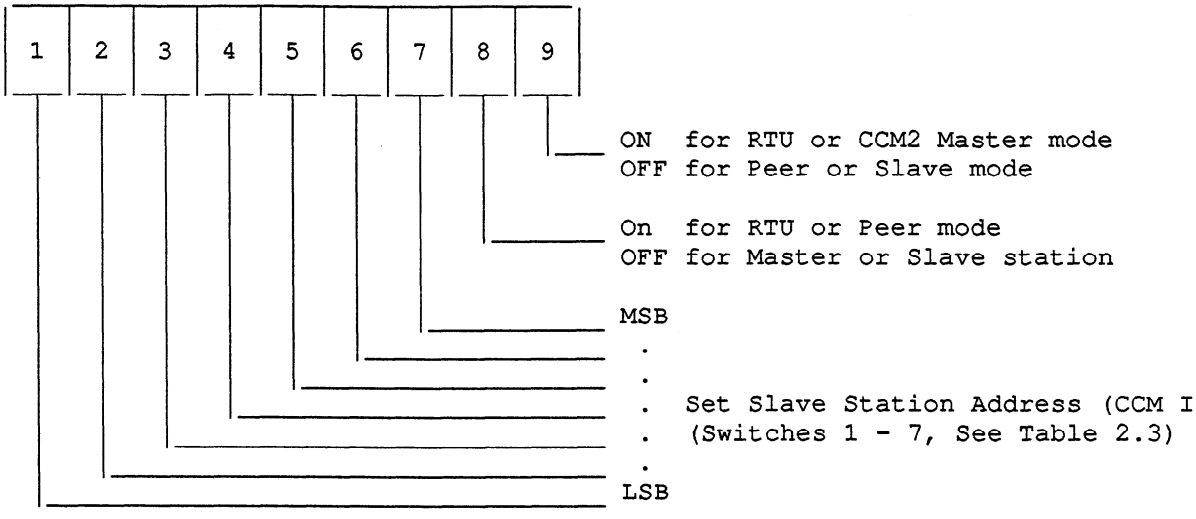


Figure 2-3. CCM, PWB DIP Switch Package

DIP Switch Setting

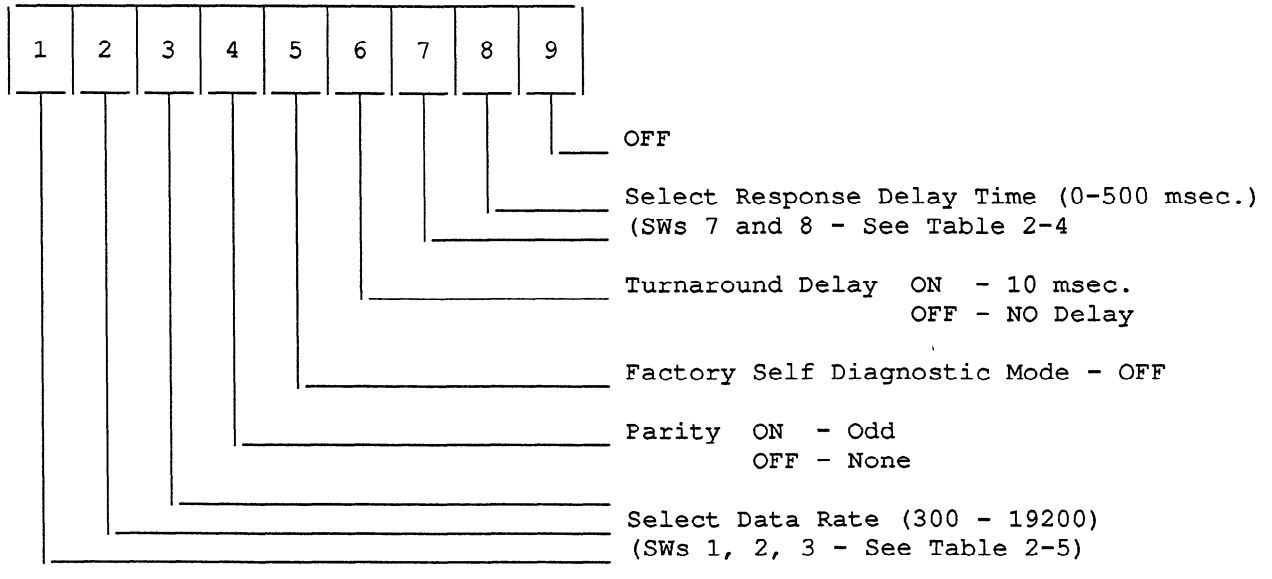
DIP Switch 1: (SW1) selects the network station address, and master/slave communication protocol.

Table 2-1. DIP Switch (SW1) Setting



DIP Switch 2: (SW2) selects the CCM timing, diagnostics, parity and data rate.

Table 2-2. DIP Switch (SW2) Setting



Slave Station Addressing

The CCM slave station addresses are selected by DIP Switch 1. Switches 1-7 select the slave station when positioned ON (X = ON). Switch 9 is always OFF for slave configuration. Table 2-3 shows the slave station designation with the corresponding switch positions. Valid IDs. for CCM2 are 1-90.

Table 2-3. CCM ID. Slave Station Address (DIP SW 1)

Binary Weight			64	32	16	8	4	2	1
			MSB						LSB
	9	8	7	6	5	4	3	2	1
Station 1									X
Station 24					X	X			
Station 90			X		X	X		X	

Response Delay Time

The "Response Delay Time" is the programmable delay time (in addition to the basic delay of the CCM) to avoid communication trouble due to a possible slow response time of the device connected with the CCM.

If the delay time of 500 msec. (only) is selected, the relay contact output (normally used for keying a radio transmitter) will be turned ON for a period of 500 msec. prior to the actual data transmission. Reference Figures 2-4

Table 2-4. Response Delay Time (DIP SW 2)

Delay Time (msec.)	SW. 7	SW. 8
0	OFF	OFF
20	ON	OFF
100	OFF	ON
500	ON	ON

Data Rate Selection

One of seven different communication data rates (300 to 19.2 K bits/sec) may be selected by DIP Switch 2. Table 2-5 below lists the selectable data rates with applicable switch settings. Also, refer to Figure 2-4 for the Data rate transfer timing sequence.

Table 2-5. Data Rate Selection (DIP SW 2)

Data Rate (bits/sec)	SW. 1	SW. 2	SW. 3
300	ON	OFF	OFF
600	OFF	ON	OFF
1200	ON	ON	OFF
2400	OFF	OFF	ON
4800	ON	OFF	ON
9600	OFF	ON	ON
19200	ON	ON	ON

Series Five CCM Timing

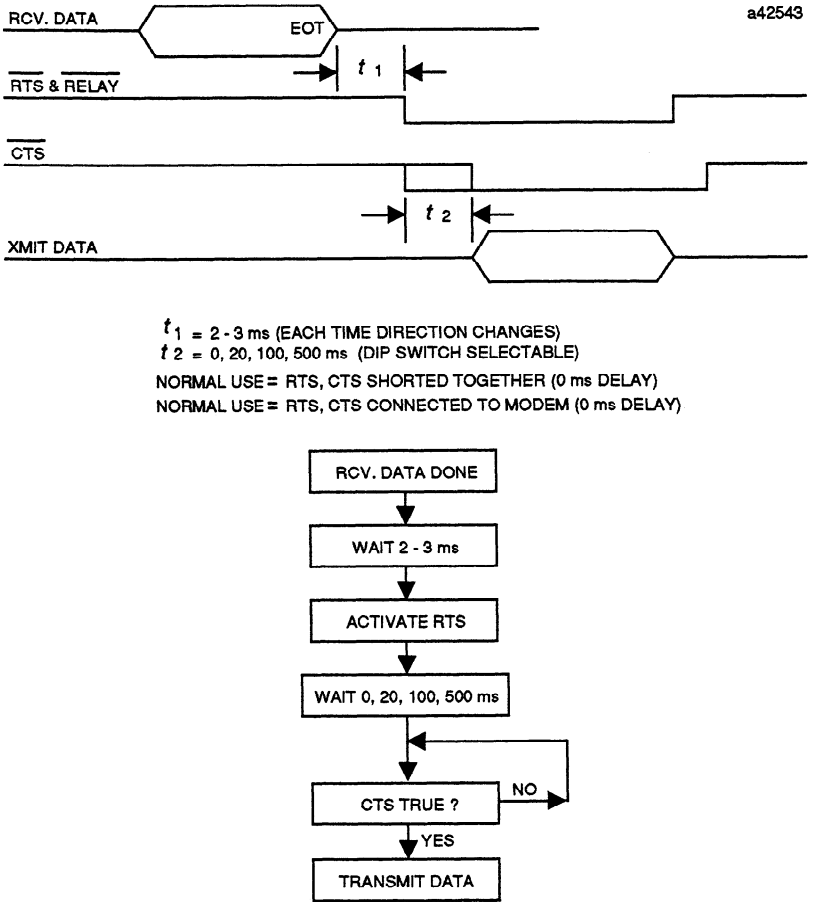


Figure 2-4. Series Five CCM Timing Sequence

CCM Front Panel

The front panel of the CCM module contains the following user items:

- An 8-Segment LED Diagnostic Display
- An ON Line / OFF Line Toggle Switch
- Two Connector Ports (J1) for RS-232C or RS-422), and (J2) for RS-422

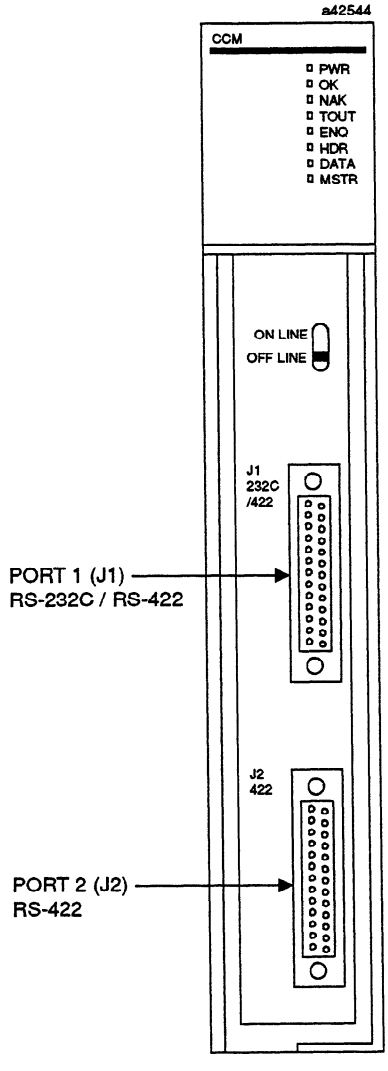


Figure 2-5. Communication Control Module (CCM) Front Panel

LED Description *

The CCM front panel Light-Emitting-Diode (LED) indicators display when the following conditions occur. These LEDs can be used to determine the cause of many problems in the communication network.

* The LED definitions are different for RTU mode. (Refer to Table 6-3)

Table 2-6. LEDs Diagnostic Display (CCM2 Mode)

PWR	ON OFF	This LED is ON when power is properly supplied by the internal DC-DC converter. It is OFF when no power is applied to the unit.
OK	ON OFF	This LED is ON if the module is in normal condition. - ROM is OK - RAM is OK (Read/Write check) - Communications with the CPU is OK It is OFF when any of the above conditions are not true.
NAK	ON OFF	This LED is ON when a NAK is sent back to the device at the other end of the link. Turns OFF when the next ENQ is received. (OFF for valid)
TOUT	ON OFF	This LED is ON when a communication Timeout is detected. Turns OFF when the next ENQ is received.
ENQ**	ON OFF	This LED is ON when an ENQ received with the correct ID. It is OFF when a header starts being sent.
HDR**	ON OFF	This LED comes ON when a header is being sent or received. Otherwise it is OFF.
DATA**	ON OFF	This LED is ON when data is being sent or received. Otherwise it is OFF.
MSTR**	ON OFF	This LED is ON when the module is used as a master. It is OFF when the module is used as a slave.

** During normal operation, these LEDs will blink.

On-Line/Off-Line Selector Switch

The CCM "ON-Line/OFF-Line" Selector Switch is used to logically disconnect the module from the communication link. This is useful when you wish to prevent the CCM/CPU from receiving new data but do not wish to physically remove the connector. Also, when the selector switch is moved to the OFF-Line position, the CCM will send a "NAK" right away instead of waiting for a timeout period.

Switch "UP" position ON-LINE

Switch "DOWN" position OFF-LINE

ON-LINE: The CCM is connected to the data communication bus to the host computer or master station and is able to react to communication instructions from the CPU, or master station.

OFF-LINE: If the CCM is set as a master unit -- the CCM will not react to the communication instructions from the CPU.

If the CCM is set as a slave unit -- the CCM will immediately return a NAK in response to a valid ENQ from the master station.

Communication Ports

The CCM has two physical ports (connectors) located on the front edge of the module. Internally, however, the CCM has only one "logical" port.

The RS-422 port (J2) is located to the bottom of the module, and the RS-232C/RS-422 port (J1) is located at mid-point on the module. Refer to Figure 2-5 for relative port location on the CCM, and Figures 2-6 and 2-7 for configuration of the ports.

- Port J1 is for RS-232C/RS-422 use.
- Port J2 is for RS-422 use.

Each port is isolated from the logic circuits by a photo-coupler and a DC-DC converter.

As shipped, the connector housings for both the RS-232C and RS-422 connectors are connected with Earth Ground (F-G) electrically. F-G is the jumper factory ship position.

NOTE

By changing the shorting plug (Jumper) to G-G, it is possible to disconnect the 0V line to Earth Ground. Consult GE Fanuc Automation before moving the jumper to the G-G position.

For more information concerning Port 1, refer to a later section of this chapter "Using Port 1 (J1) for RS-232C Signals".

RS-232 (J1) Connection

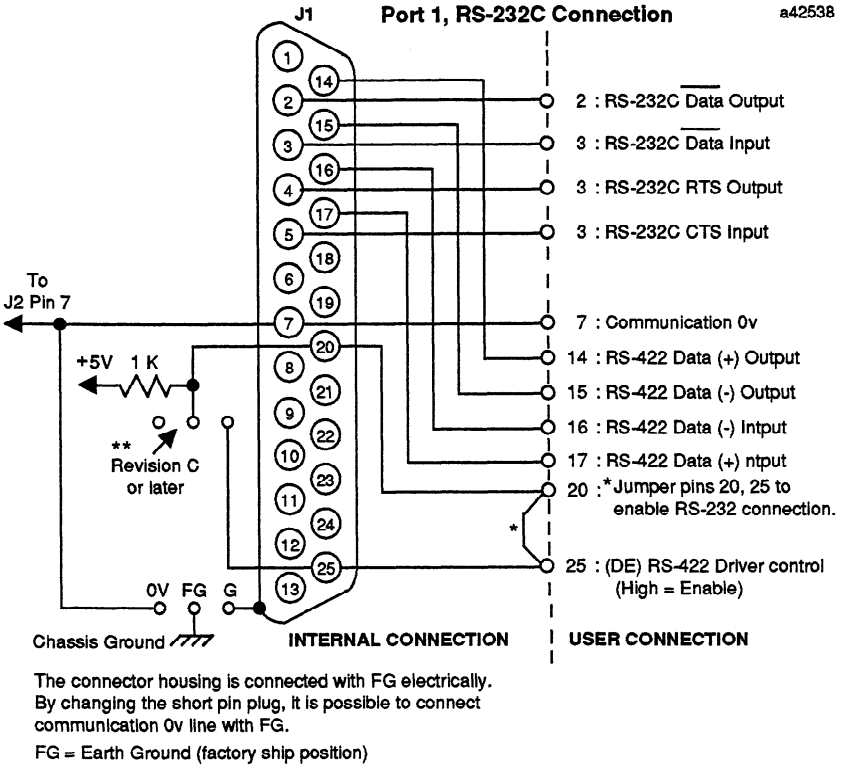


Figure 2-6. CCM, RS-232C Connector Layout

Table 2-7. Port 1 (J1) Pin Definition for RS-232C Connector

Pin No.	Definition	Function
2 3 4 5	RS-232C Data Output RS-232C Data Input RS-232C RTS Output RS-232C CTS Input	
7 14 15	0V RS-422 Data (+) Output RS-422 Data (-) Output	
16 17 20 *	RS-422 Data (-) Input RS-422 Data (+) Input 5V	(5V through 1K ohm pull-up resistor)
25 *	(DE)	Driver Control (High = ENABLED)

* Pins 20 and 25 must be jumpered together for RS-232C use on Revision A or B modules.

** An internal Jumper must be set for Revision C modules.

RS-422 (J2) Connection

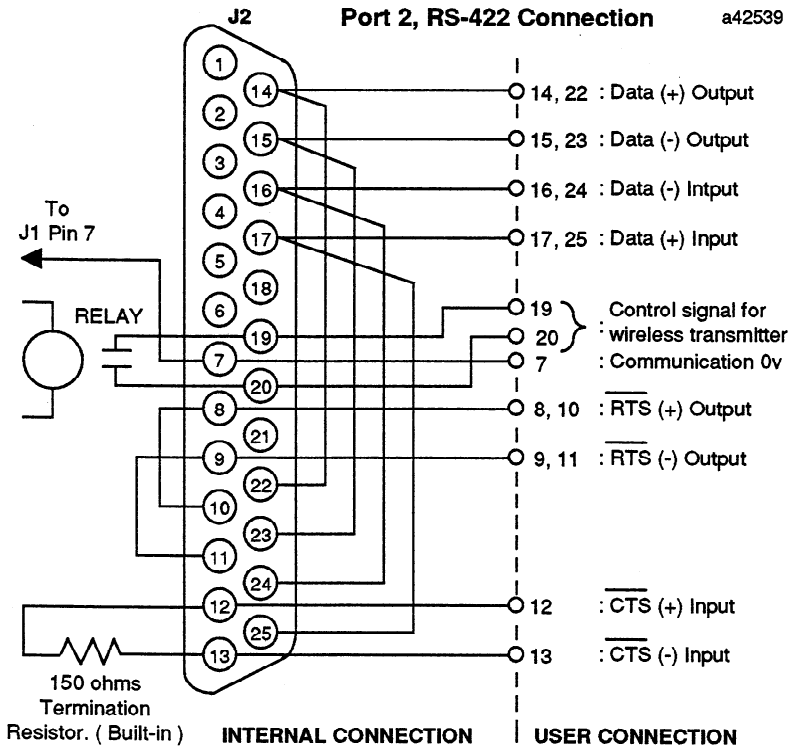


Figure 2-7. CCM, RS-422 Connector Layout

Table 2-8. Port 2 (J2) Pin Definition for RS-422 Connector

Pin No.	Definition	Function
7	0V	150 ohm terminating resistor built-in (between Pins 12 and 13)
8, 10	$\overline{\text{RTS}}$ (+) Output	
9, 11	$\overline{\text{RTS}}$ (-) Output	
12	$\overline{\text{CTS}}$ (+) Input	
13	$\overline{\text{CTS}}$ (-) Input	
14, 22	Data (+) Output	
15, 23	Data (-) Output	
16, 24	Data (-) Input	
17, 25	Data (+) Input	
19 *	TXD Control Signal	(Control signal for wireless transmission communications 0V)
20 *	TDX Control Signal	

*An internal Jumper connects Pins 19, 20 for Control Signal.

Using Port 1 (J1) for RS-232C Signals

To use Port 1 for RS-232C signals, Pins 20 and 25 must be connected together. Refer to the following figures.

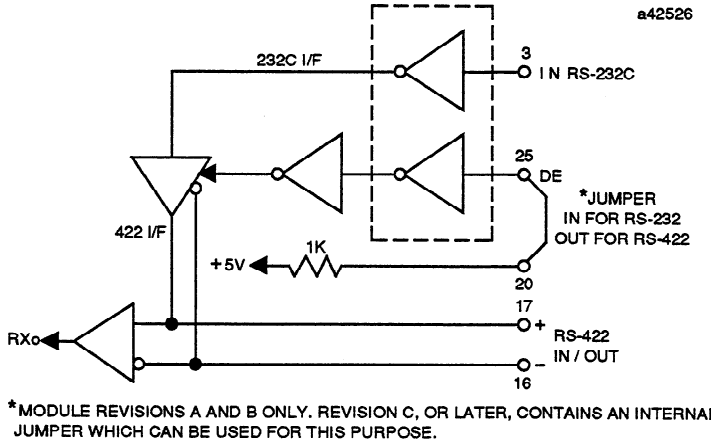


Figure 2-8. Port 1, RS-232C, Example Wiring Connection

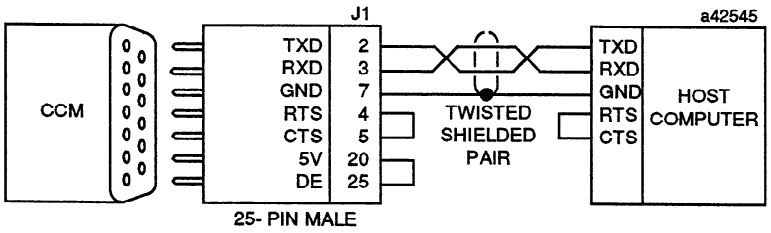


Figure 2-9. Port 1, Connection to Host Computer

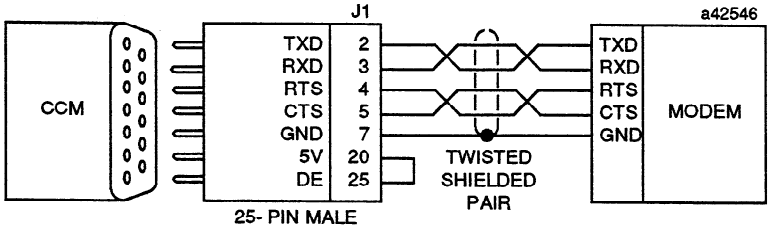


Figure 2-10. Port 1, Connection to Modem

Cable Configuration

The communication cable assembly presents one of the most common causes of communication failure. For best performance construct the RS-232C and RS-422 cable assembly according to the recommended connector parts and cable specifications. Refer to the appropriate wire table (Port 1, or Port 2) for the specific port wiring information.

Connector Specification

The recommended mating connector for Port 1 (RS-232C) and Port 2 (RS-422) is the GE Fanuc Automation (GE Fanuc IC655ACC525) 25-pin, solder pot, D-Subminiature connector.

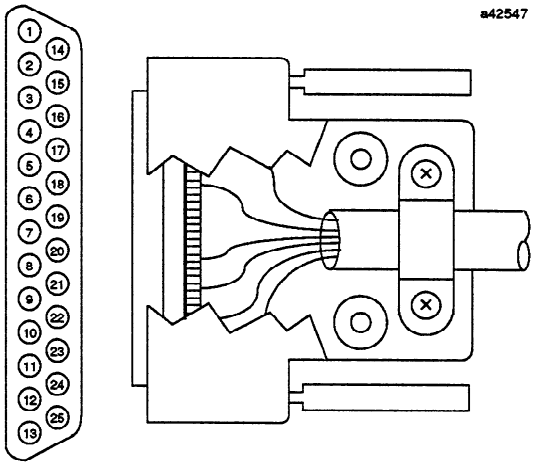


Figure 2-11. Connector Configuration

Cable Specifications

Recommended cable conductor types and fabrication instructions are as follows:

Table 2-9. Cable Specifications

Length, Maximum:	50 feet (15 meters) for RS-232C 4000 feet (1.2 Km) for RS-422
24 AWG Minimum	(Overall Shield Recommended) For RS-422, use a low capacitance, high quality, twisted pair wire. For example: Belden 9184, 9729, 9804 or 9842 Twisted pairs must be matched so that both transmit signals make up one twisted pair and both receive signals make up the other twisted pair. If this is ignored, then cross-talk will result from the mis-matching which will affect the performance of the communication system. For RS-232, use a high quality "computer" grade cable. Twisted pairs should not be used.

Cable Wiring

The pin connections are different for the RS-232C and RS-422 connections. Refer to the following figures and wire tables for the specific port (J1, J2) wiring information.

Port 1,	RS-232C (J1)	Table 2-7	Figure 2-6
Port 2,	RS-422 (J2)	Table 2-8	Figure 2-7

GFK-0244

Installing the CCM

The CCM plugs directly into the Series Five PLC logic rack with the CPU. The backplane mating connector is a 96 pin male (3 rows, 32 pin/row) connector located uppermost on the back side of the module.

Series Five CCM Installation

Carefully position the CCM module into the Series Five CPU rack. Any of the I/O slots in the CPU rack may be used.

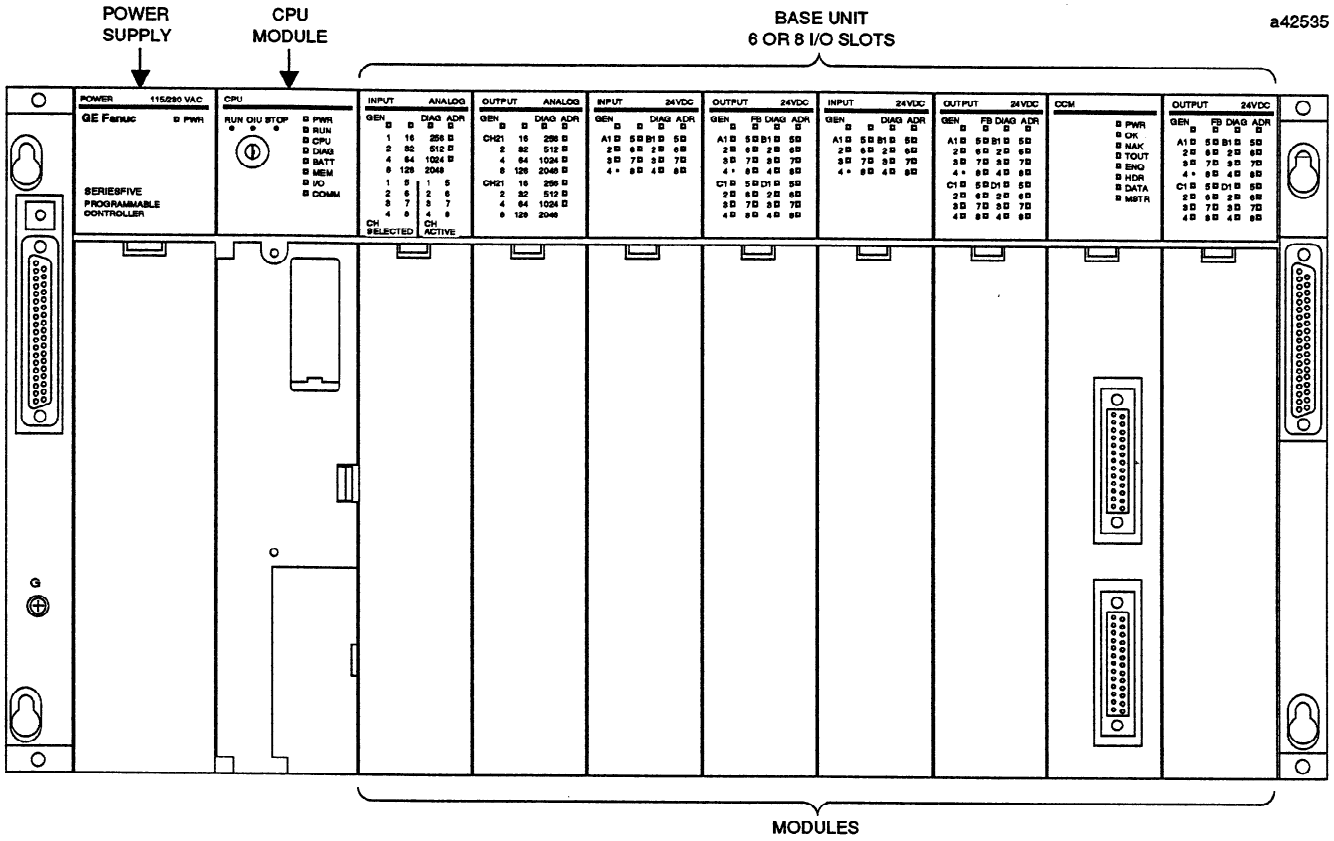


Figure 2-12. Series Five PLC Rack

Communication Network Configurations

The following pages contain several examples of the many possible communication network configurations using the CCM.

- Multidrop Network
- Using the Internal RS-232C to RS-422 Converter (CCM as Slave)
- CCM as Master Station within a Network
- Multiple CCM Buses (CCM as Master, CCM as Slave) within the same CPU rack
- Complex Networks (Multiple Master and Slave Stations)
- Illegal Networks

NOTE

In the figures which follow, the Series Five CPUs built in slave CCM port can be used in some cases instead of the CCM master module as shown.

Multidrop Network

In the case of a Multidrop network, all of the stations except one are configured as slaves. The remaining station is configured as a master. A typical network configuration and wiring diagram for the RS-422 multidrop connection (using the IC630CCM390 or IC655CCM590 converter) is shown below.

The converter is required for this application only to provide isolation between the master host computer and the first CCM slave station. If this isolation is not required, the next example may be used, when no additional RS-232 to RS-422 converter is required.

NOTE

In general, all the CCMs on a multidrop link must be referenced to the same ground point. If this cannot be guaranteed, the IC630CCM390 or IC655CCM590 converter/repeater box may be required to provide ground isolation between units.

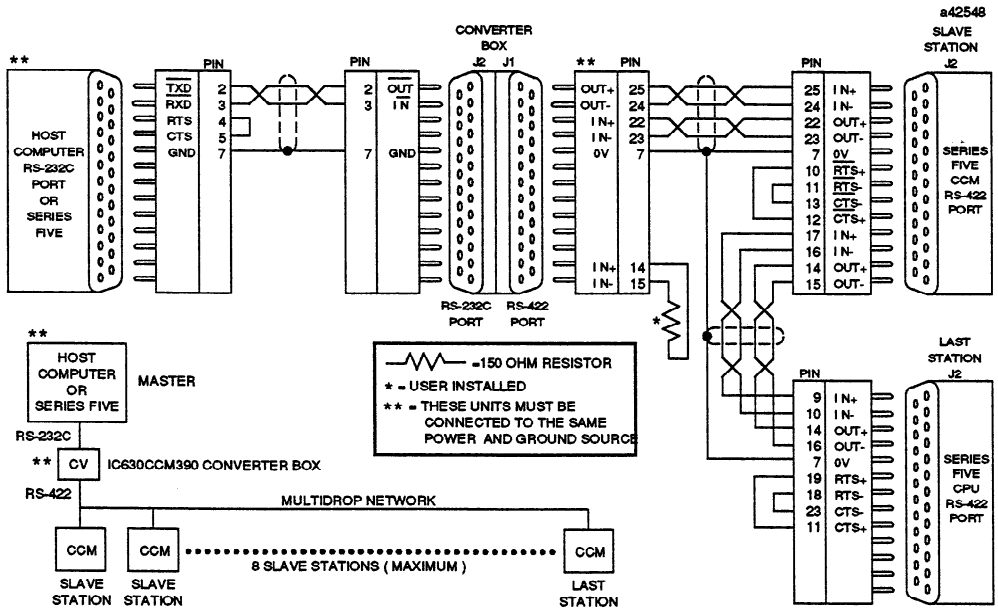


Figure 2-13. RS-422 Network Using the IC630CCM390 or IC655CCM590 Converter Box

RS-232C to RS-422 Converter (CCM as Slave)

The internal circuits of the CCM can also be used as a non-isolating RS-232C to RS-422 converter. In this way, a computer which has only a RS-232C output can drive a RS-422 multidrop network without the need for a converter box.

The host (master) computer should be within 50 feet of the slave which is doing the conversion. Also the computer and this slave need to be connected to the same power source and ground. If these conditions cannot be guaranteed, the previous configuration using the IC630CCM390 or IC655CCM590 converter box should be used.

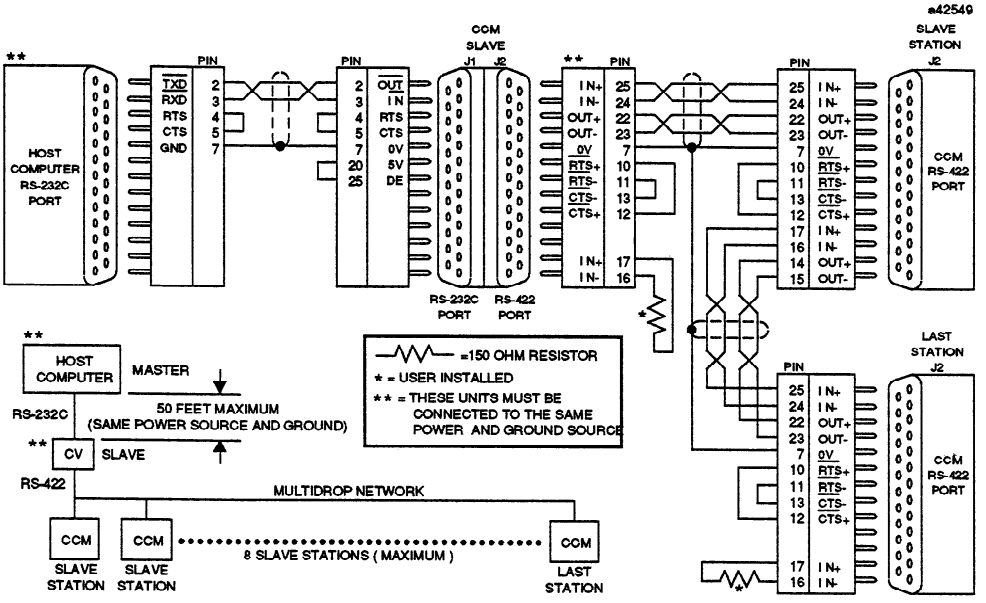


Figure 2-14. Network Using the CCM as a RS-232C to RS-422 Converter

CCM Connected as Master Station

This example shows the CCM as the master station in a RS-422 configuration.

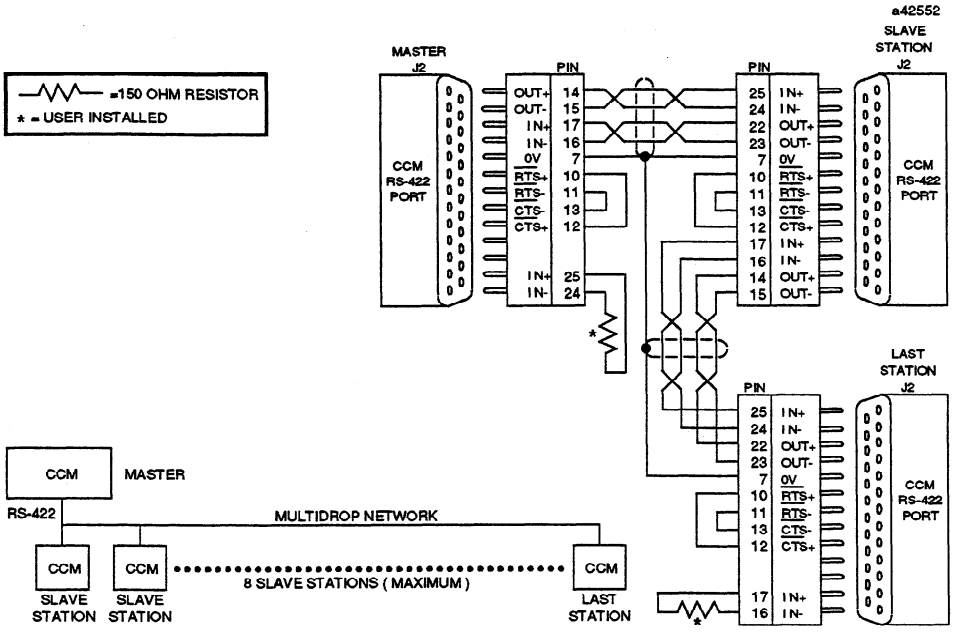


Figure 2-15. CCM as the Master Station

NOTE

In the figures which follow, the Series Five CPUs built in slave CCM port can be used in some cases instead of the CCM master module as shown.

CCM Connected to Other PLCs

When the multidrop network is used (as in the following example where other PLC types are included), all of the stations except one are configured as slaves. The remaining station is configured as a master. A typical network configuration and wiring diagram showing the Series 90-30 PLC connected is shown below.

NOTE

Series One Junior PLCs cannot be used as slaves when using the Series Five CCM master module as the master device.

All interface modules on a multidrop link must be referenced to the same ground point. If this cannot be guaranteed, the IC630CCM390 or IC655CCM590 converter/repeater box may be required to provide ground isolation between units.

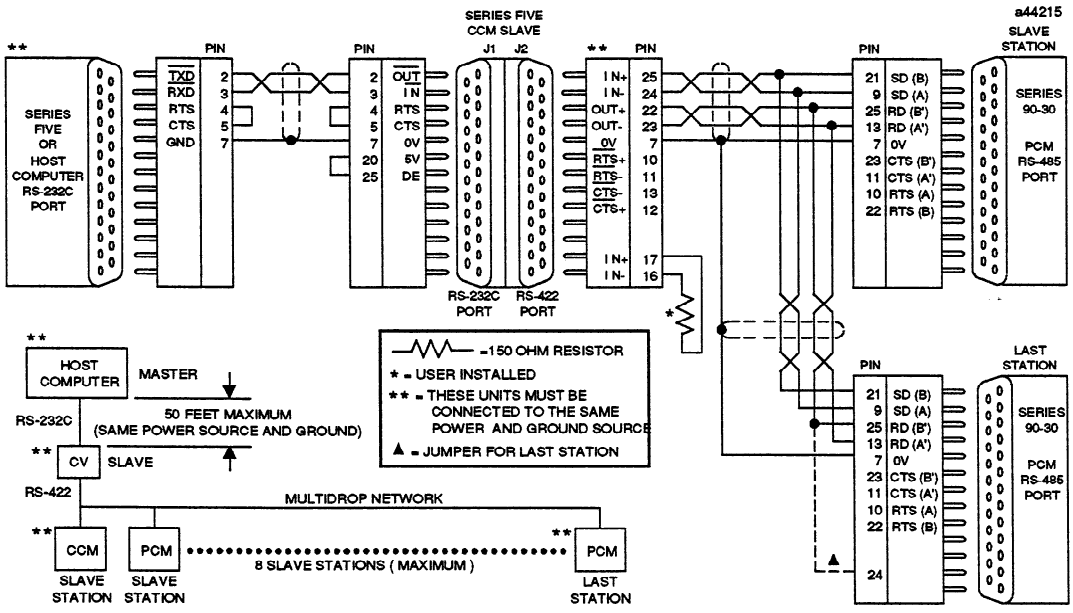


Figure 2-16. CCM Connected to the Series 90-30 PLC

You will notice in the figure above that the Series 90-30 Programmable Coprocessor Module (PCM) connections vary from the Series Five CCM as follows:

- SD (Send Data) and RD (Receive Data) are the same as TXD and RXD (used in the Series Five® CCM).
- (A) and (B) are the same as - and + (A and B denote outputs, and A' and B' denote inputs)
- Jumper pins 24 and 25 to terminate the Series 90-30 PCM RS-485 the RD input signal.

Multiple CCM Buses (CCM as Master, CCM as Slave)

Up to eight CCMs can reside in a single CPU rack. Therefore, up to eight independent CCM networks can originate or pass through a single CPU rack.

The example below shows two CCM modules in the same base. CCM 1 is treated as a slave of the host computer (master of that network) as well as CCM 2 and CCM 3. The CCMs also perform a RS-232C to RS-422 conversion. Refer to previous pages for interconnect wiring.

The second CCM network consists of CCM master #2 with slaves 3, 4, and 5. Since the networks are totally independent, CCM target addresses may be duplicated between networks with no difficulties, and the different networks communicate only through the CPUs user logic.

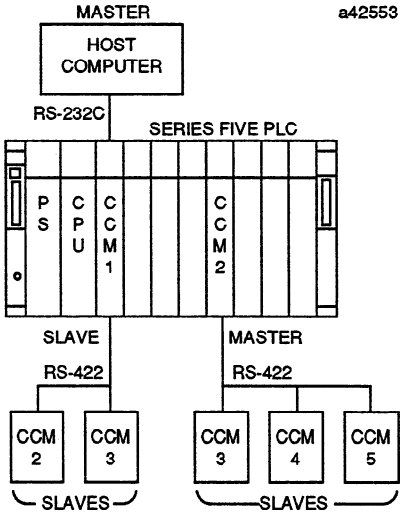


Figure 2-17. RS-232C to RS-422 Converter

Complex Networks

Since the Series Five PLC can have multiple CCM ports (one in the CPU and up to eight additional CCMs), it is possible for several external devices to write data to the CPU. When writing data to the CPU, the user must ensure that undesirable system interactions do not occur as a result of more than one external CCM device writing to the same CPU. Reading the CPU by multiple devices does not create any problems, but writing to the CPU by more than one device is not recommended.

A description of the (Set/Reset and Data Monitoring) functions that allow external devices to write data through a CCM request to the CPU's scratch pad is explained in Chapter 3. They are used by Logicmaster 5 during normal operation. If Logicmaster 5 is not in use, a single external device may use these functions.

CAUTION

Extreme care must be exercised by the user to ensure that one and only one device uses the Set/Reset and Data Monitoring functions. If Logicmaster 5 is operating, no other devices may use these functions. Also it should be noted that Logicmaster 5, when on-line, has the ability to write to most of the tables in the CPU.

The following examples of complex networks show a number of master and slave stations. (M = Master, S = Slave) Within one base unit, multiple CCMs are used as masters in different CCM networks.

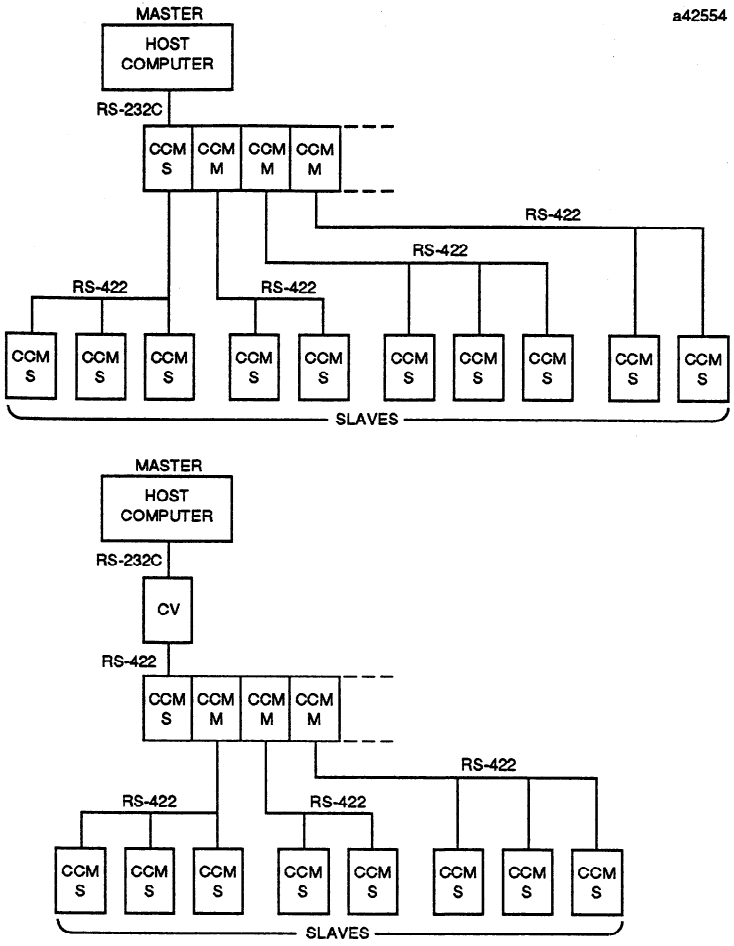


Figure 2-18. Complex Networks (Multiple Masters)

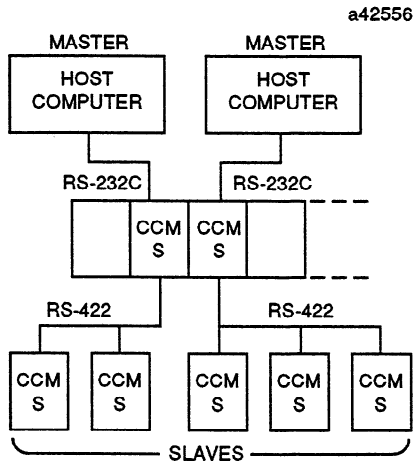


Figure 2-19. Complex Networks (Multiple Hosts)

Non-Practical Networks

Some examples of non-practical CCM networks are shown below. The first example (Figure 2-shows a network where two or more master stations exist within the network. This becomes a "multi-master" network which cannot be done with CCM compatible devices. The second example (Figure 2-20) is incorrectly wired as a master station and set as a slave.

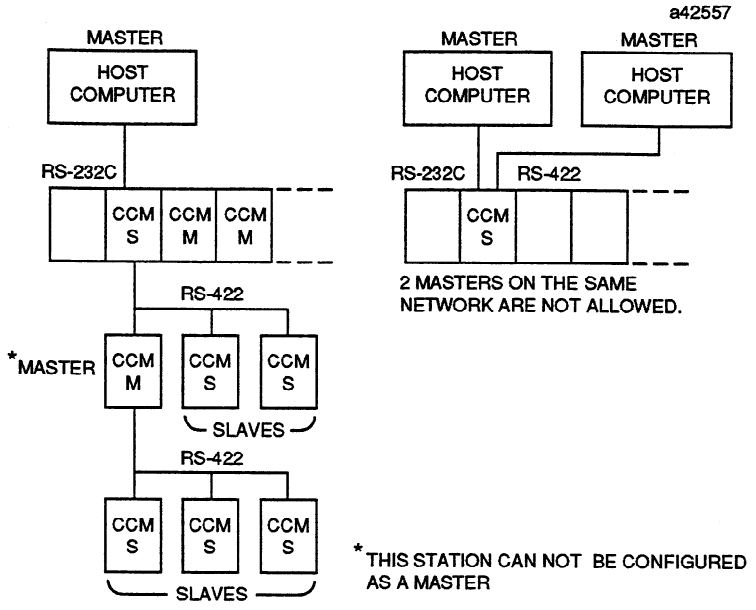


Figure 2-20. Illegal Network (Multi - Master Stations)

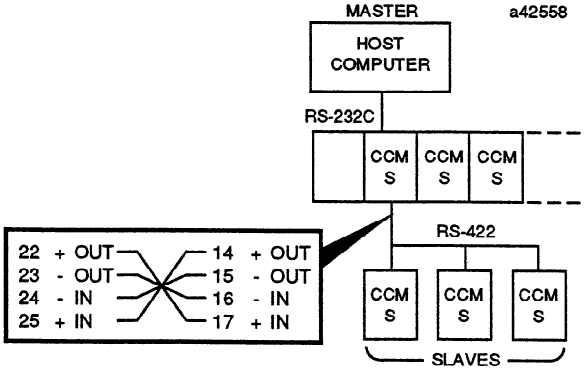


Figure 2-21. Wired as a Master and Set as a Slave

If the network wiring is as shown in figure above, where both stations are set as slaves, the host computer is not able to access the network beyond this wiring point.

Chapter 3 Memory Map

When a reference is programmed and transferred to the CPU, it is entered into a special storage area in the CPU called a table. Each reference group has its own table to keep track of the references used that correspond to the table. The table maintains the status of each reference, keeping track of whether its corresponding bit or group of bits is turned ON or OFF.

Series Five CCM/CPU Mapping

The Series Five PLC has several areas of memory which provide either special functions, system status information, or error reporting information. These are the special purpose coils, dedicated registers, and the CPU's scratch pad memory. Definitions for these items in memory are provided in the tables on the following pages.

The Communications Control Module (CCM) protocol must be given the start address (shown as reference number below) and a length. The start address plus the length should not go past the end of a table boundary. The CCM for the Series Five interprets length as BYTES, however, WORD lengths are used for the Series Five "WRITE CCM" and "READ CCM" instructions.

Table 3-1. Series Five CCM/CPU Mapping

Table Name	Reference	CCM Target Address Decimal	CCM Target Address Hexadecimal	CCM Table Memory Type	CCM Ovrd Type	Data Format
Registers *	R00001 - R16384	1-16384	0001-4000H	1	N/A	2 Bytes/Register
I1+ Inputs	I1+0001 - I1+1024	1-128	0001-0080H	2 †	4 †	8 Inputs/Byte
I2+ Inputs	I2+0001 - I2+1024	129-256	0081-0100H	2	4	8 Inputs/Byte
Local Inputs	I0001 - I1024	257-384	0101-0180H	2	4	8 Inputs/Byte
Spec. Inputs *	I1-0001 - I1-0512	385-448	0181-01C0H	2	N/A	8 Inputs/Byte
O1+ Outputs	O1+0001 - O1+1024	1-128	0001-0080H	3 †	5 †	8 Outputs/Byte
O2+ Outputs	O2+0001 - O2+1024	129-256	0081-0100H	3	5	8 Outputs/Byte
Local Outputs	O0001 - O1024	257-384	0101-0180H	3	5	8 Outputs/Byte
Internal Coils	O1-0001 - O1-1024	385-512	0181-0200H	3	5	8 Outputs/Byte
Internal Coils	O2-0001 - O2-1024	513-640	0201-0280H	3	5	8 outputs/Byte
Scratch Pad **	0000 - 0900H	0-2304	0000-0900H	6	N/A	1 Byte/Byte
User Logic	0000 - 16383	0-16383	0000-3FFFH	7	N/A	2 Bytes/Word
CCM Diagnostic *	0000 - 0009	0-9	0000-0009H	9	N/A	1 Byte/Byte

* Refer to the following pages for detailed information concerning the use of these tables.

** Do NOT write to the Scratch Pad without first consulting GE Fanuc Automation.

† Refer to Appendix A for an expanded listing of CCM Memory Types 2 and 4, 3 and 5.

Important operational NOTE on the next page.

NOTE

Even though memory type zero is not inhibited from use with CCM it is not intended to be used. Any Series Six PLC applications which use memory type zero will need to be re-written for the Series Five PLC to use the other memory types.

Register Definition

Registers R4000 - R4096 are generally used by the Series Five CPU and are not available for general use. Refer to GFK-0122 *Series Five PLC User's Manual* for a detailed explanation of these registers. Also, registers 3850 through 3999 (when using Genius I/O default values) are reserved by the system and the data to be stored in them has special meaning as listed below.

Some of the registers contain valuable information pertaining to various system errors. For example, when certain system errors or conditions occur, information relative to the error or condition is stored in specified registers, and is available for user intervention. These registers should not be used as general purpose registers for data storage or data manipulation.

Table 3-2. Reserved Register Definition

Register Ref.	Definition - Contents of Register	Data Format
R3850 - R3999	Default Genius diagnostics faults (10 per fault)	
R4000	OIU Memory Cartridge Data Transfer register	Binary
R4001	OIU Printer register	Binary
R4002	OIU Printer Port Setup Register	Binary
R4003	OIU Display Register Printer	
R4040	Reserved for future use	
R4041	Genius bus scan time (slot 0)	Binary
R4042	Genius bus scan time (slot 1)	Binary
R4043	Genius bus scan time (slot 2)	Binary
R4044	Genius bus scan time (slot 3)	Binary
R4045	Genius bus scan time (slot 4)	Binary
R4046	Genius bus scan time (slot 5)	Binary
R4047	Genius bus scan time (slot 6)	Binary
R4048	Genius bus scan time (slot 7)	Binary
R4049	Genius diag - starting register for fault table (DEF=R3850)	Binary
R4050	Genius diag - number of faults to be registered (DEF=max=15)	Binary
R4051	Genius diagnostics -- number of actual faults	Binary
R4052 - R4056	Reserved for future use	Binary
R4057	GENI Slot 0 SBA conflict address/CCM RCV Buffer Pointer	Binary
R4058	GENI Slot 1 SBA conflict address/ :	Binary
R4059	GENI Slot 2 SBA conflict address/ :	Binary
R4060	GENI Slot 3 SBA conflict address/ :	Binary
R4061	GENI Slot 4 SBA conflict address/ :	Binary
R4062	GENI Slot 5 SBA conflict address/ :	Binary
R4063	GENI Slot 6 SBA conflict address/ :	Binary
R4064	GENI Slot 7 SBA conflict address/ :	Binary
R4065	OIU Start address of timer register area	Binary
R4066	OIU Number of timers	Binary
R4067	OIU Start address of counter register area	Binary
R4068	OIU Number of counters	Binary

Table 3-2. Reserved Register Definition - Continued

Register Ref.	Definition - Contents of Register	Data Format
R4069	OIU Message pointer for ASCII display	Binary
R4070	OIU Key code buffer for operator key entry	Binary
R4071	OIU Message pointer for ASCII display (upper line)	Binary
R4072	OIU Mode register	Binary
R4073	Address of math error	
R4074	Not used	n/a
R4075	Current ID of module - I/O CONFIG ERROR	Binary
R4076	Previous ID of module - I/O CONFIG ERROR	Binary
R4077	Rack and Slot numbers - I/O CONFIG ERROR	BCD
R4078	Critical system error; (e.g., no memory cartridge) CPU stops	BCD
R4079	Somewhat critical system error; (e.g., blown fuse) CPU continues running.	BCD
R4080	System error - not serious; (e.g., battery voltage low) CPU continues running.	BCD
R4081	Error Code - MODULE ERROR	BCD
R4082	Circuit number - MODULE ERROR	BCD
R4083	Rack and Slot number - MODULE ERROR	BCD
R4084 - R4085	Reserved for future use	n/a
R4086	Scan counter	Binary
R4087	Seconds - calendar/clock	BCD
R4088	Minutes - calendar/clock	BCD
R4089	Hour - calendar/clock	BCD
R4090	Day of Week - calendar/clock (2)	BCD
R4091	Day - calendar/clock	BCD
R4092	Month - calendar/clock	BCD
R4093	Year - calendar/clock	BCD
R4094	Last Scan Time (in milliseconds)	Binary
R4095	Minimum Scan Time (in milliseconds)	Binary
R4096	Maximum Scan Time (in milliseconds)	Binary

Special Inputs Definition

A group of bits in the internal status table are reserved for use by the Series Five PLC system. These internal inputs are references I1-0001 through I1-0464. When "1" appears in the Table 3-3 (under definition) it indicates a logic 1 (coil ON).

These special Input bits (running consecutively) are designated to be used as follows:

- I1-0001 to I1-0066 System Operation/Status
- I1-0081 to I1-0208 Smart Module Communication Status
- I1-0209 to I1-0464 Genius Communications Status

Inputs not shown in the following table are reserved for internal use by the Series Five CPU only. Refer to GFK-0122, *Series Five PLC User's Manual* for a detailed explanation of these inputs.

System Operation/Status

I1-0001 to I1-0066 are special internal input contacts for system operation.

Table 3-3. Special Internal Inputs Definition (Run Time)

Reference	Purpose	Definition
I1-0001	Initial Reset	OFF in stop mode and ON for first scan after going to RUN; OFF thereafter.
I1-0002	Always ON	Used as an "always ON" conditional contact.
If the minimum pulse width is longer than the scan time -- the following clocks are no longer accurate.		
I1-0004	1 Minute Clock	Provides a pulse 30 seconds OFF, 30 seconds ON.
I1-0005	1 Second Clock	Provides a pulse .5 Seconds OFF, .5 seconds ON.
I1-0006	100 msec. Clock	Provides a pulse 50 msec. OFF, 50 msec. ON.
I1-0007	50 msec. Clock	Provides a pulse 24 msec. OFF, 26 msec. ON.
I1-0008	Scan time Clock	Provides a pulse ON for 1 scan, OFF for 1 scan.
I1-0009	Not used	- Not available for program use -
I1-0010	Forced RUN	1 = running, CPU keyswitch in the RUN position.
I1-0011	OIU RUN	1 = running, CPU keyswitch in the OIU position.
I1-0015	OIU STOP	1 = CPU stopped, keyswitch in the OIU position.
I1-0016	Stop Relay	CPU stopped by keyswitch in STOP position or by external device (switch in RUN position).
I1-0020	Suspend I/O	1 = I/O is suspended
I1-0033	Critical System Error	1 = error, CPU goes to STOP mode.
I1-0034	Non-Critical System Error	1 = error, CPU remains in RUN mode.
I1-0035	Diagnostic Error	1 = error detected
I1-0036	Battery Not Normal	1 = CPU or memory cartridge battery voltage low.
I1-0037	Memory Error	1 = latches if a memory cartridge error occurs.
I1-0038	I/O Error	1 = latches if I/O bus error is detected.
I1-0039	Communications Error	1 = Turned ON by a CCM error. next successful communications will turn it OFF.
I1-0040	I/O Configuration Error	1 = error detected, I/O configuration has changed since last power-up.
I1-0042	Watchdog Timeout	1 = Watchdog timer has timed out.
I1-0043	Internal Program Error	1 = Error (this should never occur.
I1-0044	Internal Math Error	1 = Error
I1-0045	Smart Module Comm. Error	1 = Error

Table 3-3. Special Internal Inputs Definition (Run Time) - Continued

Reference	Purpose	Definition
I1-0065	I/O Retry Parity Status	1 = Parity error after specified number of retries 0 = OK, no parity error detected
I1-0066	Non-Critical Rack	1 = Non-critical rack, 0 = Critical rack (that reported parity error setting I1-0065)
I1-0077	RD CCM	1 = RD CCM thru CPU is busy; 0 = not busy
I1-0078	RD CCM	1 = RD CCM thru CPU error (see below); 0 =no error
I1-0079	WR CCM	1 = WR CCM thru CPU is busy; 0 = not busy
I1-0080	WR CCM	1 = WR CCM thru CPU error (see below); 0 =no error

Possible causes of the Read or Write CCM thru CPU error, listed in the table above, include the following:

1. The port is already in use by OIU.
2. The data length of the Read or Write command is less than 1 or greater than 128 bytes.
3. Trap mode, which is only legal for the Read CCM function, is specified for the Write CCM function.
4. Framing, parity or overrun error detected during read.
5. An illegal register is specified as the operand.

Smart Module Communication Status

I1-0081 to I1-0208 are special purpose contacts for the smart module communication status.

There are two special purpose contact references for each slot in each rack, as shown in the table. Each reference is a status bit for smart module communications status, if a smart module is installed in that slot. Each reference has two possible conditions, 1 (ON) and 0 (OFF).

Status definition for the first reference (e.g., I1-81) is: 0 = not executing, 1 = executing.
Status definition for the second reference (e.g., I1-82) is: 0 = no error, 1 = error.

Table 3-4. Special Internal Inputs Definition (Smart Module)

Slot	Rack 0	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5	Rack 6	Rack 7
0	I1-81 I1-82	I1-97 I1-98	I1-113 I1-114	I1-129 I1-130	I1-145 I1-146	I1-161 I1-162	I1-177 I1-178	I1-193 I1-194
1	I1-83 I1-84	I1-99 I1-100	I1-115 I1-116	I1-131 I1-132	I1-147 I1-148	I1-163 I1-164	I1-179 I1-180	I1-195 I1-196
2	I1-85 I1-86	I1-101 I1-102	I1-117 I1-118	I1-133 I1-134	I1-149 I1-150	I1-165 I1-166	I1-181 I1-182	I1-197 I1-198
3	I1-87 I1-88	I1-103 I1-104	I1-119 I1-120	I1-135 I1-136	I1-151 I1-152	I1-167 I1-168	I1-183 I1-184	I1-199 I1-200
4	I1-89 I1-90	I1-105 I1-106	I1-121 I1-122	I1-137 I1-138	I1-153 I1-154	I1-169 I1-170	I1-185 I1-186	I1-201 I1-202

Table 3-4. Special Internal Inputs Definition (Smart Module) - Continued

Slot	Rack 0	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5	Rack 6	Rack 7
5	I1-91 I1-92	I1-107 I1-108	I1-123 I1-124	I1-139 I1-140	I1-155 I1-156	I1-171 I1-172	I1-187 I1-188	I1-203 I1-204
6	I1-93 I1-94	I1-109 I1-110	I1-125 I1-126	I1-141 I1-142	I1-157 I1-158	I1-173 I1-174	I1-189 I1-190	I1-205 I1-206
7	I1-95 I1-96	I1-111 I1-112	I1-127 I1-128	I1-143 I1-144	I1-159 I1-160	I1-175 I1-176	I1-191 I1-192	I1-207 I1-208

Genius Communications Status

Input bits I1-0209 to I1-0464, running consecutively, are designated to show the status of Genius communications.

(0 = Not Communicating, 1 = Communicating)

Table 3-5. Special Internal Inputs Definition (Genius)

Serial Bus Adr	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
0	I1-0209	I1-0241	I1-0273	I1-0305	I1-0337	I1-0369	I1-0401	I1-0433
1	I1-0210	I1-0242	I1-0274	I1-0306	I1-0338	I1-0370	I1-0402	I1-0434
2	I1-0211	I1-0243	I1-0275	I1-0307	I1-0339	I1-0371	I1-0403	I1-0435
3	I1-0212	I1-0244	I1-0276	I1-0308	I1-0340	I1-0372	I1-0404	I1-0436
4	I1-0213	I1-0245	I1-0277	I1-0309	I1-0341	I1-0373	I1-0405	I1-0437
5	I1-0214	I1-0246	I1-0278	I1-0310	I1-0342	I1-0374	I1-0406	I1-0438
6	I1-0215	I1-0247	I1-0279	I1-0311	I1-0343	I1-0375	I1-0407	I1-0439
7	I1-0216	I1-0248	I1-0280	I1-0312	I1-0344	I1-0376	I1-0408	I1-0440
8	I1-0217	I1-0249	I1-0281	I1-0313	I1-0345	I1-0377	I1-0409	I1-0441
9	I1-0218	I1-0250	I1-0282	I1-0314	I1-0346	I1-0378	I1-0410	I1-0442
10	I1-0219	I1-0251	I1-0283	I1-0315	I1-0347	I1-0379	I1-0411	I1-0443
11	I1-0220	I1-0252	I1-0284	I1-0316	I1-0348	I1-0380	I1-0412	I1-0444
12	I1-0221	I1-0253	I1-0285	I1-0317	I1-0349	I1-0381	I1-0413	I1-0445
13	I1-0222	I1-0254	I1-0286	I1-0318	I1-0350	I1-0382	I1-0414	I1-0446
14	I1-0223	I1-0255	I1-0287	I1-0319	I1-0351	I1-0383	I1-0415	I1-0447
15	I1-0224	I1-0256	I1-0288	I1-0320	I1-0352	I1-0384	I1-0416	I1-0448
16	I1-0225	I1-0257	I1-0289	I1-0321	I1-0353	I1-0385	I1-0417	I1-0449
17	I1-0226	I1-0258	I1-0290	I1-0322	I1-0354	I1-0386	I1-0418	I1-0450
18	I1-0227	I1-0259	I1-0291	I1-0323	I1-0355	I1-0387	I1-0419	I1-0451
19	I1-0228	I1-0260	I1-0292	I1-0324	I1-0356	I1-0388	I1-0420	I1-0452
20	I1-0229	I1-0261	I1-0293	I1-0325	I1-0357	I1-0389	I1-0421	I1-0453
21	I1-0230	I1-0262	I1-0294	I1-0326	I1-0358	I1-0390	I1-0422	I1-0454
22	I1-0231	I1-0263	I1-0295	I1-0327	I1-0359	I1-0391	I1-0423	I1-0455
23	I1-0232	I1-0264	I1-0296	I1-0328	I1-0360	I1-0392	I1-0424	I1-0456
24	I1-0233	I1-0265	I1-0297	I1-0329	I1-0361	I1-0393	I1-0425	I1-0457
25	I1-0234	I1-0266	I1-0298	I1-0330	I1-0362	I1-0394	I1-0426	I1-0458
26	I1-0235	I1-0267	I1-0299	I1-0331	I1-0363	I1-0395	I1-0427	I1-0459
27	I1-0236	I1-0268	I1-0300	I1-0332	I1-0364	I1-0396	I1-0428	I1-0460
28	I1-0237	I1-0269	I1-0301	I1-0333	I1-0365	I1-0397	I1-0429	I1-0461
29	I1-0238	I1-0270	I1-0302	I1-0334	I1-0366	I1-0398	I1-0430	I1-0462
30	I1-0239	I1-0271	I1-0303	I1-0335	I1-0367	I1-0399	I1-0431	I1-0463
31	I1-0240	I1-0272	I1-0304	I1-0336	I1-0368	I1-0400	I1-0432	I1-0464

Special Outputs Definition

A group of bits in the internal output status table are reserved for use with the Series Five Genius bus controller or Operator Interface Unit. These internal outputs are references O2-1000 through O2-1024.

Table 3-6. Special Internal Output Definition

Reference	Written By		Definition
	CPU	User	
02-1000			ESC key locked (1), unlocked (0); can be set by user program.
02-1001			Buzzer On (1), buzzer Off (0); can be set by user program.
02-1002			Keyclick On (1), keyclick Off (0);can be set by user program.
02-1003			Backlight On (1), backlight Off (0); can be set by user program.
02-1004			Printer Output Start (1), set by program; Ready (0) set by OIU.
02-1005			Transfer Cartridge Identification Start (1), set by user program; Ready (0) set by OIU.
02-1006			Transfer Data from OIU to CPU (1), set by user program; Ready (0) set by OIU.
02-1007			Transfer Data from CPU to OIU (1), set by user program; Ready (0) set by OIU.
02-1008			Clear Memory Cartridge (1), set by user program; Done (0), set by OIU.
02-1009			Printer Error (1), Printer OK (0), set by OIU.
02-1010			Memory Cartridge Error (1), Memory Cartridge OK (0), set by OIU.
02-1019	X		SETUP ERROR - If set by the CPU, the data in registers R04049 -R04051 is incorrect. The register data is only checked when the ENABLE Genius DIAGNOSTICS bit (02-1024) transitions from CLEARED to SET, or during power-up if the CPU is in RUN or RUN DISABLED mode and the ENABLE Genius DIAGNOSTICS bit is SET. If a SETUP ERROR exists, the Genius diagnostic routine is not executed.
02-1020	X		FAULT TABLE OVERFLOW - If SET by the CPU, this bit indicates that a fault table overflow exists. An overflow occurs if the value of the NUMBER OF FAULTS register (R04051) is greater than the value of the LENGTH OF FAULT TABLE register (R4050). This bit remains set as long as the overflow condition exists. The CPU will CLEAR the bit when there is no overflow condition. If diagnostics are enabled this bit will be updated every sweep when the CPU is in the RUN or RUN DISABLED mode.
02-1021	X	X	PULSE TEST - If set by user logic, the CPU will send a PULSE TEST datagram to all logged in devices in the system. The CPU monitors this bit, and if it is SET the CPU will direct a PULSE TEST datagram to each logged-in device in the system (a maximum of one PULSE TEST datagram is sent by each Genius bus controller in the system per CPU sweep.) When the PULSE TEST datagram has been sent all devices in the system, the CPU will CLEAR this bit. If diagnostics are enabled, this bit will be checked every sweep when the CPU is in the RUN or RUN/DISABLED mode, except when the pulse test or clear all faults function is in progress. If the PULSE TEST bit and the CLEAR ALL FAULTS bit are SET at the same time, the clear all faults function will be performed first.

Table 3-6. Special Internal Output Definition - Continued

Reference	Written By		Definition
	CPU	User	
02-1022	X	X	CLEAR ALL FAULTS - To initiate a Clear All Faults command to all logged in devices in the system, this bit should be set by user logic. The CPU monitors this bit, and if it is SET the CPU will CLEAR the NUMBER OF FAULTS register (R04051). The CPU will then direct a CLEAR ALL FAULTS datagram to each logged-in device in the system (a maximum of one CLEAR ALL FAULTS datagram is sent by each GENI in the system per CPU sweep). When the CLEAR ALL FAULTS datagram has been sent to all devices in the system, the CPU will CLEAR this bit. If diagnostics are enabled, this bit will be checked every sweep when the CPU is in the RUN or RUN DISABLED mode, except when the pulse test or clear all faults function is in progress. If the PULSE TEST bit and the CLEAR ALL FAULTS bit are SET at the same time, the clear all faults function will be performed first.
02-1023		X	REPORT ADD/LOSS OF BLOCK AS FAULT - If this bit is SET by the logic, the addition or loss of a device in the system will be reported as a fault in the fault table, otherwise these conditions are not entered into the fault table. If diagnostics are enabled, this bit will be checked every sweep when the CPU is in the RUN or RUN DISABLED mode.
02-1024		X	ENABLE Genius DIAGNOSTICS - If this bit is SET, then Genius diagnostics will be processed by the CPU. If the bit is CLEARED, then Genius diagnostics routine is not executed, and the CPU will not write to internal outputs 02-1019 through 02-1022. This bit will be checked every sweep when the CPU is in the RUN or RUN Disabled mode.

Diagnostic Status Words

A group of five Diagnostic Status Words can be obtained by using CCM memory type 9, start address 0000, length 10 bytes. For detailed information on status word meanings (00H - 09H) and port error codes (00H - 01H), refer to the following tables.

Table 3-7. Diagnostic Status Words

Address	Meaning
00H - 01H	Port error codes (Refer to Table 3-8)
02H - 03H	Number of successful conversions on communications port
04H - 05H	Number of aborted conversions on communications port
06H - 07H	Number of header block retries on communications port
08H - 09H	Number of data block retries on communications port

NOTE

The least significant byte shows the result of the most recent communication, the most significant byte shows the result of the next most recent communication.

It is also possible for the host CPU to read the CCM diagnostic status words. A TRANSFER instruction must be used. The TRANSFER instruction should be set up as follows:

```
                    R100
|------][---------[  TRANSFER  ]-----
```

Prior to initiating the TRANSFER, the following registers need to be set up as follows.

R100 = See Table 3-8
R101 = 7648 decimal
R102 = 32773 decimal
R103 = 1998 decimal *
R104 = 10 decimal

```
*(R#-1) X 2 = value e.g., for R1000 (1000-1) X 2 = 1998
```

Table 3-8. Register Content

CCM Slot Number	Register 100 (Decimal)
0	32774
1	32775
2	32776
3	32277
4	32778
5	32779
6	32780
7	32781

The value of R100 depends on the slot number in which the CCM Master module is installed, as shown in the chart. The value in R103 determines the starting register for the data to be stored, and R104 indicates the number of diagnostic words to fetch (10 words in this case). For this example, R1000-R1009 would contain the 10 diagnostic status words.

Of course, any bank of five consecutive registers may be used in conjunction with the TRANSFER instruction.

THE VALUES FOR R100-R102 MUST BE AS SHOWN IN TABLE 3-8, OR YOUR CPU MAY MALFUNCTION.

Table 3-9. Port Error Codes Description

Error (dec)	Code (hex)	Description
0	00	Successful transfer.
1	01	A time out occurred on the serial link.
2	02	An external device attempted to write data to a section of the CPU scratch pad that is not allowed.
3	03	An external device attempted to read or write a nonexistent I/O point.
4	04	An external device attempted to access more data than is available in a particular memory type.
5	05	An external device attempted to read or write an odd number of bytes to Timer/Counter or register memory, user-logic memory, or the diagnostic status words.
6	06	An external device attempted to read or write one or more nonexistent Timer/Counter accumulated or register values.
7	07	An external device specified the transfer of zero data bytes.
8	08	An external device attempted to write to protected memory. This will be the error code if any attempt is made to Write to user-logic memory while the CPU is in the RUN mode. This is also returned if the password is active and the CPU is locked.
9	09	An external device attempted to transfer data to or from an invalid memory type.
10	0A	An external device attempted to read or write one or more nonexistent diagnostic status words.
11	0B	An external device attempted to transfer data beginning at an invalid user-logic memory or scratch pad address.
12	0C	Serial communication was aborted after a data block transfer was retried three times.
13	0D	Serial communication was aborted after a header transfer was retried three times.
15	0F	Unit address in ENQUIRY was correct but does not agree with unit address specified in the HEADER block.
20	14	One or more of the following errors occurred during a data block transfer: A. An invalid STX character was received, B. An invalid ETB character was received, C. An invalid ETX character was received, D. An invalid LRC character was received, E. A parity, framing, or overrun error occurred.
21	15	The CCM expected to receive an EOT character from an external device and did not receive it.
22	16	The CCM expected to receive an ACK or NAK character and did not receive it.
26	1A	A time out occurred during an attempt to transmit on a port due to CTS being in an inactive state too long.
29	1D	An error occurred when data was being transferred.
30	1E	A parity, framing, or overrun error occurred during a serial data block transfer.

Series Five Scratch Pad

The Scratch Pad stores information about the current Series Five CPU system status. This information is provided in abbreviated and preliminary form for illustration only. If additional information is required contact GE Fanuc Automation.

CAUTION

Extreme care must be used when writing to any Scratch Pad location. It is strongly recommended that you consult GE Fanuc Field Service before doing this.

Table 3-10. Series Five Scratch Pad Definition

Address	Read/Write	Stored in Memory Cartridge	Preliminary Description
00H	R/W	NO	Run/Stop command area
			Mode / Write / Read RUN 01H 03H RUN/DISABLE 81H 83H STOP 80H 80H
01H	R	NO	Run/Stop status area RUN = 03H RUN disabled = 83H STOP = 80H
02H	R	NO	Password status, 00H = unlocked, 01H = locked
05H	R	NO	Memory cartridge type (same as 102H) Bit 7 - unused Bit 6 - unused Bit 5 - 1 = user logic, 0 = system Bit 4,3 - 11 = CMOS RAM, 10 = UVPROM, 01 = EEPROM Bit 2,1,0 - 001 = 8K, 010 = 16K, 011 = 32K, 100 = 64K
06H	R	NO	CPU status flags Bit 7 - keyswitch, 1 = stop, 0 = run Bit 6,5 - 11 Bit 4 - 0 = write protected, 1 = write enabled Bit 3 - 1 Bit 2 - keyswitch, 1 = OIU, 0 = not OIU Bit 1,0 - 00 = 16K registers, 10 = 4K registers
10H	R	NO	CPU error status (same as 473H - 476H) Bit 7 - I/O config error Bit 6 - not memory cartridge Bit 5 - I/O bus error Bit 4 - Genius setup error Bit 3 - I/O parity error Bit 2 - grammar error Bit 1 - gate array parity error Bit 0 - compilation error

Table 3-10. Series Five Scratch Pad Definition - Continued

Address	Read/Write	Stored in Memory Cartridge	Preliminary Description
11H	R	NO	CPU error status Bit 7 - I/O address assigned past end of table Bit 6 - I/O address conflict Bit 5 - I/O module withdrawn Bit 4-2 - unused Bit 1 - memory cartridge has only data registers Bit 0 - memory cartridge has only system parameters
12H	R	NO	CPU error status Bit 7 - not CPU battery Bit 6 - not memory cartridge battery Bit 5 - CPU battery voltage is low Bit 4 - memory cartridge battery voltage is low Bit 3 - I/O error (blown fuse, terminal strip loose, etc) Bit 2 - Watchdog timeout Bit 1 - internal F20 Bit 0 - not used
13H	R	NO	CPU error status Bits 7 - 0 unused
14H	R	NO	Microcode revision number
15H	R	NO	Firmware revision number
16H	R	YES	CCM address/parity Bits 7 - 1 = no parity after next power-up, 0 = odd parity *Parity will not actually change until the next CPU power-up or reset.*
75H-B3H	R	NO	Error detail for 10H - 13H 75H has error address or rack/slot # of 10H bit 1 error, etc B3H has error address or rack/slot # of 13H bit 7 error
C0H-C7H	R/W	YES	File name of the program in the memory cartridge
101H	R	NO	CPU/CCM data rate type (same as 05H) Bits 7-3 - unused Bits 2,1,0 - 000 = 300 Bps, 010 = 1200 Bps, 101 = 9600 Bps, 110 = 19200 Bps
102H	R	NO	Memory cartridge type (same as 05H)
103H	R	NO	CPU keyswitch (use 06H instead) Bits 7-3 - unused Bits 2,1,0 - 001 = run, 010 - OIU, 100 = stop

Table 3-10. Series Five Scratch Pad Definition - Continued

Address	Read/Write	Stored in Memory Cartridge	Preliminary Description
10BH-162H	R	YES	I/O config loaded from memory cartridge 10B = Id for rack 0 10C = Id for power supply in rack 0 10D = Id for CPU or local I/O I/F in rack 0 10E = Id for module in slot 0 : : 162 = Id code for module in rack 7, slot 7 Loads from memory cartridge on power-up, CPU init forces all bytes to 0.
163H-1BAH	R	NO	I/O configuration of the actual installed items -same format as 10B - 162H. On power-up, or CPU init, CPU reads the real I/O and loads this area. CPU compares this area with 10B-162 area for I/O configuration check.
1BBH-212H	R	NO	I/O module diagnostic error codes per slot, same addressing as 10B-162H. CPU loads this area on power-up, after CPU init, and dynamically during execution.
213H	R/W	NO	CPU initialize request (write 0A5H) CAUTION
214H	R/W	NO	I/O configuration command Write 5AH = create new I/O configuration Write A5H = OK to run with memory cartridge configuration even though real I/O is different
215H	R/W	NO	CPU mode change request -- 1 causes listed function Bit 7 - "N" Scans (216H, 217H must be setup first) Bit 6 - Scan Stop Bit 5 - Restart scan Bit 4 - Break (requires 216H, 217H setup first) Bit 3 - 1 step Bit 2 - Test Bit 1 - Stop Bit 0 - Run
216H & 217H	R/W	NO	Refer to 215H -- break address, or number of scans to execute.
218H			Bits 7 - 5 unused Bit 4 - 1 = decimal display, 0 = octal display (non-S5) Bits 3,2 - 00 = both I/O, 01 = I only, 10 = O only Bits 1,0 - 00 = on/off status, 10 = I/O address, Bit 1,0 - 00 = 16K registers, 10 = 4K registers 11 = Register address (non Series Five)
219H & 21CH	R/W	NO	Password registration area

Table 3-10. Series Five Scratch Pad Definition - Continued

Address	Read/Write	Stored in Memory Cartridge	Preliminary Description
240H	R/W	YES	I/O parity retry counter
241H	R/W	YES	Critical Bases 1 = Non-Critical
256H to 2B5H	R/W	YES	Genius setup area 256H = Slot # 0 257H = Serial bus address 258H = Start status table address - 1st byte 259H = Start status table address - 2nd byte 25AH = Number of Inputs 25BH = Number of Outputs 25CH = Data Rate 25DH = Unused 25EH-261H = 4 bytes of output control, 1 bit/node Repeat for 7 more slots.
262H to 26DH			262H = Slot # 1
26EH to 279H			26EH = Slot # 2
27AH to 285H			27AH = Slot # 3
286H to 291H			286H = Slot # 4
292H to 29DH			292H = Slot # 5
29EH to 2A9H			29EH = Slot # 6
2AAH to 2B5H			2AAH = Slot # 7
2B6H	R/W	YES	Watchdog timer value Copied from memory cartridge on power-up, range is 20 to 999 msec., default is 200. 64H data = 200 msec. Data is in 2 msec. increments.
2B8H to 1BEH	R/W	NO	Calendar clock setting area 2B8H = Seconds 2B9H = Minutes 2BAH = Hours 2BBH = Day of week, 0 = Sunday, 1 = Monday, etc. 2BCH = Day 2BDH = Month 2BEH = Year All of the above values must be written together.
4FEH	R/W	YES	I/O configuration selection - C3H = check, else no check

Table 3-10. Series Five Scratch Pad Definition - Continued

Address	Read/Write	Stored in Memory Cartridge	Preliminary Description
500H to 5FFH	R	YES	I/O map - start I/O address for each slot 500H = rack 0, slot 0 : 5FFH = rack 7, slot 7 LSB = 1 means manual address, 0 = automatic address

Module ID. and Error Codes

The Series Five CPU scratch pad bytes (1BBH - 212H) are used for the module ID. and error checking. They are filled by the CPU after performing its error checking routine. The following table defines some of the errors that may be detected by the CPU.

All modules use the same error codes in common as follows:

80H = I/O configuration error
81H = Module withdrawn from this slot
83H = I/O bus error
85H = I/O address range error
86H = I/O address duplication error

Table 3-11. Internal Module ID. and Error Codes

Cat. No.	Module ID.	Additional Error Codes (In addition to those listed above)
CPU500	00H	
PRG500	01H	
PWR501	02H	
PWR500	0CH	
PWR514	0DH	
CHS508	04H	
CHS518	05H	
CHS506	0AH	
CHS516	0BH	
BEM500	08H	
BEM510	07H	
BEM511	06H	
BEM530	09H	01H = S3 I/O abnormal 02H = S3 Power supply off
MDL502	2BH	01H = terminal block loose
MDL503	33H	
MDL512	28H	01H = terminal block loose
MDL501	45H	01H = terminal block loose
MDL511	41H	01H = terminal block loose
MDL522	2AH	01H = terminal block loose
MDL521	40H	01H = terminal block loose
MDL525	40H	
MDL524	17H	(32 Inputs)
MDL524	13H	(16 Inputs)

Table 3-11. Internal Module ID. and Error Codes - Continued

Cat. No.	Module ID.	Additional Error Codes (In addition to those listed above)
MDL527	12H	
MDL526	2AH	
MDL533	37H	
MDL556	29H	01H = terminal block loose 02H = blown fuse
MDL552	2EH	01H = terminal block loose 02H = blown fuse 04H = no external power
MDL532	29H	
MDL551	47H	01H = terminal block loose 02H = blown fuse 04H = no external power
MDL552	2EH	
MDL555	46H	
MDL531	47H	01H = terminal block loose 04H = no external power
MDL542	2CH	01H = terminal block loose 02H = blown fuse
MDL541	42H	01H = terminal block loose 02H = blown fuse
MDL553	3AH	
MDL556	29H	
MDL557	1DH	
MDL577	2CH	
MDL575	42H	
MDL576	43H	
MDL586	48H	
MDL581	2DH	01H = terminal block loose 04H = no external power
MDL580	44H	01H = terminal block loose 02H = blown fuse 04H = no external power
MDL593	3BH	
APU510	30H	01H = terminal block loose 04H = no external power
APU500	34H	
APU510	30H	
APU520	31H	
APU521	32H	
ALG516	35H	01H = terminal block loose 02H = blown fuse
APU517	3FH	
APU518	3DH	
APU519	3CH	
ALG566	36H	01H = terminal block loose 02H = blown fuse
ALG568	39H	
CCM500	38H	

Functions That Write Data to CPU Memories

Since the Series Five PLC can have multiple CCM ports (one in the CPU and up to eight additional CCMs), it is possible for several external devices to write data to the CPU. When writing data to the CPU, the user must ensure that undesirable system interactions do not occur as a result of more than one external CCM device writing to the same CPU. Reading the CPU by multiple devices does not create any problems, but writing to the CPU by more than one device is not recommended.

The following pages describe two new functions (Set/Reset and Data Monitoring) that allow external devices to write data through a CCM request to the CPU's scratch pad. They are used by Logicmaster 5 during normal operation. If Logicmaster 5 is not in use, a single external device may use these functions.

CAUTION

Extreme care must be exercised by the user to ensure that one and only one device uses the Set/Reset and Data Monitoring functions. If Logicmaster 5 is operating, no other devices may use these functions.

Set/Reset Bit (Through Scratch Pad) Function

When forcing I/O points on or off through a normal serial interface, there is the possibility that other bits within the same byte may accidentally be turned on through interaction with user logic. The Series Five PLC prevents this from happening through use of the Bit Set/Reset function which uses a specific area of the scratch pad as a buffer. This function allows the forcing on or off of one or more bits in a single byte of a discrete Input or Output status table.

Data required for this function is written to a buffer in the scratch pad by a write request from a CCM device. Five consecutive bytes are required for this function. The location of these bytes in the scratch pad, the data required for the CCM request, and the return data after the function is initialized is shown below.

Scratch Pad Address (Hex)	Data at Request	Return (Same Location)
600	Function Code	Complete Code (0)
601	CCM Memory Type + 30H	OK (88H) or Error (FCH)
602	Reference Address (LSB)	Error Code (LSB)
603	Reference Address (MSB)	Error Code (MSB)
604	Bit Mask	Not Used

The function code for the first byte can be either 44H for bit set, 45H for bit reset, or 52H for a toggle. The CCM memory type for this byte can be 32H (input tables), 33H (output tables), 34H (input override tables), or 35H (output override tables). The reference address can be the values 1 to 180H for memory types 32H and 34H, or the values 1 to 280H for memory types 33H and 35H. The reference address identifies the byte in which the desired bit or bits are located. The mask identifies which bits in the byte are to be affected.

Force ON Function

If a bit is to be turned on, a corresponding bit in the mask is set to a 1. To turn on more than one bit, set all of the corresponding bits to a 1. Any bits that are not to be turned on, should be 0. The resultant mask byte can be used in an OR operation with the status byte to provide the requested status. For example, to turn on I0097 and I0103 with a CCM master device, the following data must be written to the scratch pad buffer:

Scratch Pad Address (Hex)	At Request
600	44 (Hex)
601	32 (Hex)
602	0D (Hex)
603	01 (Hex)
604	01000001 (41H)

In the example above, the memory type is 02 (input tables), with the byte including the target bit being the 0D byte into the local input table (offset =100H).

Force OFF Function

If a bit is to be turned off, a corresponding bit in the mask is set to a 0. To turn off more than one bit, set all of the corresponding bits to be turned off to a 0. Any bits that are not to be turned off, should be a 1. The resultant mask byte can be used in an AND operation with the status byte to provide the requested status. For example, to turn off O0050 and O0053 with Logicmaster 5, the following data must be written to the scratch pad buffer:

Scratch Pad Address (Hex)	At Request
600	45 (Hex)
601	33 (Hex)
602	07 (Hex)
603	01 (Hex)
604	11101101 (EDH)

Executing the Bit Set/Reset Function

The procedure for initiating and executing the Bit Set/Reset function is described below.

- Initialize the Force Bit Buffer in the scratch pad as described previously.
- The Series Five CPU detects a request of 44H (Bit Set) or 45H (Bit Reset) and performs the function in the same CPU sweep in which the request was detected.
- If no error is detected, the Series Five CPU sets byte 2 (location 601H) in the scratch pad buffer to an "OK" code (88H).
- If an error is detected, the CPU sets byte 2 to an error code (FCH), and writes the error code values into bytes 3 (LSB) and bytes 4 (MSB).
- When the operation is complete, the CPU sets byte 1 (Complete Code) to a 0.

Data Monitoring Through a CCM Device

A function is available which allows fast monitoring of a specified number of items (from 1 to 88 items) in a single CCM communication, even if the items are widely distributed in the CCM target memory space. These items (coils, contacts, etc.) can be located anywhere that is accessible to the external CCM device. There are no qualifiers, such as being in sequential order.

The addresses of the items to be monitored are written to a consecutive area of scratch pad memory called the Request Buffer, and the values of those items are written by the CPU to a like number of locations in another area of scratch pad memory called the Return Buffer. The location of these buffers in scratch pad memory and the data contained in the buffers is as follows:

Scratch Pad Address (Hex)	Request Buffer
610	Function code
611	Number of items
612	Item 1 Address (LSB)
613	Item 1 Address (MSB)
614	Item 1 Address (LSB)
615	Item 1 Address (MSB)
.	.
.	.
.	.
6BE	Item 87 Address (LSB)
6BF	Item 87 Address (MSB)
6C0	Item 88 Address (LSB)
6C1	Item 88 Address (MSB)

WRITE ONLY

Scratch Pad Address (Hex)	Return Buffer
752	OK/Error Code
753	Item 1 Value (LSB)
754	Item 1 (1) (MSB)
755	Item 2 Value (LSB)
756	Item 2 (1) (MSB)
.	.
.	.
.	.
801	Item 87 Value (LSB)
802	Item 87 (1) (MSB)
803	Item 88 Value (LSB)
804	Item 88 (1) (MSB)

READ ONLY

(1) Override table value if an I/O table, or MSB if a register value.

Request Buffer

The data required to be entered in the Request Buffer is:

- The function code value must be 42 (Hexadecimal) to request this function.
- The number of items that can be requested to be monitored is from 1 to 88 (decimal).
- The reference address for each of the requested items must be entered in the remaining buffer locations. Each reference address requires two locations, the first is the Least Significant Byte of the reference address, the second is the Most Significant Byte of the reference address. The valid entries for the addresses are as follows:

Table 3-12. Request Buffer Entries

Reference Address	Byte Address MSB/LSB	Reference Address	Byte Address MSB/LSB
R1	4001H	O1+0001	C001H
R3FFFH	7FFFH	O1+1024	C080H
I1+0001	8001H	O2+0001	C081H
I1+1024	8080H	O2+1024	C100H
I2+0001	8081H	O0001	C101H
I2+1024	8100H	O1024	C180H
I0001	8101H	O1-0001	C181H
I1024	8180H	O1-1024	C200H
		O2-0001	C201H
Scratch Pad	0H	O2-1024	C280H

Return Buffer

The return buffer will contain the values of the items specified in the request buffer after the function has been executed. The first location will contain the "OK" code 88H if no errors are detected during execution of the function, or if an error is detected, the value FCH will be written to the first location. The data values that will be written to the return buffer can take two forms:

1. Register Data - a word value with the Least Significant Byte in the lower byte and the Most Significant Byte in the higher byte.
2. Discrete data - The lower byte will contain a byte value of the discrete byte containing the requested address. The high byte will contain the byte value of the override byte containing the requested address.

Executing the Data Monitoring Function

The procedure for initiating and executing the Data Monitoring function is described below.

- The CCM master device initializes the Request Buffer as previously described. The function code (42H) must be the last buffer entry to be initialized.
- The Series Five CPU checks for a request of 42H every sweep and begins getting the data and writing it to the scratch pad Return Buffer. The update of the Return Buffer is completed within the same sweep after the logic solution to ensure data integrity.
- If no error is detected, the CPU sets scratch pad location 752H in the Return Buffer to 88H (OK code).
- If an error is detected the CPU sets scratch pad location 752H in the return buffer to an error code (FCH).
- When the operation is complete, the CPU sets byte 1 (Function Code) in the Request Buffer to a 0.
- For the next request of the same data, the master device only needs to change the Function Code from OH to 42H. The Series Five CPU updates the data once per request. If different data is required, the Request Buffer must be loaded with the new addresses.

Example of Data Monitor Request Function

The following examples show the data contained in the two scratch pad buffers at the time of the function request and at completion. The example request is for references R12 and I1+1024.

Content of buffer at function request:

Ref.	Scratch Pad Address (Hex)	Request Buffer	Scratch Pad Address (Hex)	Return Buffer
	610	42H		
	611	02H	752	88H
R12	612	OCH	753	0
	613	40H	754	0
I1+1024	614	80H	755	0
	615	80H	756	0
	616	0	757	0
	617	0	758	0
	.	.	.	.
	.	.	.	.
	.	.	.	.
	6C0	0	801	0
	6C1	0	802	0

Content of buffer after function is complete:

Ref.	Scratch Pad Address (Hex)	Request Buffer	Scratch Pad Address (Hex)	Return Buffer
	610	0		
	611	02H	752	88H
R12	612	OCH	753	12H
	613	40H	754	20H
I1+1024	614	80H	755	00110111 status
	615	80H	756	00000000 override
	616	0	757	0
	617	0	758	0
	.	.	.	.
	.	.	.	.
	.	.	.	.
	6C0	0	801	0
	6C1	0	802	0

In the above example, R12 has the value 2012H and I1+1024 is not overridden. Note that the status of the entire byte, I1+1017 through 1024 is returned.

Chapter 4
Communication Examples

GFK-0244

This chapter explains the Series Five® PLC ladder logic programming and register setup for serial data communications between the Communications Control Module (CCM), or Genius bus controller.

Read/Write CCM

As shown in Figure 4-1, the Read CCM and Write CCM instructions move data through a CCM or Genius bus controller to or from a remote device. You can use RD or WR CCM to send and receive data with any slave CCM compatible device. RD or WR CCM can also be used to communicate with other Series Five using Genius Bus Controllers.

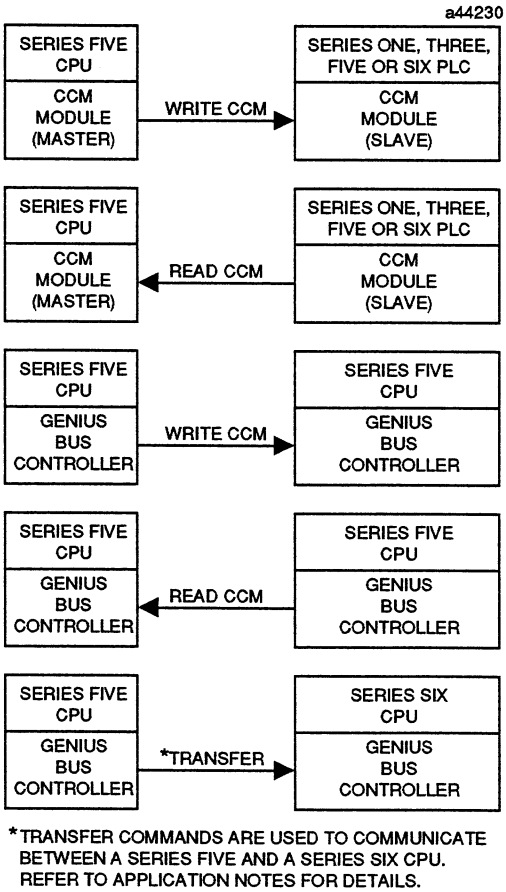


Figure 4-1. Read CCM/Write CCM, Move Data Instruction

Each of the instructions in Figure 4-1 has a single register operand which specifies the starting address of a block of six registers containing the following information:

- Slot number of the local CCM or Genius I/O bus controller.
- Target memory type.
- Starting address within the memory type of the remote target device.
- Length of the data to be transferred.
- Starting register in the local CPU for the data buffer.

NOTE

The Read/Write (RDCCM, WRCCM) command cannot be used to communicate on a Genius link between a Series Five and a Series Six CPU. The TRANSFER command must be used instead. Refer to GFK-0248 *Series Five Programmable Controller Genius Bus Controller User's Manual* for details.

Command Symbology

The Read (RDCCM) ladder logic instruction is used to request data from a remote location to the data register buffer in the local CPU. Likewise, the Write (WRCCM) instruction is used to send data from the local CPU to a remote device.

```
   R*****                    R*****
--|RDCCM|--      or      --|WRCCM|--
```

CCM/CPU Memory Mapping

Several areas of memory in the Series Five PLC provide special functions. These functions include the special purpose coils, dedicated registers, and the CPU's scratch pad memory. See Chapter 3 for details.

The CCM protocol must be given a start address and a length. The start address plus the length should not go past the end of a table boundary. Definitions for these items in memory are provided in Chapter 3, Table 3-1 CCM/CPU Memory Mapping.

CAUTION

Extreme care must be used when writing to any Scratch Pad location. It is strongly recommended that you consult GE Fanuc Field Service before doing this.

Setup Registers for Read/Write

Before execution, the following registers must contain the data listed below. (Refer to the Table 3-1)

Read CCM

Rxxxxx	=	Slot number of the local CCM or Genius bus controller.
+ 1	=	CCM ID or serial bus address of Genius bus controller.
+ 2	=	CCM memory type to read from (see Table 3-1).
+ 3	=	Start address in target (See table 3-1).
+ 4	=	Length to read (words = registers) maximum = 64.
+ 5	=	First register in local receive buffer.

Write CCM

Rxxxxx	=	Slot number of local CCM or Genius bus controller.
+ 1	=	Destination CCM ID or serial bus address of Genius bus controller.
+ 2	=	CCM memory type to write to (see Table 3-1).
+ 3	=	Start address in target (see Table 3-1).
+ 4	=	Length to write (words = registers) maximum = 64.
+ 5	=	First register in local transmit buffer.

After execution, the register data buffer pointed to by Rxxxxx +5 will contain the data from the remote device.

Special internal bits affected are I1-0081 to I1-0204 and I1-0045 for the CCM and Genius bus controller. (see Chapter 3)

Programming the Read/Write Instruction

1. From the Supervisor menu, select EDIT Prog (F2). The Edit Program function keys will be displayed at the bottom of the screen.
2. Select either INSERT RUNG (F5) or EDIT RUNG (F6), depending upon whether you wish to begin a new rung or edit an existing rung.
3. Enter any logic required to control power flow to the function.
4. With the cursor at the location for the element, select ADVNCD MN GR (F7), CONTRL FUNC (F6), READ WRITE (F5), and then READ CCM (F1) or WRITE CCM (F2). The Read/Write CCM display will appear.
5. Using the numeric keypad, type in the reference for the transfer.
6. After entering the reference, press the Enter key.
7. Complete the logic for the rung; then press the Accept key. The Edit key functions will reappear at the bottom of the screen.

Programming Examples

The Read CCM (RDCCM) and Write CCM (WRCCM) ladder logic instructions move data to and from the local CPU or Genius Bus Controller via the CCM. Each of these instructions has a single register operand which specifies the starting address of a block of six registers.

Reading from a Remote CCM Device

1. Read inputs I0017-I0048 from the remote Series Five CPU/CCM with CCM ID number 20.
2. Load the data into registers R00200-R00201. Assume that the local CCM module is in slot 5.
3. Preset the registers with the following data:

R1001	=	05	(slot number of local CCM module).
R1002	=	20	(CCM ID of remote CPU).
R1003	=	02	(memory type = input table).
R1004	=	103H	(start address for input table, I0017).
R1005	=	02	(number of words to fetch; also, length of data buffer).
R1006	=	200	(data buffer to start at register R0200).

4. **Execute the following:**

```
   R01001
--|RDCCM|--
```

5. **During execution, status bit I1-0091 will indicate the transfer status:**

0	=	done
1	=	executing for slot 5

6. Status bit I1-0092 will indicate the error status:

0	=	OK
1	=	error for slot 5

7. The data buffer for this example will contain the following data:

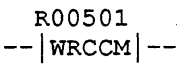
R00200	=	0002	(inputs 17, 19-32 are 0, input 0018 is 1).
R00201	=	0004	(inputs 33, 34, 36-48 are 0, input 35 = 1).

Writing to a Remote CCM Device

1. Write data buffer (R00220-00222) to outputs 1-48 in a remote Series Five CPU/CCM with CCM ID number 4. The local CCM module is in slot 2.
2. Preset the registers as follows:

R00501	=	02	(local CCM master module is in slot 2).
R00502	=	04	(target CCM ID).
R00503	=	03	(target memory type = output table).
R00504	=	0101H	(starting address for outputs 1-48).
R00505	=	03	(number of words to send).
R00506	=	220	(start data buffer at register 220).
R000220-R00222	=		data to send.

3. Execute the following:



4. During execution, status bit I1-0085 will indicate the transfer status:

   0 = done
   1 = executing for slot 2

5. Status bit I1-0086 will indicate the error status:

   0 = OK
   1 = error for slot 2

Chapter 5
Serial Interface Protocol

The purpose of this chapter is to provide complete information on the Series Five® PLC Communications Control Module (CCM) serial interface protocol and timing. Information to allow the user to write a serial communications driver for a host computer or microprocessor.

Introduction, Master-Slave Protocol

The serial interface protocol, used for CCM data communications, is based on the Master-Slave portion of CCM protocol developed for Series Six® PLC data communications. As used with the Series Five PLC, the CCM module can function either as a master or slave. For a complete description of all aspects of CCM protocol, see GEK-25364, Chapter 4, of the *Series Six Data Communications Manual.*

Asynchronous Data Format

Data transferred across the physical channel will be sent serially one bit at a time. The data is divided into 8-bit bytes and is transferred using an asynchronous format. Figure 5-1 shows the data format. If parity is selected, an additional parity bit is sent.

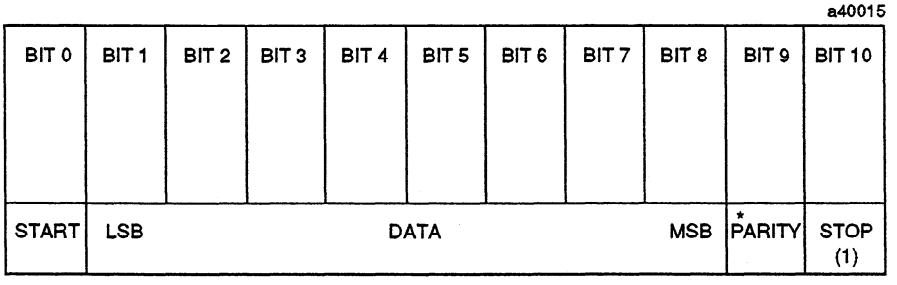


Figure 5-1. Serial Data Format

The 8-bit binary data is transferred with parity and block check codes. As will be explained in detail later, the data transfer consists of a 17-byte header followed by data blocks. The data transfers can be in either direction and are specified by the header.

Control Character Coding

The control characters used in the serial interface protocol and their meaning are given in Table 5-1.

Table 5-1. Control Character Codes

Abbreviation	Hex Value	Meaning
SOH	01	Start of Header
STX	02	Start of Text
ETX	03	End of Text
EOT	04	End of Transmission
ENQ	05	Enquiry
ACK	06	Acknowledgment
NAK	15	Negative Acknowledgment
ETB	17	End of Transmission Block

Enquiry Response Delay

The enquiry response delay is a timed delay inserted between the receipt of an enquiry sequence from a master and the response by a slave. This is done so that idle slaves, which monitor any active link between the master and a slave, will not be confused by enquiry sequences occurring during transmission of the data text. When an idle slave recognizes an apparent enquiry sequence, it starts an internal timer of 10 msec. plus 4 character times.

If any other character is received before the timer times out, the idle slave disregards the enquiry. Therefore, any device transmitting data text on a multidrop link should ensure that there will be no gaps in the text greater than 2 character times so an idle slave will not misinterpret data as an enquiry sequence.

Normal Sequence*, Master-Slave

The form of the Normal (N) Enquiry Sequence from the master to the target slave CCM and the response by the target slave CCM is shown below.

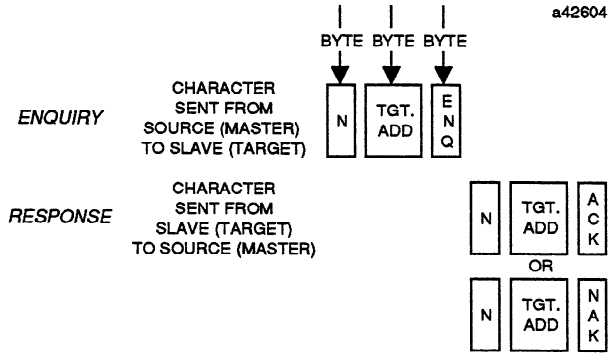


Figure 5-2. Enquiry Sequence from Master to Target Slave

* The term, Normal Sequence, is retained from the explanation of CCM protocol in GEK-25364, Series Six Data Communications Manual.

N	:	ASCII coded "N" (4E in HEX coding) used to specify Normal Sequence operation--sent as a single byte.
Target Address	:	Target address is the target ID number (set with the CCM master or CPU ID DIP switches) to which the master is attempting communications plus 20H (ASCII coded "!" though "z" or 21 through 7A in HEX coding)--sent as a single byte.
ENQ	:	ASCII control character meaning enquire--sent as a single byte.

ACK or NAK : Response from slave meaning acknowledge or negative acknowledgment--sent as a single byte.

If the slave response to a master enquiry is invalid, the master will delay a short time and retry the enquiry. The master will retry the enquiry 3 times before aborting the communication.

Normal Sequence Protocol Format

The general format for a successful communication is shown in Figures 5-3 and 5-4. Figure 5-3 shows a data transfer from the source device to the target device and Figure 5-4 shows a data transfer from the target device to the source device. The source device is always the initiator of the request; the target device receives the request.

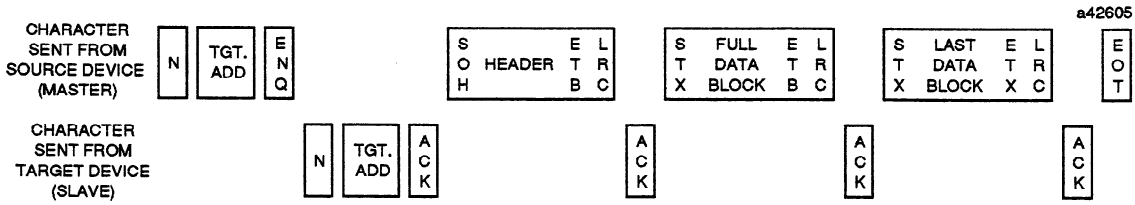


Figure 5-3. Data Transfer from Master to Slave

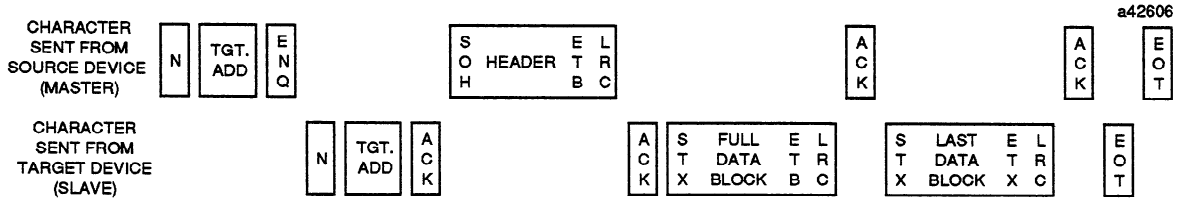


Figure 5-4. Data Transfer from Slave to Master

Master-Slave Normal Sequence Flow Charts

To fully understand how the protocol operates under error conditions see the flow charts and accompanying explanation.

Normal Sequence, Master (See Figure 5-5)

```
Start N Sequence.
Start N. Enquiry.
Has enquiry been retired 3 times?
  If YES, send EOT
  If NO, send N Enquiry
Read N Enquiry response.
Is there a timeout or error in response (response not an ACK or a NAK)?
  If YES, delay 10 msec. or the turn-around delay if it is not 0 msec.,
  Increment enquiry count and return to Start N Enquiry
  If NO, send the header to the slave.
Read response to header.
Is there a timeout on the response?  (Condition 4, Table 5-3)
  If YES, send an EOT and exit the initiate sequence.
  If NO, is response an ACK or NAK?
    If YES, has header been retried 3 times?
      If YES, send EOT and exit initiate sequence.
      If NO, return to "Send Header".
    If NO, go to "Read or Write Data Blocks" depending on the direction of
    data transfer.
```

Normal Response, Slave (See Figure 5-6)

```
Start N Response.
Read N Enquiry.
Is N Enquiry sequence correct?
  If NO, return to "Read N Enquiry".
  If YES, Start timer of 10 ms plus 4 character times.
Is timer done?
  If NO, have any characters arrived?
    If NO, go to "Is Timer Done?".
    If YES, go to "Read N Enquiry".
  If YES, send N Enquiry Response.
Read header.
Is there a timeout between ENQ response and the first character of the header?
  If YES, send EOT and exit.
  If NO, is header OK?
    If NO, has header been retried 3 times?
      If YES, send EOT and exit.
      If NO, send NAK and return to "Read Header".
    If YES, send ACK and go to "Read and Write Data Blocks" depending on the
    direction of data transfer.
```

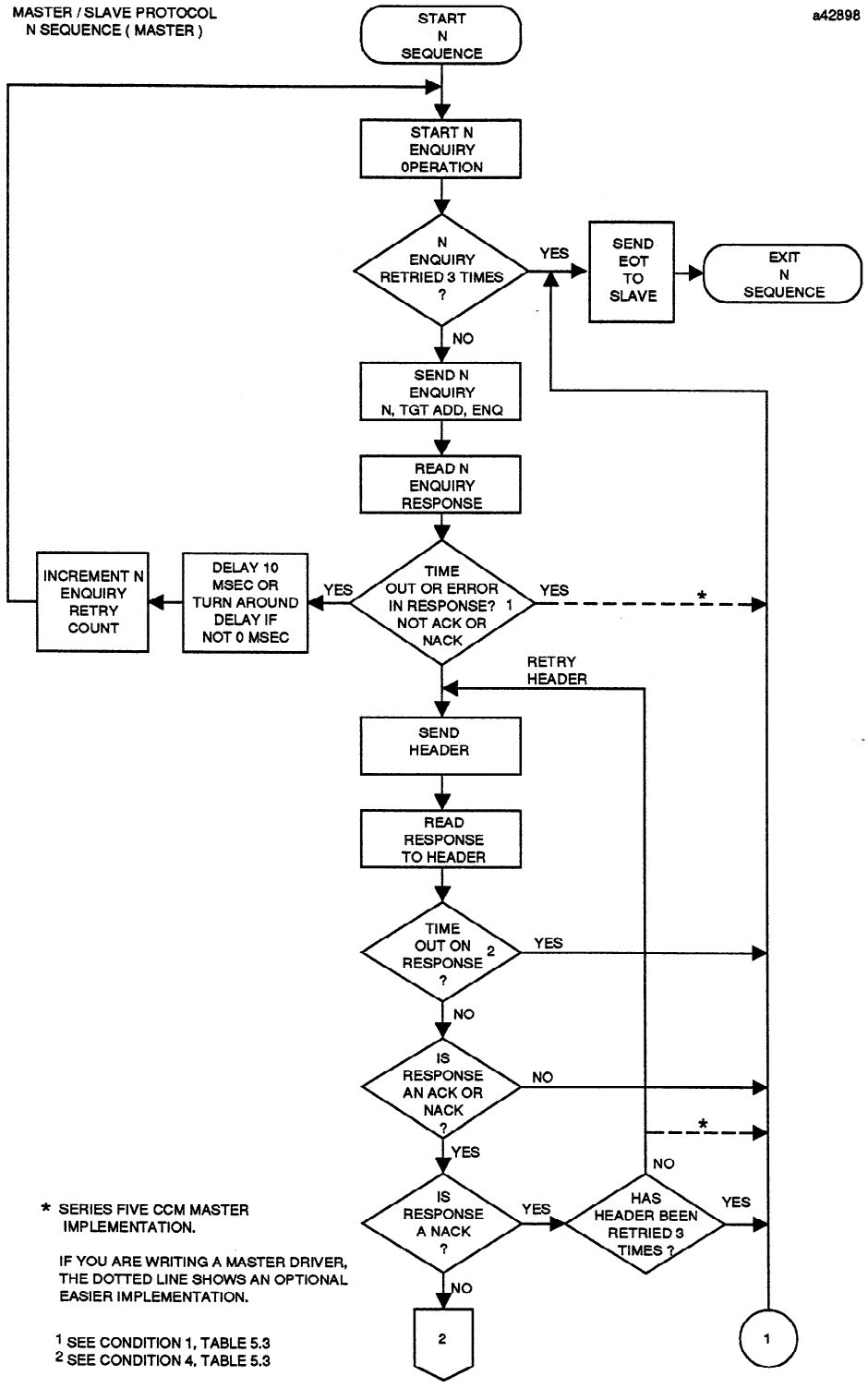


Figure 5-5. N Sequence, Master

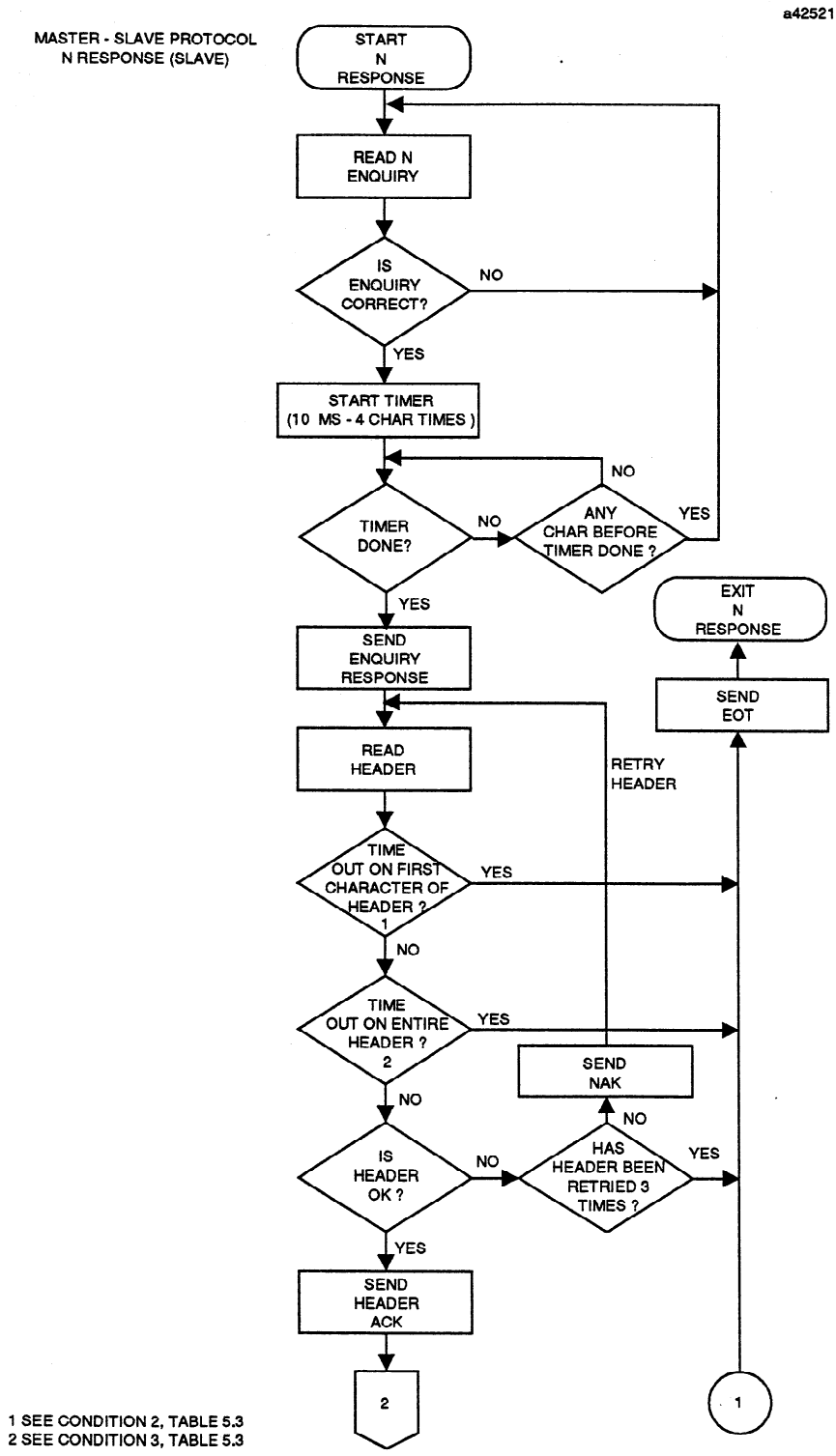


Figure 5-6. N Response, Slave

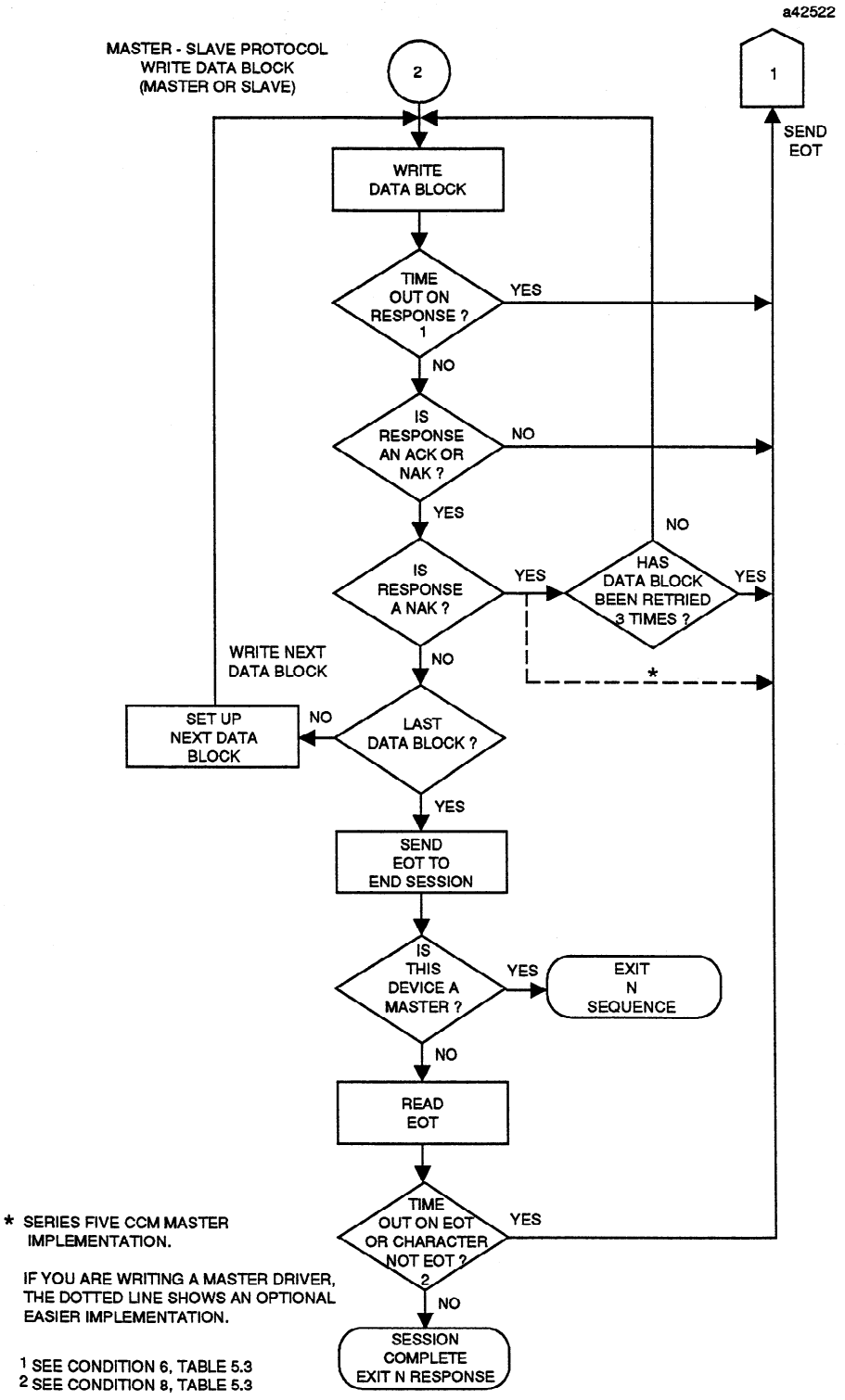


Figure 5-7. Write Data Blocks, Master or Slave

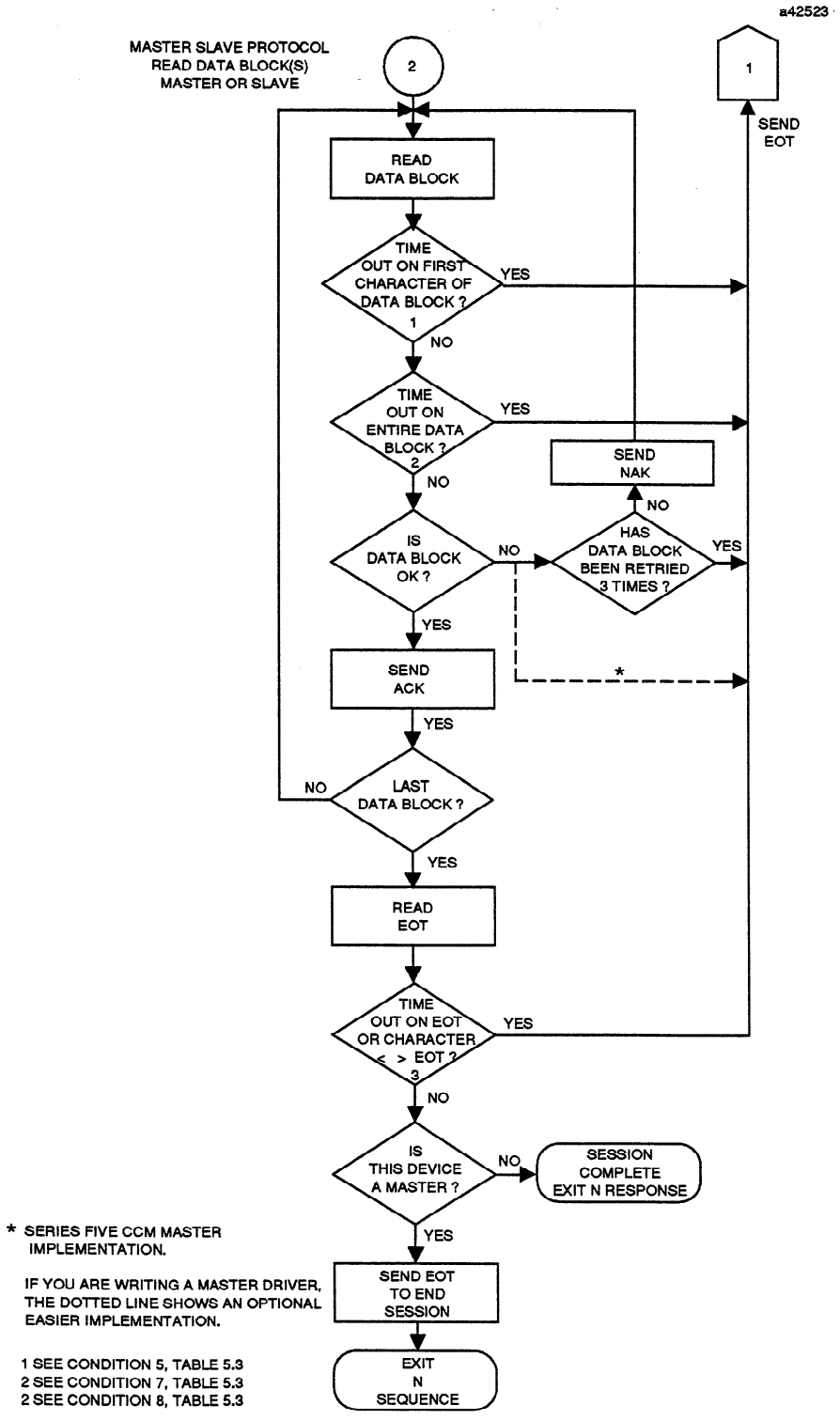


Figure 5-8. Read Data Blocks, Master or Slave

Write Data Blocks, Master or Slave (See Figure 5-7)

```
Write data block.
Is there a timeout on the data block response?  (Condition 6, Table 5-3)
  If YES, is data block response ACK or NAK?
  If NO, is data block response ACK or NAK?
    If not ACK or NAK, send EOT to other device and exit.
    If ACK or NAK, is it a NAK?
      If YES, has data block been retried 3 times?
      If NO, return to "Write Data Block".
   *If NO, is it last data block?
      If NO, set up next data block and return to "Write Data Block".
      If YES, send EOT to end session.
Is this device a Master?
  If YES, exit N Sequence.
  If NO, read EOT.
Is there a timeout on EOT or is character not an EOT?  (Condition 8, Table 5-3)
  If there is a timeout or the character is not EOT, send EOT and exit N
  Response.
  If EOT is OK, session is complete.  Exit N response.
```

Read Data Blocks, Master or Slave (See Figure 5-8)

```
Read data block.
Is there a timeout on the first character of the data block?  (Condition 5,
Table 5-3)
  If YES, send an EOT and exit.
  If NO, is there a timeout on the entire data block?  (Condition 7,  Table 5-3)
    If YES, send an EOT and exit.
    If NO, is the data block OK?
      If NO, has the data block been retried 3 times?
        If YES, send EOT and exit.
        If NO, send NAK and return to "Read Data Block".
      If YES, send ACK.
     *Is it the last data block?
        If NO, return to "Read Data Block".
        If YES, read EOT.
        Is there a timeout on the EOT or is the character not an EOT?
          If there is a timeout or the character is not EOT, send EOT and exit.
          If EOT is OK, is this device a master?
            If NO, the session is complete, exit N Response.
            If YES, send EOT to end session, exit N Sequence.
```

Master-Slave Message Transfers

As explained before, when the master wishes to initiate a data transfer, it issues a three-character enquiry sequence. The receiving device responds by sending a three-character acknowledge or negative acknowledge sequence. This establishes a link which permits the transfer of a message. Message transfers consist of a 17-byte header, sent by the master, followed by a block of data.

Header Block

A header block is sent before the text data block to describe transfer of data. The header specifies the direction of the data transfer, the amount and location of the data to be transferred, and the destination of the transfer. The header is composed of 17 bytes; the header format is shown in Figure 5-9.

a40368

S O H	I D	DATA FLOW DIR	MEM TYPE	MEM ADD MSB	MEM ADD LSB	NO COMP DATA BLKS	NO BYTES LAST BLK	SRCE I D	E T B	L R C
1	2 3	4	5	6 7	8 9	10 11	12 13	14 15	16	17

Figure 5-9. Serial Header Format

BYTE 1 SOH	(01H)
BYTES 2 + 3	CCM (target) ID Number (not encoded the same as the target address)
BYTES 4 + 5	Data flow direction, target memory type
BYTES 6 + 7	Most significant byte of address of requested data
BYTES 8 + 9	Least significant byte of address of requested data
BYTES 10 + 11	Number of complete data blocks to follow the header
BYTES 12 + 13	Number of bytes in incomplete last block.
BYTES 14 + 15	Source ID Number
BYTE 16 ETB	(17H)
BYTE 17 LRC	(Exclusive "OR" of Bytes 2-15)

The information in bytes 2-15 are ASCII coded hexadecimal. Valid ASCII coded hexadecimal values are 30H-39H (0-9) and 41H-46H (A-F). For fields requiring more than one byte, the most significant byte is transmitted first.

Target ID Number

The CCM ID (target ID) is the identification number and it is set with DIP switches. This number can range from 1 to 90. (In ASCII coded HEX: 01 to 5A). This is not encoded the same as the Target Address in the enquiry sequence. See the section, Normal Enquiry Sequence, in this chapter.

Data Flow Direction and Memory Type

Bytes 4 and 5 inform the CCM of the direction of the transfer and the memory type involved. Refer to Table 5-2 below for (Byte 4) Data Flow Direction.

Table 5-2. Data Flow Direction (Byte 4)

Contents of Byte 4 DEC	HEX	ASCII	Data Flow Direction
48	30	0	Read from CPU or CCM
56	38	8	Write to CPU or CCM

The target memory address specifies the address within the CPU, CCM where the transfer is to begin. The mapping of reference numbers to numbers used for the target memory address is shown in Chapter 3, Table 3-1, Series Five CCM/CPU Memory Mapping.

Refer to the appropriate Series One, Series Three, or Series Six User's Manual, listed in the Preface of this manual, for other PLC applications.

Number of Complete Data Blocks to Follow Header

This specifies the number of 256-byte data blocks to be transferred following the header. For more information, refer to the following section on Text Data Blocks and also Chapter 3, Table 3-1, Series Five CCM/CPU Memory Mapping. This information will help you to determine how many 8-bit bytes are required for a particular transfer.

Number of Bytes in Incomplete Last Block

This specifies the number of bytes in the last data block. When the number of complete data blocks is zero, this number specifies the total number of bytes to be transferred.

Source ID Number

The source ID number is the identification number of the source device. For a Series Five CPU, this ranges from 1 to 5AH.

Text Data Block

The text data block always starts with a Start-Of-Text (STX) character which is followed by the text. The text is followed by an End-Of-Text (ETX) character. This is then followed by the text data checksum. This checksum is used to verify the data's integrity. The checksum, (LRC) is an exclusive "OR" of all the text data bytes.

When 16-bit information (registers or user logic) is being transferred in a text data block, the least significant byte is transferred first followed by the most significant byte.

Header and Text Data Block Response

The header and text data blocks are responded to with an acknowledge (ACK) or negative acknowledge (NAK). An ACK means that the header or text was acceptable and grants permission to the sending device to start sending the next data block.

A NAK means that the header or text was not acceptable and asks for a retransmission of the header or data. The unacceptable header or text may be retried three times.

Message Termination

After the ACK to the final text data block has been received, the device receiving the ACK sends an End-Of-Transmission (EOT) character to close the serial link. The master always terminates the link with an EOT.

Timing Considerations

A timeout occurs on a serial link when the CCM or remote CCM device does not receive a response, a header, or data from another device within a fixed amount of time.

Serial Link Timeouts

Timeouts are used on the serial link for error detection, error recovery, and to prevent missing end-of-block sequences. Whenever a serial link timeout occurs, the CCM or remote CCM device will abort the conversation and send an EOT to the other device. After an EOT, a new enquiry sequence must be sent to restore communications. Refer to Table 5-3 for timeouts at any point in the serial protocol.

Turn-Around Delays

Turn-around delay options of 0 to 10 msec. for the CCM can be selected by DIP switch. A 10-msec. turn-around delay should be selected when using modems in the half-duplex mode of operation or when using full-duplex modems in multidrop configurations. This delay allows the time needed to signal the modem to turn on and ringing on the line to stop before actual transmission of data.

The CCM will delay 10 msec. before sending a control character, the start of header, or the start of a text data block.

When the 10 msec. turn-around delay is selected, the time is automatically added to the serial timeouts in Table 5-3.

Table 5-3. Serial Link Timeouts

Condition	Timeout With Turn Around Delay	
	0 msec.	10 msec.
1. Wait on ACK/NAK following ENQ	800	810
2. Wait on start of header following ACD of ENQ	800	810
3. Wait on header to finish: Data Rate (bps)		
300	2670	2680
1200	670	680
9600	670	680
19200	670	680
4. Wait on ACK/NAK following header	2000	2010
5. Wait on start of data following ACK of header	20000	20010
6. Wait on ACK/NAK following data block	20000	20010
7. Wait on data block to finish: Data Rate (bps)		
300	33340	33350
1200	8340	8450
9600	8340	8350
19200	8340	8350
8. Wait on EOT to close link	800	810

Communication Errors

Serial Link communication errors are divided into four groups:

1. Invalid Header
2. Invalid Data
3. Invalid NAK, ACK or EOT
4. Serial Link Timeouts

The different errors are outlined in the following four sections:

NOTE

If you experience communication errors, retrieve the Diagnostic Status Words for troubleshooting information. For the format of the diagnostic status words, see the section, Diagnostic Status Words, in Chapter 5.

Invalid Header

The following errors cause the header to be invalid and therefore NAK'ed by the target device.

- Incorrect LRC (header checksum).
- No SOH.
- No ETB.
- Parity, overrun, or framing error.
- Invalid unit ID number (does not match resident unit ID number).
- Invalid memory type.
- Attempted to access memory which is password protected.
- Invalid header character (not 0-9, A-F).
- Invalid address for specified memory address (see description of memory types).
- Number of complete blocks and number of bytes in last block both = 0
- Number of bytes in last block not even when the memory type is 1 or 7.
- Writing a partial instruction to user logic.

The header may be retried a maximum of three times. If the CCM is configured as a slave and the header still has one of the errors listed, the CCM will abort the session and send and EOT to the master. The Slave then waits for an ENQ to start a new session.

Invalid Data

If any of the following errors occur, the same procedure is followed as for an invalid header.

- Incorrect LRC (checksum)
- No STX
- No ETB or ETX
  (ETX must occur in last block only)
- Parity, Overrun, or Framing Error

Invalid NAK, ACK, or EOT

If the CCM is expecting one of these control characters and a character is received that is not one of these, the CPU or CCM aborts the session and sends an EOT to the other device.

Serial Link Timeout

If at any time during the conversation the CCM times out waiting for the other device, the conversation is aborted and an EOT is send to the other device.

Chapter 6
RTU Communications Protocol

GFK-0244

Introduction

The Communications Control Module (CCM - IC655CCM500B, Revision B, or later) uses two protocols, CCM Serial Interface and Remote Terminal Unit (RTU). The CCM Serial Interface protocol is explained in Chapter 5 of this manual. When the CCM module (CCM device) is configured as an RTU slave, it uses the protocol as explained in this chapter.

RTU protocol is a query-response protocol used for communication between the CCM device and a host computer which is capable of communicating using RTU protocol. The host computer is the master device and it transmits a query to a RTU slave which responds to the master. The CCM device, as an RTU slave, cannot query; it can only respond to the master.

The RTU data transferred consists of 8-bit binary characters with or without parity. No control characters are used to control the flow of data, there is, however, an error check (Cyclic Redundancy Check) included as the final field of each query and response to ensure accurate transmission of data.

Message Format

The general formats for RTU message transfers are shown below.

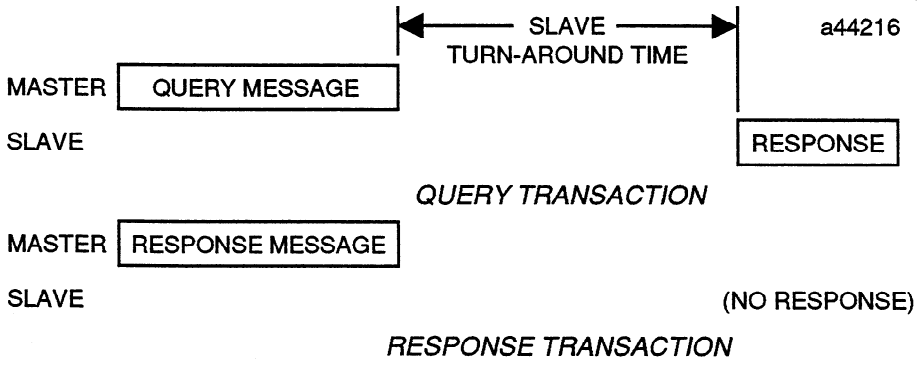


Figure 6-1. RTU Message Transfers

A distinction is made between two communicating devices. The device which initiates a data transfer is called the master and the other device is called the slave. The Series 5 device can only be a RTU slave.

The master device begins a data transfer by sending a query request message. A slave completes that data transfer by sending a response message if the master sent a query message addressed to it. No response message is sent when the master sends a broadcast request. The time between the end of a query and the beginning of the response to that query is called the slave turn-around time.

Message Types

The RTU protocol has four message types; query, normal response, error response, and broadcast.

Query

The master sends a message address to a single slave.

Normal Response

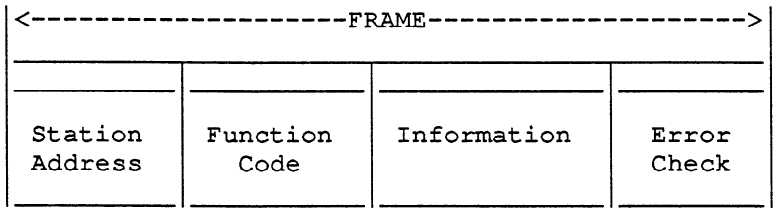
After the slave performs the function requested by the query, it sends back a normal response for that function. This indicates that the request was successful.

Error Response

The slave receives the query, but for some reason it cannot perform the requested function. The slave sends back an error response which indicates the reason the request could not be processed. (No error message will be sent for certain types of errors. For more information see section, Communication Errors).

Message Fields

The message fields for a typical message are shown below.



Station Address

The station address is the address of the slave station selected for this data transfer. It is one byte in length and has a value from 1 to 90 inclusive. The address selects a slave station with that station address.

Function Code

The function code identifies the command being issued to the station. It is one byte in length and is defined for the values 0 to 255 as follows:

0	Illegal Function	
1	Read Output Table	
2	Read Input Table	
3	Read Registers *	* These two functions are identical.
4	Read Registers *	
5	Force Single Output	
6	Preset Single Register	
7	Read Exception Status	
8	Loopback Maintenance	
9- 14	Unsupported Function	

15	Force Multiple Outputs
16	Preset Multiple Registers
17	Report Device Type
18- 64	Unsupported Function
65	Read Output Override Table
66	Read Input Override Table
67	Read Scratch Pad Memory
68	Read User Logic
69	Write Output Override Table
70	Write Input Override Table
71	Write Scratch Pad Memory (See CAUTION below)
72-127	Unsupported Function
128-255	Reserved for Exception Responses

CAUTION

Extreme care must be used when writing to any Scratch Pad location. It is strongly recommended that you consult GE Fanuc Field Service before doing this.

Information Field

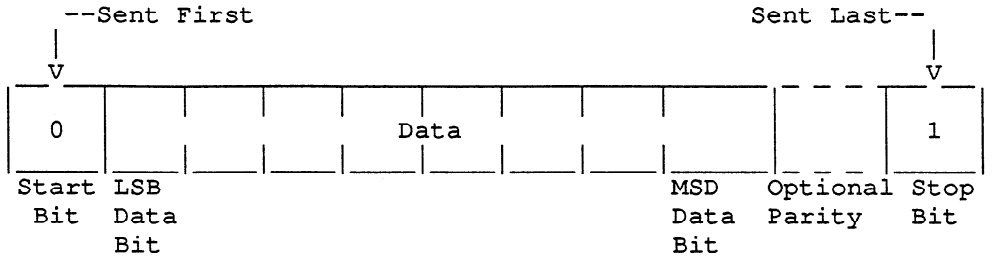
The information field contains all of the other information required to further specify or respond to a requested function. Detailed specification of the contents of the information field for each message type, query, normal response, and error response--and each function code is found in the section, Message Descriptions.

Error Check Field

The error check field is two bytes in length and contains a cyclic redundancy check (CRC-16) code. Its value is a function of the contents of the station address, function code, and information field. The details of generating the CRC-16 code are in the section, Cyclic Redundancy Check (CRC). Note that the information field is variable in length. In order to properly generate the CRC-16 code, the length of frame must be determined. See section, Calculating the Length of Frame, to calculate the length of a frame for each of the defined function codes.

Character Format

A message is sent as a series of characters. Each byte in a message is transmitted as a character. The illustration below shows the character format. A character consists of a start bit (0), eight data bits, an optional parity bit, and one stop bit (1). Between characters the line is held in the 1 state.



Message Termination

Each station monitors the time between characters. When a period of three character times elapses without the reception of a character, the end of a message is assumed. The reception of the next character is assumed to be the beginning of a new message.

The end of a frame occurs when the first of the following two events occurs:

- The number of characters received for the frame is equal to the calculated length of the frame.
- A length of 3 character times elapses without the reception of a character.

Timeout Usage

Timeouts are used on the serial link for error detection, error recovery, and to prevent the missing of the end of messages and message sequences. Note that although the module allows up to three character transmission times between each character in a message that it receives, there is no more than half a character time between each character in a message that the module transmits.

The slave turn-around times listed in Table 6-1 are the guaranteed maximum times for the communication module. In many cases the actual turn-around times will be much less.

Table 6-1. RTU Turn-Around Time

Description		RTU Turn-Around Time (Milliseconds)
Normal Responses		
Function Code	1	500
	2	500
	3	500
	4	500
	5	500
	6	500
	7	500
	8	500
	15	500
	16	500
	17	500
	65	500
	66	500
	67	500
	68	500
	69	500
	70	500
	71	500
	72	500
Error Responses		
Error Code	1	500
	2	500
	3	500
	4	500

Cyclic Redundancy Check (CRC)

The Cyclic Redundancy Check (CRC) is one of the most effective systems for checking errors. The CRC consists of 2 check characters generated at the transmitter and added at the end of the transmitted data characters. Using the same method, the receiver generates its own CRC for the incoming data and compares it to the CRC sent by the transmitter to ensure proper transmission.

A complete mathematic derivation for the CRC will not be given in this section. This information can be found in a number of texts on data communications. The essential steps which should be understood in calculating the CRC are as follows:

- The data bits which make up the message are multiplied by the number of bits in the CRC.
- The resulting product is then divided by the generating polynomial (using modulo 2 with no carries). The CRC is the remainder of this division.
- Disregard the quotient and add the remainder (CRC) to the data bits and transmit the message with CRC.
- The receiver then divides the message plus CRC by the generating polynomial and if the remainder is 0, the transmission was transmitted without error.

A generating polynomial is expressed algebraically as a string of terms in powers of X such as $X^3 + X^2 + X^0$ (or 1) which can in turn be expressed as the binary number 1101. A generating polynomial could be any length and contain any pattern of 1s and 0s as long as both the transmitter and receiver use the same value. For optimum error detection, however, certain standard generating polynomials have been developed. RTU protocol uses the polynomial $X^{16} + X^{15} + X^2 + 1$ which in binary is 1 1000 0000 0000 0101. The CRC this polynomial generates is known as CRC-16.

The discussion above can be implemented in hardware or software. One hardware implementation involves constructing a multi-section shift register based on the generating polynomial.

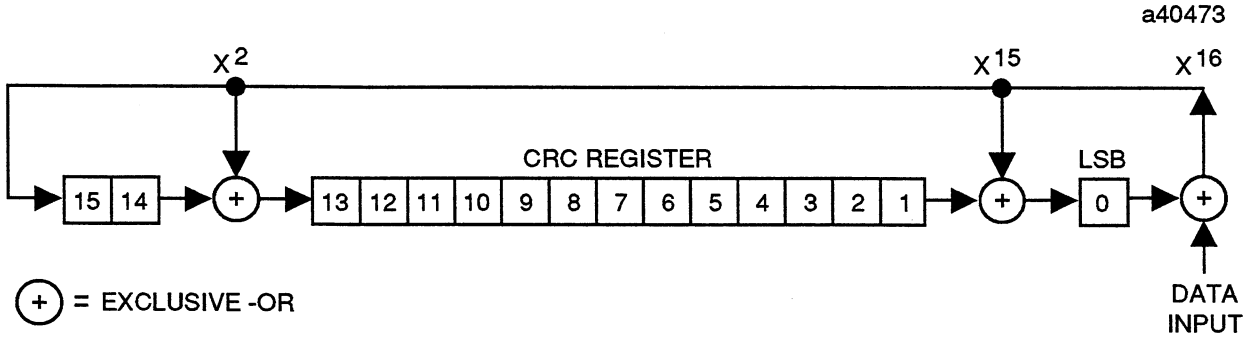


Figure 6-2. Cyclic Redundancy Check (CRC) Register

To generate the CRC, the message data bits are fed to the shift register one at a time. The CRC register contains a preset value. As each data bit is presented to the shift register, the bits are shifted to the right. The LSB is XORed with the data bit and the result is: XORed with the old contents of bit 1 (the result placed in bit 0), XORed with the old contents of bit 14 (and the result placed in bit 13), and finally, it is shifted into bit 15. This process is repeated until all data bits in a message have been processed. Software implementation of the CRC-16 is explained in the next section.

Calculating the CRC-16

The pseudo code for calculation of the CRC-16 is given below.

```
            Preset byte count for data to be sent.
            Initialize the 16-bit remainder (CRC) register to all ones.
            XOR the first 8-bit data byte with the high order byte of the
             16-bit CRC register.  The result is the current CRC.
INIT SHIFT  Initialize the shift counter to 0.
SHIFT       Shift the current CRC register 1 bit to the right.
            Increment shift count.
            Is the bit shifted out to the right (flag) a 1 or a 0?
              If it is a 1, XOR the generating polynomial with the current CRC.
              If it is a 0, continue.
            Is shift counter equal to 8?
              If NO, return to SHIFT.
              If YES, increment byte count.
            Is byte count greater than the data length?
              If NO, XOR the next 8-bit data byte with the current CRC
               and go to INIT SHIFT.
              If YES, add current CRC to end of data message for transmission
               and exit.
```

When the message is transmitted, the receiver will perform the same CRC operation on all the data bits and the transmitted CRC. If the information is received correctly the resulting remainder (receiver CRC) will be 0.

Example CRC-16 Calculation

The CCM device transmits the rightmost byte (of registers or discrete data) first. The first bit of the CRC-16 transmitted is the MSB. Therefore, in the example the MSB of the CRC polynomial is to the extreme right. The X^{16} term is dropped because it affects only the quotient (which is discarded) and not the remainder (the CRC characters). The generating polynomial is therefore 1010 0000 0000 0001. The remainder is initialized to all 1s.

As an example we will calculate the CRC-16 for RTU message, Read Exception Status (07). The message format is as follows:

Address	Func	CRC-16
01	07	

In this example we are querying device number 1 (address 01). We need to know the amount of data to be transmitted and this information can be found for every message type in the section, Calculating the Length of Frame. For this message the data length is 2 bytes.

Transmitter CRC-16 Algorithm	MSB LSB	Flag
Initial Remainder	1111 1111 1111 1111	
XOR 1st data byte	0000 0000 0000 0001	
Current CRC	1111 1111 1111 1110	
Shift 1	0111 1111 1111 1111	0
Shift 2	0011 1111 1111 1111	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1001 1111 1111 1110	
Shift 3	0100 1111 1111 1111	0
Shift 4	0010 0111 1111 1111	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1000 0111 1111 1110	
Shift 5	0100 0011 1111 1111	0
Shift 6	0010 0001 1111 1111	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1000 0001 1111 1110	
Shift 7	0100 0000 1111 1111	0
Shift 8	0010 0000 0111 1111	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1000 0000 0111 1110	
XOR 2nd data byte	0000 0000 0000 0111	
Current CRC	1000 0000 0111 1001	
Shift 1	0100 0000 0011 1100	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1110 0000 0011 1101	
Shift 2	0111 0000 0001 1110	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1101 0000 0001 1111	
Shift 3	0110 1000 0000 1111	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1100 1000 0000 1110	
Shift 4	0110 0100 0000 0111	0
Shift 5	0011 0010 0000 0011	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1001 0010 0000 0010	
Shift 6	0100 1001 0000 0001	0
Shift 7	0010 0100 1000 0000	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Current CRC	1000 0100 1000 0001	
Shift 8	0100 0010 0100 0000	1
XOR Gen. Polynomial	1010 0000 0000 0001	
Transmitted CRC	1110 0010 0100 0001	
	E 2 4 1	

Receiver * CRC-16 Algorithm	MSB LSB	Flag
Rcvr CRC after data	1110 0010 0100 0001	
XOR 1st byte Tms CRC	0000 0000 0100 0001	
Current CRC	1110 0010 0000 0000	
Shift 1	0111 0001 0000 0000	0
Shift 2	0011 1000 1000 0000	0
Shift 3	0001 1100 0100 0000	0
Shift 4	0000 1110 0010 0000	0
Shift 5	0000 0111 0001 0000	0
Shift 6	0000 0011 1000 1000	0
Shift 7	0000 0001 1100 0100	0
Shift 8	0000 0000 1110 0010	0
XOR 2nd byte tms CRC	0000 0000 1110 0010	
Current CRC	0000 0000 0000 0000	
Shift 1-8 yields	0000 0000 0000 0000	
	All Zeroes for Receiver	
	Final CRC-16 indicates	
	transmission correct!	

Example Message: Refer to the example of a transmitted message shown on the following page.

*As stated before, the receiver processes incoming data through the same CRC algorithm as the transmitter. The example for the receiver starts at the point after all the data bits but not the transmitted CRC have been received correctly. Therefore, the receiver CRC should be equal to the transmitted CRC

at this point. When this occurs, the output of the CRC algorithm will be zero indicating that the transmission is correct.

The transmitted message with CRC would then be:

```
1110 0010   0100 0001   0000 0111   0000 0001
  E    2      4    1      0    7      0    1
          Order of transmission --->
|                                           |
Transmitted last                  Transmitted first
```

Calculating the Length of Frame

To generate the CRC-16 for any message, the message length must be known. The length for all types of messages can be determined from the table below.

Table 6-2. RTU Message Length

Function Code and Name		Query Message Length Less CRC Code	Response Message Length Less CRC Code
0		Not Defined	Not Defined
1	Read Output Table	6	3 + 3rd byte *
2	Read Input Table	6	3 + 3rd byte *
3	Read Registers	6	3 + 3rd byte *
4	Read Registers	6	3 + 3rd byte *
5	Force Single Output	6	6
6	Preset Single Register	6	6
7	Read Exception Status	2	3
8	Loopback/Maintenance	6	6
9- 14		Not Defined	Not Defined
15	Force Multiple Outputs	7 + 7th byte *	6
16	Preset Multiple Registers	7 + 7th byte *	6
17	Report Device Type	2	3 + 3rd byte
18- 64		Not Defined	Not Defined
65	Read Output Override Table	6	3 + 3rd byte *
66	Read Input Override Table	6	3 + 3rd byte *
67	Read Scratch Pad Memory	6	3 + 3rd byte *
68	Read User Logic	6	3 + 3rd byte *
69	Write Output Override Table	7 + 7th byte *	6
70	Write Input Override Table	7 + 7th byte *	6
71	Write Scratch Pad Memory	7 + 7th byte *	6
72-127		Not Defined	Not Defined
128-255		Not Defined	3

* The value of this byte is the number of bytes contained in the data being transmitted.

Table Addresses

Table 6-3. RTU Table Addresses

Table Name	Range	RTU Start Address (Point, Register,Byte	Override Table Start Point Adr.
Registers	1 to 16383 (16K)	0 to 16383	N/A
	1 to 4096 (4K)	0 to 4095	N/A
Inputs			
I1 +	I1 + 1 to I1 + 1024	0 to 1023	0 to 1023
I2 +	I2 + 1 to I2 + 1024	1024 to 2047	1024 to 2047
I	I1 to 1024	2048 to 3071	2048 to 3071
I1 -	I1 - 1 to I1 - 512	3072 to 3583	N/A
Outputs			
O1 +	O1 + 1 to O1 + 1024	0 to 1023	0 to 1023
O2 +	O2 + 1 to O2 + 1024	1024 to 2047	1024 to 2047
O	O1 to O1024	2048 to 3071	2048 to 3071
O1 -	O1 - 1 to O1 - 512	3072 to 3583	3072 to 3583
O2 -	O2 - 1 to O2 - 1024	4096 to 5119	4096 to 5119
Scratch Pad	0 to 900H	0 to 900	N/A
User Logic *	0 to 16127 (16K)	0 to 16127	N/A
	0 to 3839 (4K)	0 to 3839	N/A

* The upper 256 words of the user logic memory type contains program and setup parameters, and is not available for program use.

Message Descriptions

The following pages explain the format and fields for each RTU message.

MESSAGE (01): **READ OUTPUT TABLE**

FORMAT:

Query

Address	Func 01	Starting Pt. Number		Number of Points		Error Check	
		Hi	Lo	Hi	Lo		

Normal Response

Address	Func 01	Byte Count	Data			Error Check	
					*		

* Data Length may vary

QUERY:

- The function code is 01.
- The starting point number is two bytes in length and may be any value less than the highest output point number available in the attached Series Five CPU. The starting point number is equal to one less than the number of the first output point returned in the normal response to this request. Refer to Table 6-3 for the output point mapping.
- The number of points value is two bytes in length. It specifies the number of output points returned in the normal response. The sum of the starting point value and the number of points value must be less than or equal to the highest output point number available in the attached Series Five CPU. The high order byte of the starting point number and number of bytes fields is sent as the first byte. The low order byte is the second byte in each of these fields.

RESPONSE:

- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the normal response following the byte count and preceeding the error check.
- The data field of the normal response is packed output status data. Each byte contains 8 output point values. The least significant bit (LSB) of the first byte contains the value of the output point whose number is equal to the starting point number plus one. The values of the output points are ordered by number starting with the LSB of the first byte of the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of 8, then the last data byte contains zeros in one to seven of its highest order bits.

MESSAGE (02): **READ INPUT TABLE**

FORMAT:

Query

Address	Func 02	Starting Pt. Number		Number of Points		Error Check	
		Hi	Lo	Hi	Lo		

Normal Response

Address	Func 02	Byte Count	Data			Error Check	
				*			

* Data Length May Vary

QUERY:

- The function code is 02.
- The starting point number is two bytes in length and may be any value less than the highest input point number available in the attached Series Five CPU. The starting point number is equal to one less than the number of the first input point returned in the normal response to this request.
- The number of points value is two bytes in length. It specifies the number of input points returned in the normal response. The sum of the starting point value and the number of points value must be less than or equal to the highest input point number available in the attached Series Five CPU. The high order byte of the starting point number and number of bytes fields is sent as the first byte. The low order byte is the second byte in each of these fields. Refer to Table 6-3 for the input point mapping.

RESPONSE:

- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the normal response following the byte count and preceeding the error check.
- The data field of the normal response is packed input status data. Each byte contains 8 input point values. The least significant bit (LSB) of the first byte contains the value of the input point whose number is equal to the starting point number plus one. The values of the input points are ordered by number starting with the LSB of the first byte of the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of 8, then the last data byte contains zeros in one to seven of its highest order bits.

MESSAGE (03, 04): **READ REGISTERS**

FORMAT:

Query

Address	Func 03 or 04	Starting Register No.		No. of Registers		Error Check	
		Hi	Lo	Hi	Lo		

Normal Response

Address	Func 03 or 04	Byte Count	Data					Error Check
			First Register				*	
			Hi	Lo	Hi	Lo		

* Data Length may vary

QUERY:

- The function code is equal to either 3 or 4.
- The starting register number is two bytes in length. The starting register number may be any value less than the highest register number available in the attached Series Five CPU. It is equal to one less than the number of the first register returned in the normal response to this request.
- The number of registers value is two bytes in length. It must contain a value from 1 to 125 inclusive. The sum of the starting register value and the number of registers value must be less than or equal to the highest register number available in the attached Series Five CPU. The high order byte of the starting register number and number of registers fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of these fields.

RESPONSE:

- The byte count is a binary number from 2 to 250 inclusive. It is the number of bytes in the normal response following the byte count and preceding the error check. Note that the byte count is equal to two times the number of registers returned in the response. A maximum of 250 bytes (125) registers is set so that the entire response can fit into one 256 byte data block.
- The registers are returned in the data field in order of number with the lowest number register in the first two bytes and the highest number register in the last two bytes of the datafield. The number of the first register in the data field is equal to the starting register number plus one. The high order byte is sent before the low order byte of each register.

MESSAGE (05): **FORCE SINGLE OUTPUT**

FORMAT:

Query

Address	Func 05	Point Number		Data		Error Check	
					OOH		
		Hi	Lo	Hi	Lo		

Normal Response

Address	Func 05	Point Number		Data		Error Check	
					OOH		
		Hi	Lo	Hi	Lo		

QUERY:

- The function code is equal to 05.
- The point number field is two bytes in length. It may be any value less than the highest output point number available in the attached Series Five CPU. It is equal to one less than the number of the output point to be forced on or off. Refer to Table 6-3 for the output point mapping.
- The first byte of the data field is equal to either 0 or 255 (FFH). The output point specified in the point number field is to be forced off if the first data field byte is equal to 0. It is to be forced on if the first data field byte is equal to 255 (FFH). The second byte of the data field is always equal to zero.

RESPONSE:

- The normal response to a force single output query is identical to the query.

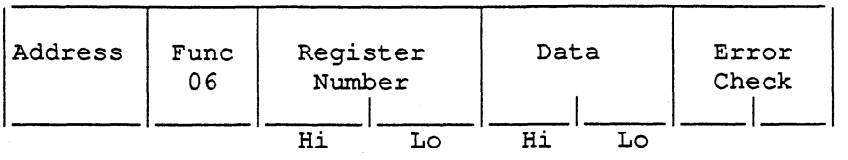
NOTE

The force single output request is not an output override command. The output specified in this request is ensured to be forced to the value specified only at the beginning of one sweep of the Series Five PLC user logic.

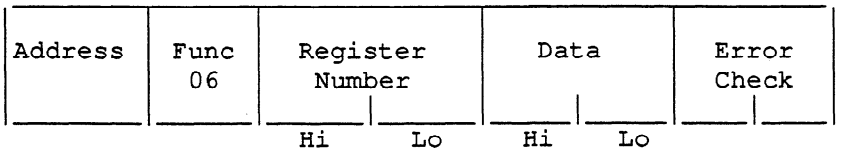
MESSAGE (06): **PRESET SINGLE REGISTER**

FORMAT:

Query



Normal Response



QUERY:

- The function code is equal to 06.
- The register number field is two bytes in length. It may be any value less than the highest register available in the attached Series Five CPU. It is equal to one less than the number of the register to be preset.
- The data field is two bytes in length and contains the value that the register specified by the register number field is to be preset to. The first byte in the data field contains the high order byte of the preset value. The second byte in the data field contains the low order byte.

RESPONSE:

- The normal response to a preset single register query is identical to the query.

MESSAGE (07): **READ EXCEPTION STATUS**

FORMAT:

Query

Address	Func 07	Error Check

Normal Response

Address	Func 07	Data	Error Check

QUERY:

This query is a short form of request for the purpose of reading the first eight output points.

- An address of zero is not allowed as this cannot be a broadcast request.
- The function code is equal to 07.

RESPONSE:

- The data field of the normal response is one byte in length and contains the states of output points O1 + 0001 through O1 + 0008. The output states are packed in order of number with output point one's state in the least significant bit and output point eight's state in the most significant bit.

MESSAGE (08): **LOOPBACK/MAINTENANCE (GENERAL)**

FORMAT:

Query

Address	Func 08	Diagnostic Code 0, 1, or 4		Data		Error Check	
				DATA1	DATA2		

Normal Response

Address	Func 08	Diagnostic Code 0, 1, or 4		Data		Error Check	
				DATA1	DATA2		
		Hi	Lo	Hi	Lo		

QUERY:

- The function code is equal to 8.
- The diagnostic code is two bytes in length. The high order byte of the diagnostic code is the first byte sent in the diagnostic code field. The low order byte is the second byte sent. The loopback/maintenance command is defined only for the diagnostic code equal to 0, 1, or 4. All other diagnostic codes are reserved.
- The data field is two bytes in length. The contents of the two data bytes are defined by the value of the diagnostic code.

RESPONSE:

- See descriptions for individual diagnostic codes.

DIAGNOSTIC CODE (00): **Return Query Data (Loopback/Maintenance)**

- A loopback/maintenance query with a diagnostic code equal to 0 is called a return query data request.
- The values of the two data field bytes in the query are arbitrary.
- The normal response is identical to the query.
- The values of the data bytes in the response are equal to the values sent in the query.

DIAGNOSTIC CODE (01): **Initiate Communication Restart (Loopback/Maintenance)**

A loopback/maintenance request (query or broadcast) with a diagnostic code equal to 1 is called an Initiate Communication Restart request.

- This request disables the listen-only mode (enables responses to be sent when queries are received so that communications can be restarted).
- The value of the first byte of the data field (DATA1) must be 0 or FF. Any other value will cause an error response to be sent. The value of the second byte of the data field (DATA2) is always equal to 0.
- The normal response to an Initiate Communication Restart query is identical to the query.

DIAGNOSTIC CODE (04): **Force Listen-Only Mode (Loopback/Maintenance)**

A loopback/maintenance request (query or broadcast) with a diagnostic code equal to 4 is called a Force Listen-Only Mode request.

- After receiving a Force Listen-Only mode request, the CCM device will go into the listen-only mode and will not send either normal or error responses to any queries. The listen-only mode is disabled when the CCM device receives an Initiate Communication Restart request, when the CCM device is powered up, or when switched off-line to on-line.
- Both bytes in the data field of a Force Listen-Only Mode request are equal to 0. The CCM device never sends a response to a Force Listen-Only Mode request.

NOTE

Upon power up, the CCM device disables the listen-only mode and is configured to continue sending responses to queries.

MESSAGE (15): **FORCE MULTIPLE OUTPUTS**

FORMAT:

Query

Address	Func 15	Starting Point No.	No. of Points	Byte Count	Data	Error Check
					*	

* Data Length may vary

Normal Response

Address	Func 15	Starting Point No.	Number of Points	Error Check

QUERY:

- The value of the function code is 15.
- The starting point number is two bytes in length and may be any value less than the highest output point number available in the attached Series Five CPU. The starting point number is equal to one less than the number of the first output point forced by this request. Refer to Table 6-3 for the output point mapping.
- The number of points value is two bytes in length. The sum of the starting point number and the number of points value must be less than or equal to the highest output point number available in the attached Series Five CPU. The high order byte of the starting point number and number of bytes fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of these fields.
- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the data field of the force multiple outputs request.
- The data field is packed data containing the values that the outputs specified by the starting point number and the number of points fields are to be forced to. Each byte in the data field contains the values that eight output points are to be forced to. The least significant bit (LSB) of the first byte contains the value that the output point whose number is equal to the starting point number plus one is to be forced to. The values for the output points are ordered by number starting with the LSB of the first byte of the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of 8, then the last data byte contains zeros in one to seven of its highest order bits.

RESPONSE:

- The description of the fields in the response are covered in the query description.

NOTE

The force multiple outputs request is not an output override command. The outputs specified in this request are ensured to be forced to the values specified only at the beginning of one sweep of the Series Five PLC user logic.

MESSAGE (16): **PRESET MULTIPLE REGISTERS**

FORMAT:

Query

Address	Func 16	Starting Register Number	No. of Registers	Byte Count	Data	Error Check
					*	

* Data Length may vary

Normal Response

Address	Func 16	Starting Register Number	Number of Registers	Error Check

QUERY:

- The value of the function code is 16.
- The starting register number is two bytes in length. The starting register number may be any value less than the highest register number available in the attached Series Five CPU. It is equal to one less than the number of the first register preset by this request.
- The number of registers value is two bytes in length. It must contain a value from 1 to 125 inclusive. The sum of the starting register number and the number of registers value must be less than or equal to the highest register number available in the attached Series Five CPU. The high order byte of the starting register number and number of registers fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of these fields.
- The byte count field is one byte in length. It is a binary number from 2 to 250 inclusive. It is equal to the number of bytes in the data field of the preset multiple registers request. Note that the byte count is equal to twice the value of the number of registers.
- The registers are returned in the data field in order of number with the lowest number register in the first two bytes and the highest number register in the last two bytes of the data field. The number of the first register in the data field is equal to the starting register number plus one. The high order byte is sent before the low order byte of each register.

RESPONSE:

- The description of the fields in the response are covered in the query description.

MESSAGE (17): **REPORT DEVICE TYPE**

FORMAT:

Query

Address	Func 17	Error Check

Normal Response

Address	Func 17	Byte Count 5	Device Type 50	Slave Run Light	Data	Error Check

QUERY:

The Report Device Type query is sent by the master to a slave in order to learn what type of programmable control or other computer it is. All models of the Series Five PLC return a device type 50 when this request is received.

- An address of zero is not allowed as this cannot be a broadcast request.
- The function code is equal to 17.

RESPONSE:

- The byte count field is one byte in length and is equal to 5.
- The device type field is one byte in length and is equal to 50.
- The slave run light field is one byte in length. The slave run light byte is equal to OFFH if the Series Five CPU is running. It is equal to 0 if the Series Five CPU is not running.
- The data field contains three bytes.

The first byte is called the system configuration byte and is shown below. Bits 4 and 5 indicate how many registers the attached Series Five CPU contains. Bits 1, 2, 3, 6, 7 and 8 are reserved for future use and are equal to 0.

The second data byte specifies the size of the attached Series Five PLC user logic memory. This value will be 4, 8, or 16 (decimal) which represents a logic memory size of 4K, 8K, or 16K. The third data byte will be 0.

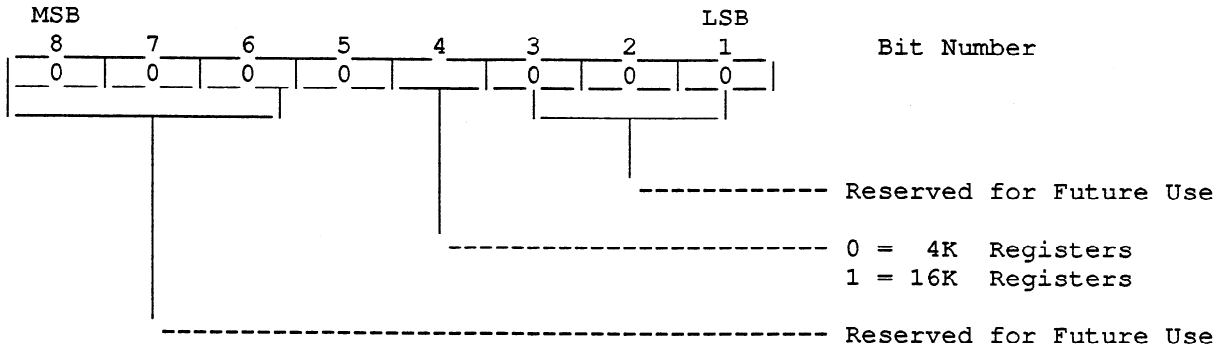


Figure 6-3. System Configuration (Byte 1)

MESSAGE (65): **READ OUTPUT OVERRIDE TABLE**

FORMAT:

Query

Address	Func 65	Starting Point No.	Number of Points	Error Check

Normal Response

Address	Func 65	Byte Count	Data	Error Check
			*	

* Data Length may vary

QUERY:

- The function code is equal to 65.
- The starting point number is two bytes in length and may be any value less than the highest output point number available in the attached Series Five CPU. The starting point number is equal to one less than the number of the first output point whose override status is returned in the normal response to this request. Refer to Table 6-3.
- The number of points value is two bytes in length. It specifies the number of output points whose override status are returned in the normal response. The sum of the starting point number and the number of points values must be less than or equal to the highest output point number available in the attached Series Five CPU. The high order byte of the starting point number and number of points fields is sent as the first byte in head of these fields. The low order byte is the second byte in each of these fields.

RESPONSE:

- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the data field of the normal response.
- The data field of the normal response is packed output override table data. Each byte contains the override status of eight output points. The least significant bit (LSB) of the first byte contains the override status of the output point whose number is equal to the starting point number plus one. The override status of the output points are ordered by number starting with the LSB of the first byte in the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of eight, then the last data byte contains zeros in one to seven of its highest order bits.

MESSAGE (66): **READ INPUT OVERRIDE TABLE**

FORMAT:

Query

Address	Func 66	Starting Point No.	Number of Points	Error Check

Normal Response

Address	Func 66	Byte Count	Data	Error Check
			*	

* Data Length may vary

QUERY:

- The function code is equal to 66.
- The starting point number is two bytes in length and may be any value less than the highest input point number available in the attached Series Five CPU. The starting point number is equal to one less than the number of the first input point whose override status is returned in the normal response to this request. Refer to Table 6-3 but notice that the I1 - table does not have an override table associated with it.
- The number of points value is two bytes in length. It specifies the number of input points whose override status are returned in the normal response. The sum of the starting point number and the number of points values must be less than or equal to the highest input point number available in the attached Series Five CPU. The high order byte of the starting point number and number of points fields is sent as the first byte in head of these fields. The low order byte is the second byte in each of these fields.

RESPONSE:

- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the data field of the normal response.
- The data field of the normal response is packed input override table data. Each byte contains the override status of eight input points. The least significant bit (LSB) of the first byte contains the override status of the input point whose number is equal to the starting point number plus one. The override status of the input points are ordered by number starting with the LSB of the first byte in the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of eight, then the last data byte contains zeros in one to seven of its highest order bits.

MESSAGE (67): **READ SCRATCH PAD MEMORY**

FORMAT:

Query

Address	Func 67	Starting Byte Number	Number Of Bytes	Error Check

Normal Response

Address	Func 67	Byte Count	Data	Error Check
			*	

* Data Length may vary

QUERY:

- The function code is equal to 67.
- The starting byte number is two bytes in length and may be any value less than or equal to the highest scratch pad memory address available in the attached Series Five CPU. The starting byte number is equal to the address of the first scratch pad memory byte returned in the normal response to this request. Refer to Table 6-3.
- The number of bytes value is two bytes in length. It specifies the number of scratch pad memory locations (bytes) returned in the normal response. The sum of the starting byte number and the number of bytes values must be less than two plus the highest scratch pad memory address available in the attached Series Five CPU. The high order byte of the starting byte number and number of bytes fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of the fields.

RESPONSE:

- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the data field of the normal response.
- The data field contains the contents of the scratch pad memory requested by the query. The scratch pad memory bytes are sent in order of address. The contents of the scratch pad memory byte whose address is equal to the starting byte number is sent in the first byte of the data field. The contents of the scratch pad memory byte whose address is equal to one less than the sum of the starting byte number and number of bytes values is sent in the last byte of the data field.

REMARKS:

Refer to the Series Five scratch pad memory map definition in Chapter 3, for more information.

MESSAGE (68): **READ USER LOGIC**

FORMAT:

Query

Address	Func 68	Starting Address	Number Of Words	Error Check

Normal Response

Address	Func 68	Byte Count	Data	Error Check
			*	

* Data Length may vary

QUERY:

- The function code is equal to 68.
- The starting address is two bytes in length and may be any value less than or equal to the highest user logic memory address available in the attached Series Five CPU. The starting address is equal to the address of the first user logic memory word returned in the normal response to this request.
- The number of words value is two bytes in length. It contains a value from 1 to 125. It specifies the number of user logic memory words returned in the normal response. The sum of the starting address and the number of words values must be less than two plus the highest user logic memory address available in the attached Series Five CPU. The high order byte of the starting address and number of words fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of these fields.

RESPONSE:

- The byte count is a binary number from 2 to 250. It is the number of bytes in the data field of the normal response.
- The contents of the user logic memory are returned in the data field in order of address. The lowest address contents are returned in the first two bytes and the highest address contents are returned in the last two bytes. The address of the first user logic memory contents returned in the data field is equal to the starting address. The high order byte of each user logic memory address is sent before the low order byte of that address.

MESSAGE (69): **WRITE OUTPUT OVERRIDE TABLE**

FORMAT:

Query

Address	Func 69	Starting Point No.	No. of Points	Byte Count	Data	Error Check
					*	

* Data Length may vary

Normal Response

Address	Func 69	Starting Point No.	Number Of Points	Error Check

QUERY:

- The value of the function code is 69.
- The starting point number is two bytes in length and may be any value less than the highest output point number available in the attached Series Five CPU. The starting point number is equal to one less than the number of the first output point whose override status is returned in the normal response to this request. Refer to Table 6-3.
- The number of points value is two bytes in length. It specifies the number of output points whose override status are returned in the normal response. The sum of the starting point number and the number of points values must be less than or equal to the highest output point number available in the attached Series Five CPU. The high order byte of the starting point number and number of points fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of these fields.
- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the data field of the normal response.
- The data field of the normal response is packed output override table data. Each byte contains the override status of eight output points. The least significant bit (LSB) of the first byte contains the override status of the output point whose number is equal to the starting point number plus one. The override status of the output points are ordered by number starting with the LSB of the first byte in the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of eight, then the last data byte contains zeros in one to seven of its highest order bits.

RESPONSE:

The description of the response fields are all covered in the description of the query fields.

MESSAGE (70): **WRITE INPUT OVERRIDE TABLE**

FORMAT:

Query

Address	Func 70	Starting Point No.	No. of Points	Byte Count	Data	Error Check
					*	

* Data Length may vary

Normal Response

Address	Func 70	Starting Point No.	Number Of Points	Error Check

QUERY:

- The function code is equal to 70 for write input override table.
- The starting point number is two bytes in length and may be any value less than the highest input point number available in the attached Series Five CPU. The starting point number is equal to one less than the number of the first input point whose override status is returned in the normal response to this request. Refer to Table 6-3 but notice that the I1 - table does not have an override table associated with it.
- The number of points value is two bytes in length. It specifies the number of input points whose override status are returned in the normal response. The sum of the starting point number and the number of points values must be less than or equal to the highest input point number available in the attached Series Five CPU. The high order byte of the starting point number and number of points fields is sent as the first byte in head of these fields. The low order byte is the second byte in each of these fields.
- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the data field of the normal response.
- The data field of the normal response is packed input override table data. Each byte contains the override status of eight input points. The least significant bit (LSB) of the first byte contains the override status of the input point whose number is equal to the starting point number plus one. The override status of the input points are ordered by number starting with the LSB of the first byte in the data field and ending with the most significant bit (MSB) of the last byte of the data field. If the number of points is not a multiple of eight, then the last data byte contains zeros in one to seven of its highest order bits.

RESPONSE:

The description of the response fields are covered in the description of the query fields.

MESSAGE (71): **WRITE SCRATCH PAD MEMORY**

FORMAT:

Query

Address	Func 71	Starting Byte No.	No. of Bytes	Byte Count	Data	Error Check
					*	

* Data Length may vary

Normal Response

Address	Func 71	Starting Byte No.	No. of Bytes	Error Check

QUERY:

- The value of the function code is 71.
- The starting byte number, number of bytes, byte count, and data fields are described in the read scratch pad memory.
- The starting byte number is two bytes in length and may be any value less than or equal to the highest scratch pad memory address available in the attached Series Five CPU. The starting byte number is equal to the address of the first scratch pad memory byte returned in the normal response to this request.
- The number of bytes value is two bytes in length. It specifies the number of scratch pad memory locations (bytes) returned in the normal response. The sum of the starting byte number and the number of bytes values must be less than two plus the highest scratch pad memory address available in the attached Series Five CPU. The high order byte of the starting byte number and number of bytes fields is sent as the first byte in each of these fields. The low order byte is the second byte in each of the fields.
- The byte count is a binary number from 1 to 256 (0 = 256). It is the number of bytes in the data field of the normal response.
- The data field contains the contents of the scratch pad memory requested by the query. The scratch pad memory bytes are sent in order of address. The contents of the scratch pad memory byte whose address is equal to the starting byte number is sent in the first byte of the data field. The contents of the scratch pad memory byte whose address is equal to one less than the sum of the starting byte number and number of bytes values is sent in the last byte of the data field.

RESPONSE:

The description of the response fields are covered in the query description.

REMARKS:

Refer to the Scratch pad memory mapping in Chapter 3 for more information.

CAUTION

Extreme care must be used when writing to any Scratch Pad location. It is strongly recommended that you consult GE Fanuc Field Service before doing this.

Communication Errors

Serial link communication errors are divided into three groups:

- Invalid Query Message
- Serial Link Timeouts
- Invalid Transaction

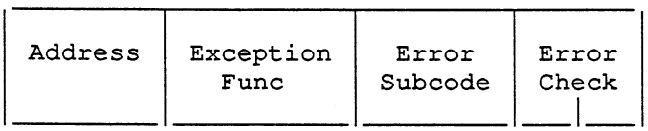
Invalid Query Message

If a query is received with a bad CRC, no response is sent.

When the communications module receives a query addressed to itself, but cannot process the query, it sends one of the following error responses:

	Subcode
• Invalid Function Code	(1)
• Invalid Address Field	(2)
• Invalid Data Field	(3)
• Query Processing Failure	(4)

The format for an error response to a query is as follows.



An address of 0 is not allowed as there is no response to a broadcast request. The exception function code is equal to the sum of the function code of the received query, plus 128. The error subcode is equal to 1, 2, 3, or 4. The value of the subcode indicates the reason that the properly received query could not be processed.

Invalid Function Code Error Response (1)

An error response with a subcode of 1 is called an invalid function code error response. This response is sent by a slave if it receives a query whose function code is not equal to 1 through 8, 15, 16, 17, or 65 through 71.

Invalid Address Error Response (2)

An error response with a subcode of 2 is called an invalid address error response. This error response is sent in the following cases:

1. The starting point number and number of points fields specify output status points or input status points that are not available in the attached Series Five CPU (returned for function codes 1, 2, 15, 65, 66, 69, 70).
2. The starting register number and number of registers fields specify registers that are not available in the attached Series Five CPU (returned for function codes 3, 4, 16).
3. The point number field specifies an output status point not available in the attached Series Five CPU (returned for function code 5).

4. The register number field specifies a register not available in the attached Series Five CPU (returned for function code 6).
5. The diagnostic code is not equal to 0, 1, or 4 (returned for function code 8).
6. The starting byte number and number of bytes fields specify a scratch pad memory address that is not available in the attached Series Five CPU (returned for function code 67).
7. The starting byte number and number of bytes fields specify a write to a scratch pad memory address other than addresses 0, 1, 60H thru 7FH, and 5CH thru 5FH (returned for function code 71).
8. The starting address and number of words fields specify a user logic memory address not available in the attached Series Five CPU (returned for function code 68).

Invalid Data Value Error Response (3)

An error response with a subcode of 3 is called an invalid data value error response. This response is sent in the following case:

The first byte of the data field is not equal to 0 or 255 (FFH) or the second byte of the data field is not equal to 0 for the force single output request (function code 5) or the initiate communication restart request (function code 8, diagnostic code 1).

Query Processing Failure Error Response (4)

An error response with a subcode of 4 is called a query processing failure response. This error response is sent by a CCM device if it properly receives a query but communication between the associated Series Five CPU and the CCM device fails.

Serial Link Timeout

The only cause for a RTU device to timeout is if an interruption to a data stream of 3 character times occurs while a message is being received. If this occurs the message is considered to have terminated and no response will be sent to the master. There are certain timing considerations due to the characteristics of the slave that should be taken into account by the master.

After sending a query message, the master should wait the length of the turn-around time before assuming that the slave did not respond to its request. See Table 6-1 for turn-around times using the various function codes.

Invalid Transactions

If an error occurs during transmission that does not fall into the category of an invalid query message or a serial link timeout, it is known as an invalid transaction. Types of errors causing an invalid transaction include:

- Bad CRC.
- The data length specified by the memory address field is longer than the data received.
- Framing or overrun errors.
- Parity errors.

If an error in this category occurs when a message is received by the CCM device, the CCM device does not return an error message. The CCM device treats the incoming message as though it was not intended for it.

Error Conditions

The 8-segment Light-Emitting-Diode (LED) display, located on the front panel of the CCM module, can be used to determine the cause of many problems in the communication network. When the CCM detects an error, corresponding LEDs will be turned ON until the next query starts.

Refer to Chapter 2, for the CCM module front panel layout and designation of the 8-segment diagnostic display. Table 6-4 below shows the LED diagnostic display patterns for the CCM module in RTU mode. The function of the PWR and OK LEDs are the same for RTU as in the CCM2 mode. The MSTR LED has no function in RTU mode.

Table 6-4. RTU Mode LED Diagnostic Display

Error Description	Led Display (° = Light OFF, • = Light ON)							
	PWR	OK	NAK	TOUT	ENQ	HDR	DATA	MSTR
Invalid Function Code (1)			•	°	•	°	°	
Invalid Address Field (2)			•	°	°	•	°	
Invalid Data Field (3)			•	°	°	°	•	
Query Processing Failure (4)			•	°	°	°	°	
Serial Link Timeout			°	•	°	°	°	
Parity, Override, Framing Error (Invalid Transaction)			°	•	•	°	°	
Too Much Data to Receive (Invalid Transaction)			°	•	°	•	°	
Bad CRC (Invalid Transaction)			°	•	°	°	•	

Chapter 7
Additional Protocols

GFK-0244

Overview

The Series Five® Communications Control Module (CCM) allows communications to and from the Series Five CPU via the CCM2 protocol. This permits communications with other CCM2 compatible systems such as Series One®, Series Three®, Series Six®, other Series Five CPUs and computers which have CCM2 drivers.

The CCM module also contains additional features which make it easy to communicate with non-CCM2 devices such as printers, bar code readers, and personal computers. It is not necessary to write a CCM2 driver in order to communicate with these types of devices. These additional features are collectively called ''non-CCM modes''. The 5-types of non-CCM modes are:

Mode 1 - Unformatted transmitting of registers.
Mode 2 - ''Bar code'' receiving (formatted with error check).*
Mode 3 - Unformatted transmitting and/or ''receiving to registers with CR LF terminator''.
Mode 4 - Unformatted transmitting and/or ''receiving with CR terminator''.
Mode 5 - Unformatted transmitting, and/or ''receiving with CR LF terminator'', and compare on reception.
(* Modes 3, 4, 5 may also be usable with bar code readers)

These non-CCM modes are initiated by setting the module configuration Dual-In-Line (DIP) switches for slave mode, with a CCM ID number above 90 (the normal limit for CCM slave devices). When a non-CCM mode is chosen, the normal CCM2 protocol is disabled. All of the above modes employ a user defined CPU register buffer as the source or destination of the data. Up to 8 Communication Control Modules (CCMs) may be used in the CPU's rack in these modes, as in the normal CCM mode.

Transmitting requires the use of the WR CCM instruction in the CPU user logic. The transmit source buffer is defined as part of the WR CCM instruction in user logic. The Logicmaster Five text entry mode makes entry of the ''message'' in the register table very easy. The special inputs (I1-81 through I1-208) used for normal CCM2 communications which indicate communications complete and error status are also functional in the non-CCM transmit modes.

Receiving is done transparently to the user program, and the receive buffer location is defined by special registers 4057-4064.

Up to 128 ASCII characters can be transmitted and received (depending on the mode), and transmission from the CCM module can be throttled either with the CTS hardware input line, or by transmitting XON / XOFF control characters to the CCM module.

Summary of Mode Types

Each of the additional mode types (Mode 1 through Mode 5) used to communicate with non-CCM2 devices are explained in this section. Unformatted transmission / receive (Modes 3, 4 and 5) use Carriage Return (CR) and Line Feed (LF) or Carriage Return (CR) terminators.

NOTE

CR = ''Carriage Return'' (0DH) LF = ''Line Feed'' (0AH)

Mode 1

Mode 1 is normally used to "dump" register information to a line printer, a display, or other "dumb" device. It can also be used to transmit data to a smart device like a personal computer (for example, a BASIC language INPUT statement can be used to read the data).

There is no automatic error checking done in this mode, so if data integrity is essential, some type of error checking must be built into the actual transmitted data. This is not normally required for most applications.

Data in the designated registers is transmitted exactly as it appears when the corresponding WR CCM instruction is executed in the CPUs user logic program. Control characters such as CR or LF can be transmitted in the same manner as printable characters, i.e. by entering their ASCII codes directly in the transmit buffer in the register table.

Mode 2

Mode 2 is used to read in data from bar code readers or other devices which utilize the mode 2 protocol as described later. The protocol used in this mode utilizes a block code error check, ACK/NAK responses, and automatic retries of data if the error check is unsuccessful. Some devices have this protocol already built in (you will need to consult the information which comes with your device). A driver for a computer can also be written which uses this protocol. This mode is normally used if your device already has this protocol, or if received data integrity is important and it is possible to modify the external transmitting device to use this protocol.

XON / XOFF and RTS /CTS flow control is not applicable in this mode.

Modes 3, 4, and 5

Modes 3,4 and 5 are used where 2-way communications is desired. These modes make communicating with an interactive operator terminal very easy. Modes 3,4 and 5 can also be used with a variety of bar code readers. These modes combine Mode 1 for transmitting, and a special receiving mode which requires almost no setup by the user. Data is transmitted without reformatting from the designated registers when the WR CCM command in user logic is executed. Data which is received by the serial port is accumulated until a CR LF sequence, or CR character is received (this depends on the mode). After the CR or CR LF terminator sequence is received, the accumulated data is stored in the receive buffer. The user logic program sets the first register in the receive buffer to 0 initially and after each received message has been processed. When the CCM module has stored the message in the buffer, it sets this register to 80H to indicate the receipt of the message.

Modes 3 and 4 differ only in the "terminator characters" which the CCM has been instructed to look for which define the end of the message (either CR LF for mode 3 or CR for mode 4).

Mode 5 requires that the received message be sent exactly the same twice in a row before the data is stored to the receive data buffer. The terminator sequence is CR LF for mode 5.

Configuring the Mode Type

The non-CCM modes are initiated by setting the CCM module configuration Dual-In-Line (DIP) switches for slave mode, with a CCM ID number above 90 (the normal limit for CCM slave devices). When a non-CCM mode is chosen, the normal CCM2 protocol is disabled. All 5-modes employ a user defined CPU register buffer as the source or destination of the data. Up to 8 CCM modules may be used in the CPU's rack in these modes, as in the normal CCM mode.

Mode 1 - Unformatted Transmitting

1. Set the CCM module configuration DIP Switch for the following (CCM ID=96D).

Table 7-1. Mode 1 - DIP Switch Setting

SWITCH NO. (SW1)	SWITCH POSITION			SWITCH NO. (SW2)	SWITCH POSITION
1	OFF			1	Set for desired
2	OFF			2	data rate.
3	OFF		CCM	3	
4	OFF		ID=96D	4	OFF = No Parity, ON=Odd
5	OFF			5	OFF
6	ON	32		6	OFF
7	ON	64	V	7	OFF
8	OFF			8	OFF
9	OFF			9	OFF

2. Set the receiving device for odd or no parity, 8 bits, 1 stop bit, desired data rate (to correspond with the CCM Master settings).
3. Store the desired ASCII characters in the register table (e.g., R2000-R2004). The Logicmaster Five text entry mode in the register table display is very useful for this.
4. Insert the following instruction in your CPU user logic program:

```
|          R0100
+--------[ BLOCK MOVE            ]----------------------(  )
|         3,1,1,1,5,2000,0
|
|       I0001  I1-87              R0100
+--------] [-----]/[------------[ WR CCM ]--------------(  )
|
```

This example uses R100-105 as the WR CCM command area, and R2000 as the start of the data buffer. Reserved input, I1-87, is used to ensure that the previous communications request has been completed before starting the new one. (Explanation of register assignment as follows, also refer to Table 7-7)

R0100 = Slot number of the CCM module (Slot 3 in this example)
R0101 = Always 1
R0102 = Always 1
R0103 = Always 1
R0104 = Number of registers to send (5 in this example, 10 characters)
R0105 = First register to send (start of the data buffer, R2000 in this example)

5. Fill the registers shown with the message to be transmitted. Maximum message size is 64 registers (128 characters).

R2000	=	4241H	"BA"
R2001	=	4443H	"DC"
R2002	=	3231H	"21"
R2003	=	1013H	"43"
R2004	=	1013H	LF CR

When displayed in Logicmaster Five text mode, and as printed on a printer or serial display the message will read:

```
"ABCD1234"
```

If entered in Logicmaster Five text mode, the message is entered from left to right as it would normally be read.

6. If the receiving device needs to halt transmission, an XOFF character should be sent to the CCM module, or CTS should be set to false by the receiving device.

Mode 2 - Bar Code Reader Receiving

The device to be connected to the CCM module must be able to transmit data using the following protocol. This can be a bar code reader, or any other device which can be programmed to send the following sequence :

```
        STX,  n BYTES OF DATA,  ETX,  BCC

Where   STX  = 02H
        ETX  = 03H
        n    = 125 bytes maximum
        BCC  = "exclusive or" of STX - ETX (inclusive) for CCM ID = 97D.
             = "exclusive or" of data - ETX (inclusive) for CCM ID = 98D.

        and parity = even,  character length = 7 bits.
```

If the external device sends the correct checksum (BCC) for the data, the CCM module will respond with an ACK character. If the BCC does not correlate with the data, a NAK will be sent to the transmitting device.

Data will be stored in the register area which starts at the register pointed to by R4057-4064 (depending on the slot in which the CCM module is installed -R4057 corresponds to slot 0).

The first register in the receive buffer area contains a status flag which is used to indicate to the user logic program that a new complete transmission has been received. If this register contains an 80H, a new transmission has been received, and the user logic program should use the information in the rest of the buffer area, or copy it to another area. If the register contains a 0, then a new transmission from the remote device has not been completed. The user logic should set this register to 0 after using the new data, or copying it for future evaluation.

XON / XOFF and RTS / CTS flow control is not used in this mode.

1. Set the CCM module configuration DIP switches for the following:

   CCM ID = 97D for BCC calculated from STX-ETX, or (* See Table 7-2, SW1)

   CCM ID = 98D for BCC calculated from DATA-ETX. (** See Table 7-2, SW1)

Table 7-2. Mode 2 - DIP Switch Setting

SWITCH NO. (SW1)	SWITCH POSITION		SWITCH NO. (SW2)	SWITCH POSITION
1	ON for 97D	CCM ID.	1	Set for desired
2	ON for 98D		2	data rate.
3	OFF		3	
4	OFF	ID = *	4	OFF = No Parity, ON=Odd
5	OFF	or	5	OFF
6	ON for both	ID = **	6	OFF
7	ON for both		7	OFF
8	OFF		8	OFF
9	OFF		9	OFF

2. Set receiving buffer pointer (R4057-4064) with the register number of the start of the receive buffer. Refer to Table 7-7

3. Await the receipt of a message.

   For example, if your CCM module is in slot 2, and you put 01234 in register 4059, then set R01234 to 0 from user logic:

   After a message is received, the register contents will be as follows:

   R01234 = 0080H (message received)
   R01235 = 4802H ("H" + STX)
   R01236 = 2049H (" I")
   R01237 = YY03H (YY=BCC, 03=ETX)

   The received message is "HI ".

4. The user logic program must use this message or copy it away, then set R01234 to 0 so that the CCM module will allow another message to be received.

Mode 3 - Unformatted Transmitting, and/or "Receiving with CR LF Terminator"

This mode is enabled by setting the DIP switches for CCM ID, to 102 decimal. Transmitting is accomplished exactly as for Mode 1 except for the DIP switch settings. To set up receiving in Mode 3, perform the following steps:

1. Set the CCM module configuration DIP switches for the following (CCM ID=102D).

Table 7-3. Mode 3 - DIP Switch Setting

SWITCH NO. (SW1)	SWITCH POSITION			SWITCH NO. (SW2)	SWITCH POSITION
1	OFF			1	Set for desired
2	ON	2		2	data rate.
3	ON	4	CCM	3	V
4	OFF		ID=102D	4	OFF = No Parity, ON=Odd
5	OFF			5	OFF
6	ON	32		6	OFF
7	ON	64	V	7	OFF
8	OFF			8	OFF
9	OFF			9	OFF

2. Set receiving buffer pointer (R4057-4064) with the register number of the start of the receive buffer. Refer to Table 7-7
3. Await the receipt of a message.

   For example, if your CCM module is in slot 2, and you put 01234 in register 4059, then set R01234 to 0 from user logic.

   After a message is received, the register contents will be as follows:

   R01234 = 0080H (message received)
   R01235 = 4948H ("HI")
   R01236 = 3520H ("5 ")
   R01237 = 1013H (LF CR)

   The received message is "HI 5".
4. The user logic program must use this message or copy it away, then set R01234 to 0 so that another message can be received.

Mode 4 - Unformatted Transmitting, and/or "Receiving with CR Terminator"

Set the CCM module configuration DIP switches for the following (CCM ID=101D).

Table 7-4. Mode 4 - DIP Switch Setting

SWITCH NO. (SW1)	SWITCH POSITION			SWITCH NO. (SW2)	SWITCH POSITION	
1	ON	1		1		Set for desired
2	OFF			2		data rate.
3	ON	4	CCM	3	V	
4	OFF		ID=101D	4	OFF	= No Parity, ON=Odd
5	OFF			5	OFF	
6	ON	32		6	OFF	
7	ON	64	V	7	OFF	
8	OFF			8	OFF	
9	OFF			9	OFF	

Transmitting is accomplished exactly the same as for mode 1 except for these DIP switch settings.

Receiving in mode 4 is identical to receiving in mode 3, except that the message is terminated with a CR rather than CR LF sequence.

Mode 5 - Unformatted Transmitting, and/or "Receiving with CR LF Terminator", and Compare on Reception

Operation in mode 5 is identical to mode 3 except that the DIP switches are set to 100D, and the received message from the external device must be received twice (with no differences) before it is stored, and the status register indicates 80H.

Set the CCM module configuration DIP switches for the following (CCM ID=100D).

Table 7-5. Mode 5 - DIP Switch Setting

SWITCH NO. (SW1)	SWITCH POSITION			SWITCH NO. (SW2)	SWITCH POSITION
1	OFF			1	Set for desired
2	OFF			2	data rate.
3	ON	4	CCM	3	
4	OFF		ID=101D	4	OFF = No Parity, ON=Odd
5	OFF			5	OFF
6	ON	32		6	OFF
7	ON	64		7	OFF
8	OFF			8	OFF
9	OFF			9	OFF

Additional Protocol Summary

Table 7-6. Additional Protocol Summary

Mode	T/R	CCM ID	RCV Term Sequence	BCC	Received Data Format
1	T	96D	N/A	N/A	N/A
2A	R	97D	N/A	STX-ETX	STATUS, STX, DATA1...DATA124, ETX, BCC (followed by ACK/NAK response)
2B	R	98D	N/A	DATA-ETX	STATUS, STX, DATA1...DATA124, ETX, BCC (followed by ACK/NAK response)
3	T/R	102D	CR-LF	N/A	STATUS, DATA1...DATA125, CR, LF
4	T/R	101D	CR	N/A	STATUS, DATA1...DATA126, CR
5	T/R	100D	CR-LF	N/A	STATUS, DATA1...DATA125, CR, LF (stored after being received the same twice).

- Transmit data format is user defined.
- All receive data is assumed to be ASCII coded. XON / XOFF control characters received will affect transmissions from the CCM module. Transmitted control characters may affect receiving device.
- CR = "Carriage Return" (0DH), LF = "Line Feed" (0AH)
- When transmitting, CTS is checked after RTS is activated by the CCM module. CTS = "false" will inhibit transmissions from the CCM module.
- When receiving, the first receive register should be set to 0 after reading the message. The CCM module will set the register to 80H when a new message has been stored in the buffer.
- All modes use 8 data bits, 1 start/stop bit, odd or no parity (except Mode 2 which is even parity).

Table 7-7. Setup and References

CPU Base Slot	RCV Buffer Pointer	
0	R4057	Communication
1	R4058	error numbers
2	R4059	will appear in
3	R4060	R4078 - 4080
4	R4061	if applicable.
5	R4062	
6	R4063	
7	R4064	

Transmit Setup

```
                RX
-------] [----[ WR CCM ]-----
RX      =   Slot Number of CCM Module
RX+1    =   1
RX+2    =   1
RX+3    =   1
RX+4    =   Number of Registers to Send
RX+5    =   Start of Transmit Buffer Reg.
```

Receive setup - set receive buffer pointer as required.

Appendix A
CCM Memory Types

An expanded listing of the Communications Control Module (CCM) memory types is provided in this appendix. This expanded listing includes the memory mapping for:

- Types 2 and 4 (Inputs/Byte) -- Table A-1
- Types 3 and 5 (Outputs/Byte) -- Table A-2

Table A-1. CCM Memory Types 2 and 4 (Inputs/Byte)

Target Address		Table Reference
Decimal	Hexadecimal	
1	1	I1+0001 - I1+0008
2	2	I1+0009 - I1+0016
3	3	I1+0017 - I1+0024
4	4	I1+0025 - I1+0032
5	5	I1+0033 - I1+0040
6	6	I1+0041 - I1+0048
7	7	I1+0049 - I1+0056
8	8	I1+0057 - I1+0064
9	9	I1+0065 - I1+0072
10	A	I1+0073 - I1+0080
11	B	I1+0081 - I1+0088
12	C	I1+0089 - I1+0096
13	D	I1+0097 - I1+0104
14	E	I1+0105 - I1+0112
15	F	I1+0113 - I1+0120
16	10	I1+0121 - I1+0128
17	11	I1+0129 - I1+0136
18	12	I1+0137 - I1+0144
19	13	I1+0145 - I1+0152
20	14	I1+0153 - I1+0160
21	15	I1+0161 - I1+0168
22	16	I1+0169 - I1+0176
23	17	I1+0177 - I1+0184
24	18	I1+0185 - I1+0192
25	19	I1+0193 - I1+0200
26	1A	I1+0201 - I1+0208
27	1B	I1+0209 - I1+0216
28	1C	I1+0217 - I1+0224
29	1D	I1+0225 - I1+0232
30	1E	I1+0233 - I1+0240
31	1F	I1+0241 - I1+0248
32	20	I1+0249 - I1+0256
33	21	I1+0257 - I1+0264
34	22	I1+0265 - I1+0272
35	23	I1+0273 - I1+0280
36	24	I1+0281 - I1+0288
37	25	I1+0289 - I1+0296
38	26	I1+0297 - I1+0304
39	27	I1+0305 - I1+0312
40	28	I1+0313 - I1+0320

Target Address		Table Reference
Decimal	Hexadecimal	
41	29	I1+0321 - I1+0328
42	2A	I1+0329 - I1+0336
43	2B	I1+0337 - I1+0344
44	2C	I1+0345 - I1+0352
45	2D	I1+0353 - I1+0360
46	2E	I1+0361 - I1+0368
47	2F	I1+0369 - I1+0376
48	30	I1+0377 - I1+0384
49	31	I1+0385 - I1+0392
50	32	I1+0393 - I1+0400
51	33	I1+0401 - I1+0408
52	34	I1+0409 - I1+0416
53	35	I1+0417 - I1+0424
54	36	I1+0425 - I1+0432
55	37	I1+0433 - I1+0440
56	38	I1+0441 - I1+0448
57	39	I1+0449 - I1+0456
58	3A	I1+0457 - I1+0464
59	3B	I1+0465 - I1+0472
60	3C	I1+0473 - I1+0480
61	3D	I1+0481 - I1+0488
62	3E	I1+0489 - I1+0496
63	3F	I1+0497 - I1+0504
64	40	I1+0505 - I1+0512
65	41	I1+0513 - I1+0520
66	42	I1+0521 - I1+0528
67	43	I1+0529 - I1+0536
68	44	I1+0537 - I1+0544
69	45	I1+0545 - I1+0552
70	46	I1+0553 - I1+0560
71	47	I1+0561 - I1+0568
72	48	I1+0569 - I1+0576
73	49	I1+0577 - I1+0584
74	4A	I1+0585 - I1+0592
75	4B	I1+0593 - I1+0600
76	4C	I1+0601 - I1+0608
77	4D	I1+0609 - I1+0616
78	4E	I1+0617 - I1+0624
79	4F	I1+0625 - I1+0632
80	50	I1+0633 - I1+0640

Target Address Decimal	Target Address Hexadecimal	Table Reference
81	51	I1+0641 - I1+0648
82	52	I1+0649 - I1+0656
83	53	I1+0657 - I1+0664
84	54	I1+0665 - I1+0672
85	55	I1+0673 - I1+0680
86	56	I1+0681 - I1+0688
87	57	I1+0689 - I1+0696
88	58	I1+0697 - I1+0704
89	59	I1+0705 - I1+0712
90	5A	I1+0713 - I1+0720
91	5B	I1+0721 - I1+0728
92	5C	I1+0729 - I1+0736
93	5D	I1+0737 - I1+0744
94	5E	I1+0745 - I1+0752
95	5F	I1+0753 - I1+0760
96	60	I1+0761 - I1+0768
97	61	I1+0769 - I1+0776
98	62	I1+0777 - I1+0784
99	63	I1+0785 - I1+0792
100	64	I1+0793 - I1+0800
101	65	I1+0801 - I1+0808
102	66	I1+0809 - I1+0816
103	67	I1+0817 - I1+0824
104	68	I1+0825 - I1+0832
105	69	I1+0833 - I1+0840
106	6A	I1+0841 - I1+0848
107	6B	I1+0849 - I1+0856
108	6C	I1+0857 - I1+0864
109	6D	I1+0865 - I1+0872
110	6E	I1+0873 - I1+0880
111	6F	I1+0881 - I1+0888
112	70	I1+0889 - I1+0896
113	71	I1+0897 - I1+0904
114	72	I1+0905 - I1+0912
115	73	I1+0913 - I1+0920
116	74	I1+0921 - I1+0928
117	75	I1+0929 - I1+0936
118	76	I1+0937 - I1+0944
119	77	I1+0945 - I1+0952
120	78	I1+0953 - I1+0960
121	79	I1+0961 - I1+0968
122	7A	I1+0969 - I1+0976
123	7B	I1+0977 - I1+0984
124	7C	I1+0985 - I1+0992
125	7D	I1+0993 - I1+1000
126	7E	I1+1001 - I1+1008
127	7F	I1+1009 - I1+1016
128	80	I1+1017 - I1+1024
129	81	I2+0001 - I2+0008
130	82	I2+0009 - I2+0016
131	83	I2+0017 - I2+0024
132	84	I2+0025 - I2+0032
133	85	I2+0033 - I2+0040
134	86	I2+0041 - I2+0048
135	87	I2+0049 - I2+0056
136	88	I2+0057 - I2+0064

Target Address Decimal	Target Address Hexadecimal	Table Reference
137	89	I2+0065 - I2+0072
138	8A	I2+0073 - I2+0080
139	8B	I2+0081 - I2+0088
140	8C	I2+0089 - I2+0096
141	8D	I2+0097 - I2+0104
142	8E	I2+0105 - I2+0112
143	8F	I2+0113 - I2+0120
144	90	I2+0121 - I2+0128
145	91	I2+0129 - I2+0136
146	92	I2+0137 - I2+0144
147	93	I2+0145 - I2+0152
148	94	I2+0153 - I2+0160
149	95	I2+0161 - I2+0168
150	96	I2+0169 - I2+0176
151	97	I2+0177 - I2+0184
152	98	I2+0185 - I2+0192
153	99	I2+0193 - I2+0200
154	9A	I2+0201 - I2+0208
155	9B	I2+0209 - I2+0216
156	9C	I2+0217 - I2+0224
157	9D	I2+0225 - I2+0232
158	9E	I2+0233 - I2+0240
159	9F	I2+0241 - I2+0248
160	A0	I2+0249 - I2+0256
161	A1	I2+0257 - I2+0264
162	A2	I2+0265 - I2+0272
163	A3	I2+0273 - I2+0280
164	A4	I2+0281 - I2+0288
165	A5	I2+0289 - I2+0296
166	A6	I2+0297 - I2+0304
167	A7	I2+0305 - I2+0312
168	A8	I2+0313 - I2+0320
169	A9	I2+0321 - I2+0328
170	AA	I2+0329 - I2+0336
171	AB	I2+0337 - I2+0344
172	AC	I2+0345 - I2+0352
173	AD	I2+0353 - I2+0360
174	AE	I2+0361 - I2+0368
175	AF	I2+0369 - I2+0376
176	B0	I2+0377 - I2+0384
177	B1	I2+0385 - I2+0392
178	B2	I2+0393 - I2+0400
179	B3	I2+0401 - I2+0408
180	B4	I2+0409 - I2+0416
181	B5	I2+0417 - I2+0424
182	B6	I2+0425 - I2+0432
183	B7	I2+0433 - I2+0440
184	B8	I2+0441 - I2+0448
185	B9	I2+0449 - I2+0456
186	BA	I2+0457 - I2+0464
187	BB	I2+0465 - I2+0472
188	BC	I2+0473 - I2+0480
189	BD	I2+0481 - I2+0488
190	BE	I2+0489 - I2+0496
191	BF	I2+0497 - I2+0504
192	C0	I2+0505 - I2+0512

Target Address Decimal	Target Address Hexadecimal	Table Reference
193	C1	I2+0513 - I2+0520
194	C2	I2+0521 - I2+0528
195	C3	I2+0529 - I2+0536
196	C4	I2+0537 - I2+0544
197	C5	I2+0545 - I2+0552
198	C6	I2+0553 - I2+0560
199	C7	I2+0561 - I2+0568
200	C8	I2+0569 - I2+0576
201	C9	I2+0577 - I2+0584
202	CA	I2+0585 - I2+0592
203	CB	I2+0593 - I2+0600
204	CC	I2+0601 - I2+0608
205	CD	I2+0609 - I2+0616
206	CE	I2+0617 - I2+0624
207	CF	I2+0625 - I2+0632
208	D0	I2+0633 - I2+0640
209	D1	I2+0641 - I2+0648
210	D2	I2+0649 - I2+0656
211	D3	I2+0657 - I2+0664
212	D4	I2+0665 - I2+0672
213	D5	I2+0673 - I2+0680
214	D6	I2+0681 - I2+0688
215	D7	I2+0689 - I2+0696
216	D8	I2+0697 - I2+0704
217	D9	I2+0705 - I2+0712
218	DA	I2+0713 - I2+0720
219	DB	I2+0721 - I2+0728
220	DC	I2+0729 - I2+0736
221	DD	I2+0737 - I2+0744
222	DE	I2+0745 - I2+0752
223	DF	I2+0753 - I2+0760
224	E0	I2+0761 - I2+0768
225	E1	I2+0769 - I2+0776
226	E2	I2+0777 - I2+0784
227	E3	I2+0785 - I2+0792
228	E4	I2+0793 - I2+0800
229	E5	I2+0801 - I2+0808
230	E6	I2+0809 - I2+0816
231	E7	I2+0817 - I2+0824
232	E8	I2+0825 - I2+0832
233	E9	I2+0833 - I2+0840
234	EA	I2+0841 - I2+0848
235	EB	I2+0849 - I2+0856
236	EC	I2+0857 - I2+0864
237	ED	I2+0865 - I2+0872
238	EE	I2+0873 - I2+0880
239	EF	I2+0881 - I2+0888
240	F0	I2+0889 - I2+0896
241	F1	I2+0897 - I2+0904
242	F2	I2+0905 - I2+0912
243	F3	I2+0913 - I2+0920
244	F4	I2+0921 - I2+0928
245	F5	I2+0929 - I2+0936
246	F6	I2+0937 - I2+0944
247	F7	I2+0945 - I2+0952
248	F8	I2+0953 - I2+0960

Target Address Decimal	Target Address Hexadecimal	Table Reference
249	F9	I2+0961 - I2+0968
250	FA	I2+0969 - I2+0976
251	FB	I2+0977 - I2+0984
252	FC	I2+0985 - I2+0992
253	FD	I2+0993 - I2+1000
254	FE	I2+1001 - I2+1008
255	FF	I2+1009 - I2+1016
256	100	I2+1017 - I2+1024
257	101	I0001 - I0008
258	102	I0009 - I0016
259	103	I0017 - I0024
260	104	I0025 - I0032
261	105	I0033 - I0040
262	106	I0041 - I0048
263	107	I0049 - I0056
264	108	I0057 - I0064
265	109	I0065 - I0072
266	10A	I0073 - I0080
267	10B	I0081 - I0088
268	10C	I0089 - I0096
269	10D	I0097 - I0104
270	10E	I0105 - I0112
271	10F	I0113 - I0120
272	110	I0121 - I0128
273	111	I0129 - I0136
274	112	I0137 - I0144
275	113	I0145 - I0152
276	114	I0153 - I0160
277	115	I0161 - I0168
278	116	I0169 - I0176
279	117	I0177 - I0184
280	118	I0185 - I0192
281	119	I0193 - I0200
282	11A	I0201 - I0208
283	11B	I0209 - I0216
284	11C	I0217 - I0224
285	11D	I0225 - I0232
286	11E	I0233 - I0240
287	11F	I0241 - I0248
288	120	I0249 - I0256
289	121	I0257 - I0264
290	122	I0265 - I0272
291	123	I0273 - I0280
292	124	I0281 - I0288
293	125	I0289 - I0296
294	126	I0297 - I0304
295	127	I0305 - I0312
296	128	I0313 - I0320
297	129	I0321 - I0328
298	12A	I0329 - I0336
299	12B	I0337 - I0344
300	12C	I0345 - I0352
301	12D	I0353 - I0360
302	12E	I0361 - I0368
303	12F	I0369 - I0376
304	130	I0377 - I0384

Target Address Decimal	Target Address Hexadecimal	Table Reference
305	131	I0385 - I0392
306	132	I0393 - I0400
307	133	I0401 - I0408
308	134	I0409 - I0416
309	135	I0417 - I0424
310	136	I0425 - I0432
311	137	I0433 - I0440
312	138	I0441 - I0448
313	139	I0449 - I0456
314	13A	I0457 - I0464
315	13B	I0465 - I0472
316	13C	I0473 - I0480
317	13D	I0481 - I0488
318	13E	I0489 - I0496
319	13F	I0497 - I0504
320	140	I0505 - I0512
321	141	I0513 - I0520
322	142	I0521 - I0528
323	143	I0529 - I0536
324	144	I0537 - I0544
325	145	I0545 - I0552
326	146	I0553 - I0560
327	147	I0561 - I0568
328	148	I0569 - I0576
329	149	I0577 - I0584
330	14A	I0585 - I0592
331	14B	I0593 - I0600
332	14C	I0601 - I0608
333	14D	I0609 - I0616
334	14E	I0617 - I0624
335	14F	I0625 - I0632
336	150	I0633 - I0640
337	151	I0641 - I0648
338	152	I0649 - I0656
339	153	I0657 - I0664
340	154	I0665 - I0672
341	155	I0673 - I0680
342	156	I0681 - I0688
343	157	I0689 - I0696
344	158	I0697 - I0704
345	159	I0705 - I0712
346	15A	I0713 - I0720
347	15B	I0721 - I0728
348	15C	I0729 - I0736
349	15D	I0737 - I0744
350	15E	I0745 - I0752
351	15F	I0753 - I0760
352	160	I0761 - I0768
353	161	I0769 - I0776
354	162	I0777 - I0784
355	163	I0785 - I0792
356	164	I0793 - I0800
357	165	I0801 - I0808
358	166	I0809 - I0816
359	167	I0817 - I0824
360	168	I0825 - I0832

Target Address Decimal	Target Address Hexadecimal	Table Reference
361	169	I0833 - I0840
362	16A	I0841 - I0848
363	16B	I0849 - I0856
364	16C	I0857 - I0864
365	16D	I0865 - I0872
366	16E	I0873 - I0880
367	16F	I0881 - I0888
368	170	I0889 - I0896
369	171	I0897 - I0904
370	172	I0905 - I0912
371	173	I0913 - I0920
372	174	I0921 - I0928
373	175	I0929 - I0936
374	176	I0937 - I0944
375	177	I0945 - I0952
376	178	I0953 - I0960
377	179	I0961 - I0968
378	17A	I0969 - I0976
379	17B	I0977 - I0984
380	17C	I0985 - I0992
381	17D	I0993 - I1000
382	17E	I1001 - I1008
383	17F	I1009 - I1016
384	180	I1017 - I1024
385	181	I1-0001 - I1-0008
386	182	I1-0009 - I1-0016
387	183	I1-0017 - I1-0024
388	184	I1-0025 - I1-0032
389	185	I1-0033 - I1-0040
390	186	I1-0041 - I1-0048
391	187	I1-0049 - I1-0056
392	188	I1-0057 - I1-0064
393	189	I1-0065 - I1-0072
394	18A	I1-0073 - I1-0080
395	18B	I1-0081 - I1-0088
396	18C	I1-0089 - I1-0096
397	18D	I1-0097 - I1-0104
398	18E	I1-0105 - I1-0112
399	18F	I1-0113 - I1-0120
400	190	I1-0121 - I1-0128
401	191	I1-0129 - I1-0136
402	192	I1-0137 - I1-0144
403	193	I1-0145 - I1-0152
404	194	I1-0153 - I1-0160
405	195	I1-0161 - I1-0168
406	196	I1-0169 - I1-0176
407	197	I1-0177 - I1-0184
408	198	I1-0185 - I1-0192
409	199	I1-0193 - I1-0200
410	19A	I1-0201 - I1-0208
411	19B	I1-0209 - I1-0216
412	19C	I1-0217 - I1-0224
413	19D	I1-0225 - I1-0232
414	19E	I1-0233 - I1-0240
415	19F	I1-0241 - I1-0248
416	1A0	I1-0249 - I1-0256

GFK-0244

Target Address Decimal	Target Address Hexadecimal	Table Reference
417	1A1	I1-0257 - I1-0264
418	1A2	I1-0265 - I1-0272
419	1A3	I1-0273 - I1-0280
420	1A4	I1-0281 - I1-0288
421	1A5	I1-0289 - I1-0296
422	1A6	I1-0297 - I1-0304
423	1A7	I1-0305 - I1-0312
424	1A8	I1-0313 - I1-0320
425	1A9	I1-0321 - I1-0328
426	1AA	I1-0329 - I1-0336
427	1AB	I1-0337 - I1-0344
428	1AC	I1-0345 - I1-0352
429	1AD	I1-0353 - I1-0360
430	1AE	I1-0361 - I1-0368
431	1AF	I1-0369 - I1-0376
432	1B0	I1-0377 - I1-0384

Target Address Decimal	Target Address Hexadecimal	Table Reference
433	1B1	I1-0385 - I1-0392
434	1B2	I1-0393 - I1-0400
435	1B3	I1-0401 - I1-0408
436	1B4	I1-0409 - I1-0416
437	1B5	I1-0417 - I1-0424
438	1B6	I1-0425 - I1-0432
439	1B7	I1-0433 - I1-0440
440	1B8	I1-0441 - I1-0448
441	1B9	I1-0449 - I1-0456
442	1BA	I1-0457 - I1-0464
443	1BB	I1-0465 - I1-0472
444	1BC	I1-0473 - I1-0480
445	1BD	I1-0481 - I1-0488
446	1BE	I1-0489 - I1-0496
447	1BF	I1-0497 - I1-0504
448	1C0	I1-0505 - I1-0512

Table A-2. CCM Memory Types 3 and 5 (Outputs/Byte)

Target Address Decimal	Target Address Hexadecimal	Table Reference
1	1	O1+0001 - O1+0008
2	2	O1+0009 - O1+0016
3	3	O1+0017 - O1+0024
4	4	O1+0025 - O1+0032
5	5	O1+0033 - O1+0040
6	6	O1+0041 - O1+0048
7	7	O1+0049 - O1+0056
8	8	O1+0057 - O1+0064
9	9	O1+0065 - O1+0072
10	A	O1+0073 - O1+0080
11	B	O1+0081 - O1+0088
12	C	O1+0089 - O1+0096
13	D	O1+0097 - O1+0104
14	E	O1+0105 - O1+0112
15	F	O1+0113 - O1+0120
16	10	O1+0121 - O1+0128
17	11	O1+0129 - O1+0136
18	12	O1+0137 - O1+0144
19	13	O1+0145 - O1+0152
20	14	O1+0153 - O1+0160
21	15	O1+0161 - O1+0168
22	16	O1+0169 - O1+0176
23	17	O1+0177 - O1+0184
24	18	O1+0185 - O1+0192
25	19	O1+0193 - O1+0200
26	1A	O1+0201 - O1+0208
27	1B	O1+0209 - O1+0216
28	1C	O1+0217 - O1+0224
29	1D	O1+0225 - O1+0232
30	1E	O1+0233 - O1+0240
31	1F	O1+0241 - O1+0248
32	20	O1+0249 - O1+0256
33	21	O1+0257 - O1+0264
34	22	O1+0265 - O1+0272
35	23	O1+0273 - O1+0280
36	24	O1+0281 - O1+0288
37	25	O1+0289 - O1+0296
38	26	O1+0297 - O1+0304
39	27	O1+0305 - O1+0312
40	28	O1+0313 - O1+0320
41	29	O1+0321 - O1+0328
42	2A	O1+0329 - O1+0336
43	2B	O1+0337 - O1+0344
44	2C	O1+0345 - O1+0352
45	2D	O1+0353 - O1+0360
46	2E	O1+0361 - O1+0368
47	2F	O1+0369 - O1+0376
48	30	O1+0377 - O1+0384

Target Address Decimal	Target Address Hexadecimal	Table Reference
49	31	O1+0385 - O1+0392
50	32	O1+0393 - O1+0400
51	33	O1+0401 - O1+0408
52	34	O1+0409 - O1+0416
53	35	O1+0417 - O1+0424
54	36	O1+0425 - O1+0432
55	37	O1+0433 - O1+0440
56	38	O1+0441 - O1+0448
57	39	O1+0449 - O1+0456
58	3A	O1+0457 - O1+0464
59	3B	O1+0465 - O1+0472
60	3C	O1+0473 - O1+0480
61	3D	O1+0481 - O1+0488
62	3E	O1+0489 - O1+0496
63	3F	O1+0497 - O1+0504
64	40	O1+0505 - O1+0512
65	41	O1+0513 - O1+0520
66	42	O1+0521 - O1+0528
67	43	O1+0529 - O1+0536
68	44	O1+0537 - O1+0544
69	45	O1+0545 - O1+0552
70	46	O1+0553 - O1+0560
71	47	O1+0561 - O1+0568
72	48	O1+0569 - O1+0576
73	49	O1+0577 - O1+0584
74	4A	O1+0585 - O1+0592
75	4B	O1+0593 - O1+0600
76	4C	O1+0601 - O1+0608
77	4D	O1+0609 - O1+0616
78	4E	O1+0617 - O1+0624
79	4F	O1+0625 - O1+0632
80	50	O1+0633 - O1+0640
81	51	O1+0641 - O1+0648
82	52	O1+0649 - O1+0656
83	53	O1+0657 - O1+0664
84	54	O1+0665 - O1+0672
85	55	O1+0673 - O1+0680
86	56	O1+0681 - O1+0688
87	57	O1+0689 - O1+0696
88	58	O1+0697 - O1+0704
89	59	O1+0705 - O1+0712
90	5A	O1+0713 - O1+0720
91	5B	O1+0721 - O1+0728
92	5C	O1+0729 - O1+0736
93	5D	O1+0737 - O1+0744
94	5E	O1+0745 - O1+0752
95	5F	O1+0753 - O1+0760
96	60	O1+0761 - O1+0768

Target Address Decimal	Target Address Hexadecimal	Table Reference
97	61	O1+0769 - O1+0776
98	62	O1+0777 - O1+0784
99	63	O1+0785 - O1+0792
100	64	O1+0793 - O1+0800
101	65	O1+0801 - O1+0808
102	66	O1+0809 - O1+0816
103	67	O1+0817 - O1+0824
104	68	O1+0825 - O1+0832
105	69	O1+0833 - O1+0840
106	6A	O1+0841 - O1+0848
107	6B	O1+0849 - O1+0856
108	6C	O1+0857 - O1+0864
109	6D	O1+0865 - O1+0872
110	6E	O1+0873 - O1+0880
111	6F	O1+0881 - O1+0888
112	70	O1+0889 - O1+0896
113	71	O1+0897 - O1+0904
114	72	O1+0905 - O1+0912
115	73	O1+0913 - O1+0920
116	74	O1+0921 - O1+0928
117	75	O1+0929 - O1+0936
118	76	O1+0937 - O1+0944
119	77	O1+0945 - O1+0952
120	78	O1+0953 - O1+0960
121	79	O1+0961 - O1+0968
122	7A	O1+0969 - O1+0976
123	7B	O1+0977 - O1+0984
124	7C	O1+0985 - O1+0992
125	7D	O1+0993 - O1+1000
126	7E	O1+1001 - O1+1008
127	7F	O1+1009 - O1+1016
128	80	O1+1017 - O1+1024
129	81	O2+0001 - O2+0008
130	82	O2+0009 - O2+0016
131	83	O2+0017 - O2+0024
132	84	O2+0025 - O2+0032
133	85	O2+0033 - O2+0040
134	86	O2+0041 - O2+0048
135	87	O2+0049 - O2+0056
136	88	O2+0057 - O2+0064
137	89	O2+0065 - O2+0072
138	8A	O2+0073 - O2+0080
139	8B	O2+0081 - O2+0088
140	8C	O2+0089 - O2+0096
141	8D	O2+0097 - O2+0104
142	8E	O2+0105 - O2+0112
143	8F	O2+0113 - O2+0120
144	90	O2+0121 - O2+0128
145	91	O2+0129 - O2+0136
146	92	O2+0137 - O2+0144
147	93	O2+0145 - O2+0152
148	94	O2+0153 - O2+0160
149	95	O2+0161 - O2+0168
150	96	O2+0169 - O2+0176
151	97	O2+0177 - O2+0184
152	98	O2+0185 - O2+0192

Target Address Decimal	Target Address Hexadecimal	Table Reference
153	99	O2+0193 - O2+0200
154	9A	O2+0201 - O2+0208
155	9B	O2+0209 - O2+0216
156	9C	O2+0217 - O2+0224
157	9D	O2+0225 - O2+0232
158	9E	O2+0233 - O2+0240
159	9F	O2+0241 - O2+0248
160	A0	O2+0249 - O2+0256
161	A1	O2+0257 - O2+0264
162	A2	O2+0265 - O2+0272
163	A3	O2+0273 - O2+0280
164	A4	O2+0281 - O2+0288
165	A5	O2+0289 - O2+0296
166	A6	O2+0297 - O2+0304
167	A7	O2+0305 - O2+0312
168	A8	O2+0313 - O2+0320
169	A9	O2+0321 - O2+0328
170	AA	O2+0329 - O2+0336
171	AB	O2+0337 - O2+0344
172	AC	O2+0345 - O2+0352
173	AD	O2+0353 - O2+0360
174	AE	O2+0361 - O2+0368
175	AF	O2+0369 - O2+0376
176	B0	O2+0377 - O2+0384
177	B1	O2+0385 - O2+0392
178	B2	O2+0393 - O2+0400
179	B3	O2+0401 - O2+0408
180	B4	O2+0409 - O2+0416
181	B5	O2+0417 - O2+0424
182	B6	O2+0425 - O2+0432
183	B7	O2+0433 - O2+0440
184	B8	O2+0441 - O2+0448
185	B9	O2+0449 - O2+0456
186	BA	O2+0457 - O2+0464
187	BB	O2+0465 - O2+0472
188	BC	O2+0473 - O2+0480
189	BD	O2+0481 - O2+0488
190	BE	O2+0489 - O2+0496
191	BF	O2+0497 - O2+0504
192	C0	O2+0505 - O2+0512
193	C1	O2+0513 - O2+0520
194	C2	O2+0521 - O2+0528
195	C3	O2+0529 - O2+0536
196	C4	O2+0537 - O2+0544
197	C5	O2+0545 - O2+0552
198	C6	O2+0553 - O2+0560
199	C7	O2+0561 - O2+0568
200	C8	O2+0569 - O2+0576
201	C9	O2+0577 - O2+0584
202	CA	O2+0585 - O2+0592
203	CB	O2+0593 - O2+0600
204	CC	O2+0601 - O2+0608
205	CD	O2+0609 - O2+0616
206	CE	O2+0617 - O2+0624
207	CF	O2+0625 - O2+0632
208	D0	O2+0633 - O2+0640

GFK-0244

Target Address Decimal	Target Address Hexadecimal	Table Reference
209	D1	O2+0641 - O2+0648
210	D2	O2+0649 - O2+0656
211	D3	O2+0657 - O2+0664
212	D4	O2+0665 - O2+0672
213	D5	O2+0673 - O2+0680
214	D6	O2+0681 - O2+0688
215	D7	O2+0689 - O2+0696
216	D8	O2+0697 - O2+0704
217	D9	O2+0705 - O2+0712
218	DA	O2+0713 - O2+0720
219	DB	O2+0721 - O2+0728
220	DC	O2+0729 - O2+0736
221	DD	O2+0737 - O2+0744
222	DE	O2+0745 - O2+0752
223	DF	O2+0753 - O2+0760
224	E0	O2+0761 - O2+0768
225	E1	O2+0769 - O2+0776
226	E2	O2+0777 - O2+0784
227	E3	O2+0785 - O2+0792
228	E4	O2+0793 - O2+0800
229	E5	O2+0801 - O2+0808
230	E6	O2+0809 - O2+0816
231	E7	O2+0817 - O2+0824
232	E8	O2+0825 - O2+0832
233	E9	O2+0833 - O2+0840
234	EA	O2+0841 - O2+0848
235	EB	O2+0849 - O2+0856
236	EC	O2+0857 - O2+0864
237	ED	O2+0865 - O2+0872
238	EE	O2+0873 - O2+0880
239	EF	O2+0881 - O2+0888
240	F0	O2+0889 - O2+0896
241	F1	O2+0897 - O2+0904
242	F2	O2+0905 - O2+0912
243	F3	O2+0913 - O2+0920
244	F4	O2+0921 - O2+0928
245	F5	O2+0929 - O2+0936
246	F6	O2+0937 - O2+0944
247	F7	O2+0945 - O2+0952
248	F8	O2+0953 - O2+0960
249	F9	O2+0961 - O2+0968
250	FA	O2+0969 - O2+0976
251	FB	O2+0977 - O2+0984
252	FC	O2+0985 - O2+0992
253	FD	O2+0993 - O2+1000
254	FE	O2+1001 - O2+1008
255	FF	O2+1009 - O2+1016
256	100	O2+1017 - O2+1024
257	101	O0001 - O0008
258	102	O0009 - O0016
259	103	O0017 - O0024
260	104	O0025 - O0032
261	105	O0033 - O0040
262	106	O0041 - O0048
263	107	O0049 - O0056
264	108	O0057 - O0064

Target Address Decimal	Target Address Hexadecimal	Table Reference
265	109	O0065 - O0072
266	10A	O0073 - O0080
267	10B	O0081 - O0088
268	10C	O0089 - O0096
269	10D	O0097 - O0104
270	10E	O0105 - O0112
271	10F	O0113 - O0120
272	110	O0121 - O0128
273	111	O0129 - O0136
274	112	O0137 - O0144
275	113	O0145 - O0152
276	114	O0153 - O0160
277	115	O0161 - O0168
278	116	O0169 - O0176
279	117	O0177 - O0184
280	118	O0185 - O0192
281	119	O0193 - O0200
282	11A	O0201 - O0208
283	11B	O0209 - O0216
284	11C	O0217 - O0224
285	11D	O0225 - O0232
286	11E	O0233 - O0240
287	11F	O0241 - O0248
288	120	O0249 - O0256
289	121	O0257 - O0264
290	122	O0265 - O0272
291	123	O0273 - O0280
292	124	O0281 - O0288
293	125	O0289 - O0296
294	126	O0297 - O0304
295	127	O0305 - O0312
296	128	O0313 - O0320
297	129	O0321 - O0328
298	12A	O0329 - O0336
299	12B	O0337 - O0344
300	12C	O0345 - O0352
301	12D	O0353 - O0360
302	12E	O0361 - O0368
303	12F	O0369 - O0376
304	130	O0377 - O0384
305	131	O0385 - O0392
306	132	O0393 - O0400
307	133	O0401 - O0408
308	134	O0409 - O0416
309	135	O0417 - O0424
310	136	O0425 - O0432
311	137	O0433 - O0440
312	138	O0441 - O0448
313	139	O0449 - O0456
314	13A	O0457 - O0464
315	13B	O0465 - O0472
316	13C	O0473 - O0480
317	13D	O0481 - O0488
318	13E	O0489 - O0496
319	13F	O0497 - O0504
320	140	O0505 - O0512

Target Address Decimal	Target Address Hexadecimal	Table Reference
321	141	O0513 - O0520
322	142	O0521 - O0528
323	143	O0529 - O0536
324	144	O0537 - O0544
325	145	O0545 - O0552
326	146	O0553 - O0560
327	147	O0561 - O0568
328	148	O0569 - O0576
329	149	O0577 - O0584
330	14A	O0585 - O0592
331	14B	O0593 - O0600
332	14C	O0601 - O0608
333	14D	O0609 - O0616
334	14E	O0617 - O0624
335	14F	O0625 - O0632
336	150	O0633 - O0640
337	151	O0641 - O0648
338	152	O0649 - O0656
339	153	O0657 - O0664
340	154	O0665 - O0672
341	155	O0673 - O0680
342	156	O0681 - O0688
343	157	O0689 - O0696
344	158	O0697 - O0704
345	159	O0705 - O0712
346	15A	O0713 - O0720
347	15B	O0721 - O0728
348	15C	O0729 - O0736
349	15D	O0737 - O0744
350	15E	O0745 - O0752
351	15F	O0753 - O0760
352	160	O0761 - O0768
353	161	O0769 - O0776
354	162	O0777 - O0784
355	163	O0785 - O0792
356	164	O0793 - O0800
357	165	O0801 - O0808
358	166	O0809 - O0816
359	167	O0817 - O0824
360	168	O0825 - O0832
361	169	O0833 - O0840
362	16A	O0841 - O0848
363	16B	O0849 - O0856
364	16C	O0857 - O0864
365	16D	O0865 - O0872
366	16E	O0873 - O0880
367	16F	O0881 - O0888
368	170	O0889 - O0896
369	171	O0897 - O0904
370	172	O0905 - O0912
371	173	O0913 - O0920
372	174	O0921 - O0928
373	175	O0929 - O0936
374	176	O0937 - O0944
375	177	O0945 - O0952
376	178	O0953 - O0960

Target Address Decimal	Target Address Hexadecimal	Table Reference
377	179	O0961 - O0968
378	17A	O0969 - O0976
379	17B	O0977 - O0984
380	17C	O0985 - O0992
381	17D	O0993 - O1000
382	17E	O1001 - O1008
383	17F	O1009 - O1016
384	180	O1017 - O1024
385	181	O1-0001 - O1-0008
386	182	O1-0009 - O1-0016
387	183	O1-0017 - O1-0024
388	184	O1-0025 - O1-0032
389	185	O1-0033 - O1-0040
390	186	O1-0041 - O1-0048
391	187	O1-0049 - O1-0056
392	188	O1-0057 - O1-0064
393	189	O1-0065 - O1-0072
394	18A	O1-0073 - O1-0080
395	18B	O1-0081 - O1-0088
396	18C	O1-0089 - O1-0096
397	18D	O1-0097 - O1-0104
398	18E	O1-0105 - O1-0112
399	18F	O1-0113 - O1-0120
400	190	O1-0121 - O1-0128
401	191	O1-0129 - O1-0136
402	192	O1-0137 - O1-0144
403	193	O1-0145 - O1-0152
404	194	O1-0153 - O1-0160
405	195	O1-0161 - O1-0168
406	196	O1-0169 - O1-0176
407	197	O1-0177 - O1-0184
408	198	O1-0185 - O1-0192
409	199	O1-0193 - O1-0200
410	19A	O1-0201 - O1-0208
411	19B	O1-0209 - O1-0216
412	19C	O1-0217 - O1-0224
413	19D	O1-0225 - O1-0232
414	19E	O1-0233 - O1-0240
415	19F	O1-0241 - O1-0248
416	1A0	O1-0249 - O1-0256
417	1A1	O1-0257 - O1-0264
418	1A2	O1-0265 - O1-0272
419	1A3	O1-0273 - O1-0280
420	1A4	O1-0281 - O1-0288
421	1A5	O1-0289 - O1-0296
422	1A6	O1-0297 - O1-0304
423	1A7	O1-0305 - O1-0312
424	1A8	O1-0313 - O1-0320
425	1A9	O1-0321 - O1-0328
426	1AA	O1-0329 - O1-0336
427	1AB	O1-0337 - O1-0344
428	1AC	O1-0345 - O1-0352
429	1AD	O1-0353 - O1-0360
430	1AE	O1-0361 - O1-0368
431	1AF	O1-0369 - O1-0376
432	1B0	O1-0377 - O1-0384

Target Address Decimal	Target Address Hexadecimal	Table Reference
433	1B1	O1-0385 - O1-0392
434	1B2	O1-0393 - O1-0400
435	1B3	O1-0401 - O1-0408
436	1B4	O1-0409 - O1-0416
437	1B5	O1-0417 - O1-0424
438	1B6	O1-0425 - O1-0432
439	1B7	O1-0433 - O1-0440
440	1B8	O1-0441 - O1-0448
441	1B9	O1-0449 - O1-0456
442	1BA	O1-0457 - O1-0464
443	1BB	O1-0465 - O1-0472
444	1BC	O1-0473 - O1-0480
445	1BD	O1-0481 - O1-0488
446	1BE	O1-0489 - O1-0496
447	1BF	O1-0497 - O1-0504
448	1C0	O1-0505 - O1-0512
449	1C1	O1-0513 - O1-0520
450	1C2	O1-0521 - O1-0528
451	1C3	O1-0529 - O1-0536
452	1C4	O1-0537 - O1-0544
453	1C5	O1-0545 - O1-0552
454	1C6	O1-0553 - O1-0560
455	1C7	O1-0561 - O1-0568
456	1C8	O1-0569 - O1-0576
457	1C9	O1-0577 - O1-0584
458	1CA	O1-0585 - O1-0592
459	1CB	O1-0593 - O1-0600
460	1CC	O1-0601 - O1-0608
461	1CD	O1-0609 - O1-0616
462	1CE	O1-0617 - O1-0624
463	1CF	O1-0625 - O1-0632
464	1D0	O1-0633 - O1-0640
465	1D1	O1-0641 - O1-0648
466	1D2	O1-0649 - O1-0656
467	1D3	O1-0657 - O1-0664
468	1D4	O1-0665 - O1-0672
469	1D5	O1-0673 - O1-0680
470	1D6	O1-0681 - O1-0688
471	1D7	O1-0689 - O1-0696
472	1D8	O1-0697 - O1-0704
473	1D9	O1-0705 - O1-0712
474	1DA	O1-0713 - O1-0720
475	1DB	O1-0721 - O1-0728
476	1DC	O1-0729 - O1-0736
477	1DD	O1-0737 - O1-0744
478	1DE	O1-0745 - O1-0752
479	1DF	O1-0753 - O1-0760
480	1E0	O1-0761 - O1-0768
481	1E1	O1-0769 - O1-0776
482	1E2	O1-0777 - O1-0784
483	1E3	O1-0785 - O1-0792
484	1E4	O1-0793 - O1-0800
485	1E5	O1-0801 - O1-0808
486	1E6	O1-0809 - O1-0816
487	1E7	O1-0817 - O1-0824
488	1E8	O1-0825 - O1-0832

Target Address Decimal	Target Address Hexadecimal	Table Reference
489	1E9	O1-0833 - O1-0840
490	1EA	O1-0841 - O1-0848
491	1EB	O1-0849 - O1-0856
492	1EC	O1-0857 - O1-0864
493	1ED	O1-0865 - O1-0872
494	1EE	O1-0873 - O1-0880
495	1EF	O1-0881 - O1-0888
496	1F0	O1-0889 - O1-0896
497	1F1	O1-0897 - O1-0904
498	1F2	O1-0905 - O1-0912
499	1F3	O1-0913 - O1-0920
500	1F4	O1-0921 - O1-0928
501	1F5	O1-0929 - O1-0936
502	1F6	O1-0937 - O1-0944
503	1F7	O1-0945 - O1-0952
504	1F8	O1-0953 - O1-0960
505	1F9	O1-0961 - O1-0968
506	1FA	O1-0969 - O1-0976
507	1FB	O1-0977 - O1-0984
508	1FC	O1-0985 - O1-0992
509	1FD	O1-0993 - O1-1000
510	1FE	O1-1001 - O1-1008
511	1FF	O1-1009 - O1-1016
512	200	O1-1017 - O1-1024
513	201	O2-0001 - O2-0008
514	202	O2-0009 - O2-0016
515	203	O2-0017 - O2-0024
516	204	O2-0025 - O2-0032
517	205	O2-0033 - O2-0040
518	206	O2-0041 - O2-0048
519	207	O2-0049 - O2-0056
520	208	O2-0057 - O2-0064
521	209	O2-0065 - O2-0072
522	20A	O2-0073 - O2-0080
523	20B	O2-0081 - O2-0088
524	20C	O2-0089 - O2-0096
525	20D	O2-0097 - O2-0104
526	20E	O2-0105 - O2-0112
527	20F	O2-0113 - O2-0120
528	210	O2-0121 - O2-0128
529	211	O2-0129 - O2-0136
530	212	O2-0137 - O2-0144
531	213	O2-0145 - O2-0152
532	214	O2-0153 - O2-0160
533	215	O2-0161 - O2-0168
534	216	O2-0169 - O2-0176
535	217	O2-0177 - O2-0184
536	218	O2-0185 - O2-0192
537	219	O2-0193 - O2-0200
538	21A	O2-0201 - O2-0208
539	21B	O2-0209 - O2-0216
540	21C	O2-0217 - O2-0224
541	21D	O2-0225 - O2-0232
542	21E	O2-0233 - O2-0240
543	21F	O2-0241 - O2-0248
544	220	O2-0249 - O2-0256

Target Address Decimal	Target Address Hexadecimal	Table Reference
545	221	O2-0257 - O2-0264
546	222	O2-0265 - O2-0272
547	223	O2-0273 - O2-0280
548	224	O2-0281 - O2-0288
549	225	O2-0289 - O2-0296
550	226	O2-0297 - O2-0304
551	227	O2-0305 - O2-0312
552	228	O2-0313 - O2-0320
553	229	O2-0321 - O2-0328
554	22A	O2-0329 - O2-0336
555	22B	O2-0337 - O2-0344
556	22C	O2-0345 - O2-0352
557	22D	O2-0353 - O2-0360
558	22E	O2-0361 - O2-0368
559	22F	O2-0369 - O2-0376
560	230	O2-0377 - O2-0384
561	231	O2-0385 - O2-0392
562	232	O2-0393 - O2-0400
563	233	O2-0401 - O2-0408
564	234	O2-0409 - O2-0416
565	235	O2-0417 - O2-0424
566	236	O2-0425 - O2-0432
567	237	O2-0433 - O2-0440
568	238	O2-0441 - O2-0448
569	239	O2-0449 - O2-0456
570	23A	O2-0457 - O2-0464
571	23B	O2-0465 - O2-0472
572	23C	O2-0473 - O2-0480
573	23D	O2-0481 - O2-0488
574	23E	O2-0489 - O2-0496
575	23F	O2-0497 - O2-0504
576	240	O2-0505 - O2-0512
577	241	O2-0513 - O2-0520
578	242	O2-0521 - O2-0528
579	243	O2-0529 - O2-0536
580	244	O2-0537 - O2-0544
581	245	O2-0545 - O2-0552
582	246	O2-0553 - O2-0560
583	247	O2-0561 - O2-0568
584	248	O2-0569 - O2-0576
585	249	O2-0577 - O2-0584
586	24A	O2-0585 - O2-0592
587	24B	O2-0593 - O2-0600
588	24C	O2-0601 - O2-0608
589	24D	O2-0609 - O2-0616
590	24E	O2-0617 - O2-0624
591	24F	O2-0625 - O2-0632
592	250	O2-0633 - O2-0640

Target Address Decimal	Target Address Hexadecimal	Table Reference
593	251	O2-0641 - O2-0648
594	252	O2-0649 - O2-0656
595	253	O2-0657 - O2-0664
596	254	O2-0665 - O2-0672
597	255	O2-0673 - O2-0680
598	256	O2-0681 - O2-0688
599	257	O2-0689 - O2-0696
600	258	O2-0697 - O2-0704
601	259	O2-0705 - O2-0712
602	25A	O2-0713 - O2-0720
603	25B	O2-0721 - O2-0728
604	25C	O2-0729 - O2-0736
605	25D	O2-0737 - O2-0744
606	25E	O2-0745 - O2-0752
607	25F	O2-0753 - O2-0760
608	260	O2-0761 - O2-0768
609	261	O2-0769 - O2-0776
610	262	O2-0777 - O2-0784
611	263	O2-0785 - O2-0792
612	264	O2-0793 - O2-0800
613	265	O2-0801 - O2-0808
614	266	O2-0809 - O2-0816
615	267	O2-0817 - O2-0824
616	268	O2-0825 - O2-0832
617	269	O2-0833 - O2-0840
618	26A	O2-0841 - O2-0848
619	26B	O2-0849 - O2-0856
620	26C	O2-0857 - O2-0864
621	26D	O2-0865 - O2-0872
622	26E	O2-0873 - O2-0880
623	26F	O2-0881 - O2-0888
624	270	O2-0889 - O2-0896
625	271	O2-0897 - O2-0904
626	272	O2-0905 - O2-0912
627	273	O2-0913 - O2-0920
628	274	O2-0921 - O2-0928
629	275	O2-0929 - O2-0936
630	276	O2-0937 - O2-0944
631	277	O2-0945 - O2-0952
632	278	O2-0953 - O2-0960
633	279	O2-0961 - O2-0968
634	27A	O2-0969 - O2-0976
635	27B	O2-0977 - O2-0984
636	27C	O2-0985 - O2-0992
637	27D	O2-0993 - O2-1000
638	27E	O2-1001 - O2-1008
639	27F	O2-1009 - O2-1016
640	280	O2-1017 - O2-1024

A

B

C

D

E

F

H

I

J

L

M

N

O

P

R

S

T

U

W

GE Fanuc Automation North America, Inc., Charlottesville, Virginia