GEK-90477A Series One/Series Three Data Communications User's Manual

GE FANUC

GE Fanuc Automation

Programmable Control Products

Series One™/Series Three™ Data Communications

User's Manual

GEK-90477A December, 1986

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.

PREFACE

This manual provides information necessary to implement a serial communications link between a Series Six PC or host computer and a Series One, Series One Junior, Series One Plus, or Series Three PC.

You should become familiar with the operation of the Series One, Series One Junior, Series One Plus, or Series Three PCs (depending on your application) before reading this manual. Also, if a Series Six is to be included in your communications link, you may wish to refer to the Series Six Data Communication Manual, GEK-25364, for complete information on Series Six Data Communications.

Chapter 1, Introduction, describes the capabilities of the Data Communications Unit (DCU) and the Data Communications Module (DCM) and possible system configurations of Series One, Series One Junior, Series One Plus, and Series Three PCs with a Series Six PC or host computer.

Chapter 2, Installation and Operation of the Data Communications Unit for the Series One, Series One Junior, and Series One Plus PCs, describes the operation of the Data Communication Unit's user interfaces and the installation of the DCU.

Chapter 3, Installation and Operation of the Data Communications Module for the Series Three PC, describes the operation of the Data Communication Module's user interfaces and the installation of the DCM.

Chapter 4, Electrical Interface Circuits, provides the information needed to construct cables to connect the DCU or DCM to other devices.

Chapter 5, Communication Examples, explains how to build the Series Six ladder diagram to initiate communications between a Series Six PC and a Series One, Series One Junior, Series One Plus, or Series Three PC.

Chapter 6, Serial Interface Protocol, provides complete reference information on DCU and DCM serial interace protocol and timing to allow the user to write a serial communications driver for a host computer or microprocessor.

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	☐	☐	☐	☐

CUT ON DOTTED LINE

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

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

CHAPTER 5: COMMUNICATION EXAMPLES USING THE SERIES SIX PC AS A MASTER DEVICE (Continued)

TABLE OF CONTENTS

TABLES

FIGURES

CHAPTER 1
INTRODUCTION

The serial interface to the Series One™ family of PCs is essentially the same as the interface to the Series Three™ PC. For this reason the user information for both have been combined into one manual. The differences are primarily related to the physical package which affects the installation of the interface. To differentiate between the two interfaces, the terms below are used throughout this manual.

Data Communications Unit (DCU)	-	Series One, Series One Junior, and Series One Plus PC Interface
Data Communications Module (DCM)	-	Series Three PC Interface

This chapter describes the capabilities and system configurations for serial communications with the Series One Family of programmable controllers and Series Three programmable controllers.

COMMUNICATIONS CAPABILITIES USING THE DCU OR DCM

The DCU and DCM provide a serial, RS-422 interface between a Series One, Series One Junior, Series One Plus, or Series Three PC and a device such as a Series Six™ PC, Workmaster™ computer or other host computer. Memory types that can be accessed through the DCU or DCM include:

- Discrete input and output points,
- Timer and counter accumulator references (and Series One Plus PC and Series Three PC data registers),
- Scratchpad (including using the password and the user logic error checking capability for the Series One Plus PC),
- User logic, and
- Diagnostic information.

Using the CCM2 protocol, the host computer or Series Six PC can have supervisory control over one or more PCs of the Series One family or one or more Series Three PCs. The data transfer rates as well as other communications parameters for the DCU and DCM are DIP-switch selectable. The primary data transfer rate for direct connections is 19.2 kBps. Other data transfer rates are provided for special purpose interfaces which include modem configurations.

SYSTEM CONFIGURATIONS USING THE DCU OR DCM

A system configuration refers to the way in which various devices are combined to form a communications network. As explained below, both point-to-point and multidrop configurations are possible through the DCU or DCM. For details on constructing cables, see Chapter 4, Electrical Interface Circuits.

In all configurations, the Series One, Series One Junior, Series One Plus, or Series Three PC is the slave device, and the host computer, Workmaster, or Series Six PC is the master device. A slave can respond only to requests from a master.

™ Trademark of General Electric Company.

When a Workmaster computer or other host computer is the master device, host software must be written to handle the protocol requirements as explained in Chapter 6, Serial Interface Protocol.

POINT-TO-POINT CONFIGURATIONS

In the point-to-point configuration, only two elements can be connected to the same communication line. The communication line can be connected directly using the RS-422 electrical interface capability (4000 feet, 1200 meters, maximum), or connected through modems and an RS-232 to RS-422 adapter unit for longer distances over telephone lines.

TPK.A.40369

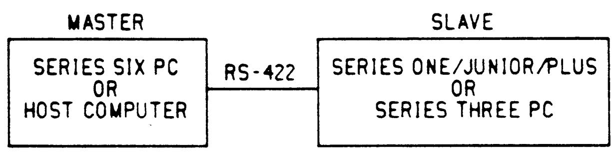


Figure 1.1 POINT-TO-POINT CONFIGURATION (DIRECT)

TPK.A.40370

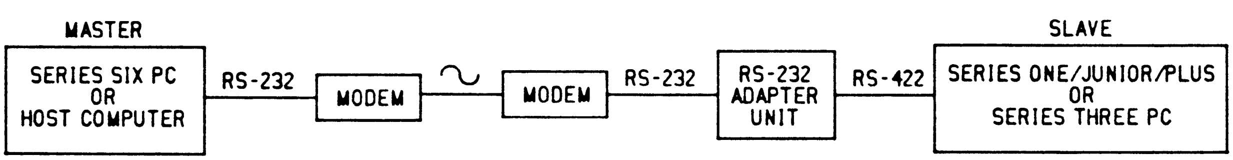


Figure 1.2 POINT-TO-POINT CONFIGURATION (USING MODEMS)

MULTIDROP CONFIGURATIONS

This configuration permits the connection of a host computer or Series Six PC to a group of Series One, Series One Junior, Series One Plus, or Series Three PCs. As with point-to-point connections, either RS-422 capability or modems can be used. A maximum of 8 slaves can be connected using RS-422. The maximum distance between the two end devices in the multidrop is 4000 feet (1200 meters).

When RS-232 modems are used, an RS-232 adapter unit must be included to convert RS-422 signals from the DCU or DCM to RS-232 signals for the modems.

TPK.A.40371

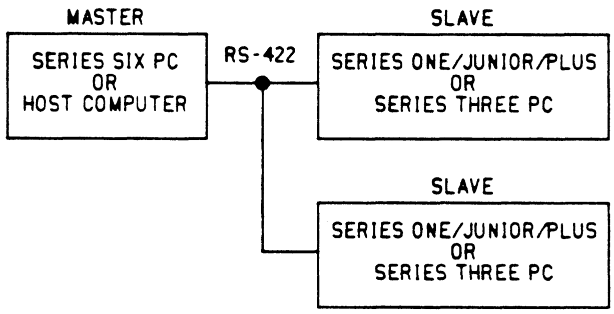
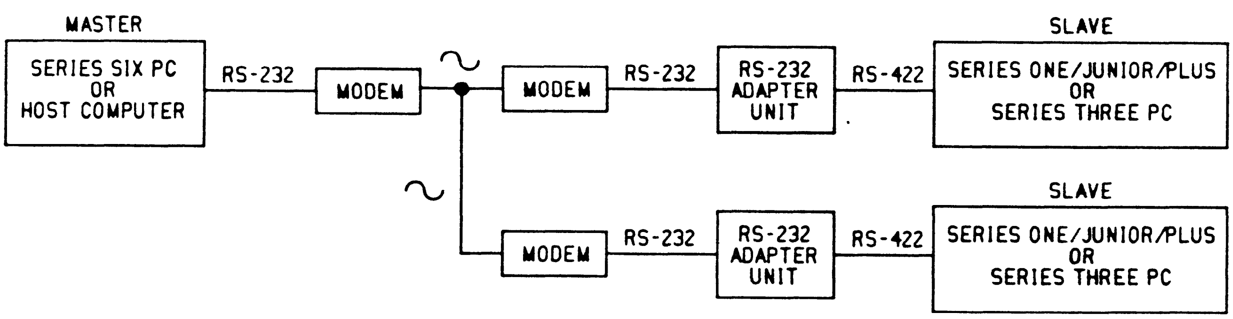


Figure 1.3 MULTIDROP CONFIGURATION (DIRECT)

TPK.A.40372



*Up to 8 slave devices can be multidropped from the RS-232 Adapter Unit.

Figure 1.4 MULTIDROP CONFIGURATION (USING MODEMS)

CHAPTER 2
INSTALLATION AND OPERATION OF THE DATA COMMUNICATIONS UNIT FOR THE SERIES ONE FAMILY OF PCS

This chapter describes the operation of the user interfaces (LEDs, switches, and ports) and the installation of the Data Communications Unit (DCU) (IC610CCM100A, IC610CCM105A).

NOTE TO SERIES ONE PLUS USERS

Use only the Data Communications Unit (IC610CCM105A) for communications with the Series One Plus PC.

DESCRIPTION AND OPERATION OF THE USER INTERFACES FOR THE DCU

The various indicator lights, connectors, and configuration DIP switches for the DCU are shown in Figure 2.1.

TPA.C.40221

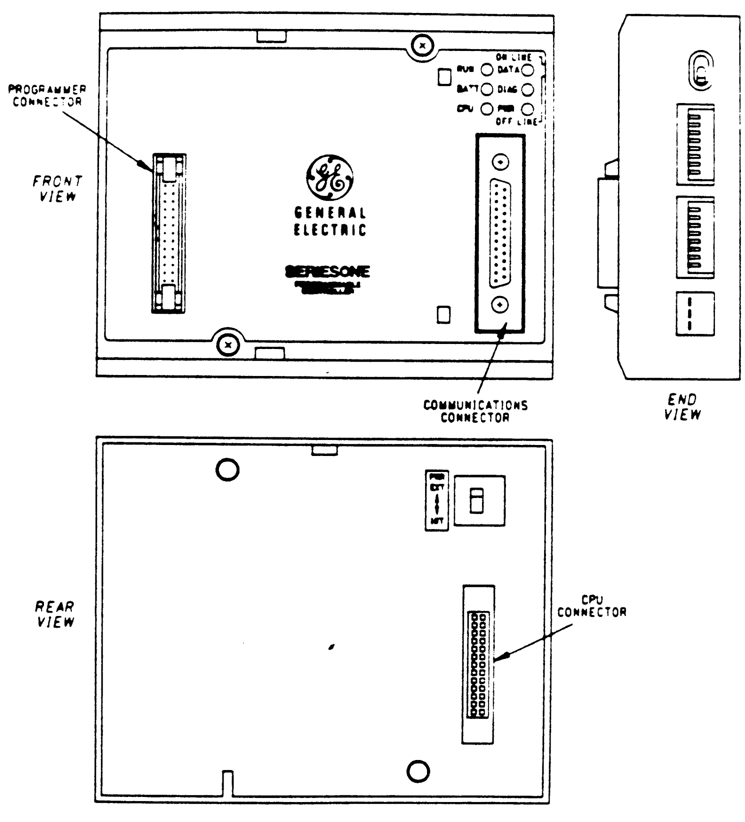


Figure 2.1 FRONT, END, AND REAR VIEW OF THE DCU

LED INDICATORS

The six status LED's on the front of the DCU convey the following information:

Status LED	State	Description
DATA	On	Data being transferred to and from the communication port.
	Off	Data not being transferred to and from the communication port or data incorrect due to: 1. Parity overrun or framing errors; 2. Invalid header, data block, control character, or checksum; 3. Time out on serial link. (Refer to Chapter 6 for more information on the protocol used).
DIAG	On	Power-up hardware diagnostics have passed.
	Off	Power-up hardware diagnostics have failed.
PWR	On	5 V dc power to DCU is connected.
	Off	5 V dc power to DCU is not connected.

NOTE

Power to the DCU can be supplied from the rack power supply or an external supply. When the power supply select switch is in the EXT position, power must be supplied through the external power supply connector on the side of the DCU. See Figures 2.1 and 2.2.

Status LED	State	Description
RUN	On	The CPU is in the RUN mode.
	Off	The CPU is not in the RUN mode.
BATT	On	The battery which provides memory back-up in the CPU is not OK.
	Off	The battery which provides memory back-up in the CPU is OK.
CPU	On	There is an error; check the error code on the programmmer display and take the appropriate action.
	Off	There is no CPU error.

FRONT PANEL CONNECTORS

Two connectors on the front of the DCU provide an interface to:

1. Programmer (Programmer Connector),
2. External serial device (Communications Connector).

Programmer Connector

The programmer connector is the mating connector which mates with the programmer and connects with the CPU. This permits use of the programmer while the DCU is connected to the CPU. See Figures 2.1 and 2.4.

Communications Connector

The communications connector (25-pin female, D-type) provides a serial interface to external devices. A pin-by-pin description of this connector is shown in Chapter 4.

DCU CONFIGURATION SWITCHES

The configuration switches are located on the right side of the DCU as shown below.

TPK.A.40373

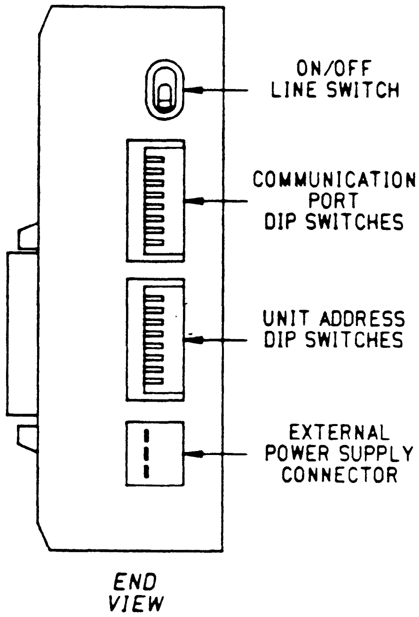


Figure 2.2 LOCATION OF THE DCU CONFIGURATION SWITCHES

ON/OFF-LINE Switch

The ON/OFF-LINE switch, which is directly above the DIP switches on the right side of the DCU, enables or disables serial communications with the Series One, Series One Junior, or Series One Plus CPU.

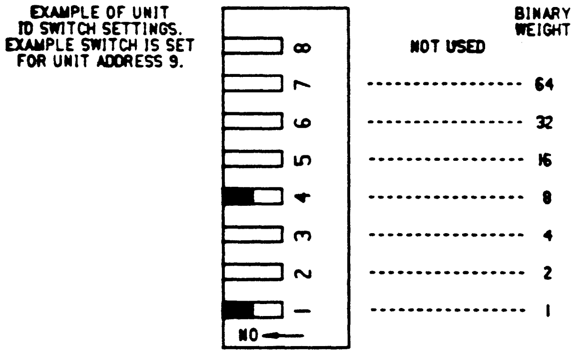
OFF LINE: Serial communication between the DCU and the CPU is disabled and the CPU is under control of the attached programmmer.

ON LINE: Serial communication between the DCU and CPU is enabled and the programmer is disabled if attached.

CPU (Unit) ID DIP Switches

The bottom group of eight DIP switches located on the right side of the DCU determines the CPU ID of 1-90. The switch configuration associated with each ID is shown in Figure 2.3.

TPA.A.40223



UNIT ID	DIP SWITCH POSITION								UNIT ID	DIP SWITCH POSITION								UNIT ID	DIP SWITCH POSITION							
	8	7	6	5	4	3	2	1		8	7	6	5	4	3	2	1		8	7	6	5	4	3	2	1
1								X	31				X	X	X	X	X	61			X	X	X	X		X
2							X		32			X						62			X	X	X	X	X	
3							X	X	33			X					X	63			X	X	X	X	X	X
4						X			34			X				X		64		X						
5						X		X	35			X				X	X	65		X						X
6						X	X		36			X			X			66		X					X	
7						X	X	X	37			X			X		X	67		X					X	X
8					X				38			X			X	X		68		X				X		
9					X			X	39			X			X	X	X	69		X				X		X
10					X		X		40			X		X				70		X				X	X	
11					X		X	X	41			X		X			X	71		X				X	X	X
12					X	X			42			X		X		X		72		X			X			
13					X	X		X	43			X		X		X	X	73		X			X			X
14					X	X	X		44			X		X	X			74		X			X		X	
15					X	X	X	X	45			X		X	X		X	75		X			X		X	X
16				X					46			X		X	X	X		76		X			X	X		
17				X				X	47			X		X	X	X	X	77		X			X	X		X
18				X			X		48			X	X					78		X			X	X	X	
19				X			X	X	49			X	X				X	79		X			X	X	X	X
20				X		X			50			X	X			X		80		X		X				
21				X		X		X	51			X	X			X	X	81		X		X				X
22				X		X	X		52			X	X		X			82		X		X			X	
23				X		X	X	X	53			X	X		X		X	83		X		X			X	X
24				X	X				54			X	X		X	X		84		X		X		X		
25				X	X			X	55			X	X		X	X	X	85		X		X		X		X
26				X	X		X		56			X	X	X				86		X		X		X	X	
27				X	X		X	X	57			X	X	X			X	87		X		X		X	X	X
28				X	X	X			58			X	X	X		X		88		X		X	X			
29				X	X	X		X	59			X	X	X		X	X	89		X		X	X			X
30				X	X	X	X		60			X	X	X	X			90		X		X	X		X	

X = Switch in the ON position

Figure 2.3 DIP-SWITCH SETTINGS FOR CPU ID SELECTION

Communication Port Configuration DIP Switches

The top group of eight DIP switches on the right side of the DCU determines the set-up parameters for the communication port (refer to Figure 2.2 for location of the switches). The settings for the communication set-up parameters are shown in Table 2.1. To execute the Loop Back Test, the ON/OFF-LINE SWITCH must be in the OFF-LINE mode. Switches 7 and 8 are not used.

Table 2.1 COMMUNICATIONS PORT CONFIGURATION DIP-SWITCH SETTINGS

DATA RATE SELECTION (BPS)	DIP-SWITCH NUMBER 1	DIP-SWITCH NUMBER 2
*300	OFF	OFF
1200	ON	OFF
9600	OFF	ON
19.2 k	ON	ON

PARITY SELECTION	DIP-SWITCH NUMBER 3
Parity ENABLED (Odd parity generated and checked).	ON
*Parity DISABLED (No parity is generated or checked).	OFF

LOOP-BACK TEST (Special Connector Required)	DIP-SWITCH NUMBER 4
Enabled	ON
*Disabled	OFF

TURN-AROUND DELAY	DIP-SWITCH NUMBER 5
* 0 ms delay	OFF
10 ms delay	ON

POWER-UP MODE**	DIP-SWITCH NUMBER 6
Program/Stop Mode	ON
*Run Mode	OFF

*Factory set default position.
**See section, Power Cycle Conditions Affecting System Operation.

EXTERNAL POWER SUPPLY CONNECTOR

The external power supply connector (see Figure 2.2) allows the DCU to receive its operating power (5 V dc at 0.5 A) from an external power supply. A three conductor cable is provided with the DCU for external power supply connection. Its color code is as follows:

White: +5 V dc (±5%) at 0.5 amps
Black: Logic ground of power supply
Green: Power system ground

POWER SUPPLY SELECT SWITCH

There is a power supply select switch on the back of the module to select internal (CPU) or external power for the DCU. An adjacent label indicates correct switch orientation for each selection.

USING THE DCU WITH CPU RACK POWER

It is recommended that a Series One high-capacity power supply, IC610CHS110A, 114A, 120A, or 124A be used when installing a DCU in a system. If a high-capacity power supply is not used, then the DCU should be powered by an external +5 V dc power supply. If a standard (low capacity) Series One power supply is used with the DCU, inconsistent CPU or communications operation will result.

NOTE

Even if a high-capacity power supply is being used in the CPU rack, inconsistent CPU or communications operation may be observed depending on the number and unit load of I/O modules installed in the rack. Refer to Tables 2.2 and 2.3 for units of load supplied by the different racks and used by I/O modules and other system devices.

Table 2.2 SERIES ONE UNITS* OF LOAD (SUPPLIED)

CATALOG NUMBER	DESCRIPTION	POWER SUPPLIED IN UNITS OF LOAD			
		+5 V	+9 V	+24 V	+24 V EXT
IC610CHS100A	5-slot std cap	40	80	20	-
IC610CHS110A	5-slot hi cap	140	80	40**	10
IC610CHS114A	5-slot hi cap 24 V dc	140	80	40	-
IC610CHS120A	10-slot hi cap	140	160	40**	10
IC610CHS124A	10-slot hi cap 24 V dc	140	160	40	-
IC610CHS130A	10-slot hi cap	140	170	50**	10
IC610CHS134A	10-slot hi cap 24 V dc	140	170	50	-

* 1 unit = 10 mA
** If an external sensor is connected to the 24 V + and - terminals on the power supply, the current used by the sensor (up to a maximum of 100 mA), should be deducted from the available listed units of load.

Table 2.3 SERIES ONE UNITS* OF LOAD (USED)

CATALOG NUMBER	DESCRIPTION (CIRCUITS)	POWER USED IN UNITS OF LOAD +5 V	+9 V	+24 V
IC610CPU101C	CPU	25	-	-
IC610CPU105A	CPU	25	-	-
IC610PRG100B	Programmer	6	5	-
IC610PRG105A	Programmer	6	5	-
IC610MDL101A	Inp 24 V dc sink (8)	-	1	10
IC610MDL102A	Inp 24 V dc src (16)	-	2	19
IC610MDL103A	I/O 24 V dc (4/4)	-	2	7
IC610MDL104A	I/Relay Out 24 V dc (4/4)	-	20	6
IC610MDL105A	Thumbwheel Interf (4x16)	-	1	10
IC610MDL106A	Inp 24 V dc sink w LEDs (16)	-	3	24
IC610MDL107A	Inp 24 V dc sink load (16)	-	3	23
IC610MDL110A	Hi Speed Counter (1)	-	7	-
IC610MDL111A	Inp 24 V ac/dc (8)	-	1	-
IC610MDL112A	Inp 24 V ac/dc souce (16)	-	13	-
IC610MDL115A	I/O Fast Response (4/2)	-	8	6
IC610MDL124A	I/O Simulator (8)	-	1	11
IC610MDL125A	Inp 115 V ac (8)	-	1	-
IC610MDL126A	Inp 115 V ac isolated (4)	-	1	-
IC610MDL127A	Inp 230 V ac (8)	-	1	-
IC610MDL151A	Out 24 V dc sink (8)	-	2	3
IC610MDL152A	Out 24 V dc sink (16)	-	5	4
IC610MDL153A	Out 24 V dc sink (4)	-	1	1
IC610MDL154A	Out 24 V dc sink/src (4)	-	1	10
IC610MDL155A	Out 24 V dc src (8)	-	3	-
IC610MDL156A	Out 24 V dc sink w LEDs (16)	-	4	10
IC610MDL157A	Out 24 V dc sink w LEDs (16)	-	4	10
IC610MDL158A	Out 24 V dc src w LEDs (16)	-	20	-
IC610MDL175A	Out 115/230 V ac (8)	-	16	-
IC610MDL176A	Out 115/230 V ac isol (4)	-	8	-
IC610MDL180A	Out Relay (8)	-	34	-
IC610MDL182A	Out Relay (16)	-	48	-
IC610CCM100A	Data Comms Unit	30	-	-
IC610CCM105A	Data Comms Unit	30	-	-
IC610CCM110A	I/O Link Local Module	60	-	-
IC610CCM111A	I/O Link Remote Module	60	-	-
IC610PER151A	Printer I/F	26	-	-
IC610PER154A	Low Cost PROM Writer	80	-	-

*1 unit = 10 mA. Calculations are based on the worst case--all inputs and outputs on.

INSTALLING THE DCU

To install the DCU:

1. Set the internal/external power switch to the desired position.
2. Position the CPU (unit) ID and port configuration DIP switches to the desired position (see Figure 2.3 and Table 2.1).
3. With the Series One, Series One Junior, or Series One Plus CPU power off, connect the DCU to the CPU and the programmer to the DCU (if desired) as shown in Figure 2.4.

If, before powering up, the ON-LINE/OFF-LINE switch is placed in the ON-LINE position, after power up the PWR, RUN, and DIAG indicators should light in that order. For more information on power-up conditions affecting the CPU and communications status see Table 2.4.

NOTE

The Series One CPU version must be Revision B or later.

TPK.A.40374

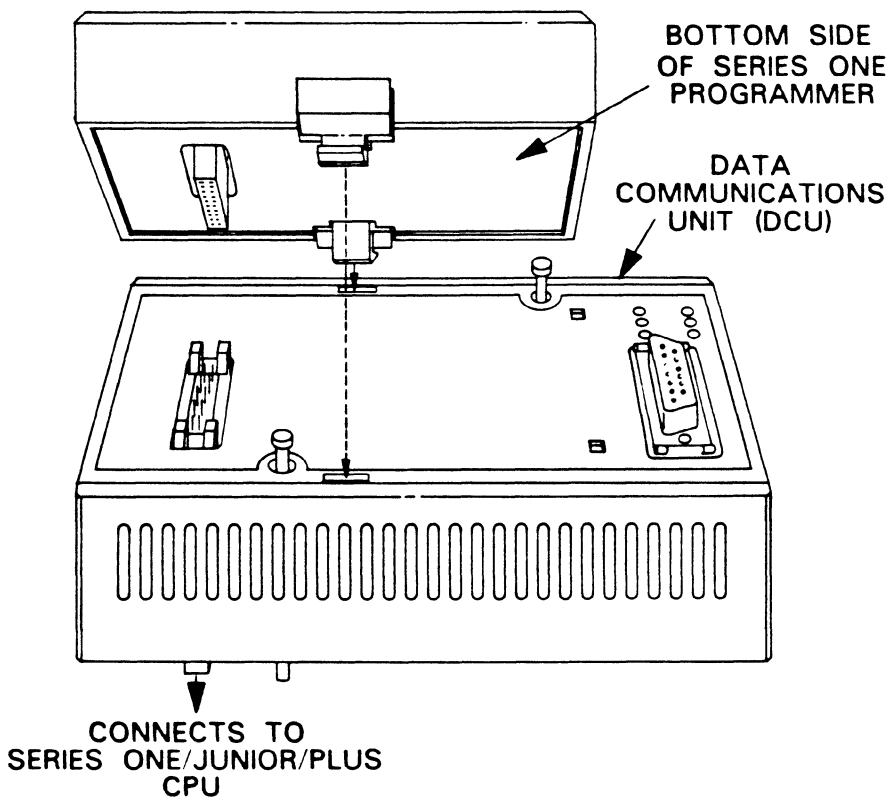


Figure 2.4 CONNECTING THE PROGRAMMER, DCU, AND CPU

POWER CYCLE CONDITIONS AFFECTING SYSTEM OPERATION

When power is cycled, the resulting CPU and communications status depends upon the position of the DCU ON-LINE/OFF-LINE switch and power-up mode switch, whether the programmer is attached or detached, the programmer mode switch position, and the condition of the CPU battery. See Table 2.4.

Table 2.4 POWER CYCLE CONDITIONS AFFECTING SYSTEM OPERATION
(The user program is assumed to be in CMOS RAM).

DCU		PROGRAMMER		RESULTING CPU AND COMMUNICATIONS STATUS ON POWER CYCLE
ON-LINE/ OFF-LINE SWITCH	POWER-UP MODE DIP SWITCH 6	ATTACHED/ DETACHED	MODE KEY-SWITCH POSITION	
On-Line	Off(Run)	Attached	Run	CPU in Run mode with communications active.
On-Line	On(Prog)	Attached	Run	CPU in Program mode with communications active only for the following serial requests: Read or command Run/Program and Read Diagnostic Status Words. The DCU will return in the status code a hexadecimal 10 to indicate that a power cycle has occurred.
Off-Line	On or Off	Not Attached	-	CPU is in the same mode in which it powered down, communication is inactive since the unit is off line. (Communications will be active on off line to on line transition).*
Off-Line	On or Off	Attached	-	CPU is in whatever mode the key-switch is set for with communications not active.

* For Series One CPUs versions A or B, the resulting communications status is the same, but the resulting CPU status is that the CPU is in Program mode with the communications inactive.

NOTE

The following statuses result <u>only</u> when there is a low battery condition in the CPU.

Table 2.4 POWER CYCLE CONDITIONS AFFECTING SYSTEM OPERATION (continued)

DCU		PROGRAMMER		RESULTING CPU AND COMMUNICATIONS STATUS ON POWER CYCLE
ON-LINE/ OFF-LINE SWITCH	POWER UP MODE DIP SWITCH 6	ATTACHED/ DETACHED	MODE KEY-SWITCH POSITION	
Off Line	On or Off	Not Attached	-	CPU in Program mode with communications inactive since unit is off line.
On Line	On(Prog)	Not Attached	-	CPU in Program mode with communications active. DIAG LED will be ON and and RUN LED will be OFF.
On Line	Off (Run)	Not Attached	-	CPU in Program mode with communications inactive. DIAG and RUN will be OFF. Unit must be manually set to Program/Stop mode and the E-21 error cleared (if it has occurred) before commmunications can resume.

CHAPTER 3
INSTALLATION AND OPERATION OF THE DATA COMMUNICATIONS MODULE FOR THE SERIES THREE PC

This chapter describes the operation of the DCM's user interfaces (LEDs, switches, and ports) and the installation of the Data Communications Module (DCM) (IC630CCM300).

DESCRIPTION AND OPERATION OF THE DCM'S USER INTERFACES

The various indicator lights, connectors, and configuration DIP switches for the DCM are shown in Figure 3.1.

TPK.A.40007

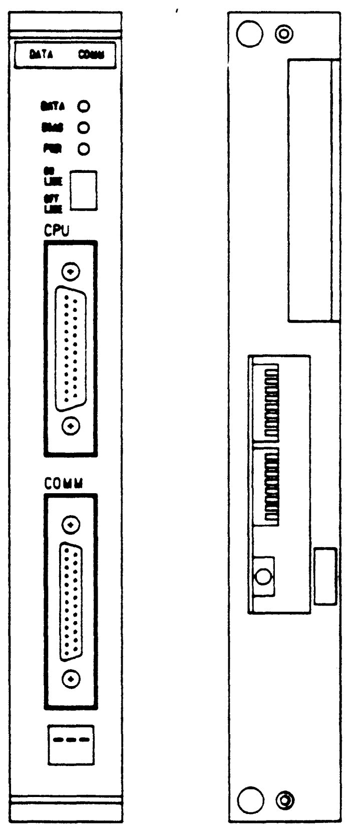


Figure 3.1 FRONT AND REAR VIEW OF THE DCM

LED INDICATORS

The three status LED's on the front of the DCM convey the following information:

Status LED	State	Description
DATA	On	Data being transferred to/from the communication port.
	Off	Data not being transferred to/from the communication port or data incorrect due to: 1. Parity overrun or framing errors; 2. Invalid header, data block, control character, or checksum; 3. Time out on serial sink.
DIAG	On	Power-up hardware diagnostics have passed.
	Off	Power-up hardware diagnostics have failed.
PWR	On	5 V dc power to DCM is connected.
	Off	5 V dc power to DCM is not connected.

NOTE

Power to the DCM can come from the Series Three CPU or external supply depending on position of power select switch. When the power supply select switch is in the EXT position, power must be supplied through the external power supply connector on the front of the DCM. See Figure 3.1.

FRONT PANEL CONNECTORS

Three connectors on the front of the DCM provide an interface to:

1. Series Three CPU (CPU Connector);
2. External serial device (Communications Connector); and
3. External power supply.

Each of these interfaces are described below.

Series Three CPU Connector

The CPU connector (25-pin male, D-type) ties the DCM to the Series Three CPU. All communication with the Series Three, as well as operating power (if the power supply select switch is set to internal) is transmitted through this interface. The cable (IC630CBL395A) is provided with each DCM for the link.

Communications Connector

The communications connector (25-pin female, D-type) connects the DCM to external devices. A detailed description (pin by pin) of this connector is shown in Chapter 4.

External Power Supply Connector

The external power supply connector allows the DCM to receive its operating power (5 V dc at 0.5 A) from an external supply. Users with Series Three power supply IC630PWR300A require an external power supply to operate a DCM. Other Series Three power supplies may or may not necessitate the use of an external power supply for proper operation of the DCM. This is dependent on the number and type of I/O modules in the CPU rack. Refer to Tables 3.2 and 3.3. A three conductor cable is provided with the DCM for external power supply connection. Its color code is as follows:

WHITE: +5V DC (± 5%) at 0.5 amps
BLACK: Logic ground of power supply
GREEN: Power system ground.

DCM CONFIGURATION SWITCHES

The ON/OFF line switch is located on the front of the DCM. The other configuration switches are located on the back of the DCM as shown below.

TPK.A.40226

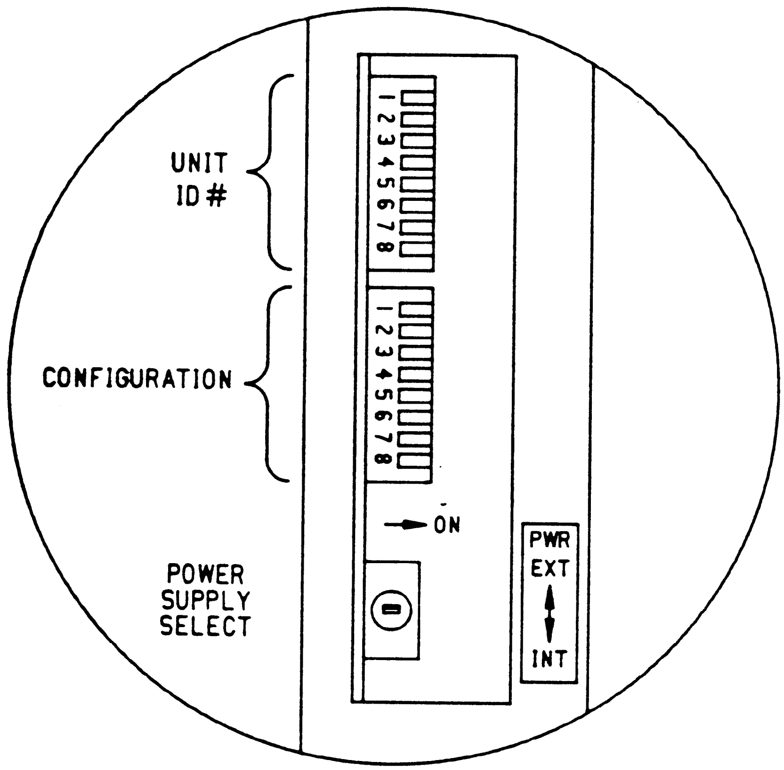


Figure 3.2 LOCATION OF THE DCM CONFIGURATION SWITCHES

ON/OFF-LINE Switch

The ON/OFF-line switch which is recessed on the front panel of the DCM enables or disables the serial communications with the Series Three CPU.

OFF LINE: Serial communication between the DCM and CPU is disabled, and the CPU is under control of the programmer.

ON LINE: Serial communication between the DCM and CPU is enabled, and the programmer is not functional.

NOTE

The terminal LED indicator on the face of the Series Three identifies the status of the serial link between the DCM and CPU.

Terminal LED ON:	DCM/CPU interface enabled.
Terminal LED OFF:	DCM/CPU interface disabled.

Interaction between the DCM ON/OFF-LINE switch and the CPU Keyswitch

In order to establish or maintain the serial link between the DCM and the Series Three CPU, the CPU keyswitch must be in the Run 1 or Run position, and the DCM ON/OFF-LINE switch in the ON-LINE position. If the CPU keyswitch is ever taken out of the Run 1/Run position when the serial link is enabled, the link will become disabled and the TERMINAL LED will turn off.

To re-enable communications:

1. Put the CPU keyswitch back in Run 1 or Run position.
2. Cycle the ON/OFF-LINE switch on the DCM with the final position being ON LINE.

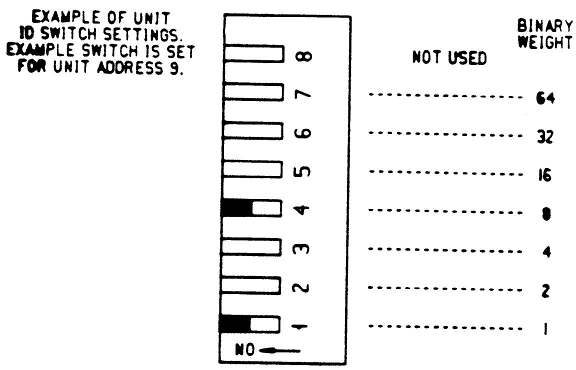
NOTE

Once the link is established and the TERMINAL LED is on, the Series Three CPU can be put in either Stop/Program or Run mode by a serial request from the master device on the link. See the application examples in Chapter 5.

CPU (Unit) ID DIP Switches

The top group of eight DIP switches located on the back of the DCM (see Figure 3.2), determine the CPU ID of 1-90. The switch configuration associated with each ID is shown in Figure 3.3.

TPK.A.40223



UNIT ID	DIP SWITCH POSITION								UNIT ID	DIP SWITCH POSITION								UNIT ID	DIP SWITCH POSITION							
	8	7	6	5	4	3	2	1		8	7	6	5	4	3	2	1		8	7	6	5	4	3	2	1
1								X	31				X	X	X	X	X	61			X	X	X	X		X
2							X		32			X						62			X	X	X	X	X	
3							X	X	33			X					X	63			X	X	X	X	X	X
4						X			34			X				X		64		X						
5						X		X	35			X				X	X	65		X						X
6						X	X		36			X			X			66		X					X	
7						X	X	X	37			X			X		X	67		X					X	X
8					X				38			X			X	X		68		X				X		
9					X			X	39			X			X	X	X	69		X				X		X
10					X		X		40			X		X				70		X				X	X	
11					X		X	X	41			X		X			X	71		X				X	X	X
12					X	X			42			X		X		X		72		X			X			
13					X	X		X	43			X		X		X	X	73		X			X			X
14					X	X	X		44			X		X	X			74		X			X		X	
15					X	X	X	X	45			X		X	X		X	75		X			X		X	X
16				X					46			X		X	X	X		76		X			X	X		
17				X				X	47			X		X	X	X	X	77		X			X	X		X
18				X			X		48			X	X					78		X			X	X	X	
19				X			X	X	49			X	X				X	79		X			X	X	X	X
20				X		X			50			X	X			X		80		X		X				
21				X		X		X	51			X	X			X	X	81		X		X				X
22				X		X	X		52			X	X		X			82		X		X			X	
23				X		X	X	X	53			X	X		X		X	83		X		X			X	X
24				X	X				54			X	X		X	X		84		X		X		X		
25				X	X			X	55			X	X		X	X	X	85		X		X		X		X
26				X	X		X		56			X	X	X				86		X		X		X	X	
27				X	X		X	X	57			X	X	X			X	87		X		X		X	X	X
28				X	X	X			58			X	X	X		X		88		X		X	X			
29				X	X	X		X	59			X	X	X		X	X	89		X		X	X			X
30				X	X	X	X		60			X	X	X	X			90		X		X	X		X	

X = Switch in the ON position

Figure 3.3 DIP-SWITCH SETTINGS FOR CPU ID SELECTION

Communication Port Configuration DIP Switches

The bottom group of eight DIP switches on the back of the DCM selects the mode of operation for the communication port (refer to Figure 3.2 for location of switches). The various settings for the communication set up parameters are shown in Table 3.1. To execute the loop-back test the ON/OFF-LINE switch must in the Off-Line mode.

Table 3.1 COMMUNICATIONS PORT CONFIGURATION DIP-SWITCH SETTINGS

DATA RATE SELECTION (BPS)	DIP-SWITCH NUMBER 1	DIP-SWITCH NUMBER 2
*300	OFF	OFF
1200	ON	OFF
9600	OFF	ON
19.2 k	ON	ON

PARITY SELECTION	DIP-SWITCH NUMBER 3
Parity ENABLED (Odd parity generated and checked).	ON
*Parity DISABLED (No parity is generated or checked).	OFF

LOOP-BACK TEST (Special Connector Required)	DIP-SWITCH NUMBER 4
Enabled	ON
*Disabled	OFF

TURN-AROUND DELAY	DIP-SWITCH NUMBER 5
* 0 ms delay	OFF
10 ms delay	ON

KEYING SIGNAL	DIP-SWITCH NUMBER 6
Enabled	ON
*Disabled	OFF

*Factory set default position.

POWER SUPPLY SELECT SWITCH

There is a power supply select switch on the back of the module for the selection of internal (CPU) or external power for the DCM. An adjacent label indicates correct switch orientation for each selection. See the section, External Power Supply Connector, in this chapter for information on the installation of an external power supply.

USING THE DCM WITH CPU RACK POWER

Users with Series Three power supply IC630PWR300A require an external 5 V dc power supply to operate the DCM. If power supply IC630PWR300A is used with the DCM, inconsistent CPU or communications operation will result.

NOTE

Even if a high-capacity power supply is being used in the CPU rack, inconsistent CPU or communications operation may be observed depending on the number and unit load of I/O modules installed in the rack. Refer to Tables 3.2 and 3.3 for units of load supplied by the different racks and used by I/O modules and other system devices.

Table 3.2 SERIES THREE UNITS* OF LOAD (SUPPLIED)

CATALOG NUMBER	DESCRIPTION	POWER SUPPLIED IN UNITS OF LOAD +5V	+12V
IC630PWR300A	Standard P.S. 115/230 Vac	250	100
IC630PWR310A	Hi Cap. P.S. 115/230 Vac	300	200
IC630PWR314A	Hi Cap. P.S. 24 Vdc	300	200
IC630PWR320A	Hi Cap. P.S. Remote I/O 115/230Vac	300	200
IC630PWR324A	Hi Cap. P.S. Remote I/O 24 Vdc	300	200

* 1 unit = 10 mA.

Table 3.3 SERIES THREE UNITS* OF LOAD (USED)

CATALOG NUMBER	DESCRIPTION (CIRCUITS)	POWER USED IN UNITS OF LOAD +5 V	+12 V
IC630CPU301A	CPU/Programmer Unit	150	-
IC630MDL301A	Inp 24 V dc sink (16)	11	-
IC630MDL302A	Inp 24 V dc sink (32)	5	-
IC630MDL303A	Inp 5-12 V dc sink (32)	5	-
IC630MDL304A	I/O 24 V dc sink (16/16)	9	-
IC630MDL306A	Inp 24 V dc sink w LEDs (32)	8	-
IC630MDL310A	Hi Speed Counter (1)	20	-
IC630MDL311A	Inp 24 V ac/dc src (16)	9	-
IC630MDL316B	Analog Inp 1-5, 1-10 V dc (2)	30	-
IC630MDL324A	I/O Simulator (16)	11	-
IC630MDL325A	Inp 115 V ac (16)	17	-
IC630MDL326A	Inp 115 V ac isolated (8)	6	-
IC630MDL327A	Inp 230 V ac (16)	9	-
IC630MDL351A	Out 24 V dc sink (8)	2	10
IC630MDL352A	Out 24 V dc sink (16)	4	18
IC630MDL353A	Out 24 V dc sink (32)	10	-
IC630MDL354A	Out 5-12 V dc sink (32)	16	-
IC630MDL356A	Out 24 V dc sink w LEDs (32)	10	-
IC630MDL357A	Out 24 V dc src (16)	2	39
IC630MDL366A	Analog Out 1-5 V dc, 4-20 mA (2)	33	-
IC630MDL367A	Analog Out -10 to +10 V dc (2)	33	-
IC630MDL368A	Analog Out 0-10 V dc, 4-20 mA (2)	33	-
IC630MDL375B	Out 115 V ac (16)	4	30
IC630MDL376B	Out 115 V ac isolated (8)	4	39
IC630MDL380A	Out Relay 5-265 ac/dc (16)	4	64
IC630CCM300A	Data Comms Module	50	-
IC630CCM310A	I/O Link Local	80	-
IC630CCM311A	I/O Link Remote	80	-
IC630PER320A	I/O Link Local Fbr Opt P-P	80	-
IC630PER321A	I/O Link Remote Fbr Opt P-P	80	-
IC630PER330A	I/O Link Local Fbr Opt M-P	80	-
IC630PER331A	I/O Link Remote Fbr Opt M-P	80	-

* 1 unit = 10 mA. Calculations are based on the worst case--all inputs and outputs on.

INSTALLING THE DCM

To install the DCM:

1. Set the internal/external power switch to the desired position.
2. Position the unit address (ID) and port configuration DIP switches to the desired position (see Figure 3.3 and Table 3.1).
3. Mount the DCM in the Series Three rack or outside the rack within about 5 feet of the CPU.
4. With the Series Three power off, connect the DCM to the CPU using cable IC630CBL395A as shown in Figure 3.4.

If, before powering up, the DCM ON-LINE/OFF-LINE switch is placed in the ON-LINE position and the Series Three CPU switch is in the RUN position, after power up the PWR and DIAG indicators on the DCM should light in that order. In addition, the RUN and TERMINAL indicators on the CPU should light. For more information on power-up conditions affecting the CPU and communications status see Table 3.4.

TPK.A.40458

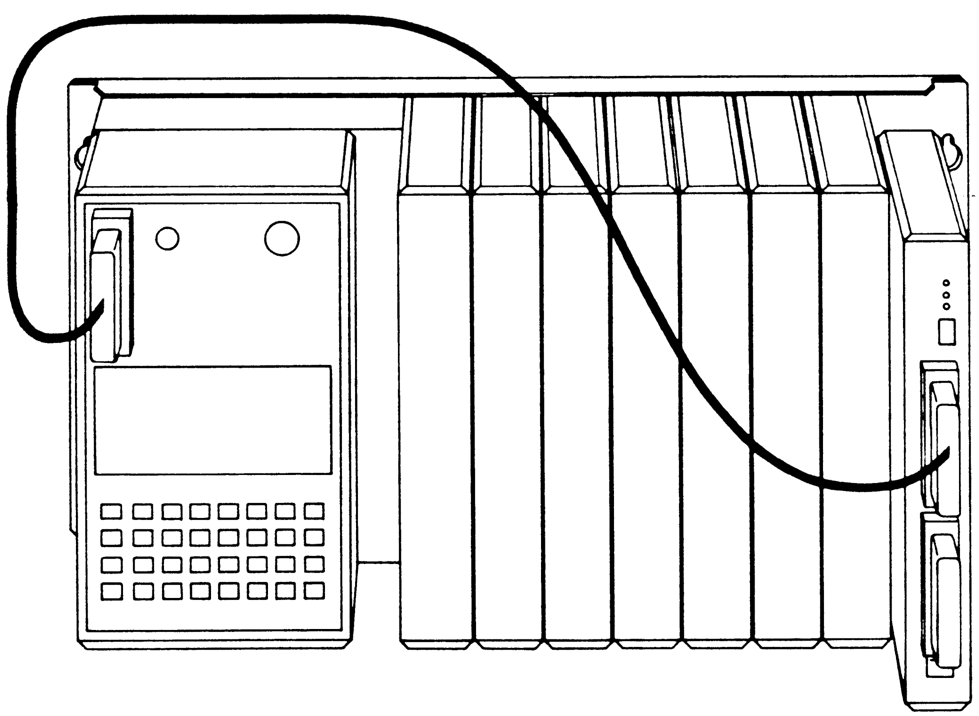


Figure 3.4 CONNECTING THE DCM TO THE CPU

POWER CYCLE CONDITIONS AFFECTING SYSTEM OPERATION

When the power is cycled, the resulting CPU and communications status depends upon the position of the DCM ON-LINE/OFF-LINE switch as shown in Table 3.4.

Table 3.4 POWER CYCLE CONDITIONS AFFECTING SYSTEM OPERATION
(The user program is assumed to be in CMOS RAM).

DCM	CPU KEY-SWITCH	RESULTING CPU AND COMMUNICATIONS STATUS ON POWER CYCLE
On-Line	Run	CPU in Run mode with TERMINAL mode indicator ON.
Off-Line	Run	CPU in Run mode with TERMINAL mode indicator OFF.

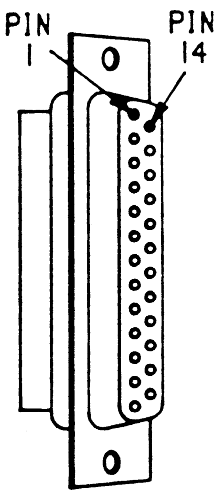
CHAPTER 4
ELECTRICAL INTERFACE CIRCUITS AND DIAGNOSTICS FOR THE DCU AND DCM

This chapter describes the port characteristics, cables, and diagnostics for the DCU and DCM. Since the characteristics of the communications port on the DCU and DCM are nearly identical, the information in this chapter (with marked exceptions) applies to both.

PORT CHARACTERISTICS

The communications port on the DCU and DCM is a 25-pin, female, D-type connector. The pin definitions for the port are given below.

TPK.A.40375



PIN *	SIGNAL DEFINITION
7	LOGIC GROUND
10	RTS, RS-422 OUTPUT+ (DAISY CHAIN OUT)
11	RTS, RS-422 OUTPUT− (DAISY CHAIN OUT)
12	CTS (CLEAR TO SEND), RS-422 INPUT +
13	CTS, RS-422 INPUT −
14	TRANSMIT DATA, RS-422 OUTPUT + (DAISY CHAIN OUT)
15	TRANSMIT DATA, RS-422 OUTPUT − (DAISY CHAIN OUT)
16	RECEIVE DATA, RS-422 INPUT − (DAISY CHAIN OUT)
17	RECEIVE DATA, RS-422 INPUT + (DAISY CHAIN OUT)
19	KEYOUT RELAY (+) } -FOR DCM (SERIES THREE ONLY)
20	KEYOUT RELAY (−) } NO CONNECTION FOR DCU
22	TRANSMIT DATA, RS-422 OUTPUT + (DAISY CHAIN IN)
23	TRANSMIT DATA, RS-422 OUTPUT − (DAISY CHAIN IN)
24	RECEIVE DATA, RS-422 INPUT − (DAISY CHAIN IN)
25	RECEIVE DATA, RS-422 INPUT + (DAISY CHAIN IN)

* ONLY PINS WITH SIGNAL CONNECTIONS ARE LISTED

Figure 4.1 COMMUNICATIONS CONNECTOR PIN ASSIGNMENTS

COMMUNICATIONS PORT MATING CONNECTOR

A mating 25-pin male, D-type connector is provided with each DCM and DCU. Use Figure 4.2 as a guide to assemble this connector.

TPK.A.40009

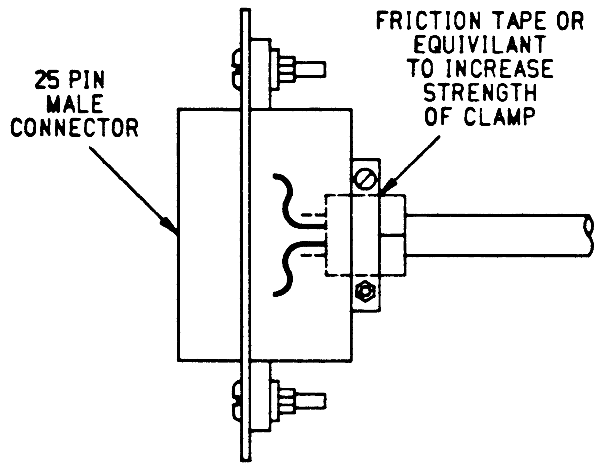


Figure 4.2 ASSEMBLY OF MATING CONNECTOR

CABLE SELECTION

The following cables will provide acceptable operation, at a maximum of 4000 feet (1200 meters) and a maximum transmission rate of 19.2 kbps, for an RS-422 communication system using DCUs or DCMs when other guidelines are followed:

Manufacturer	Manufacturer's Number
BELDEN	9184
BELDEN	9302
NEC	222PISLCBT

(Equivalents of these cables will provide acceptable operation).

Under conditions where electrical noise is low, it may be possible to extend the maximum distances.

CATALOG NUMBERS FOR GE SUPPLIED CABLES

Some fixed length cables as listed below can be purchased through GE.

DESCRIPTION	CATALOG NUMBER	LENGTH
Workmaster to Adapter Unit	IC630CBL390B	3 feet (1 meter)
DCU or DCM to Asynchronous/Joystick Card	IC630CBL391A	13 feet (4 meters)
DCU or DCM to Adapter Unit	IC630CBL392A	10 feet (3 meters)
Comms Link/Test Connector	IC630CCM394A	-

GROUNDING

CAUTION

CARE SHOULD BE EXERCISED TO ENSURE THAT BOTH THE DCU OR DCM AND THE DEVICE TO WHICH IT IS CONNECTED ARE GROUNDED TO A COMMON POINT IN DIRECT CONNECTIONS. FAILURE TO DO SO COULD RESULT IN DAMAGE TO THE EQUIPMENT.

RS-422 DIRECT CABLE DIAGRAMS

The RS-422 signal nomenclature used in this manual can be cross referenced to the RS-422 EIA standard as follows:

CCM SIGNAL NAME	RS-422 STANDARD SIGNAL NAME
RS-422 out + (TXD+)	B
RS-422 out - (TXD-)	A
RS-422 in + (RXD+)	B'
RS-422 in - (RXD-)	A'

During a mark condition (logic 1), B will be positive with respect to A. During a space condition (logic 0), B will be negative with respect to A.

When connecting the DCU or DCM to a non-Series Six master device using the RS-422 standard, the non-Series Six device's line receiver must contain "fail safe" capabilitiy. This means that in an idle, open, or shorted line condition, the output of the line receiver chip must assume the "marking" state.

NOTE

> When using RS-422, the twisted pairs should 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 not done, cross-talk can occur and severely affect the performance of the communication system.

SELECTION OF TERMINATING RESISTORS

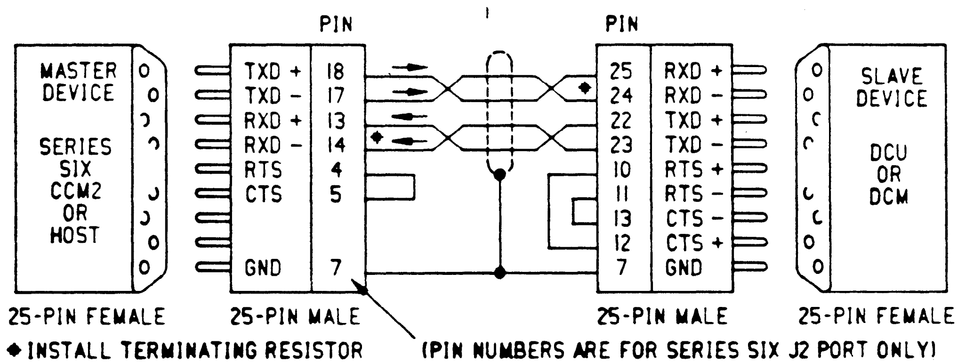
It is necessary to terminate an RS-422 link with the proper resistance in order to minimize reflection on the line. For point-to-point links with a master and a single slave, the factory-supplied resistor with a value of 150 ohms has been found to provide satisfactory termination for cable lengths of 10 feet to 4000 feet.

This resistor should be installed in the connector at either end of a point-to-point or multidrop link between the receive data (+) and receive data (-) pins. No termination resistor is needed for intermediate drops on a multidrop link. The daisy chain out connections are provided to allow direct soldering of the terminating resistor.

In a multidrop configuration (where terminating resistors are installed at the first and last drops only), it may be necessary to replace the factory supplied terminating resistor at the last active receiver in the communication link. This resistor should be between 120 ohms and 240 ohms; its actual value will vary with the distance from the master transmitter and the number of drops on the multidrop link.

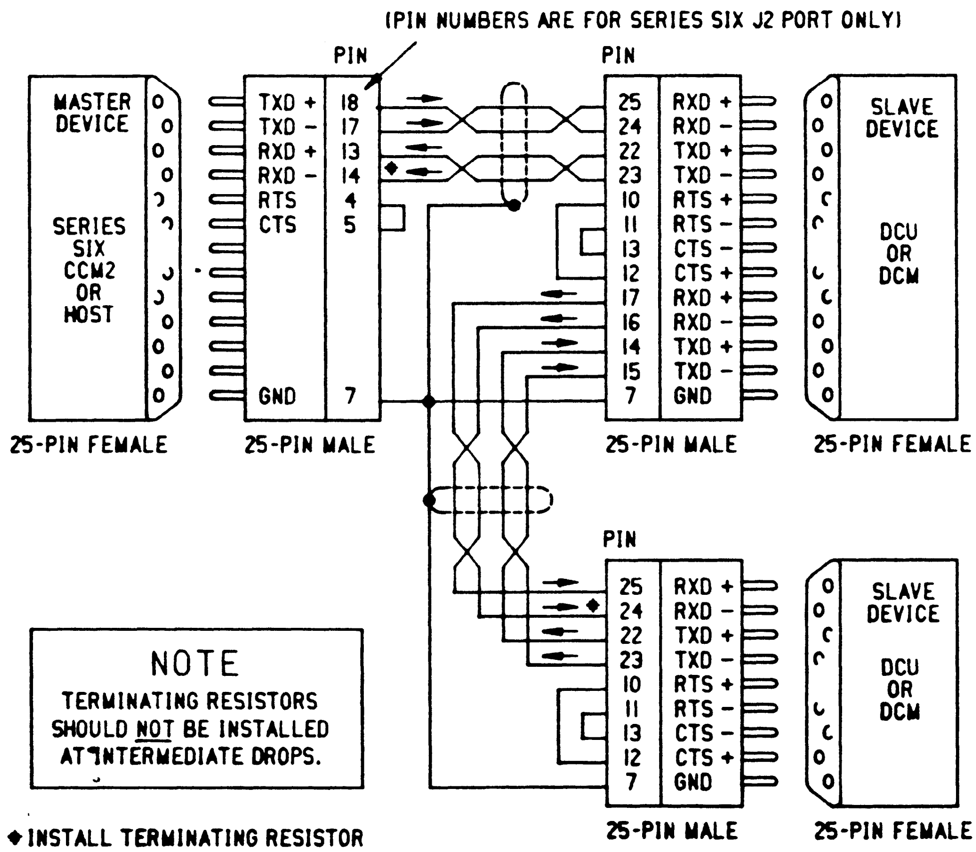
POINT-TO-POINT DCU OR DCM TO SERIES SIX CCM OR HOST COMPUTER

TPK.A.40225



MULTIDROP RS-422 CABLE, 4-WIRE

TPK.A.40227



RS-422 LINK CONNECTOR

To simplify the user wiring associated with 4-wire multidrop configurations, two sets of RS-422 terminations are provided in the connector (daisy chain in and daisy chain out). This allows you to have only one wire or solder connection per pin. In the event that a DCU or DCM on an intermediate drop is disconnected from the chain, however, a link connector (catalog number IC630CCM394A) must be installed on the connector of the disconnected drop to enable communications further down the link. Figure 4.3 illustrates the link connector.

TPK.A.40008

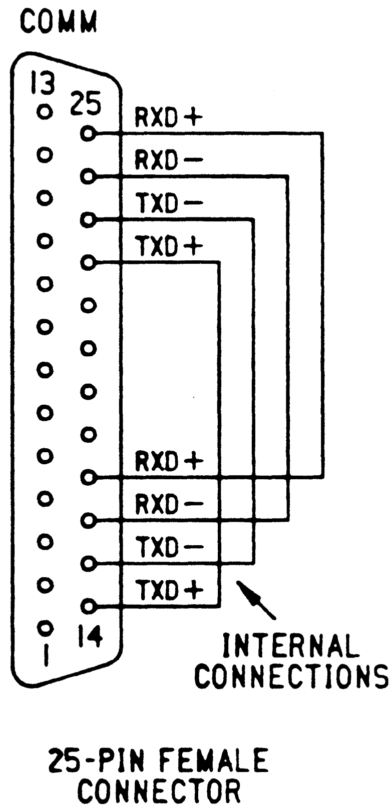


Figure 4.3 LINK CONNECTOR USED WHEN A DCU OR DCM IS REMOVED FROM A MULTIDROP CHAIN

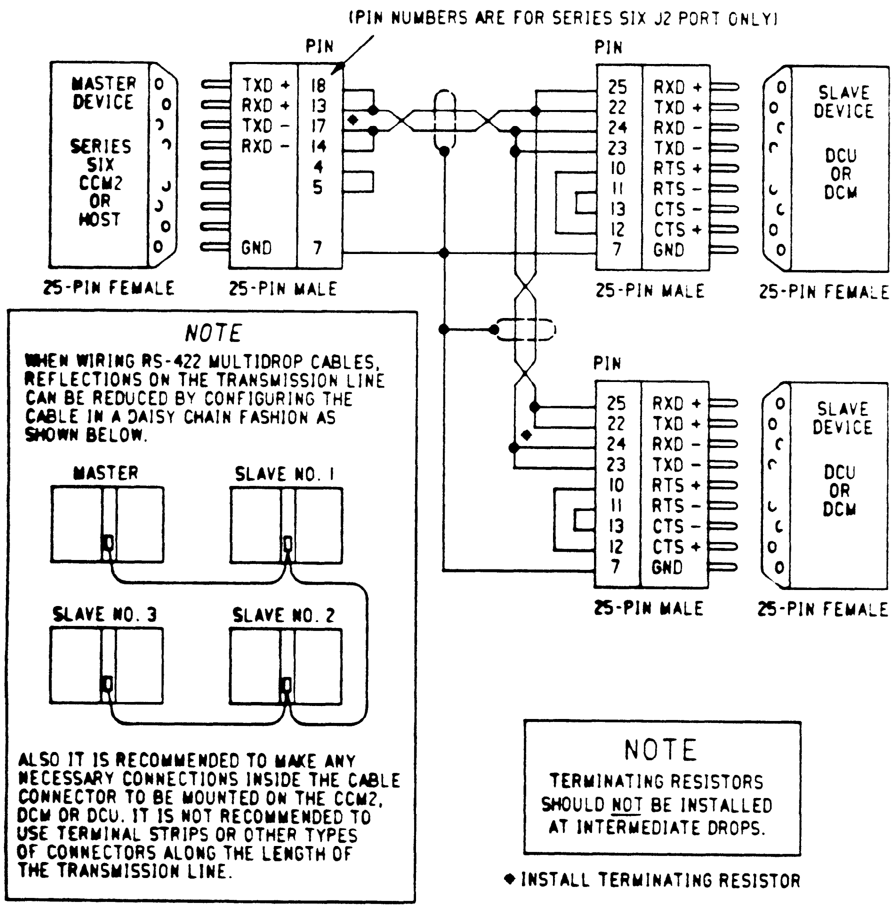
MULTIDROP RS-422 CABLE, 2-WIRE

NOTE

A two-wire RS-422 multidrop link may be implemented by tying RXD+ and TXD+ together at the DCU or DCM. This results in one signal path which is a 2-wire RS-422 multidrop.

When implementing a 2-wire RS-422 link with a host, the host must contain a tri-state transmitter which maintains idle lines in a high-impedance state. Also, some host equipment may not allow tying RXD and TXD together. In this case, the user must use the 4-wire multidrop.

TPK.A.40228



MODEM CONFIGURATION CABLE DIAGRAMS

In many cases, it is impossible to obtain a direct connection between elements of a communications system. If greater distance between elements is needed, modems can be introduced into the configuration.

The modems used on multidrop links must be switched-carrier, carrier-sense, full-duplex modems. These modems allow Request-to-Send/Clear-to-Send control of the modem. The modem carrier is turned on by the same signal that controls data transmission in the direct connection.

The RTS and CTS signals correspond to the Standard Data Terminal Equipment usage as explained below.

- When the DCU or DCM is not transmitting, the handshake output line (RTS) is in the false state.
- When the DCU or DCM has received a command to transmit some data, the handshake output line is set to true.
- After an optional turn-around delay, the DCU or DCM will check the handshake input line (CTS) and begin transmitting the data if the handshake input line is true.
- When the DCU or DCM has completed transmitting data, the handshake output line (RTS) will be set false.
- If the handshake input line (CTS) changes back to false before the DCU or DCM is finished transmitting, the DCU or DCM will stop transmitting at a character boundary and wait for the handshake input line (CTS) to change back to true.
- When flow control is used, the device implementing it must also guarantee that (CTS) will become false anytime (RTS) is set to false at the end of a data block.

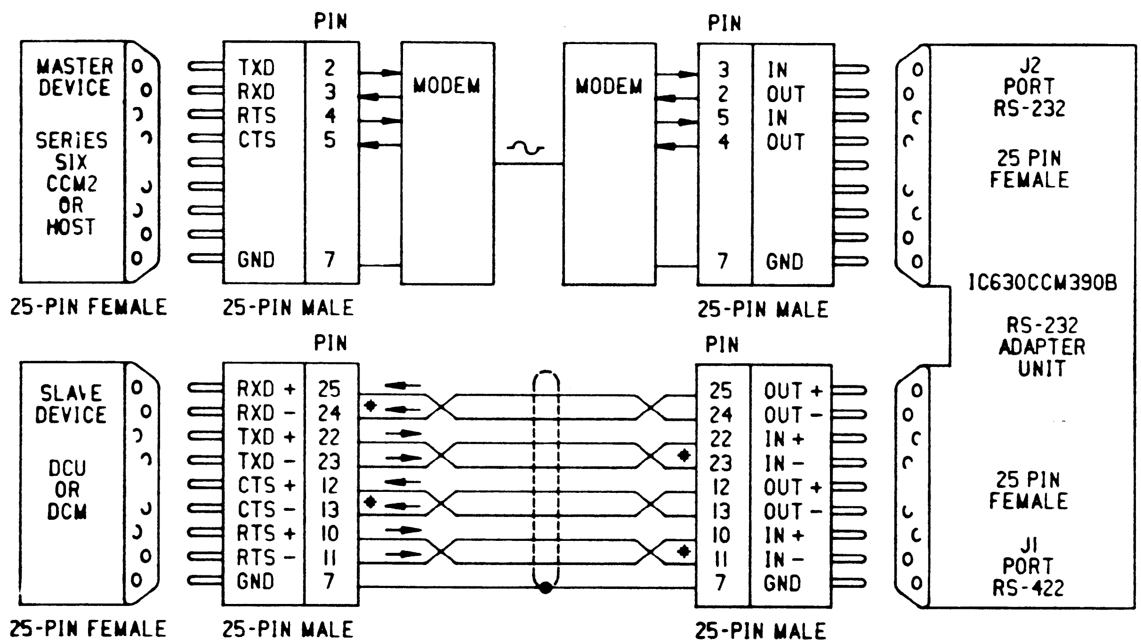
These rules explain the transmit function only. The standard DTE data receive function is independent of the RTS and CTS handshake lines. The DTE is able to receive data at any time.

NOTE

If RTS and CTS are not being used for modem control, these signals must be jumpered together at the DCU or DCM connector or the RS-232 connector of the adapter unit.

POINT-TO-POINT MODEM CONFIGURATION CABLE DIAGRAM

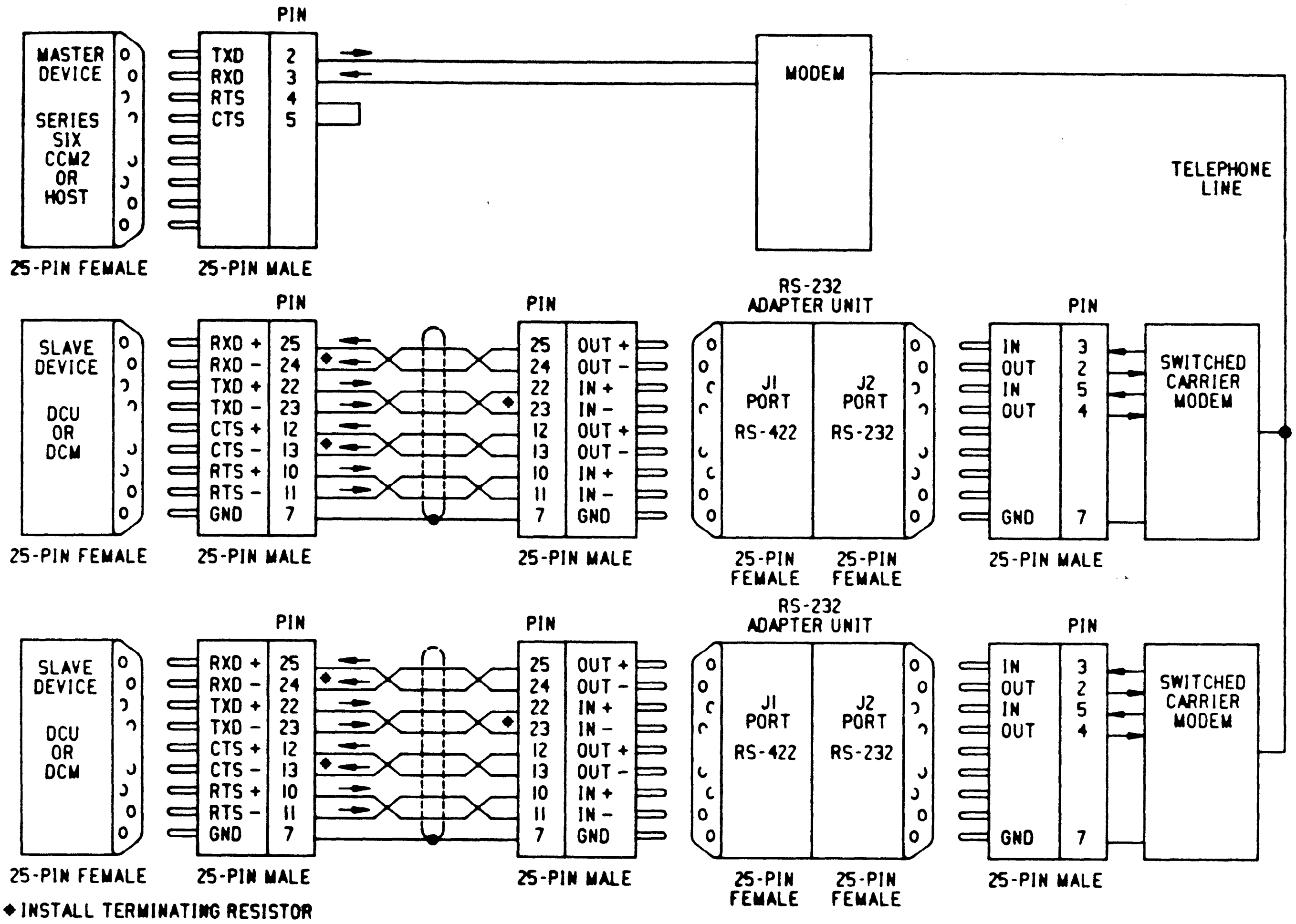
TPK.A.40229



◆ INSTALL TERMINATING RESISTOR

MULTIDROP MODEM CONFIGURATION CABLE DIAGRAM

TPK.B.40230



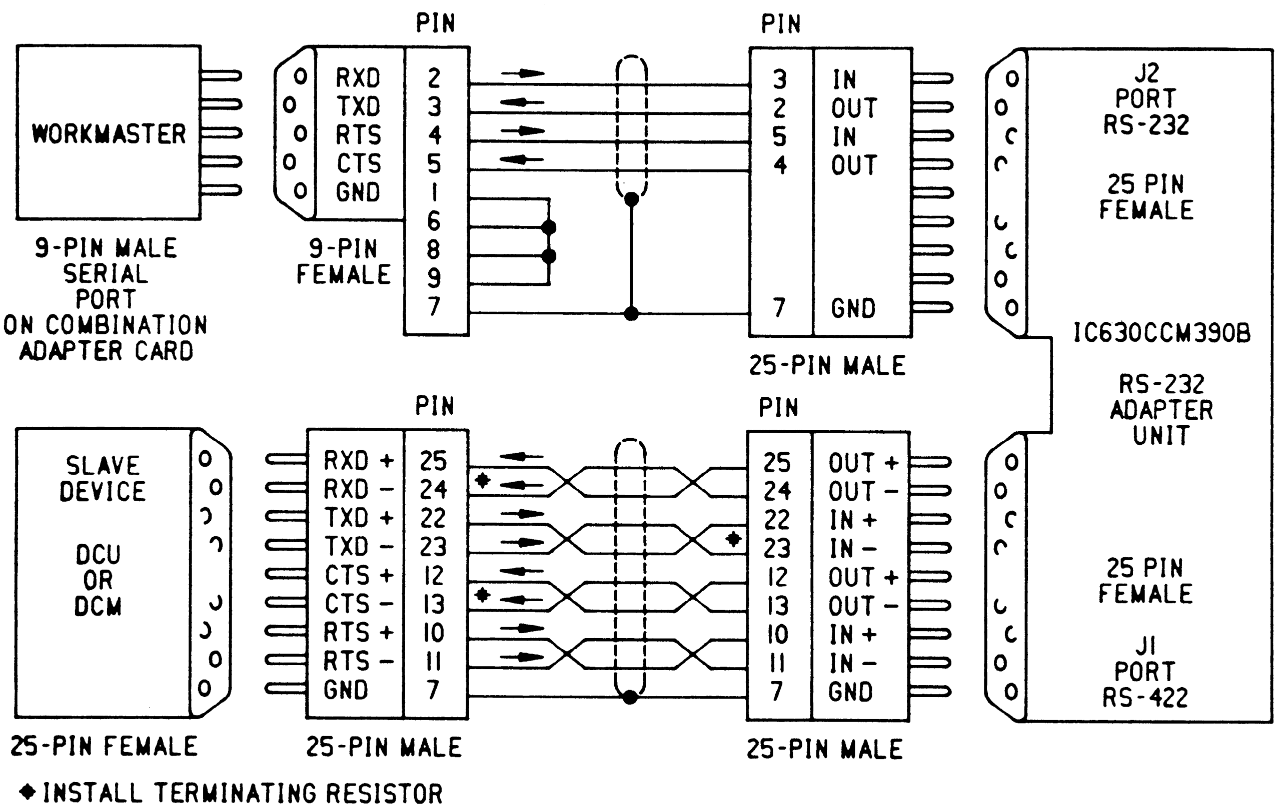
DCU OR DCM TO WORKMASTER COMPUTER CABLE DIAGRAMS

The DCU or DCM can be connected to a Workmaster computer (operating as a host) in two ways:

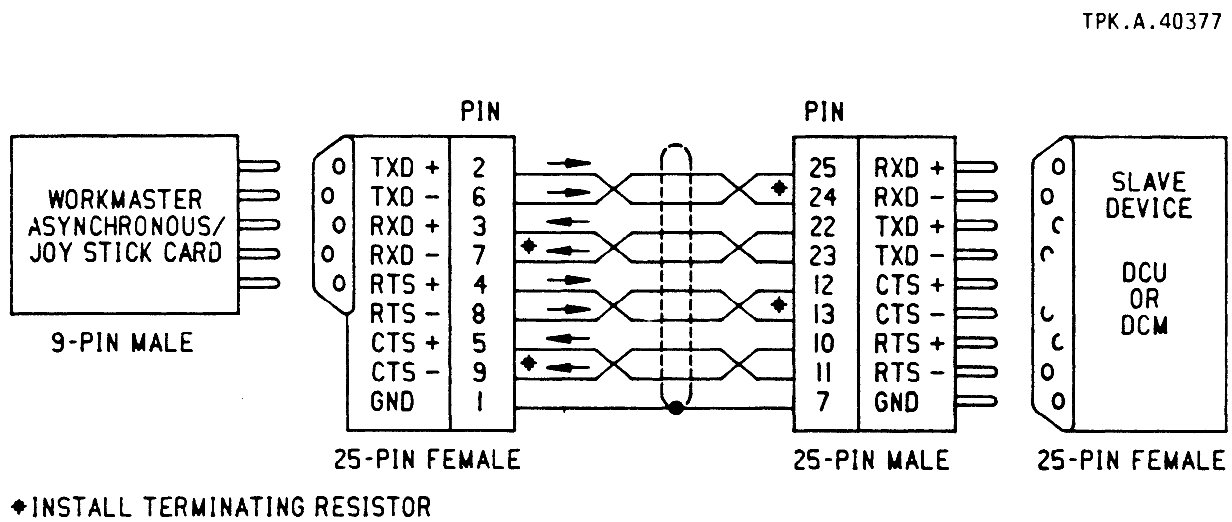
- From the DCU or DCM through the Adapter Unit (IC630CCM390B) to the Workmaster RS-232 port on the Combination Adapter Card, or
- Directly from the DCU or DCM through the RS-422 port on the optional Workmaster Asynchronous/Joystick Interface card (IC640BGB311A).

DCU OR DCM TO WORKMASTER THROUGH THE INTERFACE ADAPTER

TPK.A.40376



DCU OR DCM TO WORKMASTER DIRECTLY THROUGH THE RS-422 PORT



TEST DIAGNOSTICS

There are three sets of diagnostics which are performed upon the DCU and DCM hardware. These tests verify that the on board hardware is in working order.

POWER-UP DIAGNOSTICS

When the DCU or DCM is powered up, the following diagnostic test is run.

1. A write/read test is performed on all of the DCU or DCM RAM.
2. A checksum is calculated on all of the DCU or DCM PROM. The result is compared to a pre-calculated value that is stored in PROM.
3. The communication USARTS are programmed and checked for proper operation.

If any of the above tests fail, the DIAG LED is turned off and the DCU or DCM is inoperable. When the power is cycled, the DCU or DCM is reset and the above tests are performed.

LOOP-BACK DIAGNOSTICS

The loop-back diagnostics test the DCU or DCM hardware and communications connector. To execute the diagnostics, the DCU or DCM must be Off Line and connected to the CPU. Also, the loop-back test must be selected by placing configuration DIP switch 5 in the ON position.

The loop-back test performs the following test sequence:

1. The power-up diagnostics above are performed. If these diagnostics fail, the DIAG LED will be turned OFF and if the diagnostics pass, the DIAG LED will be ON.

2. A serial loop-back test using the special test connector shown in Figure 4.4 is performed. This procedure verifies that all of the serial interface hardware is operational.

   A test pattern is written to the communications port. The received pattern is then compared to the transmitted pattern for error detection.

   When executing the Loop-Back Diagnostics, the DATA LED Will be ON if the diagnostic testing is passing and BLINKING if the loop-back verification is being attempted but is not passing.

3. With the DCU connected to the Series One or Series One Junior CPU or the DCM connected to the Series Three CPU, a request will be made for data from the CPU. If this request is honored, the DATA LED will remain ON and if the request fails, the DATA LED will be turned OFF.

TPK.A.40158

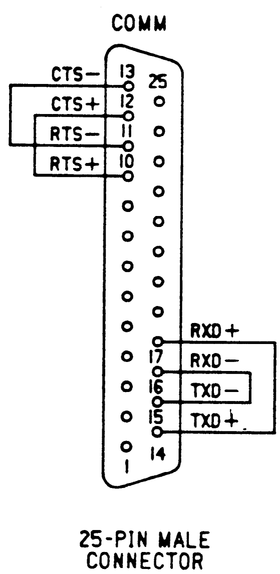


Figure 4.4 LOOP-BACK TEST CONNECTOR

CHAPTER 5
COMMUNICATION EXAMPLES USING THE SERIES SIX PC AS A MASTER DEVICE

This chapter explains how to build the Series Six ladder diagram to initiate communications between a Series Six PC and a Series One, Series One Junior, Series One Plus, or Series Three PC.

INTRODUCTION

When a Series Six PC is part of a communications link with a Series One/Junior/Plus or Series Three PC, the Series Six PC is the master and therefore the only initiator of communications.

The SCREQ function programmed into the Series Six CPU must be executed to initiate communications. The Communications Control Module (CCM2 or CCM3 in CCM2 mode) in the Series Six CPU rack uses the information supplied by this function to establish communications with the DCU or DCM and execute a transfer of data to or from the Series One, Series One Junior, Series One Plus, or Series Three PC.

Refer to the Series Six Data Communications Manual, GEK-25364, for details on using the SCREQ command. There are differences in memory types between Series One, Series One Junior, Series One Plus, or Series Three PCs and the Series Six PC which affect the programming of the SCREQ command registers. The differences are explained in this chapter, and a number of application examples are included to assist the reader.

NOTE

CCM2 PROM Revision D or later is required for communications with the DCU or DCM.

CCM3 PROM Revision C or later is required for communciations with the DCU or DCM.

The revision letter can be found on the labels attached to the socketed PROMS located on the component side of the module. On this label is a 3-digit number followed by a dash followed by a 3-digit number. The revision letter is after the second 3-digit number, and it may differ from PROM to PROM on the module. The correct revision letter is the highest of the letters.

SCREQ REGISTERS

The six SCREQ registers are defined as follows:

Rnnnn	Command Number (must be valid for DCU or DCM)
Rnnnn + 1	Target* ID
Rnnnn + 2	Target Memory Type
Rnnnn + 3	Target Memory Address
Rnnnn + 4	Data Length
Rnnnn + 5	Source* Memory Address

* In Series Six to Series One/Junior/Plus or Series Three communications, the target is always the Series One/Junior/Plus or Series Three PC and the source is always the Series Six PC.

Rnnnn: COMMAND NUMBERS

Port J1 of CCM2

06100 (17D4H)	No Op	
06101 (17D5H)	READ from target to source	Register Table
06102 (17D6H)	READ from target to source	Input Table
06103 (17D7H)	READ from target to source	Output Table
06111 (17DFH)	WRITE to target from source	Register Table
06112 (17E0H)	WRITE to target from source	Input Table
06113 (17E1H)	WRITE to target from source	Output Table

Port J2 of CCM2

06200 (1838H)	No Op	
06201 (1839H)	READ from target to source	Register Table
06202 (183AH)	READ from target to source	Input Table
06203 (183BH)	READ from target to source	Output Table
06211 (1843H)	WRITE to target from source	Register Table
06212 (1844H)	WRITE to target from source	Input Table
06213 (1845H)	WRITE to target from source	Output Table

Rnnnn + 1: TARGET ID

This is the identification number of the target device. For a Series One/Junior/Plus or Series Three CPU, this number is the DCU or DCM ID number and can range from 1 to 90.

Rnnnn + 2: **TARGET MEMORY TYPE**

The target memory types used with the Series One/Junior/Plus and Series Three PCs are:

Number	Type
1	Timer/Counter Accumulators and Data Registers*
3	Discrete I/O
6	CPU Scratch Pad Memory
7	User Logic Memory
9	DCU or DCM Diagnostic Status Words

*Data Registers exist in the Series Three CPU only.

Rnnnn + 3: **TARGET MEMORY ADDRESS**

The target memory address specifies the relative address within the Series One, Series One Junior, Series One Plus, or Series Three CPU where the transfer is to begin. The valid ranges given below are for communications initiated by the Series Six PC.

Memory Type 1: The target memory address specifies the Timer/Counter or Data Register where the data transfer is to begin. See Tables 5.1, 5.2, 5.3 and 5.4 for the mapping of Series One, Series One Junior, Series One Plus and Series Three reference numbers into reference numbers used for communication. Also, see application examples 1, 6, 7, 8, 12, 13, 14 and 15.

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	1-64 decimal	1-21 decimal	1-128 decimal	1-192 decimal

Memory Type 3: The target memory address specifies the group of 8 discrete I/O points where the data transfer is to begin. See Tables 5.1, 5.2, 5.3, and 5.4 for mapping of Series One, Series One Junior, Series One Plus, and Series Three discrete I/O reference numbes into reference numbers used for communication. Also, see application examples 2, 3, 16, and 17.

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	1-48 decimal	1-32 decimal	1-64 decimal	1-128 decimal

Memory Type 6: The target memory address specifies the CPU Scratch-Pad byte (8-bits) at which the data transfer is to begin. Address 0 is used to access the RUN/STOP mode and address 22 is used to access the PC type. Two bytes must be read or written. See application examples 9, 10, 11, 20, 21, and 22.

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	0 or 22 dec	0 or 22 dec	0, 2, 4, or 22 dec	0 to 22 dec

The Scratch-Pad for the Series One Plus has been expanded to accommodate the password and program error check features. See the section "Using the Password and Error Checking Features of the Series One Plus PC", later in this chapter.

Memory Type 7: The target memory address specifies the User-Logic memory word (16 bits) at which the data transfer is to begin. See application examples 4, 5, 18, and 19.

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	0-1723 dec	0-699 dec	0-1723 dec	0-4094 dec

Memory Type 9: The Target Address specifies the DCU or DCM Diagnostic Status Word (16 bits) at which the data transfer is to begin. The only valid starting address for Series One, Series One Junior, Series One Plus, and Series Three is 0. See application examples 23, 24.

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	0 decimal	0 decimal	0 decimal	0 decimal

Table 5.1 MAPPING OF SERIES ONE REFERENCES TO TARGET ADDRESSES (MEMORY TYPES 1 AND 3)

MEMORY TYPE	SERIES ONE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES ONE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES ONE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES ONE REFERENCE	MAPPED ADDRESS DEC	HEX
Type 1												
Timers/	600 ...	01	01	620 ...	17	11	640 ...	33	21	660 ...	49	31
Counters	601 ...	02	02	621 ...	18	12	641 ...	34	22	661 ...	50	32
.	602 ...	03	03	622 ...	19	13	642 ...	35	23	662 ...	51	33
.	603 ...	04	04	623 ...	20	14	643 ...	36	24	663 ...	52	34
.	604 ...	05	05	624 ...	21	15	644 ...	37	25	664 ...	53	35
.	605 ...	06	06	625 ...	22	16	645 ...	38	26	665 ...	54	36
.	606 ...	07	07	626 ...	23	17	646 ...	39	27	666 ...	55	37
.	607 ...	08	08	627 ...	24	18	647 ...	40	28	667 ...	56	38
.	610 ...	09	09	630 ...	25	19	650 ...	41	29	670 ...	57	39
.	611 ...	10	0A	631 ...	26	1A	651 ...	42	2A	671 ...	58	3A
.	612 ...	11	0B	632 ...	27	1B	652 ...	43	2B	672 ...	59	3B
.	613 ...	12	0C	633 ...	28	1C	653 ...	44	2C	673 ...	60	3C
.	614 ...	13	0D	634 ...	29	1D	654 ...	45	2D	674 ...	61	3D
.	615 ...	14	0E	635 ...	30	1E	655 ...	46	2E	675 ...	62	3E
.	616 ...	15	0F	636 ...	31	1F	656 ...	47	2F	676 ...	63	3F
.	617 ...	16	10	637 ...	32	20	657 ...	48	30	677 ...	64	40
Type 3												
External	000-007...	01	01	100-107...	09	09						
I/O	010-017...	02	02	110-117...	10	0A						
.	020-027...	03	03	120-127...	11	0B						
.	030-037...	04	04	130-137...	12	0C						
.	040-047...	05	05	140-147...	13	0D						
.	050-057...	06	06	150-157...	14	0E						
.	060-067...	07	07									
.	070-077...	08	08									
Internal	160-167...	15	0F	260-267...	23	17	360-367...	31	1F			
Coils	170-177...	16	10	270-277...	24	18	370-377...	32	20			
.	200-207...	17	11	300-307...	25	19						
.	210-217...	18	12	310-317...	26	1A						
.	220-227...	19	13	320-327...	27	1B						
.	230-237...	20	14	330-337...	28	1C						
.	240-247...	21	15	340-347...	29	1D						
.	250-257...	22	16	350-357...	30	1E						
Shift	400-407...	33	21	500-507...	41	29						
Register	410-417...	34	22	510-517...	42	2A						
Points	420-427...	35	23	520-527...	43	2B						
.	430-437...	36	24	530-537...	44	2C						
.	440-447...	37	25	540-547...	45	2D						
.	450-457...	38	26	550-557...	46	2E						
.	460-467...	39	27	560-567...	47	2F						
.	470-477...	40	28	570-577...	48	30						

Table 5.2 MAPPING OF SERIES ONE JR REFERENCES TO TARGET ADDRESSES (MEMORY TYPES 1 AND 3)

MEMORY TYPE	SERIES ONE JUNIOR REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES ONE JUNIOR REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES ONE JUNIOR REFERENCE	MAPPED ADDRESS DEC	HEX
Type (1)									
Timers/	600 ...	01	01	610 ...	09	09	620 ...	17	11
Counters	601 ...	02	02	611 ...	10	0A	621 ...	18	12
.	602 ...	03	03	612 ...	11	0B	622 ...	19	13
.	603 ...	04	04	613 ...	12	0C	623 ...	20	14
.	604 ...	05	05	614 ...	13	0D	624 ...	21	15
.	605 ...	06	06	615 ...	14	0E			
.	606 ...	07	07	616 ...	15	0F			
.	607 ...	08	08	617 ...	16	10			
Type 3									
External	000-007...	01	01						
I/O	010-017...	02	02						
.	020-027...	03	03						
.	030-037...	04	04						
.	040-047...	05	05						
.	050-057...	06	06						
.	060-067...	07	07						
.	070-077...	08	08						
.	130-137...	12	0C						
Internal	140-147...	13	0D	240-247...	21	15	340-347...	29	1D
Coils	150-157...	14	0E	250-257...	22	16	350-357...	30	1E
.	160-167...	15	0F	260-267...	23	17	360-367...	31	1F
.	170-177...	16	10	270-277...	24	18	370-377...	32	20
.	200-207...	17	11	300-307...	25	19			
.	210-217...	18	12	310-317...	26	1A			
.	220-227...	19	13	320-327...	27	1B			
.	230-237...	20	14	330-337...	28	1C			
Shift	140-147...	13	0D	240-247...	21	15	340-347...	29	1D
Register	150-157...	14	0E	250-257...	22	16	350-357...	30	1E
Points	160-167...	15	0F	260-267...	23	17	360-367...	31	1F
.	170-177...	16	10	270-277...	24	18	370-377...	32	20
.	200-207...	17	11	300-307...	25	19			
.	210-217...	18	12	310-317...	26	1A			
.	220-227...	19	13	320-327...	27	1B			
.	230-237...	20	14	330-337...	28	1C			

Table 5.3 MAPPING OF SERIES ONE PLUS REFERENCES TO TARGET ADDRESSES (MEMORY TYPE 1)

MEMORY TYPE	SERIES ONE PLUS REFERENCE	MAPPED ADDRESS DEC	MAPPED ADDRESS HEX	SERIES ONE PLUS REFERENCE	MAPPED ADDRESS DEC	MAPPED ADDRESS HEX	SERIES ONE PLUS REFERENCE	MAPPED ADDRESS DEC	MAPPED ADDRESS HEX	SERIES ONE PLUS REFERENCE	MAPPED ADDRESS DEC	MAPPED ADDRESS HEX
Type 1												
Timers/	600	01	01	620	17	11	640	33	21	660	49	31
Counters	601	02	02	621	18	12	641	34	22	661	50	32
.	602	03	03	622	19	13	642	35	23	662	51	33
.	603	04	04	623	20	14	643	36	24	663	52	34
.	604	05	05	624	21	15	644	37	25	664	53	35
.	605	06	06	625	22	16	645	38	26	665	54	36
.	606	07	07	626	23	17	646	39	27	666	55	37
.	607	08	08	627	24	18	647	40	28	667	56	38
.	610	09	09	630	25	19	650	41	29	670	57	39
.	611	10	0A	631	26	1A	651	42	2A	671	58	3A
.	612	11	0B	632	27	1B	652	43	2B	672	59	3B
.	613	12	0C	633	28	1C	653	44	2C	673	60	3C
.	614	13	0D	634	29	1D	654	45	2D	674	61	3D
.	615	14	0E	635	30	1E	655	46	2E	675	62	3E
.	616	15	0F	636	31	1F	656	47	2F	676	63	3F
.	617	16	10	637	32	20	657	48	30	677	64	40
Data	400-401	65	41	440-441	81	51	500-501	97	61	540-541	113	71
Registers	402-403	66	42	442-443	82	52	502-503	98	62	542-543	114	72
.	404-405	67	43	444-445	83	53	504-505	99	63	544-545	115	73
.	406-407	68	44	446-447	84	54	506-507	100	64	546-547	116	74
.	410-411	69	45	450-451	85	55	510-511	101	65	550-551	117	75
.	412-413	70	46	452-453	86	56	512-513	102	66	552-553	118	76
.	414-415	71	47	454-455	87	53	514-515	103	67	554-555	119	77
.	416-417	72	48	456-457	88	58	516-517	104	68	556-557	120	78
.	420-421	73	49	460-461	89	59	520-521	105	69	560-561	121	79
.	422-423	74	4A	462-463	90	5A	522-523	106	6A	562-563	122	7A
.	424-425	75	4B	464-465	91	5B	524-525	107	6B	564-565	123	7B
.	426-427	76	4C	466-467	92	5C	526-527	108	6C	566-567	124	7C
.	430-431	77	4D	470-471	93	5D	530-531	109	6D	570-571	125	7D
.	432-433	78	4E	472-473	94	5E	532-533	110	6E	572-573	126	7E
.	434-435	79	4F	474-475	95	5F	534-535	111	6F	574-575	127	7F
.	436-437	80	50	476-477	96	60	536-537	112	70	576-577	128	80

Table 5.3 (Cont.) MAPPING OF SERIES ONE PLUS REFERENCES TO TARGET ADDRESSES (MEMORY TYPE 3)

MEMORY TYPE	SERIES ONE PLUS REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES ONE PLUS REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES ONE PLUS REFERENCE	MAPPED ADDRESS DEC	HEX
Type 3									
External	000-007...	01	01	100-107...	09	09	700-707...	57	39
I/O	010-017...	02	02	110-117...	10	0A	710-717...	58	3A
.	020-027...	03	03	120-127...	11	0B	720-727...	59	3B
.	030-037...	04	04	130-137...	12	0C	730-737...	60	3C
.	040-047...	05	05	140-147...	13	0D	740-747...	61	3D
.	050-057...	06	06	150-157...	14	0E	750-757...	62	3E
.	060-067...	07	07				760-767...	63	3F
.	070-077...	08	08				770-777...	64	40
Internal	160-167...	15	0F	260-267...	23	17	360-367...	31	1F
Coils	170-177...	16	10	270-277...	24	18	370-377...	32	20
.	200-207...	17	11	300-307...	25	19			
.	210-217...	18	12	310-317...	26	1A			
.	220-227...	19	13	320-327...	27	1B			
.	230-237...	20	14	330-337...	28	1C			
.	240-247...	21	15	340-347...	29	1D			
.	250-257...	22	16	350-357...	30	1E			
Shift	400-407...	33	21	500-507...	41	29			
Register	410-417...	34	22	510-517...	42	2A			
Points	420-427...	35	23	520-527...	43	2B			
.	430-437...	36	24	530-537...	44	2C			
.	440-447...	37	25	540-547...	45	2D			
.	450-457...	38	26	550-557...	46	2E			
.	460-467...	39	27	560-567...	47	2F			
.	470-477...	40	28	570-577...	48	30			
Timer/	600-607...	49	31						
Counter	610-617...	50	32						
Up Status	620-627...	55	33						
.	630-637...	52	34						
.	640-647...	53	35						
.	650-657...	54	36						
.	660-667...	55	37						
.	670-677...	56	38						

Table 5.4 MAPPING OF SERIES THREE REFERENCES TO TARGET ADDRESSES (MEMORY TYPE 1)

MEMORY TYPE	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX
Type 1												
Data	500-501	01	01	540-541	17	11	600-601	33	21	640-641	49	31
Registers	502-503	02	02	542-543	18	12	602-603	34	22	642-643	50	32
.	504-505	03	03	544-545	19	13	604-605	35	23	644-645	51	33
.	506-507	04	04	546-547	20	14	606-607	36	24	646-647	52	34
.	510-511	05	05	550-551	21	15	610-611	37	25	650-651	53	35
.	512-513	06	06	552-553	22	16	612-613	38	26	652-653	54	36
.	514-515	07	07	554-555	23	17	614-615	39	27	654-655	55	37
.	516-517	08	08	556-557	24	18	616-617	40	28	656-657	56	38
.	520-521	09	09	560-561	25	19	620-621	41	29	660-661	57	39
.	522-523	10	0A	562-563	26	1A	622-623	42	2A	662-663	58	3A
.	524-525	11	0B	564-565	27	1B	624-625	43	2B	664-665	59	3B
.	526-527	12	0C	566-567	28	1C	626-627	44	2C	666-667	60	3C
.	530-531	13	0D	570-571	29	1D	630-631	45	2D	670-671	61	3D
.	532-533	14	0E	572-573	30	1E	632-633	46	2E	672-673	62	3E
.	534-535	15	0F	574-575	31	1F	634-635	47	2F	674-675	63	3F
.	536-537	16	10	576-577	32	20	636-637	48	30	676-677	64	40
Timer/	200	65	41	240	97	61	300	129	81	340	161	A1
Counter	201	66	42	241	98	62	301	130	82	341	162	A2
Accumulators	202	67	43	242	99	63	302	131	83	342	163	A3
.	203	68	44	243	100	64	303	132	84	343	164	A4
.	204	69	45	244	101	65	304	133	85	344	165	A5
.	205	70	46	245	102	66	305	134	86	345	166	A6
.	206	71	47	246	103	67	306	135	87	346	167	A7
.	207	72	48	247	104	68	307	136	88	347	168	A8
.	210	73	49	250	105	69	310	137	89	350	169	A9
.	211	74	4A	251	106	6A	311	138	8A	351	170	A0
.	212	75	4B	252	107	6B	312	139	8B	352	171	AB
.	213	76	4C	253	108	6C	313	140	8C	353	172	AC
.	214	77	4D	254	109	6D	314	141	8D	354	173	AD
.	215	78	4E	255	110	6E	315	142	8E	355	174	AE
.	216	79	4F	256	111	6F	316	143	8F	356	175	AF
.	217	80	50	257	112	70	317	144	90	357	176	B0
.	220	81	51	260	113	71	320	145	91	360	177	B1
.	221	82	52	261	114	72	321	146	92	361	178	B2
.	222	83	53	262	115	73	322	147	93	362	179	B3
.	223	84	54	263	116	74	323	148	94	363	180	B4
.	224	85	55	264	117	75	324	149	95	364	181	B5
.	225	86	56	265	118	76	325	150	96	365	182	B6
.	226	87	57	266	119	77	326	151	97	366	183	B7
.	227	88	58	267	120	78	327	152	98	367	184	B8
.	230	89	59	270	121	79	330	153	99	370	185	B9
.	231	90	5A	271	122	7A	331	154	9A	371	186	BA
.	232	91	5B	272	123	7B	332	155	9B	372	187	BB
.	233	92	5C	273	124	7C	333	156	9C	373	188	BC
.	234	93	5D	274	125	7D	334	157	9D	374	189	BD
.	235	94	5E	275	126	7E	335	158	9E	375	190	BE
.	236	95	5F	276	127	7F	336	159	9F	376	191	BF
.	237	96	60	277	128	80	337	160	A0	377	192	C0

Table 5.4 (Cont.) MAPPING OF SERIES THREE REFERENCES TO TARGET ADDRESSES (MEMORY TYPE 3)

MEMORY TYPE	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX
Type 3												
External	000-007	01	01	200-207	17	11	400-407	33	21	600-607	49	31
I/O	010-017	02	02	210-217	18	12	410-417	34	22	610-617	50	32
.	020-027	03	03	220-227	19	13	420-427	35	23			
.	030-037	04	04	230-237	20	14	430-437	36	24			
.	040-047	05	05	240-247	21	15	440-447	37	25			
.	050-057	06	06	250-257	22	16	450-457	38	26			
.	060-067	07	07	260-267	23	17	460-467	39	27			
.	070-077	08	08	270-277	24	18	470-477	40	28			
.	100-107	09	09	300-307	25	19	500-507	41	29			
.	110-117	10	0A	310-317	26	1A	510-517	42	2A			
.	120-127	11	0B	320-327	27	1B	520-527	43	2B			
.	130-137	12	0C	330-337	28	1C	530-537	44	2C			
.	140-147	13	0D	340-347	29	1D	540-547	45	2D			
.	150-157	14	0E	350-357	30	1E	550-557	46	2E			
.	160-167	15	0F	360-367	31	1F	560-567	47	2F			
.	170-177	16	10	370-377	32	20	570-577	48	30			
Internal	4000-4007	51	33	4200-4207	67	43	4400-4407	83	53			
I/O	4010-4017	52	34	4210-4217	68	44	4410-4417	84	54			
.	4020-4027	53	35	4220-4227	69	45	4420-4427	85	55			
.	4030-4037	54	36	4230-4237	70	46	4430-4437	86	56			
.	4040-4047	55	37	4240-4247	71	47	4440-4447	87	57			
.	4050-4057	56	38	4250-4257	72	48	4450-4457	88	58			
.	4060-4067	57	39	4260-4267	73	49	7000-7007	89	59			
.	4070-4077	58	3A	4270-4277	74	4A	7010-7017	90	5A			
.	4100-4107	59	3B	4300-4307	75	4B	7020-7027	91	5B			
.	4110-4117	60	3C	4310-4317	76	4C	7030-7037	92	5C			
.	4120-4127	61	3D	4320-4327	77	4D	7040-7047	93	5D			
.	4130-4137	62	3E	4330-4337	78	4E	7050-7057	94	5E			
.	4140-4147	63	3F	4340-4347	79	4F	7060-7067	95	5F			
.	4150-4157	64	40	4350-4357	80	50	7070-7077	96	60			
.	4160-4167	65	41	4360-4367	81	51						
.	4170-4177	66	42	4370-4377	82	52						

Table 5.4 (Cont.) MAPPING OF SERIES THREE REFERENCES TO TARGET ADDRESSES (MEMORY TYPE 3)

MEMORY TYPE	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX	SERIES THREE REFERENCE	MAPPED ADDRESS DEC	HEX
Type 3												
Shift	9000-9007	.. 97	61	9100-9107	..105	69						
Registers	9010-9017	.. 98	62	9110-9117	..106	6A						
.	9020-9027	.. 99	63	9120-9127	..107	6B						
.	9030-9037	..100	64	9130-9137	..108	6C						
.	9040-9047	..101	65	9140-9147	..109	6D						
.	9050-9057	..102	66	9150-9157	..110	6E						
.	9060-9067	..103	67	9160-9167	..111	6F						
.	9070-9077	..104	68	9170-9177	..112	70						
Timer/	000-007	...113	71	100-107	...121	79						
Counter	010-017	...114	72	110-117	...122	7A						
Up Status	020-027	...115	73	120-127	...123	7B						
.	030-037	...116	74	130-137	...124	7C						
.	040-047	...117	75	140-147	...125	7D						
.	050-057	...118	76	150-157	...126	7E						
.	060-067	...119	77	160-167	...127	7F						
.	070-077	...120	78	170-177	...128	80						

Rnnnn + 4: **DATA LENGTH**

This is the data length of the source (Series Six) memory type.

To determine the source data length, it is necessary to compare the unit lengths of the source and target memory types.

Table 5.5 UNIT LENGTHS OF SOURCE AND TARGET MEMORY TYPES

SOURCE (SERIES SIX) MEMORY TYPE	UNIT LENGTH	LENGTH ACCESSIBLE
1: Registers	1 Reg = 16 bits	Register(s)
2, 3: Inputs and Outputs	1 Point = 1 bit	Multiples of 8 Points
SERIES ONE, JUNIOR/PLUS, SERIES THREE MEMORY TYPE	**UNIT LENGTH**	**LENGTH ACCESSIBLE**
1: Timer/Counter Accumulator	1 Accum = 16 bits	Accumulator(s)
1: Data Registers (Series One Plus and Three Only)	1 Data Reg = 8 bits	Multiples of 2 Reg
3: Discrete I/O	1 Point = 1 bit	Multiples of 8 Points
6: Scratch Pad Bytes	1 Byte = 8 bits	2 Bytes
7: User Logic Word	1 Word = 16 bits	Word(s)
9: Diagnostic Status Word	1 Word = 16 bits	5 Words

Example: If you want to read 5 target Timer/Counter accumulators into Series Six registers, the Data Length is 5 registers since the unit length is the same for each. However, if you want to read the 5 target Timer/Counter accumulators into Series Six inputs, the Data Length is 5 Accum. x 16 Points/Accum. = 80 Points.

Example: If you want to read 8 target discrete I/O into Series Six inputs, the Data Lenth is 8 points since the unit length is the same for each. Discrete I/O and Series Six I/O can only be accessed in multiples of 8.

Refer to the communication examples in this chapter for other combinations of target and source memory types.

Limitations on Amount of Data for the Series One and Series One Junior PCs

For communications with the Series One Plus and Series Three PCs, the maximum amount of data which can be transferred is limited only by the maximum size of the Series One Plus or Series Three memory type being accessed.

For communications with the Series One and Series One Junior PCs, the maximum amount of data which can be transferred is limited by the maximum size of memory types 6 (Scratch Pad) and 9 (Diagnostic Status Words). But the maximum amount of data which can be transferred is limited further for memory types 1 (T/C Accumulators), 3 (I/O and Shift Registers), and 7 (User Logic) as shown in Table 5.6.

Table 5.6 MAXIMUM AMOUNT OF DATA FOR SERIES ONE AND SERIES ONE JUNIOR MEMORY TYPES 1, 3, AND 7

TYPE OF COMMUNICATION REQUEST	MAXIMUM AMOUNT OF DATA FOR EACH COMMUNICATION			
	SERIES ONE PC*		SERIES ONE JR PC	
Read from Memory Type 1 (T/C Accumulators)	58 Acc	116 Bytes	All 21 Acc	42 Bytes
Write to Memory Type 1 (T/C Accumulators)	Communication Not Supported		Communication Not Supported	
Read from Memory Type 3 (I/O and Shift Reg)	368 I/O	46 Bytes	176 I/O	22 Bytes
Write to Memory Type 3 (I/O and Shift Reg)	24 I/O	3 Bytes	No I/O, Communication Times Out	
Read from Memory Type 7 (User Logic)	75 Words	150 Bytes	25 Words	50 Bytes
Write to Memory Type 7 (User Logic)	45 Words	90 Bytes	20 Words	40 Bytes

* CPU Revision C or later.

Rnnnn + 5: **SOURCE MEMORY ADDRESS**

This is the memory address of the source device (Series Six CPU) at which the transfer is to begin. The command number specifies the source memory type.

Table 5.7 SOURCE MEMORY ADDRESS

MEMORY TYPE	DESCRIPTION	SOURCE ADDRESS RANGE
Register Table	Model 60, 2K memory	1-256
	Model 60, 4K memory	1-1024
	Model 600, Model 6000 & Series Six Plus	1-1024*
Input Table	Input or output. The number must begin on the beginning of a byte boundary: 1, 9, 17	1-1024
Output Table		1-1024

*If the Series Six Model 600, 6000, or the Series Six Plus contains 8K of registers, then the range is 1-8192. If it contains 16K of registers, then the range is 1-16384.

USING THE PASSWORD AND ERROR CHECKING FEATURES OF THE SERIES ONE PLUS PC

The addressing for the Series One Plus Scratch-Pad is as follows:

Table 5.8 SERIES ONE PLUS CPU SCRATCH-PAD ADDRESSES

SERIES ONE PLUS ADDRESSES (Hex)	SUB-COMMAND (Hex)	DESCRIPTION
0000		PC Mode
0002		Sub-command for executing the functions:
	0009	Logging-In with the Password
	000A	Changing the Password
	0003	Grammar Checking
	0006	Reading Error Address
0004		Location of the error code generated by Grammar check and of the error location in the user program
000A		Password Write Location
0016		PC Type

Reading or writing the PC mode (RUN/STOP) and reading the PC type are the same for the Series One Plus as for the Series One/Junior and Series Three PCs (see application examples 20-22). The password and error checking features are available only for the Series One Plus PC and require the use of a sub-command written to 0002H of the Scratch-Pad (see explanation below).

LOGGING-IN ON THE SERIES ONE PLUS CPU USING THE PASSWORD

If a password has been assigned, either using the manual programmer or through communications, you must log in before executing each communications request to memory type 1 (T/C Accumulators), 3 (I/O and Shift Registers), or 7 (User Logic). If you do not log in, the commmunications request for these memory types will fail. It is not required to log in for communications requests to memory types 6 (Scratch-Pad) and 9 (Diagnostic Status Words). Logging in is done by executing a write command from registers to the Scratch-Pad beginning at address 0002H. The write command will write 5 registers of information as follows:

Rn	0900*(Hex)	Where 0009H is the subcommand written to Scratch-Pad
Rn+1	0000	address 0002H, and where xxxx is the existing password
Rn+2	0000	in BCD (Valid range 0-9999). A value of 0 is equivalent
Rn+3	0000	to no password.
Rn+4	xxxx**(BCD)	

Also see application example 9.

* The least significant byte of the subcommand occupies the most significant byte of the Series Six register.

** The most significant byte of the password occupies the most significant byte of the Series Six register.

CHANGING THE PASSWORD OF THE SERIES ONE PLUS PC

Changing the password is a 2-step operation. First, you must log in as explained in the preceding section. Then you must execute another write command from registers to the Series One Plus Scratch-Pad beginnng at address 0002H. The write command will write 5 registers of information as follows:

Rn	0A00*(Hex)	Where 000AH is the subcommand written to Scratch-Pad
Rn+1	0000	address 0002H, and where xxxx is the new password entered
Rn+2	0000	in BCD. Valid range 0-9999. A value of zero is
Rn+3	0000	equivalent to no password.
Rn+4	xxxx**(BCD)	

Also see application example 10.

* The least significant byte of the subcommand occupies the most significant byte of the Series Six register.

** The most significant byte of the password occupies the most significant byte of the Series Six register.

USER PROGRAM ERROR CHECKING

A complete program error check can be initiated at any time on a program in the Series One Plus CPU as explained below.

Initiating the error check and reading the error code is a 4 step operation.

1. To initiate the error check, write the subcommand, 0003H, from a Series Six register to the Series One Plus Scratch-Pad address 0002H.

2. Read the error code from Scratch-Pad address 0004H. If the contents of address 0004H is zero, there is no error code. If the contents of address 0004H is not zero, then this is the error code. Go to the next step to find the location of the first error in the user program.

3. To find the location of the error, write the subcommand, 0006H, to Scratch-Pad addess 0002H.

4. Read the location from the Scratch-Pad address 0004H. The contents of address 0004 is the location of the first error in user memory.

Table 5.9 defines the errors which may be found in a user program when the Series One Plus CPU is transitioned from PROGRAM to RUN. Also see application example 11.

DIAGNOSTIC STATUS WORDS

There are 5 Diagnostic Status Words in the DCU and DCM which store information regarding the communications activity on their ports.

When reading the Diagnostic Status Words, the transfer can start only with address 0 (word number 1) and all 5 words must be read. An external device can read or write/clear the Diagnostic Status Words by specifying memory type 9.

Diagnostic Status Word Number	Bits 16–9	Bits 8–1
1	Communications Port Most recent communication (Error Code)	Communications Port Next most recent communication (Error Code)
2	Number of Successful Conversations on Communications Port	
3	Number of Aborted Conversations on Communications Port	
4	Number of Header Re-tries on Communications Port	
5	Number of Data Block Re-tries on Communications Port	

NOTE

If you experience unexpected difficulties in communications, retrieve the Diagnostic Status Words from the Series One/Plus/Junior or Series Three and compare the value in the upper and lower bytes of Diagnostic Status Word 1 with the error codes listed in Table 5.9

DIAGNOSTIC STATUS WORD 1 ERROR CODES

Table 5.9 contains a list of all of the error codes that are reported in Diagnostic Status Word 1.

Table 5.9 DIAGNOSTIC STATUS WORD ERROR CODES

ERROR CODE DEC	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 an 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.

Table 5.9 (Cont.) DIAGNOSTIC STATUS WORD ERROR CODES

ERROR CODE DEC	HEX	DESCRIPTION
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 DCU or DCM expected to receive an EOT character from an external device and did not receive it.
22	16	The DCU or DCM expected to receive an ACK or NAK character and did not receive either one.
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 between the DCU and the Series One, Series One Junior, or the Series One Plus CPU or the DCM and the Series Three CPU.
30	1E	A parity, framing, or overrun error occurred during a serial header transfer.
31	1F	A parity, framing, or overrun error occurred during a serial data block transfer.

SERIES ONE, SERIES ONE JUNIOR, SERIES ONE PLUS, AND SERIES THREE ERROR CODES

There are certain errors detected by the Series One/Junior/Plus or Series Three CPU during communication attempts. If this error occurs it will be displayed on the Series One/Junior/Plus or Series Three programmer display with the following codes. In addition, these error codes can be obtained from the Series One Plus CPU by an external device using a serial request. See application example 11.

Table 5.10 SERIES ONE, SERIES ONE JUNIOR, SERIES ONE PLUS, SERIES THREE CPU ERROR CODES

DIAGNOSTIC STATUS CODE REPORTED	ERROR CODE DISPLAYED ON PROGRAMMER	TYPE OF ERROR CONDITION
1D Hex	E02	Instruction and I/O data wrong. Input programmed as an Output.
	E21	Parity error in user program memory.
	E31	Watchdog timer timed out.
	E41*	I/O module configuration change since last power up. Invalid I/O to CPU transfer.
	No Error Code	PROGRAM/RUN keyswitch set to program; DCU to CPU cable disconnected; CPU not accepting communication request.
00 Hex	E01	Incorrect entry of instruction and data wrong, operand on write to user program instruction and/or data has parity error on write to user program, cannot write to user program memory. (Program in PROM or RAM defective).
	E10	All user program memory locations used.

* Series Three only.

Some of the above error conditions also cause diagnostic status code "1D hex" to be returned in Diagnostic Status Word 1. Other conditions will be reported as successful transfers (diagnostic status code "00 hex").

SCREQ COMMAND EXAMPLES

This section contains application examples for programming the Series Six to initiate serial communications with a Series One, Series One Junior, Series One Plus, and Series Three PC. Examples 1-5 apply to Series One, Series One Junior, and Series One Plus PCs only; examples 6-11, to Series One Plus PCs only; examples 12-19, to Series Three PCs only; and examples 20-24, to Series One, Series One Junior, Series One Plus, and Series Three PCs.

NOTE

Users with Series Three CPUs with date codes prior to 8408XXXX will be unable to write directly to DATA REGISTERS or TIMER/COUNTER accumulated values - the DCM will NAK the HEADER block. Contact G.E. Product Service for information on upgrading the CPU. The catalog number for the Series Three CPU upgrade kit is IC630CPU390A. An alternate way of writing this kind of data into the register table is to write it into the unused portion of the I/O table and then move it into the register table via user logic.

Example 1: **READ FROM TARGET TIMERS AND COUNTERS (SERIES ONE/JR/PLUS)**

Read 5 Series One/Junior/Plus Timer/Counter accumulated values and store in the Series Six data registers starting at Register 1. The target ID is 10. The communication is to take place on the J1 port of the Series Six CCM.

Rnnnn	=	06101 (decimal) COMMAND NUMBER - read from target to source Register Table.
Rnnnn (+1)	=	00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	=	00001 MEMORY TYPE OF TARGET - Timers/Counters.
Rnnnn (+3)	=	00001 MEMORY ADDRESS OF TARGET - Start reading from Series One/Junior/Plus T/C accumulator reference 600. See Tables 5.1, 5.2, and 5.3 for mapping of Series One/Junior/Plus Timer/Counter reference numbers to reference numbers used for communication.
Rnnnn (+4)	=	00005 DATA LENGTH - 5 words (5 registers)
Rnnnn (+5)	=	00001 Memory Address of Source - Start storing of data in Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
| I0001                                                              O0101
+--] [---------------------------------------------------------------(OS)-+
|
|
| O0101   R0100
+--] [---[                     BLOCK MOVE                          ]+-(  )-+
|           +06101  +00010  +00001  +00001  +00005  +00001  +00000
|
| O0101   R0100
+--] [---[SCREQ]
|
```

The low byte of a Series One/Junior/Plus Timer or Counter accumulator is stored in the low byte of a Series Six register. The high byte is stored in the high byte of a Series Six register.

Example 2: **READ FROM TARGET I/O (SERIES ONE/JR/PLUS)**

Read the first 16 Series One/Junior/Plus external I/O points and store in Series Six Input Status Table starting at Input point 1. The target ID is 10. The communication is to take place on CCM port J1.

Rnnnn	= 06102 COMMAND NUMBER - Read from target to source Input Status Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00003 MEMORY TYPE OF TARGET - Discrete I/O.
Rnnnn (+3)	= 00001 MEMORY ADDRESS OF TARGET - Beginning address in Series One/Junior/Plus/I/O table - I/O point number 1. See Tables 5.1, 5.2, and 5.3 for mapping of Discrete I/O reference numbers to reference numbers used for communication.
Rnnnn (+4)	= 00016 DATA LENGTH - 16 Input points.
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Begin storing in Input Status Table at address 1.

The Series Six ladder logic is shown below:

```
|
|
| I0002                                                                00102
+--] [-----------------------------------------------------------------(OS)-+
|
|
|
|
| 00102   R0100
+--] [---[                      BLOCK MOVE                          ]+-(  )-+
|           +06102  +00010  +00003  +00001  +00016  +00001  +00000
|
|
|
| 00102   R0100
+--] [---[SCREQ]
|
|
```

I/O from the Series One/Junior will be stored in the Series Six Input Status Table in the following format (only the first 8 I/O in the example are shown).

Series Six Inputs 1-8							
8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	0

Series Three I/O points 0-7

Example 3: WRITE TO TARGET I/O (SERIES ONE/PLUS)

Write 16 Series One/Plus internal I/O points (points 160-177) from Series Six Output Status Table starting at output point 33. The target ID is 10. The communication is to take place on CCM port J1.

Rnnnn	= 06113 COMMAND NUMBER - write to target from source Output Status Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00003 MEMORY TYPE OF TARGET - Inputs or outputs.
Rnnnn (+3)	= 00015 MEMORY ADDRESS OF TARGET - Start writing to Series One internal I/O point 160. See Tables 5.1 and 5.3 for mapping of Series One and Series One Plus discrete I/O reference numbers to reference numbers used for communication.
Rnnnn (+4)	= 00016 DATA LENGTH - 16 points.
Rnnnn (+5)	= 00033 MEMORY ADDRESS OF SOURCE - Start transfer in Series Six at Output Status Table reference 33.

The Series Six ladder logic is shown below:

```
|
|
| I0003                                                                   00103
+--] [--------------------------------------------------------------------(OS)-+
|
|
|
|
| 00103  R0100
+--] [---[                          BLOCK MOVE                          ]+-(  )-+
|          +06113  +00010  +00003  +00015  +00016  +00033  +00000
|
|
|
| 00103   R0100
+--] [---[SCREQ]
|
```

Outputs to be sent to Series One/Junior/Plus I/O must be stored in the Series Six Output Table before execution of the serial request. The sample format shows the relationship above of Series Six Outputs to their corresponding Series One/Junior/Plus I/O (only first 8 outputs in the example are shown).

Series Six Inputs 33-40

40	39	38	37	36	35	34	33
167	166	165	164	163	162	161	160

Series Three I/O points 160-167

NOTE

Based on the timeouts in Table 6.2, Series One Junior I/O cannot be writeen-to from the Series Six PC.

Example 4: **READ FROM TARGET USER MEMORY (SERIES ONE/JR/PLUS)**

Read the first 16 words of the user program in the Series One/Junior/Plus CPU and store them in Series Six data registers starting at Register 1. The Target ID is 10. The communication is to take place on CCM port J1.

Rnnnn	= 06101 COMMAND NUMBER - Read from target to source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00007 MEMORY TYPE OF TARGET - User-Logic memory.
Rnnnn (+3)	= 00000 MEMORY ADDRESS OF TARGET - Read from Series One/Junior/Plus user program beginning at address 0.
Rnnnn (+4)	= 00016 DATA LENGTH - 16 words (16 registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Store in Series Six beginning at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0004                                                                  O0104
+--[ ]-----------------------------------------------------------------(OS)-+
|
|
|
|
| O0104    R0100
+--] [---[                        BLOCK MOVE                           ]+-(  )-+
|          +06101  +00010  +00007  +00000  +00016  +00001  +00000
|
|
|
| O0104    R0100
+--] [---[SCREQ]
|
|
```

Example 5: **WRITE TO TARGET USER MEMORY (SERIES ONE/JR/PLUS)**

Write to the first 16 words of the user program in the Series One/Junior/Plus CPU from program data stored in Series Six data Registers 1-16. The target ID is 10. The communications will take place on the J1 port of the CCM in the Series Six. The CPU must be placed in Program/Stop mode either using the keyswitch or a serial request before writing to User Logic.

Rnnnn	= 06111 COMMAND NUMBER - Write to target from source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00007 MEMORY TYPE OF TARGET - User-Logic memory.
Rnnnn (+3)	= 00000 MEMORY ADDRESS OF TARGET - Start writing to User-Logic memory address 0.
Rnnnn (+4)	= 00016 DATA LENGTH - 16 words (16 registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start transfer from Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0005                                                              00105
+--] [-------------------------------------------------------------(OS)-+
|
|
|
|
| 00105   R0100
+--] [---[                        BLOCK MOVE                        ]+-(  )-+
|             +06111  +00010  +00007  +00000  +00016  +00001  +00000
|
|
|
| 00105   R0100
+--] [---[SCREQ]
|
|
```

Example 6: READ FROM TARGET DATA REGISTERS (SERIES ONE PLUS)

Read 64 Series One Plus Data Registers and store in Series Six data registers starting at Series Six Register 1. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06101 (decimal) COMMAND NUMBER - Read from target to source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00001 MEMORY TYPE OF TARGET - Register memory.
Rnnnn (+3)	= 00065 MEMORY ADDRESS OF TARGET - Start reading from Series One Plus Register 400. See Table 5.3 for mapping of Series One Plus data register reference numbers to reference numbers used for communication.
Rnnnn (+4)	= 00032 DATA LENGTH - 64 Series One Plus registers (32 Series Six registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start storing in Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0006                                                            O0106
+--] [-----------------------------------------------------------(OS)-+
|
|
|
|
| O0106   R0100
+--] [---[                    BLOCK MOVE                        ]+-(  )-+
|          +06101  +00010  +00001  +00065  +00032  +00001  +00000
|
|
|
| O0106   R0100
+--] [---[SCREQ]
|
|
```

Series One Plus data registers are 8-bits long therefore two of these registers will be transferred to one 16-bit Series Six register. The least significant of the two Series One Plus data registers will be transferred to the least significant byte of the corresponding Series Six register (see sample format below):

Series Six Register 1

High Byte	Low Byte
Series One Plus Register 2 (address 401)	Series One Plus Register 1 (address 400)

Example 7: **WRITE TO TARGET DATA REGISTERS (SERIES ONE PLUS)**

Write to the 64 Series One Plus data registers from Series Six data registers starting at Series Six Register 1. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06111 COMMAND NUMBER - Write to target from source Register Table.
Rnnnn(+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn(+2)	= 00001 MEMORY TYPE OF TARGET - Register memory.
Rnnnn(+3)	= 00065 MEMORY ADDRESS OF TARGET - Start reading from Series One Plus Register 400. See Table 5.3 for mapping of Series One Plus data register reference numbers to reference numbers used for communication.
Rnnnn(+4)	= 00032 DATA LENGTH - 64 Series One Plus registers (32 Series Six registers).
Rnnnn(+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start sending from Series Six Register 1.

The Series Six ladder logic is shown below:

```
|
| I0007                                                                00107
+--] [---------------------------------------------------------------(OS)-+
|
|
|
| 00107   R0100
+--] [---[                          BLOCK MOVE                       ]+-(  )-+
|           +06111  +00010  +00001  +00065  +00032  +00001  +00000
|
|
| 00107   R0100
+--] [---[SCREQ]
|
|
```

Series One Plus data registers are 8-bits long therefore two of these registers will be written-to from one 16-bit Series Six register. The least significant of the two Series One Plus data registers will be written-to from the least significant byte of the corresponding Series Six register (see sample format below):

```
                  Series Six Register 1
           High Byte                      Low Byte
Series One Plus Register 2   Series One Plus Register 1
      (address 401)                (address 400)
```

Example 8: **WRITE TO TARGET TIMER/COUNTER ACCUMULATORS (SERIES ONE PLUS)**

Write to 2 Series One Plus Timer/Counter accumulators from Series Six registers starting at Series Six Register 1. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06111 (decimal) COMMAND NUMBER - Write to target from source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00001 MEMORY TYPE OF TARGET - Register memory.
Rnnnn (+3)	= 00001 MEMORY ADDRESS OF TARGET - Start writing to Timer/Counter 1 (referenced as T/C 600 in the user program). See Table 5.3 for mapping of Timer/Counter reference numbers to reference numbers used for communication.
Rnnnn (+4)	= 00002 DATA LENGTH - 2 accumulators (2 registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start writing from Series Six Register 1.

The Series Six ladder logic is shown below:

```
|
| I0009                                                                 O0109
+--] [-----------------------------------------------------------------(OS)-+
|
|
|
| O0109    R0100
+--] [---[                       BLOCK MOVE                          ]+-(   )-+
|          +06111  +00010  +00001  +00001  +00002  +00001  +00000
|
|
| O0109    R0100
+--] [---[SCREQ]
|
```

NOTES ON WRITING TO TIMER/COUNTER ACCUMULATORS

- Values can be written at any time to Timer/Counter accumulators which are not referenced by a timer or counter in Series One Plus user logic.

- If a timer is programmed in Series One Plus user logic and the input to that timer is open, the value of the accumulator will always be zero. If, however, the input to the timer is closed and the timer is timing, the accumulator will assume the value written to it and will resume timing out from that value. Once the timer has timed out, the accumulator will accept new values, and if the value is below the preset, the timer "coil" is reset and the timer will start timing from the new accumulator value to the preset. When the timer is reset, the accumulator will always assume the value of zero.

- When a counter accumulator is programmed in Series One Plus user logic, it can be written to unless the reset input is on. Once the counter has counted out, the accumulator will accept new values, and if the value is below the preset, the counter "coil" is reset and the counter will start counting from the new accumulator value to the preset. When the counter is reset, the accumulator will always assume value of zero.

Prior to execution of the serial request, data to be transferred must be placed in Series Six registers as follows: The low byte of a Series One Plus Timer or Counter accumulator must be stored in the low byte of the corresponding Series Six register. The high byte of the Series One Plus timer or counter accumulator must be stored in the high byte of the Series Six register.

Example 9: **LOGGING-IN WITH PASSWORD (SERIES ONE PLUS)**

Log-in on the Series One Plus CPU in which the password 1234 has been assigned previously (by the manual programmer or communications).

Rnnnn	= 06111 (decimal) COMMAND NUMBER - Write to target from source Register Table.
Rnnnn (+1)	= 00002 ID OF TARGET DEVICE - 2.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Scratch-Pad.
Rnnnn (+3)	= 00002 MEMORY ADDRESS OF TARGET - Start writing to Scratch-Pad address 02.
Rnnnn (+4)	= 00005 DATA LENGTH - 5 registers.
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start sending from Series Six Register 1. See explanation below:

The Series Six ladder logic is shown below:

```
|
|
| I0009                                                            O0109
+--] [------------------------------------------------------------(OS)-+
|
|
|
|
| O0109   R0100
+--] [---[                     BLOCK MOVE                        ]+-(  )-+
|          +06111  +00002  +00006  +00002  +00005  +00001  +00000
|
|
| O0109   R0100
+--] [---[SCREQ]
|
|
```

To log-in on the Series One Plus, the contents of 5 Series Six registers as shown below must be written to the Scratch-Pad of the Series One Plus CPU starting at address 0002. In this example, Register 1 contains the subcode for logging-in and Register 5 contains the password. The password must have been assigned previously (using the manual programmer or communications as shown in application example 10).

R0001 0900 (Hex)
R0002 0000
R0003 0000
R0004 0000
R0005 1234 (Hex)

Example 10: **CHANGE PASSWORD (SERIES ONE PLUS)**

Change the password to the Series One Plus CPU to 0100 (BCD). If a password has been assigned previously, you must first log in according to the instructions in example 9. If a password has not been assigned, you do not need to log in.

Rnnnn	= 06111 (decimal) COMMAND NUMBER - Write to target from source Register Table.
Rnnnn (+1)	= 00002 ID OF TARGET DEVICE - 2.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Scratch-Pad.
Rnnnn (+3)	= 00002 MEMORY ADDRESS OF TARGET - Start writing to Scratch-Pad address 02.
Rnnnn (+4)	= 00005 DATA LENGTH - 5 registers.
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start sending from Series Six at Register 1. See explanation below:

The Series Six ladder logic is shown below:

```
|
|
| I0010                                                                00110
+--] [------------------------------------------------------------------(OS)-+
|
|
|
|
| 00110   R0100
+--] [---[                         BLOCK MOVE                          ]+-(  )-+
|          +06111  +00002  +00006  +00002  +00005  +00001  +00000
|
|
|
| 00110   R0100
+--] [---[SCREQ]
|
|
```

To change the password, the contents of 5 Series Six registers as shown below must be written to the Scratch Pad starting at address 0002. In this example, Register 1 contains the subcode for changing the password and Register 5 contains the new password.

R0001 0A00 (Hex)
R0002 0000
R0003 0000
R0004 0000
R0005 0100 (BCD*)

* Enter this BCD value in HEX mode from the Display Reference Tables function.

Example 11: **CHECK PROGRAM ERROR CODE (SERIES ONE PLUS)**

Checking for a user program error and its location requires the execution of 4 communication requests.

1. **To inititate the error check**, write Register 1 containing the subcommand (0300 Hex) to the Series One Scratch-Pad starting at address 0002. This initiates the error check.

Rnnnn	= 06111 (decimal) COMMAND NUMBER - Write to target from source Register Table.
Rnnnn (+1)	= 00002 ID OF TARGET DEVICE - 2.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Scratch-Pad.
Rnnnn (+3)	= 00002 MEMORY ADDRESS OF TARGET - Start writing to Scratch-Pad address 02.
Rnnnn (+4)	= 00001 DATA LENGTH - 1 register.
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start sending from Series Six Register 1.

2. **To read the error code**, read the Series One Plus Scratch-Pad address 0004.

Rnnnn	= 06101 (decimal) COMMAND NUMBER - Read from target to source Register Table.
Rnnnn (+1)	= 00002 ID OF TARGET DEVICE - 2.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Scratch-Pad.
Rnnnn (+3)	= 00004 MEMORY ADDRESS OF TARGET - Start reading from Scratch-Pad address 0004.
Rnnnn (+4)	= 00001 DATA LENGTH - 1 register.
Rnnnn (+5)	= 00002 MEMORY ADDRESS OF SOURCE - Start storing in Series Six Register 2.

3. If the contents of Scratch-Pad address 0004 are 0, then there is no error. If the contents are not 0, **initiate the error location check** by writing Register 3 containing the subcommand (0600 Hex) to Series One Scratch-Pad address 0002.

Rnnnn	= 06111 (decimal) COMMAND NUMBER - Write to target from source Register Table.
Rnnnn (+1)	= 00002 ID OF TARGET DEVICE - 2.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Scratch-Pad.
Rnnnn (+3)	= 00002 MEMORY ADDRESS OF TARGET - Start writing to Scratch-Pad address 0002.
Rnnnn (+4)	= 00001 DATA LENGTH - 1 register.
Rnnnn (+5)	= 00003 MEMORY ADDRESS OF SOURCE - Start sending from Series Six Register 3.

4. **To read the location of the error in user memory**, read the Series One Plus Scratch-Pad address 0004.

Rnnnn	= 06101 (decimal) COMMAND NUMBER - Read from target to source Register Table.
Rnnnn (+1)	= 00002 ID OF TARGET DEVICE - 2.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Scratch-Pad.
Rnnnn (+3)	= 00004 MEMORY ADDRESS OF TARGET - Start reading from Scratch-Pad address 004.
Rnnnn (+4)	= 00001 DATA LENGTH - 1 register.
Rnnnn (+5)	= 00004 MEMORY ADDRESS OF SOURCE - Start storing in Series Six Register 4.

The Series Six ladder logic is shown below:

1. Initiate the error check.

```
|
| I0011                                                              O0111
+--] [-------------------------------------------------------------(OS)-+
|
|
| O0111    R0100
+--] [---[                  BLOCK MOVE                          ]+-(  )-+
|          +06111  +00002  +00006  +00002  +00001  +00001  +00000
|
```

2. Read the error code.

```
|
| I0112                                                              O0112
+--] [-------------------------------------------------------------(OS)-+
|
|
| O0112    R0100
+--] [---[                  BLOCK MOVE                          ]+-(  )-+
|          +06101  +00002  +00006  +00004  +00001  +00002  +00000
|
```

3. Initiate the error location check.

```
|
| I0013                                                              O0113
+--] [-------------------------------------------------------------(OS)-+
|
|
| O0113    R0100
+--] [---[                  BLOCK MOVE                          ]+-(  )-+
|          +06111  +00002  +00006  +00002  +00001  +00003  +00000
|
```

4. Read the location of the error code.

```
|
| I0114                                                              O0114
+--] [-------------------------------------------------------------(OS)-+
|
|
| O0114    R0100
+--] [---[                  BLOCK MOVE                          ]+-(  )-+
|          +06101  +00002  +00006  +00004  +00001  +00004  +00000
|
```

```
|
| 00111      R0100
+--] [-----[SCREQ]
|       |
|       |
| 00112 |
+--] [--+
|       |
|       |
| 00113 |
+--] [--+
|       |
|       |
| 00114 |
+--] [--+
|
|
|
```

The Series Six registers used in the communications requests shown above are defined as follows:

R0001 - The subcommand (0300 Hex), written to Scratch-Pad address 0002, must be placed in this register by the programmer.

R0002 - Receives the Series One Plus error code from Scratch-Pad address 0004.

R0003 - The subcommand (0600 Hex), written to Scratch-Pad addess 0002, must be placed in this register by the programmer.

R0004 - Receives the Series One Plus error location in user memory.

Example 12: **READ FROM TARGET DATA REGISTERS (SERIES THREE)**

Read 64 Series One Plus Data Registers and store in Series Six data registers starting at Register 1. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06101 (decimal) COMMAND NUMBER - Read from target to source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00001 MEMORY TYPE OF TARGET - Register memory.
Rnnnn (+3)	= 00001 MEMORY ADDRESS OF TARGET - Start reading from Series Three Register 1. See Table 5.4 for mapping of Series Three data register reference numbers to reference numbers used for communication.
Rnnnn (+4)	= 00032 DATA LENGTH - 64 Series Three registers (32 Series Six registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start storing in Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0006                                                                  O0106
+--] [-----------------------------------------------------------------(OS)-+
|
|
|
|
| O0106   R0100
+--] [---[                        BLOCK MOVE                         ]+-(  )-+
|         +06101  +00010  +00001  +00001  +00032  +00001  +00000
|
|
|
| O0106   R0100
+--] [---[SCREQ]
|
|
```

Series Three data registers are 8-bits long therefore two of these registers will be transferred to one 16-bit Series Six register. The least significant of the two Series Three data registers will be transferred to the least significant byte of the corresponding Series Six register (see sample format below):

```
                Series Six Register 1
          High Byte                    Low Byte
Series Three Register 2     Series Three Register 1
      (address 501)               (address 500)
```

Example 13: **WRITE TO TARGET DATA REGISTERS (SERIES THREE)**

Write to the 64 Series Three data registers from Series Six data registers starting at Register 1. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06111 COMMAND NUMBER - Write to target from source Register Table.
Rnnnn(+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn(+2)	= 00001 MEMORY TYPE OF TARGET - Register memory.
Rnnnn(+3)	= 00001 MEMORY ADDRESS OF TARGET - Start reading from Series Three Register 1. See Table 5.4 for mapping of Series Three data register reference numbers to reference numbers used for communication.
Rnnnn(+4)	= 00032 DATA LENGTH - 64 Series Three registers (32 Series Six registers).
Rnnnn(+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start sending from Series Six Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0007                                                                00107
+--] [---------------------------------------------------------------(OS)-+
|
|
|
|
| 00107   R0100
+--] [---[                          BLOCK MOVE                        ]+-(  )-+
|            +06111  +00010  +00001  +00001  +00032  +00001  +00000
|
|
|
| 00107   R0100
+--] [---[SCREQ]
|
|
```

Series Three data registers are 8-bits long therefore two of these registers will be written-to from one 16-bit Series Six register. The least significant of the two Series Three data registers will be written-to from the least significant byte of the corresponding Series Six register (see sample format below):

```
                     Series Six Register 1
           High Byte                      Low Byte
  ------------------------      ------------------------
  Series Three Register 2       Series Three Register 1
       (address 501)                 (address 500)
```

Example 14: **READ FROM TARGET TIMERS AND COUNTERS (SERIES THREE)**

Read 64 Series Three Timer/Counter accumulator values and store them in Series Six registers starting at Register 1. Target ID is 10. Communications takes place through CCM port J1.

Rnnnn	= 06101 COMMAND NUMBER - Read from target to source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00001 MEMORY TYPE OF TARGET - Register memory.
Rnnnn (+3)	= 00065 MEMORY ADDRESS OF TARGET - Start reading from Series Three Timer/Counter 0 (accumulator referenced as 200). See Table 5.4 for mapping of Series Three Timer and Counter accumulator reference numbers to reference numbers used for communication.
Rnnnn (+4)	= 00064 DATA LENGTH - 64 accumulator references (64 registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start storing in Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0008                                                              00108
+--] [-------------------------------------------------------------(OS)-+
|
|
|
|
| 00108   R0100
+--] [---[                       BLOCK MOVE                      ]+-(  )-+
|            +06101  +00010  +00001  +00065  +00064  +00001  +00000
|
|
|
| 00108   R0100
+--] [---[SCREQ]
|
|
```

The low byte of a Series Three Timer or Counter accumulator is stored in the low byte of a Series Six register. The high byte of a Series Three Timer or Counter accumulator is stored in the high byte of a Series Six register.

Example 15: **WRITE TO TARGET TIMER/COUNTER ACCUMULATORS (SERIES THREE)**

Write to 2 Series Three Timer/Counter accumulator values from Series Six registers starting at Register 1. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06111 (decimal) COMMAND NUMBER - Write to target from source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00001 MEMORY TYPE OF TARGET - Register memory.
Rnnnn (+3)	= 00065 MEMORY ADDRESS OF TARGET - Start writing to Timer/Counter 0 (accumulator reference 200). See Table 5.4 for mapping of Timer/Counter accumulator reference numbers to reference numbers used for communication.
Rnnnn (+4)	= 00002 DATA LENGTH - 2 accumulators (2 registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start writing from Series Six Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0009                                                             00109
+--] [------------------------------------------------------------(OS)-+
|
|
|
|
| 00109   R0100
+--] [---[                     BLOCK MOVE                         ]+-(  )-+
|            +06111  +00010  +00001  +00065  +00002  +00001  +00000
|
|
|
| 00109   R0100
+--] [---[SCREQ]
|
```

NOTES ON WRITING TO TIMER/COUNTER ACCUMULATORS

- Values can be written at any time to Timer/Counter accumulators which are not referenced by a timer or counter in Series Three user logic.

- If a timer is programmed in Series Three user logic and the input to that timer is open, the programmed preset will always override any value written to the accumulator. If, however, the input to the timer is closed and the timer is timing, the accumulator will assume the value written to it and will resume timing down from that value. Once the timer has timed out, the accumulator will accept new values but the timer will not time down again; it must be reset first. When the timer is reset, the accumulator will always assume the preset value.

- When a counter accumulator is programmed in Series Three User Logic, it can be written to unless the reset input is on. Once the counter has counted out, the accumulator will accept new values but the counter will not count down again; it must be reset first. When the counter is reset, the accumulator will always assume the preset value.

Prior to execution of the serial request, data to be transferred must be placed in Series Six registers as follows: The low byte of a Series Three Timer or Counter accumulator must be stored in the low byte of the corresponding Series Six register. The high byte of the Series Three timer or counter accumulator must be stored in the high byte of the Series Six register.

Example 16: **READ FROM TARGET I/O (SERIES THREE)**

Read the 400 Series Three external I/O points and store in Series Six Input Status Table starting at Input 1. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06102 (decimal) COMMAND NUMBER - Read from target to source Input Status Table.
Rnnnn(+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn(+2)	= 00003 MEMORY TYPE OF TARGET - Series Three I/O.
Rnnnn(+3)	= 00001 MEMORY ADDRESS OF TARGET - Start reading from Series Three I/O point 1. See Table 5.4 for mapping of Series Three discrete I/O reference numbers to reference numbers used for communication.
Rnnnn(+4)	= 00400 DATA LENGTH - read 400 I/O points.
Rnnnn(+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start storing in Series Six at Input 1.

The Series Six ladder logic is shown below:

```
|
|
| I0010                                                              O0110
+--] [-------------------------------------------------------------(OS)-+
|
|
|
|
| O0110   R0100
+--] [---[                    BLOCK MOVE                          ]+-(  )-+
|          +06102  +00010  +00003  +00001  +00400  +00001  +00000
|
|
|
| O0110   R0100
+--] [---[SCREQ]
|
|
```

I/O from the Series Three will be stored in the Series Six Input Status Table in the following format (only the first 8 I/O in the example are shown).

Series Six Inputs 1-8							
8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	0

Series Three I/O points 0-7

Example 17: **WRITE TO TARGET I/O (SERIES THREE)**

Write 16 Series Three internal I/O points (points 7000 - 7017) from Series Six Output Status Table starting at Output point 33. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06113 (COMMAND NUMBER - Write to target from Source Output Status Table.
Rnnnn(+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn(+2)	= 00003 MEMORY TYPE OF TARGET - Series Three I/O.
Rnnnn(+3)	= 00089 MEMORY ADDRESS OF TARGET - Start writing to Series Three internal I/O point 7000. See Table 5.4 for mapping of Series Three discrete I/O reference numbers to reference numbers used for communication.
Rnnnn(+4)	= 00016 DATA LENGTH - write 16 I/O points.
Rnnnn(+5)	= 00033 MEMORY ADDRESS OF SOURCE - Start transfer in Series Six at Output point 33.

The Series Six ladder logic is shown below:

```
|
|
| I0011                                                              O0111
+--] [-------------------------------------------------------------(OS)-+
|
|
|
|
| O0111    R0100
+--] [---[                         BLOCK MOVE                        ]+-(  )-+
|         +06113  +00010  +00003  +00089  +00016  +00033  +00000
|
|
|
| O0111    R0100
+--] [---[SCREQ]
|
|
```

Outputs to be sent to Series Three I/O must be stored in the Series Six Output Table before execution of the serial request. The sample format shows the relationship above of Series Six Outputs to their corresponding Series Three I/O (only first 8 outputs in the example are shown).

Series Six Inputs 33-40

40	39	38	37	36	35	34	33
7007	7006	7005	7004	7003	7002	7001	7000

Series Three I/O points 7000-7007

Example 18: **READ FROM TARGET USER MEMORY (SERIES THREE)**

Read the first 64 words of the user program in the Series Three CPU and store in Series Six data registers starting at Register 1. Target ID is 10. Communication to take place on CCM port J1.

Rnnnn	= 06101 (decimal) COMMAND NUMBER - Read from target to source Register Table.
Rnnnn(+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn(+2)	= 00007 MEMORY TYPE OF TARGET - User-Logic memory.
Rnnnn(+3)	= 00000 MEMORY ADDRESS OF TARGET - Start reading from Series Three User Program address 0.
Rnnnn(+4)	= 00064 DATA LENGTH - 64 words (64 registers). Each Series Three user instruction is at least 2 bytes).
Rnnnn(+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start storing in Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0012                                                            00112
+--] [-----------------------------------------------------------(OS)-+
|
|
|
|
| 00112   R0100
+--] [---[                    BLOCK MOVE                    ]+-(  )-+
|         +06101  +00010  +00007  +00000  +00064  +00001  +00000
|
|
|
| 00112   R0100
+--] [---[SCREQ]
|
|
```

Example 19: **WRITE TO TARGET USER MEMORY (SERIES THREE)**

Write to the first 64 words of the user program in the Series Three CPU from program data stored in Series Six Registers 1-128. Target ID is 10. Communication to take place on CCM port J1. The CPU must be placed in Program/Stop mode using the serial request in Example 22 before writing to target user memory.

Rnnnn	= 06111 (decimal) COMMAND NUMBER - Write to target from source Register Table.
Rnnnn(+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn(+2)	= 00007 MEMORY TYPE OF TARGET - User-Logic memory.
Rnnnn(+3)	= 00000 MEMORY ADDRESS OF TARGET - Start reading from Series Three user program address 0.
Rnnnn(+4)	= 00064 DATA LENGTH - 128 bytes, each Series Three user instruction is at least 2 bytes (64 regsisters).
Rnnnn(+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start storing in Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0013                                                                    O0113
+--] [----------------------------------------------------------------------(OS)-+
|
|
|
|
| O0113   R0100
+--] [---[                        BLOCK MOVE                              ]+-(  )-+
|            +06111  +00010  +00007  +00000  +00064  +00001  +00000
|
|
|
| O0113   R0100
+--] [---[SCREQ]
|
|
```

Example 20: **READ PC TYPE (SERIES ONE/JUNIOR/PLUS OR SERIES THREE)**

Read the PC Type code from the Series One/Junior/Plus or Series Three CPU and store in Series Six Register 1. The target ID is 10. The communication will take place over the J1 port of the CCM in the Series Six.

Rnnnn	= 06101 COMMAND NUMBER - Read from target to source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Scratch Pad.
Rnnnn (+3)	= 00022 MEMORY ADDRESS OF TARGET - Read from Series One Scratch Pad beginning at address 22.
Rnnnn (+4)	= 00001 DATA LENGTH - read 2 bytes (1 register).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Store in Series Six beginning at Register 1.

The Series Six ladder logic is shown below:

```
|
| I0014                                                              00114
+--] [-------------------------------------------------------------(OS)-+
|
|
|
| 00114  R0100
+--] [---[                        BLOCK MOVE                          ]+-(  )-+
|              +06101  +00010  +00006  +00022  +00001  +00001  +00000
|
|
| 00114   R0100
+--] [---[SCREQ]
|
|
```

For this example:

The data from the Series One/Junior/Plus or Series Three will be stored in the Series Six register as follows:

Series Six Register	Content
1	0101 (Hex) (for Series One)
1	0202 (Hex) (for Series One Junior)
1	0303 (Hex) (for Series One Plus)
1	0707 (Hex) (for Series Three)

Example 21: **READ TARGET RUN/PROGRAM MODE (SERIES ONE/JUNIOR/PLUS OR SERIES THREE)**

Read the Series One/Junior/Plus or Series Three Operation mode (RUN/PROGRAM) and store in Series Six Register 1. The Target ID is 10. The communication will take place on the J1 port of the CCM.

Rnnnn	= 06101 COMMAND NUMBER - Read from target to source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Series One/Junior/Plus or Series Three Scratch Pad.
Rnnnn (+3)	= 00000 MEMORY ADDRESS OF TARGET - Start reading from Scratch Pad address 0.
Rnnnn (+4)	= 00001 DATA LENGTH - read 2 bytes (1 register).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start storing in Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0015                                                                      00115
+--] [-------------------------------------------------------------------(OS)-+
|
|
|
|
| 00115   R0100
+--] [---[                        BLOCK MOVE                            ]+-(  )-+
|            +06101  +00010  +00006  +00000  +00001  +00001  +00000
|
|
|
| 00115   R0100
+--] [---[SCREQ]
|
|
```

For this example:

The data from the Series One/Junior/Plus or Series Three will be stored in the Series Six register as follows:

Series Six Register	Contents		
1	0101 (Hex) or 00257 (Dec)	-	if CPU in RUN mode
	8080 (Hex) or 32896 (Dec)	-	if CPU in PROGRAM mode

Example 22: COMMAND TARGET RUN/PROGRAM MODE (SERIES ONE/JUNIOR/PLUS OR SERIES THREE)

Command the Series One/Junior/Plus or Series Three CPU operational mode (RUN/PROGRAM) from Series Six Register 1. The target ID is 10. The communication is to take place on the J1 port of the CCM.

Rnnnn	= 06111 COMMAND NUMBER - Write to target from source Register Table.
Rnnnn (+1)	= 00010 ID OF TARGET DEVICE - 10.
Rnnnn (+2)	= 00006 MEMORY TYPE OF TARGET - Scratch pad.
Rnnnn (+3)	= 00000 MEMORY ADDRESS OF TARGET - Start writing to Scratch-Pad address 0.
Rnnnn (+4)	= 00001 DATA LENGTH - write 2 bytes (1 register).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start reading from Series Six at Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0016                                                               00116
+--] [-------------------------------------------------------------------(OS)-+
|
|
|
|
| 00116    R0100
+--] [---[                         BLOCK MOVE                         ]+-(  )-+
|           +06111   +00010   +00006   +00000   +00001   +00001   +00000
|
|
|
| 00116    R0100
+--] [---[SCREQ]
|
|
```

For this example:

The data to be sent to the Series One/Junior/Plus, or Series Three must be stored in the Series Six register as follows:

Series Six Register	Contents	
1	0101 (Hex) or 00257 (Dec)	- to Command CPU to RUN mode
	8080 (Hex) or 32896 (Dec)	- to Command CPU to PROGRAM mode

NOTE

Users with Series Three CPUs with date codes prior to 8408xxxx and are executing user programs in PROM will experience difficulty using this request since the user program is in PROM. In this case the CCM will report the following for this request:

SERIES THREE MODE	PROM	RAM
RUN mode	DCM reports RUN mode	DCM reports RUN mode
PROGRAM/STOP mode	DCM reports RUN mode	DCM reports PROGRAM mode

Example 23: **READ TARGET DIAGNOSTIC STATUS WORDS (SERIES ONE/JUNIOR/PLUS OR SERIES THREE)**

Read from target DCU or DCM Diagnostic Status Words 1-5 to Series Six Registers 1-5. The communication is to take place on CCM port J2. The Target ID is 36.

Rnnnn	= 06201 COMMAND NUMBER - Read from DCU or DCM to source Register Table.
Rnnnn (+1)	= 00036 ID OF TARGET DEVICE - 36.
Rnnnn (+2)	= 00009 MEMORY TYPE OF TARGET - Diagnostic Status Word.
Rnnnn (+3)	= 00000 MEMORY ADDRESS OF TARGET - Begin read from Status Word 1.
Rnnnn (+4)	= 00005 DATA LENGTH - 5 words (5 registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Start storing in Series Six Register 1.

The Series Six ladder logic is shown below:

```
|
|
| I0017                                                                  00117
+--] [-----------------------------------------------------------------(OS)-+
|
|
|
|
| 00117   R0100
+--] [---[                         BLOCK MOVE                          ]+-(  )-+
|              +06201  +00036  +00009  +00000  +00005  +00001  +00000
|
|
|
| 00117   R0100
+--] [---[SCREQ]
|
|
```

For more information, see the section on Diagnostic Status Words and error codes in this chapter.

Example 24: CLEAR TARGET DIAGNOSTIC STATUS WORDS (SERIES ONE/JUNIOR/PLUS OR SERIES THREE)

Clear the target DCU or DCM Diagnostic Status Words 1-5 by writing zeroes to them from Series Six Registers 1-5. The communication is to take place on CCM port J2. The target ID is 36.

Rnnnn	= 06211 COMMAND NUMBER - Write from source Register table to DCU or DCM Diagnostic Status Words.
Rnnnn (+1)	= 00036 ID OF TARGET DEVICE - 36.
Rnnnn (+2)	= 00009 MEMORY TYPE OF TARGET - Diagnostic Status Word.
Rnnnn (+3)	= 00000 MEMORY ADDRESS OF TARGET - Start with status word 1.
Rnnnn (+4)	= 00005 DATA LENGTH - 5 words (5 registers).
Rnnnn (+5)	= 00001 MEMORY ADDRESS OF SOURCE - Begin writing from Series Six register 1. Series Six Registers 1-5 should be cleared before execution.

The Series Six ladder logic is shown below:

```
|
|
| I0018                                                              00118
+--] [-----------------------------------------------------------------(OS)-+
|
|
|
|
| 00118   R0100
+--} [---[                      BLOCK MOVE                              ]+-(  )-+
|            +06211  +00036  +00009  +00000  +00005  +00001  +00000
|
|
|
| 00118   R0100
+--] [---[SCREQ]
|
|
```

For more information, see the section on Diagnostic Status Words and error codes in this chapter.

CHAPTER 6
SERIAL INTERFACE PROTOCOL

The purpose of this chapter is to provide complete information on DCU and DCM serial interface protocol and timing 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 DCU and DCM data communications is based on the Master-Slave portion of CCM protocol developed for Series Six data communications. As used with the DCU or DCM, the host will always be the master and the DCU or DCM will always be the slave. For a complete description of all aspects of Series Six CCM protocol, see Chapter 4 of the Series Six Data Communications Manual, GEK-25364.

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 6.1 shows the data format. If parity is selected, an additional parity bit is sent.

TPK.A.40015

BIT 0	BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	BIT 8	BIT 9	BIT 10
START	LSB				DATA			MSB	* PARITY	STOP (1)

* ODD OR NONE VIA DIP SWITCH SELECTION ON DCM

NOTE: WHEN PARITY IS DISABLED, BIT 9 IS NOT INCLUDED IN THE TRANSMISSION.

Figure 6.1 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 6.1.

Table 6.1 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 ms plus 4 character times.

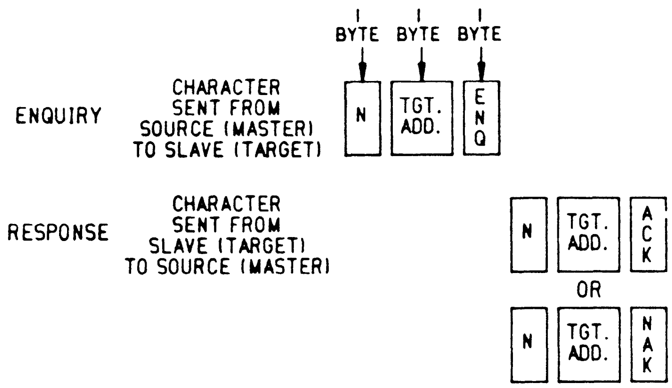
If any other character is received before the timer times out, the idle slave disregards the enquiry. Therefore, any device transmitting data text on a multidrop link should ensure that there will be no gaps in the text greater than 2 character times so an idle slave will not misinterpret data as an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

NORMAL SEQUENCE*, MASTER-SLAVE

Normal Enquiry Sequence

The form of the Normal (N) Enquiry Sequence from the master to the target slave DCU or DCM and the response by the target slave DCU or DCM is shown below. In data communications involving a DCU or DCM, the DCU or DCM is always the slave (target) and the Series Six or host computer is always the master (source).

TPK.A.40366



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 DCU Unit 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 32 times before aborting the communication.

Normal Sequence Protocol Format

The general format for a successful communication is shown in Figures 6.2 and 6.3. Figure 6.2 shows a data transfer from the source device to the target device and Figure 6.3 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.

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

TPK.A.40366

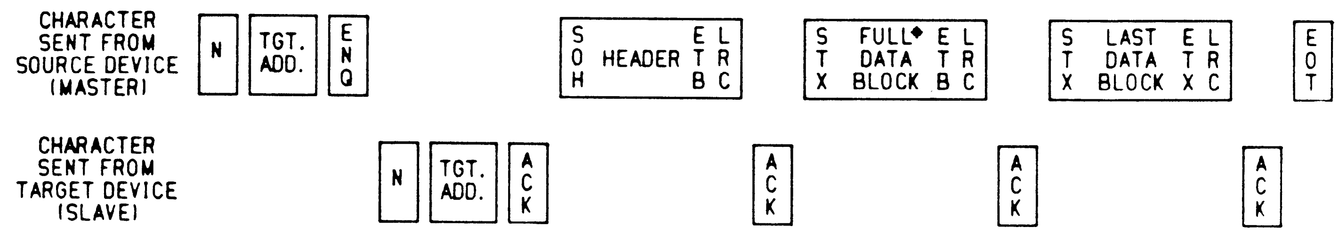


Figure 6.2 DATA TRANSFER FROM MASTER TO SLAVE

TPK.A.40367

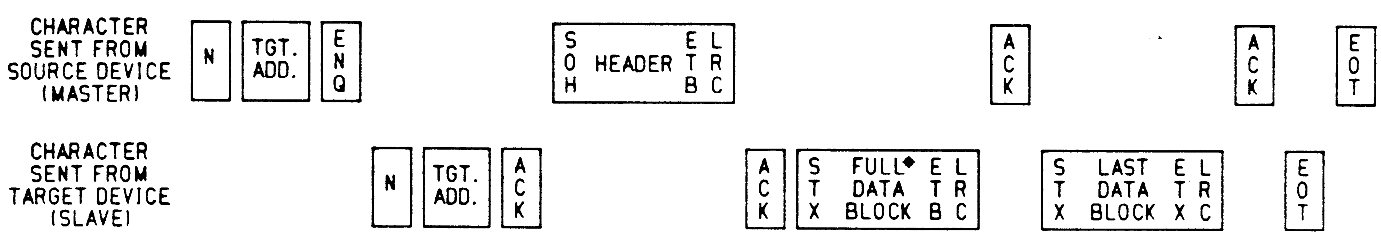


Figure 6.3 DATA TRANSFER FROM SLAVE TO MASTER

* The maximum size of a data block is 256 bytes for the Series One Plus and Series Three PCs. Because of limitations of accessing Series One or Series One Junior memory through the DCU, the standard time outs for CCM protocol shown in Table 6.2 restrict the transmission length of a single request to less than one full data block. For more information see the section, Text Data Blocks.

Master-Slave Normal Sequence Flow Charts

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

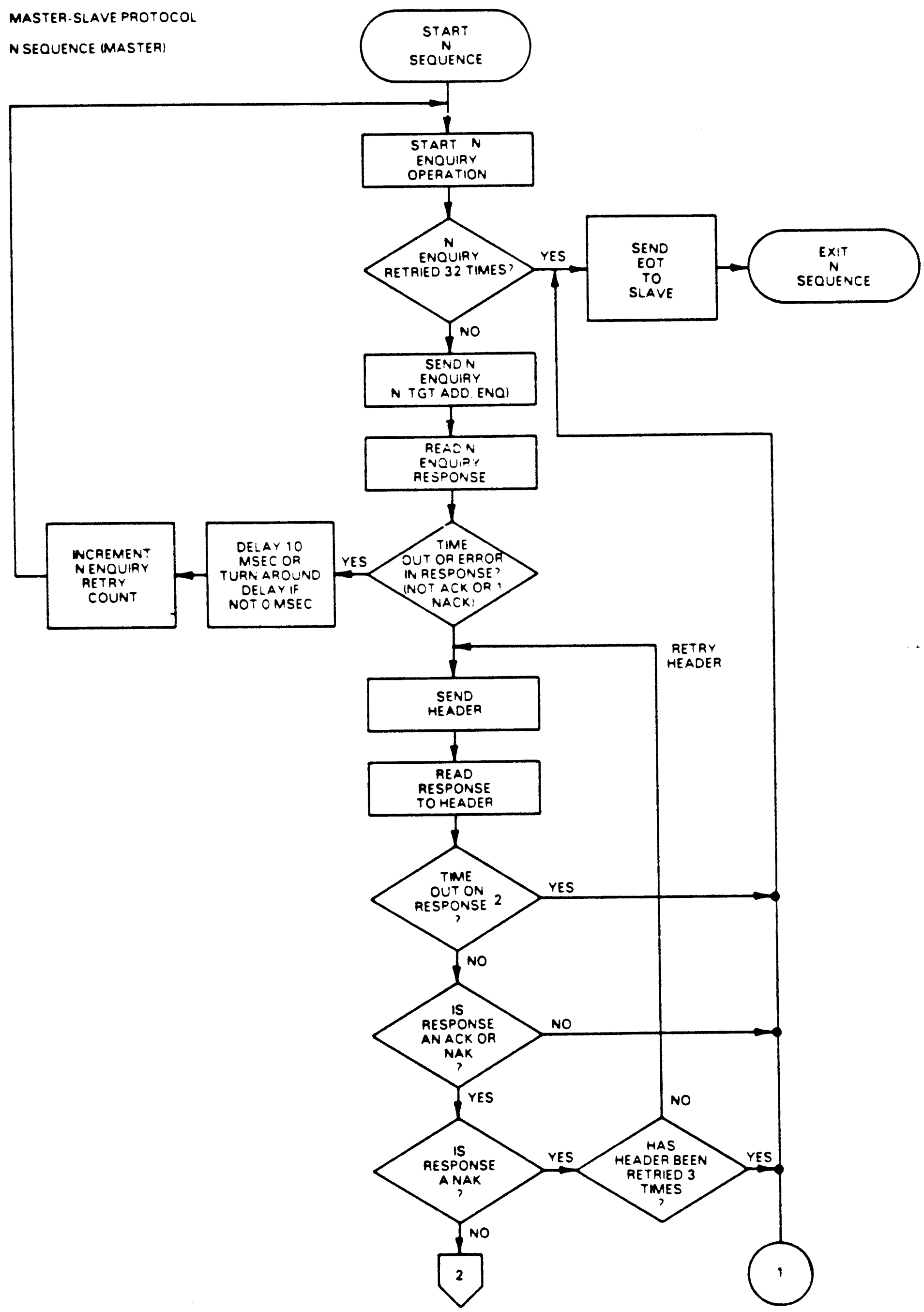
Normal Sequence, Master (See Figure 6.4)

Start N Sequence.
Start N. Enquiry.
Has enquiry been retried 32 times?
  If YES, send EOT to slave and exit N Sequence.
  If NO, send N Enquiry (N, Target Address, ENQ).
Read N Enquiry response.
Is there a time-out or error in response (response not an ACK or a NAK)?
  If yes, delay 10 ms or the turn-around delay if it is not 0 ms, increment the N Enquiry retry count, and return to "Start N Enquiry".
  If NO, send the header to the slave.
Read response to header.
Is there a time-out on the response? (Condition 4, Table 6.2)
  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 6.5)

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 time-out 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.

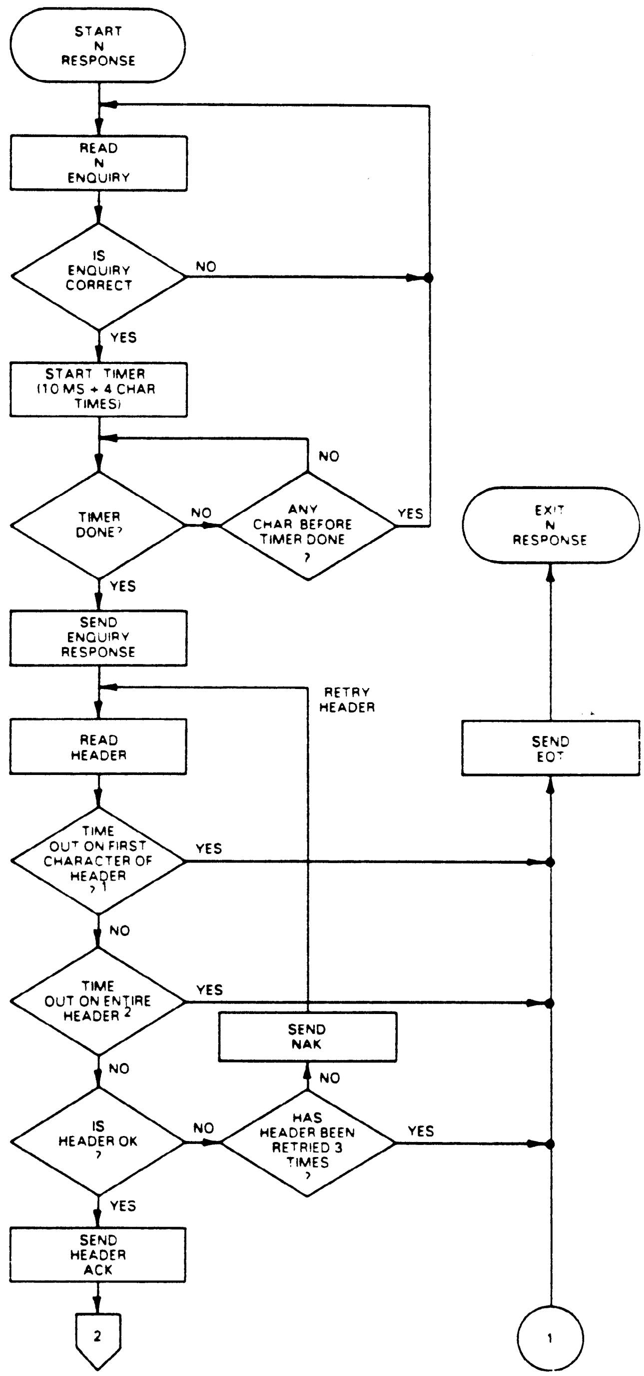
PCS6-84-0057



[1]SEE CONDITION 1, TABLE 6.2
[2]SEE CONDITION 4, TABLE 6.2

Figure 6.4 N SEQUENCE, MASTER

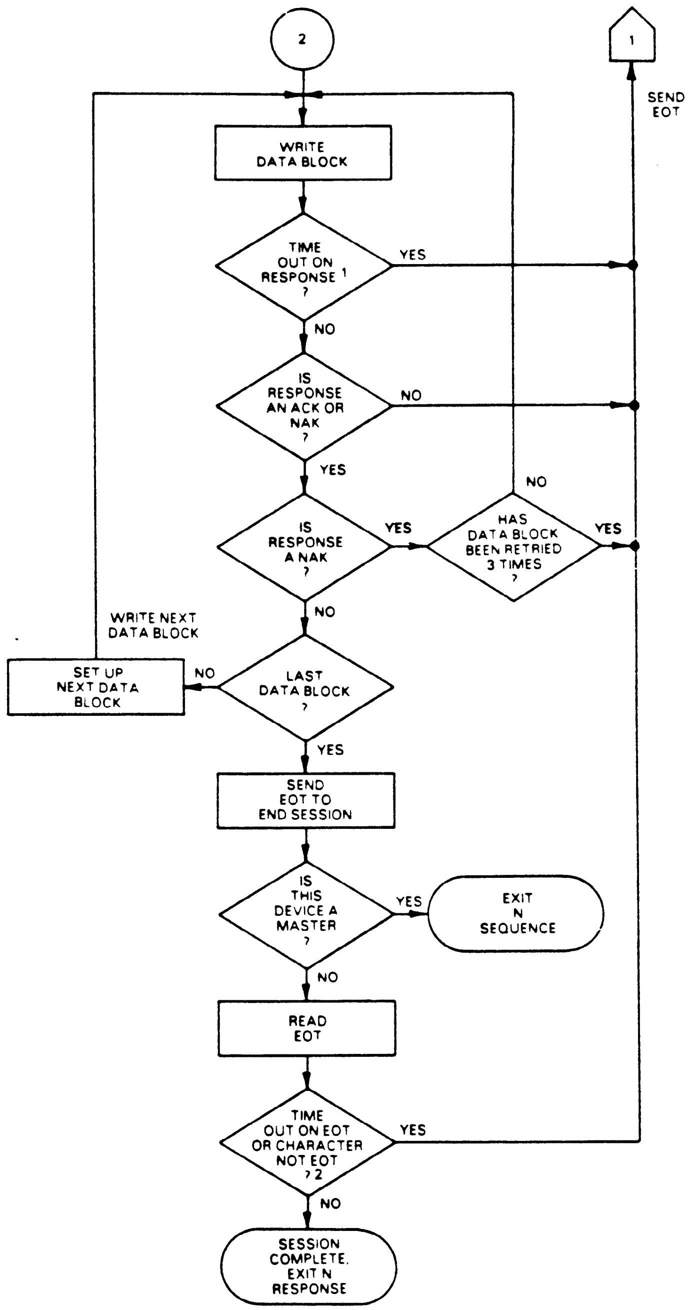
PCS6-84-0058



[1]SEE CONDITION 2, TABLE 6.2
[2]SEE CONDITION 3, TABLE 6.2

Figure 6.5 N RESPONSE, SLAVE

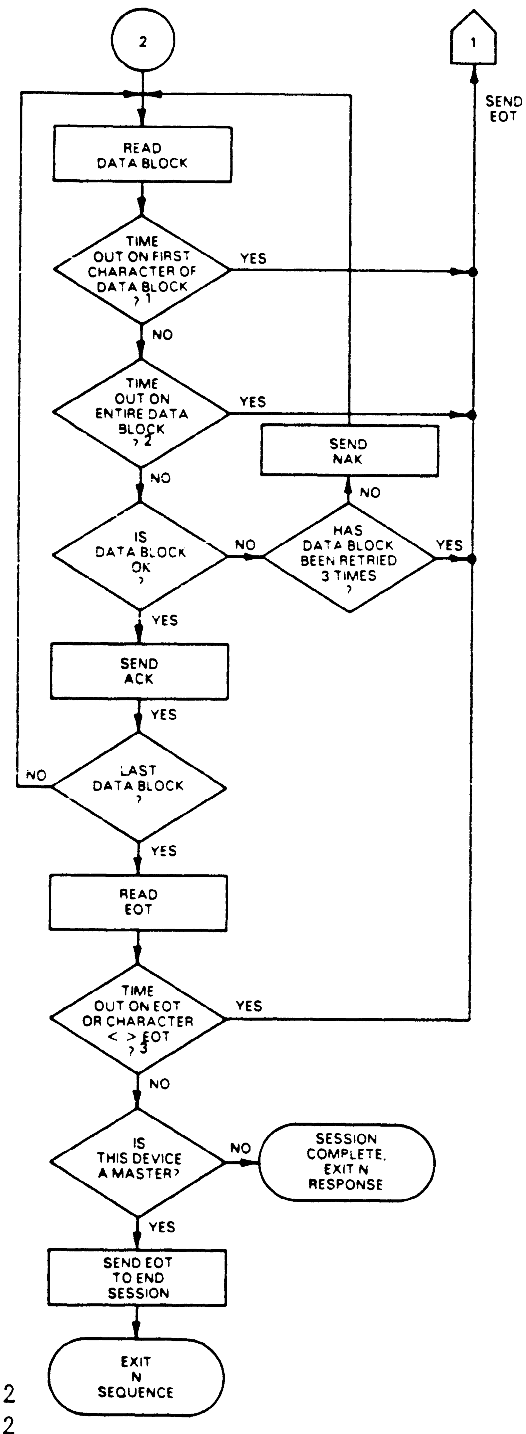
PCS6-84-0059



1SEE CONDITION 6, TABLE 6.2
2SEE CONDITION 8, TABLE 6.2

Figure 6.6 WRITE DATA BLOCKS, MASTER OR SLAVE

PCS6-84-0060



[1]SEE CONDITION 5, TABLE 6.2
[2]SEE CONDITION 7, TABLE 6.2
[2]SEE CONDITION 8, TABLE 6.2

Figure 6.7 READ DATA BLOCKS, MASTER OR SLAVE

Write Data Blocks, Master or Slave (See Figure 6.6)

Write data block.
Is there a time-out on the data block response? (Condition 6, Table 6.2)
- 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 time-out on EOT or is character not an EOT? (Condition 8, Table 6.2)
- If there is a time-out 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 6.7)

Read data block.
Is there a time-out on the first character of the data block? (Condition 5, Table 6.2)
- If YES, send an EOT and exit.
- If NO, is there a time-out on the entire data block? (Condition 7, Table 6.2)
  - 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 time-out on the EOT or is the character not an EOT?
      - If there is a time-out 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.

* For Series One and Series One Junior communications, only one partial data block can be sent per request based on the time outs in Table 6.2. Therefore, it is always the last. The flow chart and accompanying explanation describe the full functionality of CCM2 protocol.

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 6.8.

TPK.A.40368

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

BYTE 1 SOH (01H)
BYTES 2 + 3 DCU (target) ID Number (not encoded the same as the target address)
BYTES 4 + 5 Data flow direction, DCU 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 (In Series One and Series One Junior data communications, this byte is always zero).
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)

Figure 6.8 SERIAL HEADER FORMAT

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.

DCU or DCM ID Number

The DCU or DCM ID (target ID) is the identification number of the DCU or DCM 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 DCU or DCM of the direction of the transfer and the memory type involved.

Byte 4 - Direction

CONTENTS OF BYTE 4			DATA FLOW DIRECTION
DEC	HEX	ASCII	
48	30	0	Read from DCU or DCM
56	38	8	Write to DCU or DCM

Byte 5 - Memory Type

CONTENTS OF BYTE 5			TARGET MEMORY TYPE
DEC	HEX	ASCII	
49	31	1	(Memory Type 1) Data Registers* and CPU Timer/Counter Memory
50	32	3	(Memory Type 3) CPU Discrete I/O Status values (External and Internal Input/Output values, shift registers, and Timer/Counter complete* bits)
54	36	6	(Memory Type 6) CPU Scratch Pad Memory
55	37	7	(Memory Type 7) CPU User Logic Memory
57	39	9	(Memory Type 9) DCU or DCM Diagnostic Status Words

* Timer/Counter complete references and Data Registers can be accessed in Series One Plus and Series Three CPUs only.

Target Memory Address

The target memory address specifies the address within the Series One/Junior/Plus or Series Three CPU where the transfer is to begin.

Memory Type 1 The target memory address specifies the Timer/Counter accumulator or Data Register (Series One Plus and Series Three PCs only) where the data transfer is to begin. The mapping of reference numbers to numbers used for the target memory address is shown in Tables 5.1 (Series One), 5.2 (Series One Junior), 5.3 (Series One Plus), and 5.4 (Series Three).

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	0001H-0040H	0001H-0015H	0001H-0080H	0001H-00C0H

Memory Type 3 The target memory address specifies the Input or Output point where the data transfer is to begin. The transfer begins with the byte that contains the specified Input or Output. The mapping of discrete I/O reference numbers to numbers used for target memory address is shown in Tables 5.1 (Series One), 5.2 (Series One Junior). 5.3 (Series One Plus), and 5.4 (Series Three).

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	0001H-0030H	0001H-0020H (excluding 0009H-000BH)	0001H-0040H	0001H-0080H

Memory Type 6 The target memory address specifies the CPU Scratch-Pad reference at which the data transfer is to begin.

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	0000H or 0016H	0000H or 0016H	0000H, 0002H, 0004H, or 0016H	0000H or 0016H

The address can be either 0000H to access CPU operation mode status or 0016H to access PC Type status. Operation mode and PC type each consist of 2 bytes. See the section, Accessing the CPU Scratch Pad.

Memory Type 7 The target memory address specifies the User-Logic memory word at which the data transfer is to begin. Timer and Counter instructions must be written in their entirety (2 words or 4 bytes). To clear User-Logic memory, write FF to each byte to cleared.

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	0000H-06BBH	0000H-02BBH	0000H-06BBH	0000H-0FFEH

Memory Type 9 The target memory address specifies the Diagnostic Status Word at which the data transfer is to begin. The only valid target memory address is 0000H. All 5 words (10 bytes) must be read or written.

Valid	Series One	Series One Junior	Series One Plus	Series Three
Range	0000H	0000H	0000H	0000H

See the section, Diagnostic Status Words, in Chapter 5 for the definition of each word.

Number of Complete Data Blocks to Follow Header

This specifies the number of 256-byte data blocks to be transferred following the header. This number can range from 0 to 20H for Series One Plus or Series Three communications, but must be 0 for Series One or Series One Junior communications using the serial time outs in Table 6.2. For more information, see the section on Text Data Blocks.

The information below defines the unit length and accessible lengths for each Series One/Junior/Plus and Series Three memory type. This information will help you to determine how many 8-bit bytes are required for a particular transfer.

SERIES ONE/JUNIOR/PLUS SERIES THREE MEMORY TYPE	UNIT LENGTH	ACCESSIBLE LENGTHS
1: Timer/Counter Accumulator	1 Accum = 16 bits	Accumulator(s)
1: Data Registers (Series One Plus and Three Only)	1 Data Reg = 8 bits	Multiples of 2 Reg
3: Discrete I/O	1 Point = 1 bit	Multiples of 8 Points
6: Scratch Pad Bytes	1 Byte = 8 bits	2 Bytes
7: User Logic Word	1 Word = 16 bits	Word(s)
9: Diagnostic Status Word	1 Word = 16 bits	5 Words

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.

For Series One Plus and Series Three communications, this number of bytes in the last block can vary from 0001H to 00FFH. For Series One and Series One Junior communciations, this number is restricted because of the limitation of accessing memory through the DCU. For more information, see the section on Text Data Blocks.

Source ID Number

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

Text Data Block

The maximum data block size is 256 (00FFH) bytes for Series One Plus and Series Three CPUs but is less than this for Series One/Junior CPUs using the time outs in Table 6.2. This does not affect reading or writing to memory types 6 (Scratch Pad) or 9 (Diagnostic Status Words). Reading and writing to memory types 1 (T/C Accumulators), 3 (I/O and Shift Registers), and 7 (User Memory) are restricted as shown in the table below.

TYPE OF COMMUNICATION REQUEST	MAXIMUM AMOUNT OF DATA FOR EACH COMMUNICATION			
	SERIES ONE PC*		SERIES ONE JR PC	
Read from Memory Type 1 (T/C Accumulators)	58 Acc	116 Bytes	All 21 Acc	42 Bytes
Write to Memory Type 1 (T/C Accumulators)	Communication Not Supported		Communication Not Supported	
Read from Memory Type 3 (I/O and Shift Reg)	368 I/O	46 Bytes	176 I/O	22 Bytes
Write to Memory Type 3 (I/O and Shift Reg)	24 I/O	3 Bytes	No I/O, Communication Times Out	
Read from Memory Type 7 (User Memory)	75 Words	150 Bytes	25 Words	50 Bytes
Write to Memory Type 7 (User Memory)	45 Words	90 Bytes	20 Words	40 Bytes

* CPU Revision C or later.

NOTE

If you are writing a CCM protocol interface for the Series One or Series One Junior, the time outs for conditions 5, 6, and possibly 7 in Table 6.2 must be lengthened to transfer more data per request. The time outs in the Series Six CCM2 and CCM3, however, are fixed and cannot be made longer.

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 is 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

Serial Link Time-Outs

A time-out occurs on a serial link when the DCU or DCM does not receive a response, a header, or data from another device within a fixed amount of time. Time-outs are used on the serial link for error detection, error recovery, and to prevent missing end-of-block sequences. Whenever a serial link time-out occurs, the DCU or DCM 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 6.2 for time-outs at any point in the serial protocol.

Turn-Around Delays

Turn-around delay options of 0 to 10 ms for the DCU or DCM can be selected by DIP switch. A 10-ms 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 a computer or Series Six the time needed to signal the modem to turn on and ringing on the line to stop before actual transmission of data.

The DCU or DCM will delay 10 ms before sending a control character, the start of header, or the start of a text data block.

When the 10 ms turn-around delay is selected, the time is automatically added to the serial time-outs in Table 6.2.

NOTE

If a time out occurs when actual data is being transmitted to or from and Series One and Series One Junior CPU, try reducing the number of bytes to be transmitted.

Table 6.2 SERIAL LINK TIME-OUTS

CONDITION	TIME OUT WITH TURN AROUND DELAY OF 0 MS	10 MS
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	8350
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 Time Outs

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 Series One Plus 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, 6, 7, or 9.
- Reading from or writing to discrete I/O while the CPU is in Stop/Prog mode.
- Writing to PC type in the scratch pad.
- Writing to user logic while the CPU is in Run mode.
- Writing a partial instruction to user logic.
- Reading timer/counter references in Stop/Prog mode.
- Writing to timer/counter references in Stop/Prog mode or Run mode.*
- Reading timer/counter references in Stop/Prog mode.*

* Invalid but does not get NAK'ed.

The header is retried a maximum of three times. If the DCU or DCM is connected to the Series Six CCM and the header still has one of the errors listed, the CCM will abort the session and send and EOT to the DCU or DCM. The DCU or DCM 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
  Note: ETX must occur in last block only
- Parity, Overrun, or Framing Error

Invalid NAK, ACK, or EOT

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

Serial Link Time Out

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

ACCESSING THE CPU SCRATCH PAD

There are only 2 fields within the Series One, Series One Junior, or Series Three CPU Scratch Pad that can be accessed: the CPU RUN/STOP field and the PC Type field. The Series One Plus CPU Scratch Pad contains more fields which are discussed in the following sections.

The RUN/STOP field can be written to or read from using Memory Type 6 and starting address 0 with a length of 2 bytes only.

- To put the CPU in Run mode, write 0101H to address 0000H and 0001H in the Scratch Pad.
- To put the CPU in Stop mode, write 8080H to address 0000H and 0001H in the Scratch Pad.

These numbers (0101H and 8080H) also indicate the CPU mode when this field is read.

The PC Type can only be read using memory type 6 and starting address 0016H with a length of 2 bytes only. This field indicates whether the CPU is a Series One, Series One Junior, Series One Plus, or Series Three CPU.

- Series One CPU = 0101H.
- Series One Junior CPU = 0202H
- Series One Plus CPU = 0303H
- Series Three CPU = 0707H

USING THE PASSWORD AND ERROR CHECKING FEATURES OF THE SERIES ONE PLUS PC

The addressing for the Series One Plus Scratch-Pad is as follows:

Table 6.3 SERIES ONE PLUS CPU SCRATCH-PAD ADDRESSES

SERIES ONE PLUS ADDRESS (Hex)	SUB-COMMAND (Hex)	DESCRIPTION
0000		PC Mode
0002		Sub-command for executing the functions:
	0009	Logging-In with the Password
	000A	Changing the Password
	0003	Grammar checking
	0006	Reading Error Address
0004		Location of the error code generated by Grammar check and of the error location in the user program
000A		Password Write Location
0016		PC Type

Reading or writing the PC mode (RUN/STOP) and reading the PC type are the same for the Series One Plus as for the Series One/Junior and Series Three PCs (see application examples 20-22). The password and error checking features are available only for the Series One Plus PC and require the use of a sub-command written to 0002H of the Scratch-Pad (see explanation below).

Logging-in on the Series One Plus CPU using the Password

If a password has been assigned, either using the manual programmer or through communications, you must log in before executing a communications request to memory types 1, 3 and 7. If you do not log in, the communications request for these memory types will fail. It is not required to log in for communications requests to memory types 6 (Scratch Pad) and 9 (Diagnostic Status Words). Logging-In is done by executing a CCM protocol write command to the Scratch-Pad beginning at address 0002 (Hex). The write command will write 10 bytes of information as follows.

```
00 (Hex)  Where 0009H is the subcommand written to Scratch-Pad
09 (Hex)  address 0002H, and where xxxx (BCD) is the existing
00        password (Valid range 0-9999).  A value of 0 is equivalent
00        to no password.
00
00
00
00
xx (BCD)  (Password, most significant digits).
xx (BCD)  (Password, least significant digits).
```

Changing the Password of the Series One Plus PC

Changing the password is a 2-step operation. First, you must log in as explained in the preceding section. Then you must execute another write command to the Series One Plus Scratch Pad beginning at address 0002. The write command will write 10 bytes of information as follows.

```
00 (Hex)  Where 000AH is the subcommand written to Scratch-Pad
0A (Hex)  address 0002H, and where xxxx (BCD) is the new password
00        (Valid range 0-9999).  A value of 0 is equivalent to no
00        password.
00
00
00
00
xx (BCD) (Password, most significant digits).
xx (BCD) (Password, least significant digits).
```

User Program Error Checking

A complete program error check can be initiated at any time on a program in the Series One Plus CPU as explained below.

Reading the error code is a 4 step operation.

1. To initiate the error check, write the subcommand, 0003H, to the Series One Plus Scratch-Pad address 0002H.

2. Read the error code from the Scratch Pad address the contents of address 0004H. If the contents of address 0004 is zero, there is no error code. If the contents of address 0004 is not zero, then this is the error code. Go to the next step to find the location of the first error in the user program.

3. To find the location of the error, write the subcommand, 0006H, to Scratch-Pad address 0002.

4. Read the location of the error from the Scratch-Pad address 0004. The contents of address 0004 is the location of the first error in user memory.

Table 5.9 defines the errors which may be found in a user program when the Series One Plus CPU is transition from PROGRAM to RUN.

INDEX

INDEX

INDEX

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