

***Programmable Control Products***

***Series 90\* PLC  
SNP Communications Driver***

***User's Manual***

GFK-0585A

February 2010



## ***Warnings, Cautions, and Notes as Used in this Publication***

### **Warning**

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

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

### **Caution**

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

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

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

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

\* indicates a trademark of GE Intelligent Platforms, Inc. and/or its affiliates. All other trademarks are the property of their respective owners.

## Contact Information

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

### *General Contact Information*

Online technical support and GlobalCare	<a href="http://www.ge-ip.com/support">http://www.ge-ip.com/support</a>
Additional information	<a href="http://www.ge-ip.com/">http://www.ge-ip.com/</a>
Solution Provider	<a href="mailto:solutionprovider.ip@ge.com">solutionprovider.ip@ge.com</a>

### *Technical Support*

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

### *Americas*

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

### *Europe, the Middle East, and Africa*

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

### *Asia Pacific*

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

# Preface

---

## Contents of Manual

The purpose of this manual is to describe the software installation, operation, and programming of the GE Series Ninety Protocol (SNP) Driver. The Series 90 SNP protocol is used to retrieve/store data or issue commands to a given Series 90 PLC via the serial port.

A summary of the contents is provided here:

**Chapter 1. Introduction:** Provides an overview of the GE SNP Driver software and briefly discusses the possible system configurations.

**Chapter 2. Installation and Configuration:** Includes installation and configuration information for the GE SNP driver on a Personal Computer (PC) operating in the MS-DOS® environment.

**Chapter 3. SNP Driver Functions:** Describes the SNP Driver function requests for operation in the MS-DOS environment.

**Appendix A. Serial Port and Cables:** Describes the Series 90 PLCs serial port, converter, and cables used to connect to Series 90 PLCs.

**Appendix B. Files on the Diskette:** Provides a listing of the files and a description of each of the files that are supplied on the SNP Master Driver diskette.

**Appendix C. Application Notes:** Contains helpful information for the application programmer.

## Related Publications

The following documents are Series 90 PLC products that may be associated with the SNP software installation and application.

- GFK-0262      Series 90-70 Programmable Controller Installation Manual
- GFK-0263      Logicmaster 90 Programming Software User's Manual
- GFK-0356      Series 90-30 Programmable Controller Installation Manual
- GFK-0402      Series 90-30 Hand-Held Programmer User's Manual
- GFK-0582      Series 90 PLC Serial Communications User's Manual

At GE Intelligent Platforms, we strive to produce quality technical documentation. After you have used this manual, please take a few moments to complete and return the following Reader's Comment card.

®MS-DOS is a registered trademark of Microsoft Corporation.

<b>Chapter 1</b>	<b>Introduction .....</b>	<b>1-1</b>
	SNP Driver Overview .....	1-1
	DOS Version .....	1-3
	SNP Driver Implementations .....	1-3
	Systems Configurations .....	1-4
	RS-232/RS-422Point-to-Point Connection .....	1-4
	RS-422 Point-to-Point Connection .....	1-5
	Series 90 PLC Modem Serial Link .....	1-5
	Series 90 PLC Multidrop Serial Link .....	1-6
	Definition of Terms .....	1-8
	Acronyms and Abbreviations .....	1-8
	Basic Terms .....	1-8
 <b>Chapter 2</b>	 <b>Installation and Configuration .....</b>	 <b>2-1</b>
	SNP Driver Software .....	2-1
	License Agreement .....	2-1
	SNP Driver Implementations .....	2-1
	Installation of the TSR SNP Driver Software .....	2-1
	Installation of the Library Driver Software .....	2-3
	SNP Driver Configuration .....	2-3

<b>Chapter 3</b>	<b>SNP Driver Functions</b>	<b>3-1</b>
	SNP Driver Interface	3-1
	SNP Driver Functions	3-1
	Open Port	3-2
	Close Port	3-3
	Change CPU ID	3-3
	Change T1 Timer	3-3
	Change T2 Timer	3-4
	Change T3p Timer	3-4
	Change T4 Timer	3-4
	Change Modem Turnaround Time	3-5
	Change Port Parameters	3-5
	Change Maximum Data Size	3-5
	De-Install SNP Master Driver	3-6
	Process Configuration File	3-6
	Get Version	3-7
	Establish Session	3-7
	Programmer Logon	3-9
	Return Controller Type and ID	3-10
	Read/Write System Memory	3-11
	Return PLC Time/Date	3-16
	Set PLC Time/Date	3-16
	Return Fault Table	3-17
	Fault Entry Format	3-18
	Clear Fault	3-21
	Set PLC State	3-21
	Example C Program	3-22
	Error Status Codes	3-23
<b>Appendix A</b>	<b>Serial Port and Cables</b>	<b>A-1</b>
	What this Appendix Contains	A-1
	<b>Section 1: RS-422/RS-485 Interface and Cabling Information</b>	<b>A-2</b>
	Cable and Connector Specifications	A-3
	Series 90 PLC Serial Port	A-4
	Workmaster Serial Port	A-5
	IBM-AT/XTSerialPort	A-6
	RS-232/RS-422Converter	A-7
	<b>Section 2: The Isolated Repeater/Converter(IC655CCM590)</b>	<b>A-9</b>
	Description of the Isolated Repeater/Converter	A-9
	System Configurations	A-13
	Cable Diagrams	A-15
	Serial Cable Diagrams	A-19
	RS-232 Point-to-Point Connections	A-19
	RS-422 Point-to-Point Connection	A-21

<b>Appendix B</b>	<b>Diskette Files .....</b>	<b>B-1</b>
	SNP Master Driver Object Diskette Contents .....	B-1
	Executable Files .....	B-1
	C Header Files .....	B-1
	C Source File .....	B-2
	SNP Master Driver Source Diskette Contents .....	B-2
	Assembly Files: .....	B-2
	C Source File: .....	B-2
	C Header Files: .....	B-3
	Assembly Include File: .....	B-3
	Other Files: .....	B-3
<b>Appendix C</b>	<b>Application Notes .....</b>	<b>C-1</b>
	Application Note for the TSR SNP Driver .....	C-1
	Application Notes for the Library SNP Driver .....	C-1
	Application Notes for Either TSR or Library Implementation .....	C-2

Figure 1-1. RS-232/RS-422 Point-to-Point Serial Connection to Series 90-70 PLC .....	1-4
Figure 1-2. RS-232/RS-422 Point-to-Point Serial Connection to Series 90-30 PLC .....	1-4
Figure 1-3. RS-422 Point-to-Point Serial Connection to Series 90-70 PLC .....	1-5
Figure 1-4. RS-422 Point-to-Point Serial Connection to Series 90-30 PLC .....	1-5
Figure 1-5. Point-to-Point Modem Serial Link Connection for Series 90 PLCs .....	1-5
Figure 1-6. Multidrop Serial Link Connection for Series 90 PLCs .....	1-6
Figure 1-7. Multidrop System Using RS-422 Isolated Repeater/RS-232 Converter .....	1-7
Figure A-1. Series 90 PLC, RS-422 Serial Port Connector Configuration .....	A-4
Figure A-2. Workmaster RS-232 Serial Port Connector Configuration .....	A-5
Figure A-3. IBM-AT/XTSerialPort .....	A-6
Figure A-4. RS-232 to RS-422 Converter Logic Diagram .....	A-7
Figure A-5. Example RS-232 to RS-422 Converter Connection .....	A-8
Figure A-6. The Isolated/Repeater Converter .....	A-10
Figure A-7. RS-422 Isolated Repeater/RS-232 Converter Logic Diagram .....	A-11
Figure A-8. Simple System Configuration Using the Isolated Repeater/Converter .....	A-14
Figure A-9. Complex System Configuration Using the Isolated Repeater/Converter .....	A-14
Figure A-10. Cable A; RS-232 Device To Converter .....	A-15
Figure A-11. Cable B; RS-422 Device to Converter .....	A-15
Figure A-12. Cable C; RS422 Twisted Pair .....	A-16
Figure A-13. Cable D; RS422 Twisted Pair .....	A-17
Figure A-14. Cable E; RS-232 Converter to CMM .....	A-18
Figure A-15. Workmaster II (25-pin Serial Connection to Series 90 PLCs .....	A-19
Figure A-16. IBM-AT (compatibles) Personal Computer to Series 90 PLCs .....	A-20
Figure A-17. Workmaster or IBM-XT (compatibles) Personal Computer to Series 90 PLCs .....	A-20
Figure A-18. Typical RS-422, Host to PLC Connection, with Handshaking .....	A-21



Table 2-1. Default Configuration .....	2-3
Table 2-2. Port Designation .....	2-4
Table 2-3. Series 90 PLC Timer Defaults .....	2-5
Table 3-1. Local SNP Driver Functions .....	3-2
Table 3-2. Remote SNP Driver Functions .....	3-2
Table 3-3. Controller Type and ID Destination Buffer .....	3-10
Table 3-4. PLC Major/Minor Code Types .....	3-11
Table 3-5. Reference Data Segment Selectors .....	3-12
Table 3-6. PLC Time/Date Destination Buffer .....	3-16
Table 3-7. PLC Time/Date Source Buffer .....	3-17
Table 3-8. Fault Header Format .....	3-17
Table 3-9. PLC Fault Entry .....	3-18
Table 3-10. I/O Fault Entry .....	3-19
Table 3-11. Fault Action Codes .....	3-20
Table 3-12. Fault Group Codes .....	3-20
Table 3-13. Major Error Status Codes .....	3-23
Table 3-14. Minor Error Status Codes: Major Code = 5 .....	3-24
Table 3-15. Minor Error Status Codes: Major Code =10 (OAh) .....	3-25
Table A-1. Connector/Cable Specifications .....	A-3
Table A-2. Series 90 PLC, RS-422 Serial Port Pin Assignment .....	A-4
Table A-3. Workmaster RS-232 Serial Port Pin Assignment .....	A-5
Table A-4. IBM-AT/XT Serial Port Pin Assignment .....	A-6
Table A-5. RS-232/RS-422 Converter Pin Assignment .....	A-8

# Chapter 1

## Introduction

---

---

The Series Ninety Protocol (SNP) driver software is used to issue commands, to send data to, and retrieve data from, a Series 90 Programmable Logic Controller (PLC) for operator interface or other applications. The SNP driver handles the details of the SNP protocol allowing the application programmer to focus on the application software.

### SNP Driver Overview

The SNP driver will reside in the host device supporting the master side of the communication link only. The application software via the driver initiates all communications and the slave (Series 90 PLC CPU) responds to the master's requests. There is no peer-to-peer capability with the SNP protocol (i.e., a PLC cannot be the master; the PLC can only be a slave). The SNP driver supports the following communication requests.

**Open Port:** allows the application program to open the serial port using the installed SNP Driver's port configuration.

**Close Port:** allows the application program to close the serial port.

**Change CPU ID:** allows the application program to change the SNP Driver's target CPU ID.

**Change T1:** allows the application program to change the SNP Driver's T1 timer value for subsequent communication sessions.

**Change T2:** allows the application program to change the SNP Driver's T2 timer value for subsequent communication sessions.

**Change T3p:** allows the application program to change the SNP Driver's T3p timeout value for subsequent communication sessions.

**Change T4:** allows the application program to change the T4 timer value for subsequent communication sessions.

**Change Modem Turnaround Time:** allows the application program to change the modem turnaround time value for subsequent communication sessions.

**Change Port Parameters:** allows the application program to change the serial port configuration for subsequent communication sessions by changing the following parameters: baud rate, port number, number of stop bits, and parity.

**Change Maximum Data Size:** allows the application program to change the maximum number of data bytes allowed in a single SNP Data message for subsequent communication sessions.

**De-Install Driver:** allows the user to remove all SNP Master Driver Interrupt hooks: specifically, to un-hook from the Hardware Timer Interrupt, restore the Clock Frequency to its original value, and restore the SNP Interrupt to its original value. This request is only implemented in the Library Version of the SNP Master Driver; it is not valid in the TSR version.

**Process Configuration File:** allows the user to supply the SNP Master Driver with a configuration file which the driver will parse and process.

**Get Version:** allows the application program to retrieve the SNP Driver's Major and Minor software version number.

**Establish Session:** allows the application program to establish a communication session with a Series 90 PLC.

**Programmer Logon:** allows the application program to login and logout as a programmer attachment.

**Return Controller TYPE and ID:** returns the CPU controller ID and the Major and Minor TYPE of the PLC CPU. It also returns information about the current program and configuration stored within the PLC CPU.

**Return PLC Time:** returns the current time and date stored in the PLC CPU to the application program.

**Set PLC Time:** allows the application program to set the internal time and date of the PLC CPU.

**Return Fault Table:** returns fault data from either the PLC fault table or the I/O fault table to the application program. The PLC fault table has a maximum of 16 faults and the I/O fault table has a maximum of 32 faults.

**Clear Fault Table:** clears either the PLC fault table or the I/O fault table.

**Set PLC State:** allows the application program to change the execution state of the PLC. Allowable choices are:

- Run mode with I/O enabled.
- Run mode with I/O disabled (Series 90-70 only).
- Stop mode with I/O enabled.
- Stop mode with I/O disabled.

**Read System Memory:** allows the application program to read a block of PLC reference data (%I,%Q,%T,%M,%S,%SA,%SB,%SC,%G,%R,%AI,%AQ).

**Write System Memory:** allows the application program to write a block of PLC reference data (%I,%Q,%T,%M,%SA,%SB,%SC,%G,%R,%AI,%AQ).

## DOS Version

The SNP Master Driver requires MS-DOS version 2.0 or later.

## SNP Driver Implementations

Two different implementations of the Master SNP Driver are available: a Terminate and Stay Resident (TSR) implementation and a linkable library implementation.

If you choose to use the TSR implementation, you must run an executable file supplied by GE to install the TSR driver. The TSR driver starts off as a normal program, but when it exits, it invokes the Terminate and Stay Resident DOS function, thereby leaving some of the code resident. Once you install the TSR, the code remains until you either CTRL-ALT-DEL or cycle power your PC. The SNP TSR Driver takes approximately 12K bytes.

If you choose to use the linkable library supplied by GE, you must link in the SNP driver library along with your application object files to produce an executable file.

### Caution

**The SNP Master Driver hooks into the Hardware Clock Interrupt (Int 8). Normally, this interrupt is generated every 54.925 milliseconds, at which time the BIOS timer tick service is called. SNP has timers which require greater resolution than 55 milliseconds. To achieve greater clock resolution, we adjust the 8253 clock chip to run at 16 times the normal frequency when the SNP Master Driver is installed. We also hook into the Clock Interrupt (Int 8) and perform a divide-by-16 adjustment so that the BIOS timer tick service is called at the normal rate.**

**If you are using the linkable SNP Driver, the clock frequency MUST be restored upon exiting the user application program. This is accomplished by the SNP De-install function which is described in Chapter 3, SNP Driver Functions. The user MUST invoke this function on exit from the user program.**

**Currently, there is no De-install feature implemented for the TSR version of the SNP Driver. Therefore, if you are using the TSR implementation, the clock frequency is never restored until you either press CTRL-ALT-DEL or cycle power on the PC. You must be aware of the changed frequency in case you load other applications that hook into the Clock Interrupt, and assume a 54.925 millisecond frequency. These applications must be loaded PRIOR to loading the TSR SNP Driver in order for proper timing to take place. If an application which hooks into the Clock Interrupt is loaded after the TSR SNP Driver, the application's Clock Interrupt will be called every 3.433 milliseconds rather than the normal 54.925 milliseconds.**

## Systems Configurations

Several system configurations are possible. There may be just one PLC on the serial link (direct, point to point connection), or there may be many Series 90 PLCs on a serial link (multi-drop port connection). Only one SNP master may be on a multi-drop link. Communication over modems is also supported. The following figures illustrate some of the possible master/slave device connections:

If you have a host device equipped with an RS-232 Interface, the RS-232/RS-422 converter (GE catalog number IC690ACC900 or IC690ACC901) is required.

### RS-232/RS-422 Point-to-Point Connection

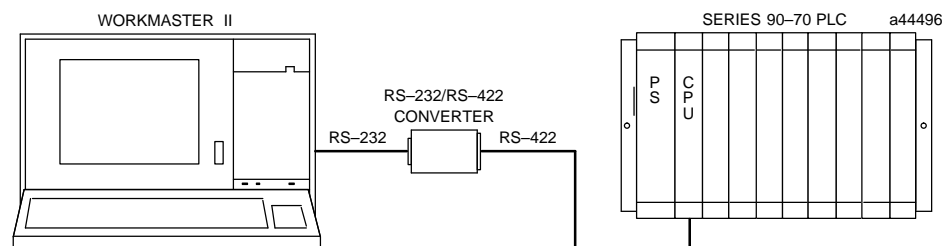


Figure 1-1. RS-232/RS-422 Point-to-Point Serial Connection to Series 90-70 PLC

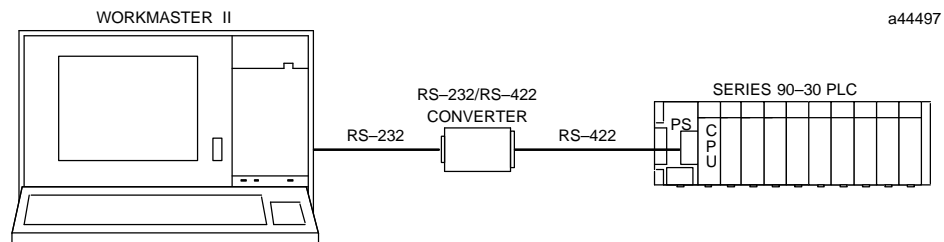


Figure 1-2. RS-232/RS-422 Point-to-Point Serial Connection to Series 90-30 PLC

**Write System Memory:** allows the application program to write a block of PLC reference data (%I,%Q,%T,%M,%SA,%SB,%SC,%G,%R,%AI,%AQ).

## DOS Version

The SNP Master Driver requires MS-DOS version 2.0 or later.

## SNP Driver Implementations

Two different implementations of the Master SNP Driver are available: a Terminate and Stay Resident (TSR) implementation and a linkable library implementation.

If you choose to use the TSR implementation, you must run an executable file supplied by GE to install the TSR driver. The TSR driver starts off as a normal program, but when it exits, it invokes the Terminate and Stay Resident DOS function, thereby leaving some of the code resident. Once you install the TSR, the code remains until you either CTRL-ALT-DEL or cycle power your PC. The SNP TSR Driver takes approximately 12K bytes.

If you choose to use the linkable library supplied by GE, you must link in the SNP driver library along with your application object files to produce an executable file.

### Caution

**The SNP Master Driver hooks into the Hardware Clock Interrupt (Int 8). Normally, this interrupt is generated every 54.925 milliseconds, at which time the BIOS timer tick service is called. SNP has timers which require greater resolution than 55 milliseconds. To achieve greater clock resolution, we adjust the 8253 clock chip to run at 16 times the normal frequency when the SNP Master Driver is installed. We also hook into the Clock Interrupt (Int 8) and perform a divide-by-16 adjustment so that the BIOS timer tick service is called at the normal rate.**

**If you are using the linkable SNP Driver, the clock frequency MUST be restored upon exiting the user application program. This is accomplished by the SNP De-install function which is described in Chapter 3, SNP Driver Functions. The user MUST invoke this function on exit from the user program.**

**Currently, there is no De-install feature implemented for the TSR version of the SNP Driver. Therefore, if you are using the TSR implementation, the clock frequency is never restored until you either press CTRL-ALT-DEL or cycle power on the PC. You must be aware of the changed frequency in case you load other applications that hook into the Clock Interrupt, and assume a 54.925 millisecond frequency. These applications must be loaded PRIOR to loading the TSR SNP Driver in order for proper timing to take place. If an application which hooks into the Clock Interrupt is loaded after the TSR SNP Driver, the application's Clock Interrupt will be called every 3.433 milliseconds rather than the normal 54.925 milliseconds.**

## Series 90 PLC Multidrop Serial Link

In the multidrop configuration, the host device is configured as the master and one or more PLCs are configured as slaves. Figure 1-6 illustrates a multidrop connection with conversion capability only, Figure 1-7 illustrates a multidrop connection having isolation, repeater and converter capabilities.

For detailed information and example multidrop connections refer to Appendix A, Serial Port and Cables.

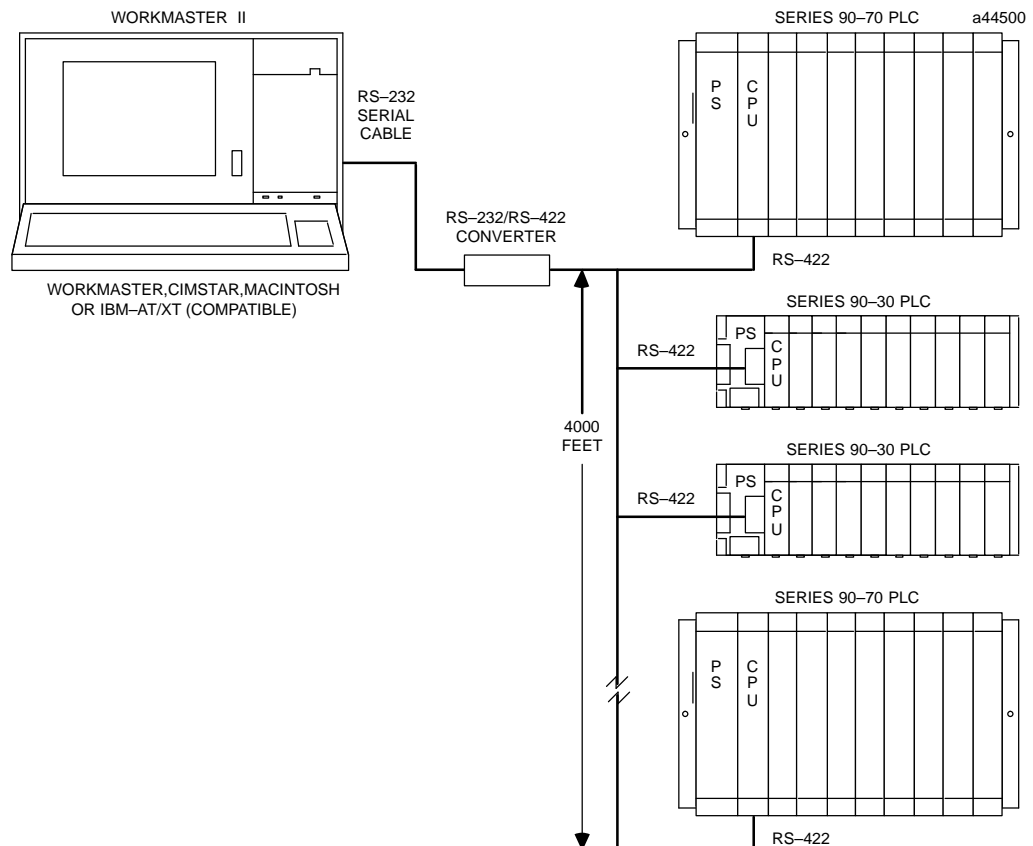
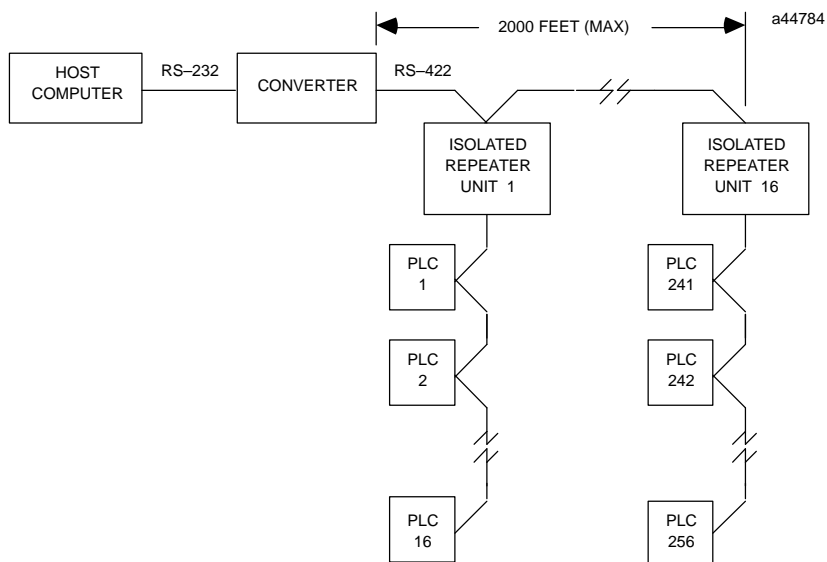


Figure 1-6. Multidrop Serial Link Connection for Series 90 PLCs

### Caution

Optical isolation networks may be necessary for certain applications.  
(Refer to Figure 1-7.)

Where isolation is required, the RS-422 Isolated Repeater/RS-232 Converter (GE catalog number IC655CCM590) can be used in place of the RS-232/RS-422 Converter (Figure 1-6). In addition to converting from RS-232 to RS-422 communications, this unit provides ground isolation where a common ground cannot otherwise be established between components.



**Figure 1-7. Multidrop System Using RS-422 Isolated Repeater/RS-232 Converter**

Refer to Appendix A for detailed information about the serial port, cable connections, and electrical signals.



## Definition of Terms

A brief list of acronyms, abbreviations, and terms used throughout this manual are introduced here in order to facilitate the reading of this manual.

### Acronyms and Abbreviations

ASCII	<b>A</b> merican <b>S</b> tandard <b>C</b> ode for <b>I</b> nformation <b>I</b> nterchange. An eight-bit (7 bits plus 1 parity bit) code used for data.
BCD	<b>B</b> inary <b>C</b> oded <b>D</b> ecimal.
CPU	<b>C</b> entral <b>P</b> rocessing <b>U</b> nit; usage in this document refers to a Series 90 family PLC CPU.
I/O	<b>I</b> nput <b>O</b> utput
K	1024
LM90	<b>L</b> ogic <b>M</b> aster <b>90</b> : GE Intelligent Platforms Programming software that can be used to create ladder logic programs and configuration for the Series 90 PLC family. The LM90 programming software connects to the target PLC via the Work Station Interface (WSIB/WS9) hardware or a serial communication port.
MS	<b>M</b> ill <b>i</b> second.
NUL	The ASCII NUL character (00h - hexadecimal).
PLC	<b>P</b> rogrammable <b>L</b> ogic <b>C</b> ontroller
SMEM	<b>S</b> ystem <b>M</b> EMory (%I, %Q, %T, %M, %SA, %SB, %SC, %S, %G, %R, %AI, %AQ) .
SNP	<b>S</b> eries <b>N</b> inety <b>P</b> rotocol.
TSR	<b>T</b> erminate and <b>S</b> tay <b>R</b> esident.
UART	<b>U</b> niversal <b>A</b> synchronous <b>R</b> eceiver <b>T</b> ransmitter; samples communications bit-stream input data.

### Basic Terms

Hexadecimal	A numbering system, having 16 as a base, represented by the digits '0' through '9', and 'A' through 'F'.
Master SNP Device	Host device on which the master implementation of the SNP protocol is running.
Programmer Attachment	A device which has identified itself as a programmer (i.e., "logged in" to the PLC) and therefore is granted access to certain extra commands such as changing the state of the PLC, and clearing the Fault tables.
Slave SNP Device	Any CPU in the Series 90 PLC family.
Series 90-30 PLC	Pertains to any of the CPU models within the Series 90-30 PLC family; specifically the 301, 311, or 331 models.
Series 90-70 PLC	Pertains to any of the CPU models within the Series 90-70 PLC family; specifically the 731, 732, 771, 772, 781, or 782 models.
VMEbus	Versa Module European - An electrical and mechanical bus specification used by Series 90-70 PLCs.

# Chapter 2

## *Installation and Configuration*

---

This chapter includes installation and configuration instructions for the GE SNP driver software on a Personal Computer (PC) operating in the MS-DOS environment.

### SNP Driver Software

The GE SNP driver software resides on a single, 5 1/4 inch or a 3 1/2 inch, diskette.

#### Caution

**Make a working copy of the SNP software diskette. Store the original diskette in a safe place and use only the working copy.**

### License Agreement

Use of software means you accept the provisions of the license packaged in the documentation and shipped with your software. Software use is restricted; no copies may be made except as provided in the license. If the license is unacceptable, do not use the software. Return the entire contents of the package to GE Intelligent Platforms.

### SNP Driver Implementations

As previously stated in Chapter 1, two implementations of the Master SNP Driver are provided: a Terminate and Stay Resident (TSR) implementation and a linkable library implementation.

Installation and Configuration procedures differ between the two implementations. The application programmer should choose between the two implementations, and then refer to the appropriate sections in this chapter for Installation and Configuration procedures. (Refer to Chapter 1 and Appendix B for descriptions/differences between the TSR and linkable library implementations.)

### Installation of the TSR SNP Driver Software

- Boot up the computer using DOS.
- Insert the driver software diskette into the computer's floppy disk drive.
- The TSR SNP Driver is contained in one file called SNPEXE, which may be copied to the hard disk using the copy command (e.g., copy A:SNPEXE C:).

Other files, such as C Header files and example C Source files, are included on the SNP Driver software diskette but are not necessary to the installation process. (See Appendix B, Diskette Files, for a description of all files included on the diskette.)

- The SNP driver may be installed from the hard disk or the driver diskette. Install the SNP driver by typing "SNP" at the DOS prompt and pressing the ENTER key.

`C>SNP`                      or                      `A>SNP`

At this point, you should see the message:

`Processing SNP Driver Configuration...`

During this time, the driver installation software validates any configuration data changes that you may have specified. Refer to the next section, "SNP Driver Configuration" for details on how to change the default.

If errors are encountered, an explicit error message will be displayed, stating which configuration parameter is in error. A separate message is displayed for each error encountered (see example below).

Upon completion of configuration processing, the message:

`SNP Driver Configuration processing complete.`

is displayed. After all configuration data is processed, the driver is installed as a Terminate Stay Resident (TSR) program. The driver stays resident until you either CTRL-ALT-DEL or cycle power on the PC.

You should now see the message:

`SNP TSR Driver Installed: Rev 1.01`

followed by a copyright message. Therefore, a first-time installation with no configuration parameter errors should yield these messages:

`Processing SNP Driver configuration...`  
`SNP Driver Configuration processing complete.`

`SNP TSR Driver Installed: Rev 1.01`  
`Copyright 1994. All Rights, including trade secret rights, are`  
`reserved. Unauthorized use of the information or program is`  
`strictly prohibited.`

`Installation of this software conforms acceptance of the terms`  
`and conditions of the license agreement distributed with this`  
`product.`

Installation of the driver is now complete. Any subsequent attempts to install the driver (e.g., to change configuration parameters) results in a message informing you that the driver is already installed.

`Processing SNP Driver configuration...`  
`SNP Driver Configuration processing complete.`

`SNP TSR Driver Already Installed: Rev 1.01`

## Example Installation Error Message

Suppose that when you first install the driver, you specify new values for the baud rate and the T1 timer, but neither new value fall within the acceptable range. The following messages will be displayed:

`Processing SNP Driver configuration...`  
`Unsupported Baud Rate. Baud rate was not changed.`  
`T1 time specified is out of range. The time was not changed.`  
`SNP Driver Configuration processing complete.`

The revision level displayed in each of these examples assumes the first revision of the driver. The actual revision level displayed when you install the driver is dependent on the revision level of your driver software.

## Installation of the Library Driver Software

- Boot up the computer using DOS.
- Insert the driver software diskette into the computer's floppy disk drive.
- The Library SNP Driver is contained in one file called SNPLIB, which may be copied to the Hard disk using the copy command (e.g., copy A:SNPLIB C:).

Other files, such as C Header files and example C source files, are contained on the SNP driver software diskette but are not necessary to the installation or usage of the library. (See Appendix B, Diskette Files, for a description of all files contained on the diskette.)

- The user application code must make a far call to a function called:

`Init_snp_driver`

in order to install the library version of the SNP Driver. This call MUST be made prior to invoking any of the SNP Driver Functions. The function returns a status, where the only error status is a bad DOS version. You must have DOS version 2.0 or later. (See Chapter 3 for error status descriptions.)

- The file, SNPLIB must be linked in with the user application object files which resolves the far call `Init_snp_driver` and contains the SNP Driver code.

Unlike the TSR implementation, which always remains resident, you can de-install the library version of the SNP driver. In fact, you MUST de-install the SNP driver on exiting the user application program in the library implementation. This is accomplished via the De-install SNP driver function (see Chapter 3).

## SNP Driver Configuration

This section explains the SNP configuration required to establish a successful communication link: baud rate, target PLC, serial port, parity, stop bits, modem turnaround time, maximum data size and timers. The installed SNP driver uses the following default configuration.

Table 2-1. Default Configuration

Mnemonic	Default Value	Description
BAUD	19200 *	The baud rate for the communication.
CPUID	EightNULs	The CPU ID of the PLC CPU that the host wishes to establish a communication session with.
PORT	1	The COM port number, from COM1 to COM4.
PARITY	ODD	The parity used, either NONE, ODD, or EVEN.
STOPBITS	1	The number of stop bits, either 1 or 2.
MODEMTIM	0	The modem turnaround time.
MAXDATA	1000	The maximum SNP data size.
T1	10	The T1 timer's value in milliseconds.
T2	1000	The T2 timer's value in milliseconds.
T3p	3000	The T3p timer's value in milliseconds.
T4	50	The T4 timer's value in milliseconds.

\* The BAUD rate can be set to any of the following values (300, 600, 1200, 2400, 4800, 9600, 19200).

## CPU ID

The CPU ID field is 8 bytes long. For the Series 90-70 PLCs, the CPU ID can be a maximum of 7 bytes followed by a NUL character (0) and can include any ASCII character. For the Series 90-30 PLCs, the CPU ID is restricted to a maximum of 6 bytes followed by a NUL character (0). The values of the 6 bytes are further restricted to the ASCII characters '0' through '9' inclusive and 'A' through 'F' inclusive (must be capital letters).

## Serial Port Designation

The ports are numbered as shown in the table below:

Table 2-2. Port Designation

Serial Port	Port Designation	Port Base Address	Interrupt Request Input (IRQ)
COM1	1	3F8h	4
COM2	2	2F8h	3
COM3	3	3E8h	4
COM4	4	2E8h	3

## Modem Turn-Around Time

The Modem Turn-Around Time is used specifically for modem applications where larger turn-around times may be required due to modem characteristics. If you are using modems to communicate, this configuration parameter must be set to at least one (1). The range of values for modem turn-around time is 0 to 2550 milliseconds.

## SNP Data Size

The SNP data size refers to the number of data bytes allowed in one SNP Data message. A SNP Data message is one of the three SNP message types: Text Buffer message, Block Transfer message, or Connection Data message. All other SNP message types have a fixed size and therefore are not affected by the SNP data size.

You may specify the maximum amount of data allowed in SNP messages. The default, and the maximum allowed data, for the SNP Driver is 1000 bytes.

You may decrease this value if needed for a given application, where the minimum allowed is 40 bytes. A smaller SNP data message may be useful in an environment where there is noise on the serial link. The smaller data messages provide more frequent error checking.

## Timers

The SNP Driver provides several timers whose values may be adjusted by the user to allow the serial communication link to run as efficiently as possible for a given situation. The following is a description of the timers:

- T4** T4 is the break processing time. It is the minimum time the master is required to wait between the end of transmission of the long break and the first character of the attach message. The break processing time is necessary in order to give the slave time to recognize the long break and prepare to receive the Attach message. This timer is used only by the master, but its value is determined by the PLC CPUs on the serial link (must be at least 50 milliseconds; 600 milliseconds with modems).
- T1** T1 is the amount of time that must elapse between the reception or transmission of the last character of one message and the transmission of the first character of the next message. It is the minimum amount of time the sending device (master or slave) must wait before transmitting either a message or an acknowledgement. This timer is used by both the master and the slave. T1 time must account for the worst case time for the sending device to turn the link around and set up to be a receiving device, *including* any modem turn around time. The range for T1 is 5 to 50 milliseconds.
- T2** T2 is the maximum amount of time that the sending device (master or slave) will wait for an acknowledgement to the message that it just transmitted. The first character of the acknowledgement must be received within T2 time after the last character of the message is sent, or else it is assumed that there has been a loss of connection, and the device goes back to its “start” state. The “start” state for the master is a state in which required conditions for establishing a communication session (based on the user’s given application) are fulfilled. This timer is used by both the master and the slave. The range for T2 is 500 milliseconds to 6.5 seconds.
- T3p** T3p is the link idle timeout time. After the last character of an acknowledge is sent, the first character of the next message must be transmitted within T3p time. If one device is waiting for a message from the other device, and T3p time is exceeded before any part of the message is received (i.e. no character was received within T3p time), a loss of communication is assumed and the device goes back to its “start” state. Once any part of a message is received, T3p time must elapse after the last character was received before the receiving device may assume loss of communication. This timer is used by both the master and the slave. The range for T3p is 1 to 60 seconds.

The following table shows the defaults for all four timers for Series 90-30 PLCs, and Series 90-70 PLCs, and Logicmaster 90.

**Table 2-3. Series 90 PLC Timer Defaults**

Timer	Series 90-30 PLC	Series 90-70 PLC
T1	5 milliseconds ( + modem turnaround)	5 milliseconds ( + modem turnaround)
T2	1 second	500milliseconds
T3p	10seconds	5 seconds (currently disabled on Series 90-70PLCs)
T4	N/A	N/A

## Changing the Configuration Parameters

Each configuration parameter has a SNP Driver Function Request by which the user can change the driver's configuration parameter value for subsequent communication sessions. These function requests include:

```
Change CPU ID
Change T1 Timer
Change T2 Timer
Change T3p Timer
Change T4 Timer
Change Modem Turnaround Time
Change Port Parameters (baud, port number, parity, stop bits)
Change Maximum Data Size
```

and are described in detail in Chapter 3, SNP Driver Functions.

There is another SNP Driver Function Request provided called Process Configuration File, by which the user can specify new configuration data in a file. The configuration file uses the mnemonics listed in Table 2-1 to specify individual parameters, and sets them equal to new values. The space, tab, carriage return, and line feed characters are used as delimiters. Any number of configuration parameters can be changed via one call to this request. This clearly has an advantage over the individual configuration function requests listed above if you have a lot of configuration data to change.

For example, suppose the user wants a baud rate of 9600, no parity, two stop bits, comm port number 3, CPU ID of "PLC0001", a maximum data size of 60, a modem turnaround time of 20 milliseconds, a T1 time of 15 milliseconds, a T2 time of 1000 milliseconds, and a T3p time of 5 seconds. Using the appropriate mnemonics from Table 2-1, a configuration file containing these parameter values delimited by a <CR><LF> has the following format:

```
BAUD=9600
PARITY=NONE
STOPBITS=2
PORT=3
CPUID=PLC0001
MAXDATA=60
MODEMTIM=20
T1=15
T2=1000
T3p=5000
```

The same parameter data in a file delimited by spaces looks as follows:

```
BAUD=9600 PARITY=NONE STOPBITS=2 PORT=3
CPUID=PLC0001 MAXDATA=60 MODEMTIM=20
T1=15 T2=1000 T3p=5000
```

The user then invokes the Process Configuration File function request, passing the name of this file to the driver. The file is opened and processed, and the configuration data is changed. (See Chapter 3 for further details.)

The method of changing the SNP Driver's configuration data via the driver function requests discussed above can be used with either the TSR or Library SNP Driver implementation. In fact, it is the only method available to the Library SNP Driver implementation. The next section discusses a method available only to the TSR implementation.

## Alternative Method to Configuration Data change with TSR version

If you use the TSR SNP Driver implementation, the driver configuration data can be changed via another method: via the command line when the driver is installed. The command line has the following form:

```
C>SNP parameter=value [parameter=value...]
```

where "parameter" is equal to a configuration mnemonic listed in Table 2-1.

For example, to change the default driver configuration to use COM2 at 9600 BAUD, type the following at the DOS prompt:

```
C>SNP BAUD=9600 PORT=2
```

The parameters not listed remain set to their default values.

A configuration file of the same format described above in the Process Configuration File function request can also be specified at the command level as follows:

```
C>SNP @filename
```

the "@" symbol must precede the filename.

Two methods of specifying a filename and specifying individual parameters can be used interchangeably on the same command line. For instance, the following is legal:

```
C>SNP BAUD=9600 @config.snp PORT=1
```

If a variable is mentioned more than once, the second definition is used. For instance, if a file named CONFIG.SNP contains:

```
BAUD=19200          PORT=3
```

and the following command line is entered,

```
C>SNP BAUD=9600 @config.snp PORT=1
```

then the baud rate is set to 19200, and the port is set to COM1.

If you specify a configuration file that does not exist, the driver installation software will display an error message. In the previous example, if the file CONFIG.SNP does not exist, the following message is displayed:

```
Configuration file not found: @config.snp
```



# Chapter 3

## SNP Driver Functions

---

---

This chapter describes the SNP driver interface and function requests when operating in the MS-DOS environment.

### SNP Driver Interface

All SNP Driver function calls are invoked by interrupt 53H (decimal 83). The individual driver functions are selected by placing the appropriate function number in the AX register. Any parameter data required for a specific driver function are passed in the other registers (e.g., BX, CX, DX, ES, DI). All SNP Driver function calls return a status in the AX register, where:

```
0    ---> Driver function call completed successfully
< 0  ---> An error in the request was encountered, where a Major Error
          Status code is returned in the AL register and the Minor Error
          Status code (if applicable) is returned in the AH register.
```

The Minor Error Status only has meaning for certain Major Error Status codes. Tables 2-13 and 3-14 at the end of this chapter, define the Major and Minor Error Status codes. Minor codes are undefined for a given Major Error Status unless otherwise stated.

### SNP Driver Functions

The SNP Driver Functions are organized into two basic groups: Local functions and Remote functions:

- The Local functions interact with the installed driver to change the driver's configuration data, to open and close the user's serial port device, and to perform other driver housekeeping functions.
- The Remote functions interact with the PLC CPU: they are the means through which the user transfers data to or from the PLC CPU and issues commands to the PLC CPU. Some of the remote functions require the requesting device to be at a minimum access privilege level before the request can be honored, and some require the requesting device to be logged in as a programmer attachment.

The Local SNP Driver Functions (Table 3-1) lists those functions that affect or interact with the installed SNP Driver, along with their function number. The Remote SNP Driver Functions (Table 3-2) lists those functions that affect or interact with the PLC CPU, along with the privilege level and logon requirements.

Table 3-1. Local SNP Driver Functions

SNP Driver Function	Function Number
Open Port	0
Close Port	1
Change CPU ID	2
Change T1 Timer	3
Change T2 Timer	4
Change T3p Timer	5
Change T4 Timer	6
Change Modem Turnaround Time	7
Change Port Parameters	8
Change Maximum Data Size	9
De-install SNP Master Driver	10
Process Configuration File	11
Get Version	12

Table 3-2. Remote SNP Driver Functions

SNP Driver Function	Function Number	Series 90-30 PLC Privilege Level (min)	Series 90-70 PLC Privilege Level (min)	Logged In
EstablishSession	22	N/A	N/A	N/A
ProgrammeLogon/Logoff	23	1	0	NO
Return Controller Type and ID	24	1	1	NO
Read System Memory	25	1	1	NO
Write System Memory	26	2	2	NO
Return PLC Time/Date	27	1	1	NO
Set PLC Time/Date	28	2*	2	NO
Return Fault Table	29	1	1	NO
Clear Fault Table	30	2	2	YES
Set PLC State	31	2	2	YES

\* For Series 90-30 PLCs with Release 1.x CPU firmware, privilege level 3 or higher is required.

## Open Port

The Open Port Request allows the application program to open the serial port using the installed SNP Driver's port configuration.

The Open Port Request is issued by setting up the AX register and invoking the driver interrupt routine.

**AX = 0      (Open Port function number)**

This request cannot fail; therefore, AX always returns zero (success).

## Close Port

The Close Port Request allows the application program to close the serial port.

The Close Port Request is issued by setting up the AX register and invoking the driver interrupt routine.

**AX = 1 (Close Port function number)**

This request cannot fail; therefore, AX always returns zero (success).

## Change CPU ID

The Change CPU ID Request allows the application program to change the SNP Driver's CPU ID. This request does not affect any communications session that may currently be active with another PLC CPU. The next time an Establish Session Request is issued, the new CPU ID specified by the Change CPU ID Request is used in the establish session sequence in order to connect to a PLC CPU whose CPU ID is a match.

The Change CPU ID Request is issued by setting up the following registers and invoking the driver interrupt routine.

**AX = 2 (Change CPU ID function number)**  
**ES:DI = pointer to the new CPU ID**

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request.

where the CPU ID pointed to by ES:DI was invalid: either the first character of the CPU ID was not alphanumeric, or it was not NUL-terminated.

## Change T1 Timer

The Change T1 Timer Request allows the application program to change the SNP Driver's T1 timer value for *subsequent* communication sessions. The T1 timer value for a current communication session is not affected. The new T1 timer value set by the Change T1 Timer Request does not become active until an Established Session Request is issued.

The Change T1 Timer Request is issued by setting up the following registers and invoking the driver interrupt routine.

**AX = 3 (Change T1 Timer function number)**  
**BX = new T1 Timer value in milliseconds**

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request

where the new T1 timer value passed in the BX register was out of range. The valid range for T1 is 5 to 50 milliseconds.

## Change T2 Timer

The Change T2 Timer Request allows the application program to change the SNP Driver's T2 timer value for *subsequent* communication sessions. The T2 timer value for a current communication session is not affected. The new T2 timer value set by the Change T2 Timer Request does not become active until an Established Session Request is issued.

The Change T2 Timer Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 4 (Change T2 Timer function number)
BX = new T2 Timer value in milliseconds
```

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request.

where the new T2 timer value passed in the BX register was out of range. The valid range for T2 is 500 to 6500 milliseconds.

## Change T3p Timer

The Change T3p Timer Request allows the application program to change the SNP Driver's T3p timer value for *subsequent* communication sessions. The T3p timer value for a current communication session is not affected. The new T3p timer value set by the Change T3p Timer Request does not become active until an Establish Session Request is issued.

The Change T3p Timer Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 5 (Change T3p Timer function number)
BX = new T3p Timer value in milliseconds
```

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request

where the new T3p timer value passed in the BX register was out of range. The valid range for T3p is 1 to 60 seconds.

## Change T4 Timer

The Change T4 Timer Request allows the application program to change the SNP Driver's T4 timer value for *subsequent* communication sessions. The T4 timer value for a current communication session is not affected. The new T4 timer value set by the Change T4 Timer Request does not become active until an Establish Session Request is issued.

The Change T4 Timer Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 6 (Change T4 Timer function number)
BX = new T4 Timer value in milliseconds
```

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request

where the new T4 timer value passed in the BX register was out of range. The valid range for T4 is 50 to 6500 milliseconds.

## Change Modem Turnaround Time

The Change Modem Turnaround Time Request allows the application program to change the SNP Driver's Modem Turnaround timer value for *subsequent* communication sessions. The Modem Turnaround timer value for a current communication session is not affected. The new Modem Turnaround timer value set by the Change Modem Turnaround Time Request does not become active until an Establish Communication Session Request is issued.

The Change Modem Turnaround Time Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 7 (Change Modem Turnaround Time function number)
BX = new Modem Turnaround Time value in milliseconds
```

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request

where the new Modem Turnaround Time value passed in the BX register was out of range. The valid range for Modem Turnaround Time is 0 to 2550 milliseconds.

## Change Port Parameters

The Change Port Parameters Request allows the application program to change the SNP Driver's port configuration for *subsequent* communication sessions. The port configuration for a current communication session is not affected. The new port configuration specified by the Change Port Parameters Request does not become active until an Open Port Request is issued.

The Change Port Parameters Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 8 (Change Port Parameters function number)
BX = baud rate (300, 600, 1200, 2400, 4200, 9600, 19200)
CL = port number (1, 2, 3, or 4)
DL = number of stop bits (1 or 2)
DH = parity (0=none, 1=even, 2=odd)
```

## Change Maximum Data Size

The Change Maximum Data Size Request allows the application program to change the SNP Driver's maximum number of data bytes allowed in one SNP Data message for *subsequent* communication sessions. The SNP Data Size for the current communication session is not affected. The new SNP Data size set by the Change Maximum Data Size Request does not become active until an Establish Communication Session Request is issued.

The Change Maximum Data Size Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 9  (Change Maximum Data Size function number)
BX = new maximum data size
```

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request

where the new Maximum Data Size value passed in the BX register was out of range. The valid range for Maximum Data Size is 40 bytes to 1000 bytes.

## De-Install SNP Master Driver

The De-Install SNP Master Driver Request allows the application program to remove all SNP Master Driver Interrupt hooks: specifically, to un-hook from the Hardware Timer Interrupt, restore the Clock Frequency to its original value, and restore the interrupt vector used by the SNP Driver to its original value. This request is only implemented in the Library version of the SNP Master Driver. It is not a valid request to the TSR version of the SNP Master Driver. If the application program makes a De-Install request to the TSR version, an error is returned in the AX register which reflects invalid parameter (i.e., invalid function number).

The De-Install SNP Master Driver Request is issued by setting up the AX register and invoking the driver interrupt routine.

```
AX = 10  (De-Install SNP Master Driver function number)
```

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request

in the case where this function is called from a TSR implementation. The De-install Driver function has not been implemented in the TSR implementation, and is therefore an invalid function to call. In the Library implementation, this function call always returns zero (success).

## Process Configuration File

The Process Configuration File Request allows the application program to change any of the driver configuration data (i.e. port configuration, SNP timer values, CPU ID, SNP Data Size) for *subsequent* communication sessions. The configuration data for a current communication session is not affected. Any port configuration (baud rate, port number, number of stop bits, or parity) changed by a Process Configuration File Request does not become active until an Open Port Request is issued. All other driver configuration data (SNP timer values, CPU ID, and SNP Data Size) changed by a Process Configuration File Request does not become active until an Establish Communication Session Request is issued.

The user passes the pathname and filename (in the form of a NUL-terminated ASCII string) of a configuration file to the driver. The driver opens the file, parses and processes it, and then closes the file.

The Process Configuration File Request is issued by setting up the following registers and invoking the driver interrupt routine.

**AX = 11** (Process Configuration File function number)  
**ES:DI** = pointer to the pathname and filename to be processed as configuration data

If the request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The only reason for an error status is:

- Invalid input parameter in request

due to an error encountered with the configuration file provided. Possible problems encountered that would produce the invalid input parameter return error include the following:

1. Pathname or filename was not found.
2. Invalid mnemonic keyword encountered within the configuration file.
3. Value specified for a particular configuration item was invalid or out of range for that item.

## Get Version

The Get Version Request allows the user to obtain the SNP Driver's Major and Minor Software version numbers.

The Get Version Request is issued by setting up the following register and invoking the driver's interrupt routine.

**AX = 12** (Get Version function number)

If the SNP Driver is installed, AX returns zero (success), the Major Version number returns in BX, and the Minor Version number returns in CX.

If AX does not return zero, the driver is not installed and the return in AX is undefined.

## Establish Session

The Establish Session Request establishes a communication session with a SNP slave device (Series 90 PLC). This request must be issued prior to any of the other SNP Driver functions that issue remote requests to a PLC CPU (e.g., Programmer Logon, Return Controller Type and ID, READ/Write System Memory, etc.) All configuration data required for this communication session must be set up prior to invoking this request. The Establish Session Request takes the following actions:

- Attaches to the Series 90 PLC whose ID matches the installed driver's configuration CPU ID.

In the case of a point to point connection, where there is only one PLC CPU on the serial link, the installed driver's configuration CPU ID may be the ASCII NUL string. The single PLC on the serial link will always respond to a NUL CPU ID regardless of its CPU ID.

In the case of a multi-drop connection, each PLC CPU must have a distinct CPU ID. The driver establishes the communication session with the PLC whose CPU ID matches the driver's configured CPU ID.

- The driver and the PLC resolve the time values for the SNP timers (T1, T2, T3p) and the maximum SNP Data Size to be used for this communication session.
- Sets the user's access privilege level to the highest unprotected level.

The Establish Session Request is issued by setting up the AX register and invoking the driver interrupt routine.

**AX = 22 (Establish Session function number)**

The Establish Session function call returns success (AX = 0) if the request successfully completes. Otherwise, AX contains an error status. The most common reason for an error status is:

- No communication.

There are several reasons why a No Communication error would be returned:

1. The port configuration for the host device on which the SNP Driver is running does not match that of the PLC (e.g., baud rates are different, parity is different, number of stop bits are different.)
2. There is not a ID match between the SNP Driver configured CPU ID and PLCs CPU ID.

This breaks down into two scenarios: point-to-point versus multi-drop.

**Point-to-Point:** If there is only one PLC on the link, a SNP Driver's configured CPU ID of NUL (default) is sufficient. The PLC will always respond to the NUL CPU ID. If the Driver's CPU ID is not NUL, then it must equal the PLC CPUs ID in order to establish a communication session.

**Multi-drop:** Each PLC CPU on a multi-drop link must have a distinct CPU ID. The user must change the SNP Driver's configured CPU ID according to the PLC it wishes to establish a communication session with.



## Programmer Logon

The PLC Programmer Logon Request allows a user to login and logout as a programmer attachment. Some service requests require that the user be logged in as a programmer attachment such as Clear Fault Table and Set PLC State. When the user establishes a communication session with a Series 90-70 PLC the user is automatically logged into the Null task. Therefore, no explicit logon request is required in order to issue the Clear Fault Table or Set PLC State service requests. There is no automatic logon when the user establishes a communication session with a Series 90-30 PLC. The user must make an explicit logon request to the Null task before issuing the Clear Fault Table or Set PLC State service requests.

### PLC Programmer Logon

The PLC Programmer Logon Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 23 (PLC Programmer Logon function number)
BX = 2 (Login Serial Mode)
    = 3 (Logout Mode used to logout from program task)
    = 4 (Logout Bye Mode used when connection is to be broken)
CX = word value (Block Transfer Buffer Size)
ES:DI = pointer to an 8 byte NUL-terminated ASCII string
        (Program Task Name)
```

The Series 90-30 PLCs do not differentiate between “logout mode” and “logout bye mode”, so either value can be used when logging out of the programmer window.

The Block Transfer Buffer Size is used by the Series 90-70 PLC only. If you are communicating to a Series 90-70 PLC the Block Transfer Buffer Size must be a non-zero, positive number (e.g., 16384=4000h). If you are communicating to the Series 90-30 PLC the Block Transfer Buffer Size field and pointer to the Program Task Name are ignored.

The Program Task Name is a 8-byte NUL-terminated ASCII string or 8 NULs if logged on to the Null Task.

If the Programmer Logon request successfully completes, the AX register returns zero (success). Otherwise, AX contains an error status. The most common reasons for an error status is:

- Invalid task name (Series 90-70 PLC only).
- No Block Transfer size specified (Series 90-70 PLC only).

### Example of the Series 90 Programmer Logon Request

The Programmer Logon Request to a Series 90 PLC would require the registers to be initialized as follows before invoking the interrupt routine.

```
AX = 23 (PLC Programmer Logon function number)
BX = 2 (Login Serial Mode)
CX = 00 40 (Block Transfer Buffer Size = 16384)
ES:DI = pointer to the Program Task Name (e.g. "STAT_1")
```

## Return Controller Type and ID

The PLC Return Controller Type and ID Request returns the CPU Controller ID and the Major and Minor type of the PLC CPU. It also returns information about the current program and configuration stored within the PLC CPU such as the Main Control Program name, number of program blocks and the master checksums for the the program and configuration.

The Return Controller Type and ID Request is issued by setting up the following registers and invoking the driver interrupt routine.

```

AX = 24 (Return Controller Type and ID function number)
ES:DI = pointer to the destination buffer in memory where the
        user wishes the returned control information to be
        stored.

```

The destination buffer must be allocated by the application software prior to invoking this request. The destination buffer is defined as follows.

**Table 3-3. Controller Type and ID Destination Buffer**

Bytes	Description	Bytes	Description
Bytes 1-8	CPU Controller ID	Bytes 23-26	Total Length of ProgramBlocks
Byte 9	PLC Series 90 Major Type	Bytes 27-28	Sum of Program Blocks AdditiveChecksums
Byte 10	PLC Series 90 Minor Type	Bytes 29-32	Sum of ProgramBlocksCRC Checksums
Byte 11	Number of Control Programs	Bytes 33-34	Length of Configuration Re- cords
Byte 12	Reserved	Bytes 35-36	Sum of Configuration Additive Checksums
Bytes 13-20	Main Control ProgramName	Bytes 37-40	Sum of Configuration CRC Checksums
Bytes 21-22	Number of ProgramBlocks		

## Major/Minor Code Types

Table 3-4. PLC Major/Minor Code Types

Major Type		Description	Minor Type		Description
Dec	Hex		Dec	Hex	
12	0Ch	Series 90-70 PLC CPU	31	1Fh	Series 90-70 Model 731 CPU
			32	20h	Series 90-70 Model 732 CPU
			71	47h	Series 90-70 Model 771 CPU
			72	48h	Series 90-70 Model 772 CPU
			80	50h	Series 90-70 Model 780 CPU
			81	51h	Series 90-70 Model 781 CPU
			82	52h	Series 90-70 Model 782 CPU
			88	58h	Series 90-70 Model 788 CPU
			89	59h	Series 90-70 Model 789 CPU
			92	5Ch	Series 90-70 Model 914 CPU
			94	5Eh	Series 90-70 Model 924 CPU
16	10h	Series 90-20 PLC CPU, Series 90-30 PLC CPU	31	1Fh	Series 90-20 Model 211 CPU
			30	1Eh	Series 90-30 Model 311 CPU
			32	20h	Series 90-30 Model 321 CPU
			33	21h	Series 90-30 Model 313 CPU
			34	22h	Series 90-30 Model 323 CPU
			35	23h	Series 90-30 Model 331 CPU
			36	24h	Series 90-30 Model 341 CPU

The Return Controller Type and ID function call should return success (AX = 0) as long as a communication session has been established and no communication errors (i.e., parity error, overrun error, framing error) have been encountered.

## Read/Write System Memory

This section describes the function requests that allow the user to read and write reference data from the Series 90 PLC.

- The Read System Memory Request
- The Write System Memory Request

Discrete data may be accessed as bit data or byte data. Registers and Analog inputs and Analog outputs may be accessed as word data only.

Each possible access mode (bit, byte or word) for all reference memory types is assigned a distinct selector. Table 6-4 lists the reference data segment selectors accessible to the SNP user along with the size of the corresponding memory type on a per model basis. Read requests to memory types within the table require a privilege level of 1 or higher. Write requests require a privilege level of 2 or higher. It should be noted that the %S discrete memory types are READ ONLY.

Table 3-5. Reference Data Segment Selectors

Memory Type <sup>1</sup>	Selector		Access Mode
	Decimal	Hex	
Discrete Inputs (%I)	70	46	bit
	16	10	byte
Discrete Outputs (%Q)	72	48	bit
	18	12	byte
Discrete Temporaries (%T)	74	4A	bit
	20	14	byte
Discrete Internals (%M)	76	4C	bit
	22	16	byte
%SA Discretes	78	4E	bit
	24	18	byte
%SB Discretes	80	50	bit
	26	1A	byte
%SC Discretes	82	52	bit
	28	1C	byte
%S Discretes	84	54	bit
	30	1E	byte
Genius Global Data (%G)	86	56	bit
	56	38	byte
Analog Inputs (%AI)	10	0A	word
Analog Outputs (%AQ)	12	0C	word
Registers (%R)	08	08	word

<sup>1</sup> The maximum addressable ranges for each memory type depends on the model of CPU and memory configuration.

**SegmentSelector:**

Byte field which specifies the memory type to be accessed. Refer to Table 3-5, Reference Data Segment Selector for valid values.

**DataOffset:**

Word field (least significant byte first followed by most significant byte) which specifies an index into the memory type where access is to begin. Data Offset is zero-based, and is defined in terms of bit, byte or word, depending on the segment selector specified.

**DataLength:**

Word field (least significant byte first followed by most significant byte) which specifies the length of data to be accessed within the specified memory type. Data Length is defined in terms of bit, byte or word, again depending on the segment selector specified.

The following examples show the values of these three fields for different modes of access (bit, byte, or word). The key items to remember when looking at these examples are that the two fields Data Offset and Data Length are defined in terms of bits, bytes, or words, depending on the segment selector, and that the field Data Offset is **zero-based**

**Examples of BIT access:**

## (1) %M35 in bit mode:

Segment Selector	=	4C	(%M memory in bit mode)
Data Offset	=	22 00	(= 0022h = 34 = bit index to %M35)
Data Length	=	01 00	(= 0001 = one bit)

## (2) %M97 to %M112 in bit mode:

Segment Selector	=	4C	(%M memory in bit mode)
Data Offset	=	60 00	(= 0060h = 96 = bit index to %M97)
Data Length	=	10 00	(= 0010h = 16 bits)

**Examples of BYTE access:**

## (1) %Q497 to %Q512 in byte mode:

Segment Selector	=	12	(%Q memory in byte mode)
Data Offset	=	3E 00	(= 003eh = 62 = byte index to %Q497)
Data Length	=	02 00	(= 0002h = 2 bytes)

## (2) %T17 to %T208 in byte mode:

Segment Selector	=	14	(%T memory in byte mode)
Data Offset	=	02 00	(= 0002h = byte index to %T17)
Data Length	=	18 00	(= 0018h = 24 bytes)

**Examples of WORD access:**

## (1) %R1 to %R78 in word mode:

Segment Selector	=	00	(%R memory in word mode)
Data Offset	=	00 00	(= 0 = word index to %R1)
Data Length	=	4E 00	(= 004Eh = 78 words)

## (2) %R93 to %R98 in word mode:

Segment Selector	=	08	(%R memory in word mode)
Data Offset	=	5C 00	(= 005Ch = 92 = word index to %R93)
Data Length	=	06 00	(= 0006h = 6 words)

## Read System Memory

The Read System Memory Request (Read SMEM) returns a block of PLC reference data (%I, %Q, %T, %M, %S, %SA, %SB, %SC, %G, %R, %AI, %AQ) from the PLC.

The Read System Memory Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 25 (Read System Memory function number)
BL = segment selector (see table)
DX = data offset
CX = data length (see table)
ES:DI = pointer to the destination buffer
```

The destination buffer must be allocated by the application software prior to invoking the Read System Memory function call. The size of the destination buffer must be large enough to handle the data being returned.

### Example of Read SMEM: %M in Bit Mode

This example reads the eleven discrete internal bits %M99 through %M109, using the %M bit mode segment selector (BL = 4Ch) and a data offset (DX = 62 00) and data length (CX = 0B 00) whose units are bit.

The example assumes that the master has already attached to the slave device, and the discrete points at %M97 through %M112 have the following values:

M112			M109						M99			M97			
1	0	1	1	0	0	1	1	1	0	0	1	0	0	1	0

Reading M99 through M109 inclusive returns the two bytes "90 13" where %M97 - %M104 = 90h and %M105 - %M112 = 13h.

M109										M99					
0	0	0	1	0	0	1	1	1	0	0	1	0	0	0	0

It should be noted that all bits **not** requested within a byte are returned as zero, regardless of their true state.

## Write System Memory

The Write System Memory Request (Write SMEM) writes a block of PLC reference data (%I, %Q, %T, %M, %S, %G, %R, %AI, %AQ) to the PLC.

The Write System Memory Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 26 (Write System Memory function number)
BL = segment selector (see table)
DX = segment offset
CX = data length
ES:DI = pointer to the source buffer
```

The source buffer must be allocated and set to the values to be written by the application software prior to invoking the Write System Memory function call.

### Example of Write SMEM: %Q in Bit Mode

This example writes to 23 discrete outputs %Q19 through %Q41 inclusive, using the %Q bit mode segment selector (BL = 48h) a data offset (DX = 12 00) and data length (CX = 17 00) whose units are bit.

The example assumes that the master has already attached to the slave device, set the privilege level to two (2), and that the current value of the discrete outputs %Q17 through %Q48 is zero. It should be noted that those bits **not** included in the specified range remain as they were, regardless of what value the master sends. When the write request is complete, %Q17 through %Q48 should look as follows:

Q32														Q17	
0	0	1	1	0	1	0	0	1	1	1	0	0	0	0	0

Q48														Q33	
0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1

## Return PLC Time/Date

The Return PLC Time/Date Request returns the current time and date stored in the PLC CPU.

The Return PLC Time/Date Request is issued by setting up the following registers and invoking the driver interrupt routine.

**AX = 27 (Return PLC Time/Date function number)**  
**ES:DI = pointer to the destination buffer**

The destination buffer must be allocated by the application software prior to invoking the driver interrupt routine. The destination buffer will be eight bytes in length and is defined as follows.

**Table 3-6. PLC Time/Date Destination Buffer**

Item	Byte	Description
Seconds	Byte 1	Current seconds in packed BCD format.
Minutes	Byte 2	Current minutes in packed BCD format.
Hour	Byte 3	Current hour in packed BCD format.
Day	Byte 4	Current day of month in packed BCD format.
Month	Byte 5	Current Month in packed BCD format.
Year	Byte 6	Current year in packed BCD format.
Day of Week	Byte 7	Valid range 1 - 7 where Sunday = 1, ... Saturday = 7.
Spare	Byte 8	Reserved - Don't Care.

Note: Packed BCD format: Tens digit in bits 4 thru 7, units digit in bits 0 thru 3.

In the following example the current time in the PLC CPU is 10:48:59, and the date is Friday, May 4, 1990. The destination buffer would contain the following values in BCD format.

59 48 10 04 05 90 06 xx

where 'xx' is the spare, don't care byte

The Return PLC Time/Date function call should return success (AX = 0) as long as a communication session has been established and no communication errors (i.e., parity error, overrun error, framing error) have been encountered.

## Set PLC Time/Date

The Set PLC Time/Date Request sets the internal time and date stored in the PLC CPU.

The Set PLC Time/Date Request is issued by setting up the following registers and invoking the driver interrupt routine.

**AX = 28 (Set PLC Time/Date function number)**  
**DX = 1 (Set both time and date)**  
       = 2 (Set time only)  
       = 3 (Set date only)  
**ES:DI = pointer to the source buffer**

The source buffer must be allocated and initialized by the application software prior to invoking the SNP driver. The source buffer will be seven bytes in length and is defined as follows.



Table 3-7. PLC Time/Date Source Buffer

Item	Byte	Description
Seconds	Byte 1	Current seconds in packed BCD format.
Minutes	Byte 2	Current minutes in packed BCD format.
Hour	Byte 3	Current hour in packed BCD format.
Day	Byte 4	Current day of month in packed BCD format.
Month	Byte 5	Current Month in packed BCD format.
Year	Byte 6	Current year in packed BCD format.
Day of Week	Byte 7	Valid range 1 - 7 where Sunday = 1, ... Saturday = 7.

Note: Packed BCD format: Tens digit in bits 4 thru 7, units digit in bits 0 thru 3.

In the following example the current time and date in the PLC CPU will be set to 09:44:10 and Monday, November 05, 1990 respectively. The DX register will be set to 1 and the source buffer would contain the following values in BCD format. The extraneous bytes are disregarded when set time only or set date only DX values are specified.

10 44 09 05 11 90 02

The Set PLC Time/Date function call returns success (AX = 0) if the request successfully completes. Otherwise, AX contains an error status. The most common reason for an error status is:

- Insufficient privilege (must be level 2 or higher\*).

Note: For Series 90-30 PLCs with Release 1.x CPU firmware, privilege level 3 or higher is required.

## Return Fault Table

The Return Fault Table Request returns the fault data from either the PLC Fault Table or the I/O Fault Table stored in the PLC CPU. The PLC Fault Table has a maximum of 16 faults and the I/O Fault Table has a maximum of 32 faults.

The Return Fault Table Request is issued by setting up the following registers and invoking the driver interrupt routine.

```

AX = 29 (Return Fault Table function number)
BX = 1 (I/O Fault Table)
    = 2 (PLC Fault Table)
DX = Fault Table index
CX = number of faults to return
ES:DI = pointer to the destination buffer in memory where
        the user wishes the returned Fault Table data to be
        stored. The destination buffer must be allocated by the
        application software prior to invoking the Return Fault
        Table function call. The size of the destination buffer
        must be large enough to handle all the fault data being
        returned.

```

The first twelve (12) bytes of the destination buffer returned to the user is the fault header which has the following format.

Table 3-8. Fault Header Format

Byte	Description
Bytes 1-6	Time Stamp of last Clear

Bytes 7-8	Faults since last Clear
Bytes 9-10	Faults in the Table
Bytes 11-12	Fault Entries in this Response

The remaining bytes in the destination buffer is the actual data for each fault requested. Each fault takes 42 bytes.

Taking into account the twelve (12) bytes of fault header,

```
Fault number 1 starts at byte 15 of the destination buffer
Fault number 2 starts at byte 57 of the destination buffer
Fault number 3 starts at byte 99 of the destination buffer
      .
      .
      .
etc.
```

## Fault Entry Format

The format of a fault table entry differs between the PLC Fault Table and the I/O Fault Table.

### PLC Fault Entry

A PLC Fault Entry has the following format.

Table 3-9. PLC Fault Entry

Byte	Description
Bytes 1-4	Spare Bytes
Byte 5	RackNumber
Byte 6	SlotNumber
Byte 7	UnitNumber
Byte 8	Spare Byte
Byte 9	Fault Group
Byte 10	Fault Action
Bytes 11-12	Fault Error Code
Bytes 13-36	Spare Bytes
Bytes 37-42	Time Error Logged

## I/O Fault Entry

An I/O Fault Entry has the following format.

**Table 3-10. I/O Fault Entry**

Byte	Description
Byte 1	Spare Byte
Bytes 2-4	Reference Address
Byte 5	Rack Number
Byte 6	Slot Number
Byte 7	I/O Bus Number
Byte 8	Bus Address
Bytes 9-10	Point Address
Byte 11	Fault Group
Byte 12	Fault Action
Byte 13	Fault Category
Byte 14	Fault Type
Byte 15	Fault Description
Bytes 16-36	Spare Bytes
Bytes 37-42	Time Error Logged

To determine the number of bytes that will be set in the destination buffer, one would take the number of fault entries requested ('N') and apply the following formula:

$$\text{total bytes returned} = 12 + (42 * N);$$

where 12 is the number of bytes in the fault header, and 42 is the number of bytes in each fault entry returned.

The Return Fault Table function call returns success (AX = 0) if the request successfully completes. Otherwise, AX contains an error status. The most common reasons for an error status is:

- Insufficient privilege (Series 90-70 PLC: must be level 1 or higher).
- Invalid input parameter in the request (Fault Table Type passed in Bx was invalid: 1 = I/O, 2 = PLC, all others invalid).

Tables 3-11 and 3-12 define the fault action and fault group fields.

## Fault Codes

Table 3-11. Fault Action Codes

Fault	Description
1	Informational
2	Diagnostic
3	Fatal

Table 3-12. Fault Group Codes

Fault Code		Description
Decimal	Hexadecimal	
1	1	Loss of or Missing Rack.
2	2	Loss of or Missing IOC.
3	3	Loss of or Missing I/O Module.
4	4	Loss of or Missing Non-I/O Module.
5	5	Addition of or Extra Rack.
6	6	Addition of or Extra IOC.
7	7	Addition of or Extra I/O Module.
8	8	Addition of or Extra Non-I/O Module.
9	9	IOC or I/O Bus Fault.
10	A	I/O Module Fault.
11	B	System Configuration Mismatch.
12	C	VME/LP System Bus Error.
13	D	PLC CPU Hardware Failure.
14	E	Module Non-Fatal Hardware Error.
15	F	IOC Software Failure.
16	10	Non-I/O Module Software Failure.
17	11	Program Block Checksum Mismatch.
18	12	Low Battery in the PLC CPU.
19	13	Constant Sweep Exceeded.
20	14	PLC Fault Table Full.
21	15	I/O Fault Table Full.
22	16	User Application Fault.
128	80	System Bus Failure.
129	81	No User's Program on Power-up.
130	82	Corrupted User's Ram Detected on Power-up.
131	83	Window Completion Failure in Constant Sweep Mode (i.e., all windows failed to receive their allotted time).
132	84	Password Access Failure.
133	85	Genius Block Address Mismatch with User Configuration Reference Address.
134	86	NULL System Configuration for RUN Mode.
135	87	PLC CPU Software Failure.
136	88	More than the allowable number of I/O Bus Controllers were found in the system.
137	89	Communication failure during a store operation by the programmer

## Clear Fault

The Clear Fault Table Request clears either the PLC Fault Table or the I/O Fault Table stored in the PLC CPU.

The Clear Fault Table Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 30 (Clear Fault Table function number)
BX = 1 (I/O Fault Table)
    = 2 (PLC Fault Table)
```

The Clear Fault Table function call returns success (AX = 0) if the request successfully completes. Otherwise, AX contains an error status. The most common reason for an error status is:

- Insufficient privilege (must be level 2 or higher).
- Not logged on as a programmer attachment (Series 90-30 PLCs only).

## Set PLC State

The Set PLC State Request sets the execution state of the PLC CPU to one of the following states: RUN/IO ENABLED, RUN/IO DISABLED, STOP/IO ENABLED or STOP/IO DISABLED.

The Set PLC State Request is issued by setting up the following registers and invoking the driver interrupt routine.

```
AX = 31 (Set PLC State function number)
BX = 0 (RUN mode I/O Enabled)
    = 1 (RUN mode I/O Disabled) Series 90-70 only
    = 2 (STOP mode I/O Disabled)
    = 6 (STOP mode I/O Enabled)
```

The Set PLC State function call returns success (AX = 0) if the request successfully completes. Otherwise, AX contains an error status. The most common reason for an error status is:

- Insufficient privilege (must be level 2 or higher).
- Not logged on as a programmer attachment (Series 90-30 PLCs only).
- Faults exist in the fault table (run mode request).
- Invalid program fault exists (run mode request).
- Invalid new state specified (New state specified in the BX register was invalid: valid values are 0, 1, 2, or 6).

## Example C Program

An example of a Microsoft C language program call for the Programmer Logon Request would be:

```
#include <dos.h>
#include "snptypes.h"
.
.
.
union REGS snpin;      /* Input arguments to SNP Driver */
union REGS snpout;     /* Output arguments from SNP Driver */
char far taskname[PROG_TSKNAME_SIZE] = "TASK_1";
.
.
.
main()
{
.
.
.
snpin.x.ax = PRG_LOGON;
snpin.x.bx = LOGIN_SERIAL;
snpin.x.cx = BLOCK_XFER_SIZE;
snpin.x.di = (int)taskname;
int86( SNP_FUNCTIONS, &snpin, &snpout );
}
```

## Error Status Codes

Tables 3-13 and 3-14 provide a summary of the Major and Minor Error Status codes that are returned if the SNP Function request is unsuccessful.

**Table 3-13. Major Error Status Codes**

Decimal	Hexadecimal	Description
01	01h	Illegal Service Request: either not defined or not supported.
02	02h	Insufficient Privilege: the minor status field contains the privilege level required for the service request.
04	04h	Protocol Sequence Error: the CPU has received a message that is out of order.
05	05h	Service Request Error: Minor status field contains the request specific error code. See table of Minor Error Status Codes.
07	07h	The PLC CPU's Service Request Queue is full. You should retry later. It is recommended that you wait a minimum of 10 milliseconds before sending another service request.
10	0Ah	SNP Driver Error: Minor status field contains the request specific error code. See table of Minor Error Status Codes.

Table 3-14. Minor Error Status Codes: Major Code = 5

Decimal	Hexadecimal	Description
-1	0FF	Service request has been aborted.
-2	0FE	No privilege for attempted operation.
-3	0FD	Unable to perform auto configuration.
-4	0FC	I/O configuration is invalid.
-5	0FB	Cannot clear I/O configuration.
-6	0FA	Cannot replace I/O module.
-7	0F9	Task address out of range.
-8	0F8	Invalid task name referenced.
-9	0F7	Required to log in to a task for service.
-10	0F6	Invalid sweep state to set.
-11	0F5	Invalid password.
-12	0F4	Invalid input parameter in request.
-13	0F3	I/O configuration mismatch.
-14	0F2	Invalid program cannot log in.
-15	0F1	Request only valid from programmer.
-16	0F0	Request only valid in stop mode.
-17	0EF	Programmer is already attached.
-18	0EE	Could not return block sizes.
-19	0ED	VMEbus error encountered.
-20	0EC	Task unable to be created.
-21	0EB	Task unable to be deleted.
-22	0EA	Not logged in to process service request.
-23	0E9	Segment selector not valid in context.
-24	0E8	No user memory is available to allocate.
-25	0E7	Configuration is not valid.
-26	0E6	CPU model number does not match.
-27	0E5	DOS file area not formatted.
-28	0E4	Segment for this selector does not exist.
-29	0E3	CPU revision number does not match.
-30	0E2	IOS could not delete configuration or bad type.
-31	0E1	No I/O configuration to read or delete.
-32	0E0	Service in process cannot login.
-33	0DF	Invalid Datagram connection address.
-34	0DE	Size of Datagram connection invalid.
-35	0DD	Unable to locate given connection ID.
-36	0DC	Unable to find connection address.
-37	0DB	Invalid segment selector in Datagram.
-38	0DA	Null pointer to data in segment selector.
-39	0D9	Transfer type invalid for this selector.
-40	0D8	Point length not allowed.
-41	0D7	Invalid Datagram type specified.
-42	0D6	Datagram connection boundary exceeded.
-43	0D5	Invalid block name specified in Datagram.
-44	0D4	Mismatch of configuration checksum.
-45	0D3	User Program Module (UPM) read or write exceeded block end.
-46	0D2	Invalid write mode parameter.
-47	0D1	Packet size or total program size does not match input.
-48	0D0	One or more PLC modules configured have unsupported revision.
-49	0CF	Specified device is not available in the system (not present).
-50	0CE	Specified device has insufficient memory to handle request.



Table 3-14. Minor Error Status Codes: Major Code = 5 - Continued

Decimal	Hexadecimal	Description
-51	0CD	Attempt was made to read a device but no data has been stored on it.
-52	0CC	Data stored on device has been corrupted and is no longer reliable.
-53	0CB	A comm or write verify error occurred during save or restore.
-54	0CA	Device is write protected.
-55	0C9	Login using non-zero buffer size required for block commands.
-56	0C8	Password(s) already enabled and cannot be forced inactive.
-57	0C7	Passwords are set to inactive and cannot be enabled or disabled.
-58	0C6	Control Program (CP) tasks exist but requestor not logged into main CP.
-59	0C5	No task-level Rack/Slot configuration to read or delete.
-60	0C4	Verify with FA Card or EEPROM failed.
-61	0C3	Text length does not match traffic type.
-62	0C2	The OEM key is NULL (inactive).
-63	0C1	Invalid block state transition.
80	50h	Problem with sending mail to the slave Service Request task. (Series 90-70 PLC CPUs only).
81	51h	Problem with getting mail from the slave Service Request task. (Series 90-70 PLC CPUs only).
85	55h	Slave SNP task timed out before receiving SRP response. (Series 90-70 PLC CPUs only).

Table 3-15. Minor Error Status Codes: Major Code =10 (0Ah)

Decimal	Hexadecimal	Description
-110	92	No SNP communication: either communication has been lost or a communication session has not been established.
-111	91	Bad SNP communication: transmission had to be aborted after sufficient retries due to communication errors (i.e., parity, overrun, or framing errors).
-112	90	Bad SNP BCC encountered: transmission had to be aborted after sufficient retries due to a bad Block Check Code.
-113	8F	Out of Sequence SNP message: SNP message type received was not the type expected.
-114	8E	Serial port configured for SNP Master driver is not open; no communication can take place.
-115	8D	Bad DOS Version: Must have DOS 2.0, or later, to support the SNP MasterDriver.

This appendix describes the serial port, converters, and cables used to connect Series 90 PLCs for Series 90 Protocol (SNP). This information is included for reference and for those users who have applications that require cable lengths different than the factory-supplied cables.

## **What this Appendix Contains**

Information in this section includes:

- Communication Interfaces
- Cable and Connector Specifications
- Serial Port Configuration
- RS-232/RS-422 Converter (Catalog No. IC690ACC900 or IC690ACC901)
- RS-422 Isolated Repeater/RS-232 Converter (Catalog No. IC655CCM590)
- Serial Cable Diagrams

Point-to-Point Connection

Multidrop Connection

Cable Termination

## Section 1: RS-422/RS-485 Interface and Cabling Information

### The RS-485 Interface

The Series 90 PLC family of products are compatible with EIA RS-422 or RS-485 specifications. RS-485 drivers and receivers are utilized to accomplish communications between several system components using multiple driver/receiver combinations on a single cable with four twisted pairs. The total cable length cannot exceed 4000 feet.

A multidrop system of a driver and 8 receivers can be configured. The maximum common mode voltage between each additional drop is the RS-485 standard of +12 Volts to -7 Volts. The driver output must be capable of " 1.5 V minimum into 60 ohms. The driver output impedance must be at least 120 K ohms in the high impedance state. The receiver input resistance is 12K ohms or greater. Receiver sensitivity is " 200 millivolt.

### Caution

**Care must be taken that common mode voltage specifications are met. Common mode conditions that exceed those specified will result in errors in transmission and/or damage to Series 90 PLC components.**

### Constructing RS-422/RS-485 Cables

When connecting the Series 90 CMM modules to a non-Series 90 device using the RS-422/RS-485 standard, the non-Series 90 device's line receiver must contain "fail safe" capability. This means that in an idle, open, or shorted line condition, the output of the line receiver chip must assume the "marking" state.

When using RS-422/RS-485, the twisted pairs should both be matched so that both transmit signals make up one twisted pair and both receive signals make up the other twisted pair.

The CMM is supplied with a 120 Ohm terminating resistor in each RS-422 receiver circuit. If the module is at either end of an RS-422 multidrop or point-to-point link, pin 25, the RD (B') signal, must be connected directly to pin 24, the internal 120 ohm terminating resistor. If the module is an intermediate drop in the multidrop link, this connection should not be made.

## Cable and Connector Specifications

The cable assembly presents one of the most common causes of communication failure. For best performance construct the cable assemblies according to the recommended connector parts and specifications.

**Table A-1. Connector/Cable Specifications**

Item	Description
Mating Connectors:	<b>Series 90 PLC:</b> Serial (RS-422) port with metric hardware Connector: 15-pin male, D-Subminiature Type, Cannon DA15S (solder pot) Hood: AMP 207470-1 connector shell Hardware Kit: AMP 207871-1 Kit includes 2 metric screws and 2 screw clips
	<b>Workmaster II:</b> Serial (RS-232) port with standard RS-232 connector Connector: 25-pin female, D-Subminiature Type, Cannon DB25S (solder pot) with DB110963-3 hood or equivalent (standard RS-232 connector)
	<b>Workmaster:</b> Serial (RS-232) port with standard RS-232 connector Connector: 9-pin female, D-Subminiature Type, Cannon DE9S (solder pot) with DE110963-1 hood or equivalent (standard RS-232 connector)
	<b>IBM-AT/XT:</b> Serial (RS-232) port with standard RS-232 connector Connector: 9-pin female, D-Subminiature Type, Cannon DE9S (solder pot) with DE110963-31 hood or equivalent (standard RS-232 connector)
	<b>RS-232/RS-422 Converter:</b> one 15-pin male, and one 25-pin male connector 15-pin male connector requires metric hardware (same connector, hood, and hardware as for Series 90 PLC listed above) 25-pin male D-Subminiature Type, Cannon DA25S (solder pot) with DB110963-3 hood or equivalent (standard RS-232 connector)
Cable:	Computer grade, 24 AWG (minimum) with overall shield Catalog Numbers: Belden 9505 Belden 9306 Belden 9832 These cables provide acceptable operation for data rates up to 19.2 Kbps as follows: RS-232: 50 feet (15 meters) maximum cable length RS-422: 4000 feet (1200 meters) maximum length. Isolation at the remote end may be used to eliminate Common Mode voltages At shorter distances under 50 feet (15 meters), almost any twisted pair or shielded twisted pair cable will work, as long as the wire pairs are connected correctly. 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 ignored, cross-talk resulting from the mismatching will affect the performance of the communication system. When routing communication cables outdoors, transient suppression devices can be used to reduce the possibility of damage due to lightning or static discharge. <i>Care should be exercised that all connected devices are grounded to a common point. Failure to do so could result in damage to the equipment.</i>

## Series 90 PLC Serial Port

The Series 90 PLC serial port is compatible with RS-422. A RS-232 to RS-422 converter is required to interface to systems that provide RS-232 compatible interfaces.

The Series 90 PLC, RS-422 serial port provides the physical connection for SNP communication. This port is a 15-pin D-type female connector located as follows:

- Series 90-70 PLC - CPU Module
- Series 90-70 PLC - Remote I/O Scanner
- Series 90-30 PLC - Power Supply

Figure A-1 shows the serial port orientation and connector layout for both PLC types. Table A-2 shows the pin numbering and signal assignment applicable to both PLCs.

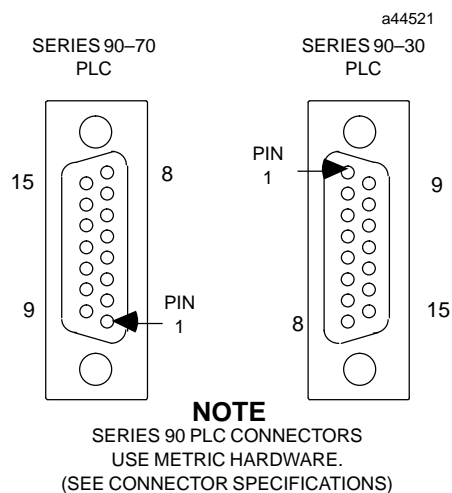


Figure A-1. Series 90 PLC, RS-422 Serial Port Connector Configuration

Table A-2. Series 90 PLC, RS-422 Serial Port Pin Assignment

Pin No.	Signal Name	Description
1	Shield	
2		NC
3		NC
4	ATCH *	Hand Held Programmer attach signal
5	+5V *	+5V Power for: Hand Held Programmer, RS-232/422 Converter
6	RTS (A)	Request To Send
7	Signal Ground	Signal Ground, OV
8	CTS (B')	Clear To Send
9	RD *	Terminating Resistor for RD (120 Ω) **
10	RD (A')	Receive Data
11	RD (B')	Receive Data
12	SD (A)	Send Data
13	SD (B)	Send Data
14	RTS (B)	Request To Send
15	CTS (A')	Clear To Send

\* Signals available at the Connector but are not included in the RS-422 specification.

SD (Send Data) and RD (Receive Data) are the same as TXD and RXD (used in the Series Six PLC).

(A) and (B) are the same as - and +. A and B denote outputs, and A' and B' denote inputs.

\*\* Termination resistance for the Receive Data (RD) signal needs to be connected only on units at the end of the lines. This termination is made on the Series 90 PLC products by connecting a jumper between pins 9 and 10 inside the 15-pin D-shell with the following exception. For Series 90-70 PLCs Cat. #'s IC697CPU731 and IC697CPU771 the termination for RD at the PLC is implemented by a jumper between pins 9 and 11.

## Workmaster Serial Port

The Workmaster II industrial computer, RS-232 serial port is a 25-pin D-type male connector, and the early model Workmaster is a 9-pin male connector.

Figure A-2 shows the serial port connector layout for both computers. Table A-3 shows the pin numbering and signal assignment for both connector types.

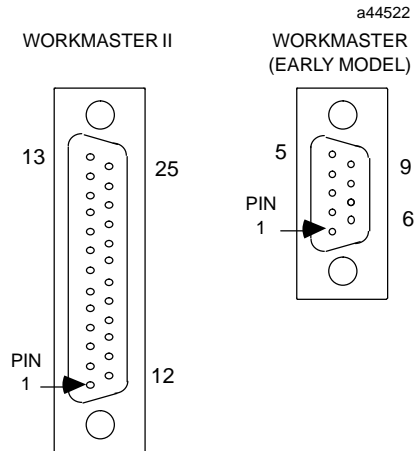


Figure A-2. Workmaster RS-232 Serial Port Connector Configuration

Table A-3. Workmaster RS-232 Serial Port Pin Assignment

Workmaster II (25-pin connector)			Workmaster (9-pin connector)		
Pin No.	Signal	Description	Pin No.	Signal	Description
1		NC	1		NC
2	TD	Transmit Data	2	TD	Transmit Data
3	RD	Receive Data	3	RD	Receive Data
4	RTS	Request to Send	4	RTS	Request to Send
5	CTS	Clear to Send	5	CTS	Clear to Send
6		NC	6		NC
7	GND	Signal Ground	7	GND	Signal Ground, 0V
8	DCD	Data Carrier Detect	8	DCD	Data Carrier Detect
9,10		NC	9	DTR	Data Terminal Ready
11		Tied to line 20			
12-19		NC			
20	DTR	Data Terminal Ready			
21		NC			
22		Ring Indicate			
23-25		NC			

NC = Not Connected

For more information about the Workmaster industrial computer serial port refer to manuals:

GFK-0401 *Workmaster II PLC Programming Unit Guide to Operation*

GEK-25373 *Workmaster Programmable Control Information Center Guide to Operation*

## IBM-AT/XT Serial Port

The IBM-AT, IBM-XT or compatible computers, RS-232 serial port is a 9-pin D-type male connector as shown in the figure below.

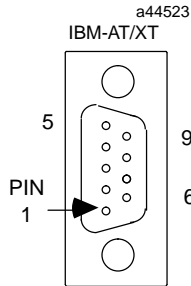


Figure A-3. IBM-AT/XT Serial Port

Table A-4. IBM-AT/XT Serial Port Pin Assignment

IBM-AT Pin No.	Signal	Description	IBM-XT Pin No.	Signal	Description
1	DCD	Data Carrier Detect	1		NC
2	RD	Receive Data	2	TD	Transmit Data
3	TD	Transmit Data	3	RD	Receive Data
4	DTR	Data Terminal Ready	4	RTS	Request to Send
5	GND	Signal Ground	5	CTS	Clear to Send
6		NC	6		NC
7	RTS	Request to Send	7	GND	Signal Ground
8	CTS	Clear to Send	8	DCD	Data Carrier Detect
9		NC	9	DTR	Data Terminal Ready

## RS-232/RS-422 Converter

The RS-232/RS-422 Converter (IC690ACC900) can be used to convert from RS-232 to RS-422 communications. The converter has one 15-pin female D-type port, and one 25-pin female D-type port.

This converter unit can be purchased from GE Intelligent Platforms. Please contact any GE Intelligent Platforms sales office or field service representative.

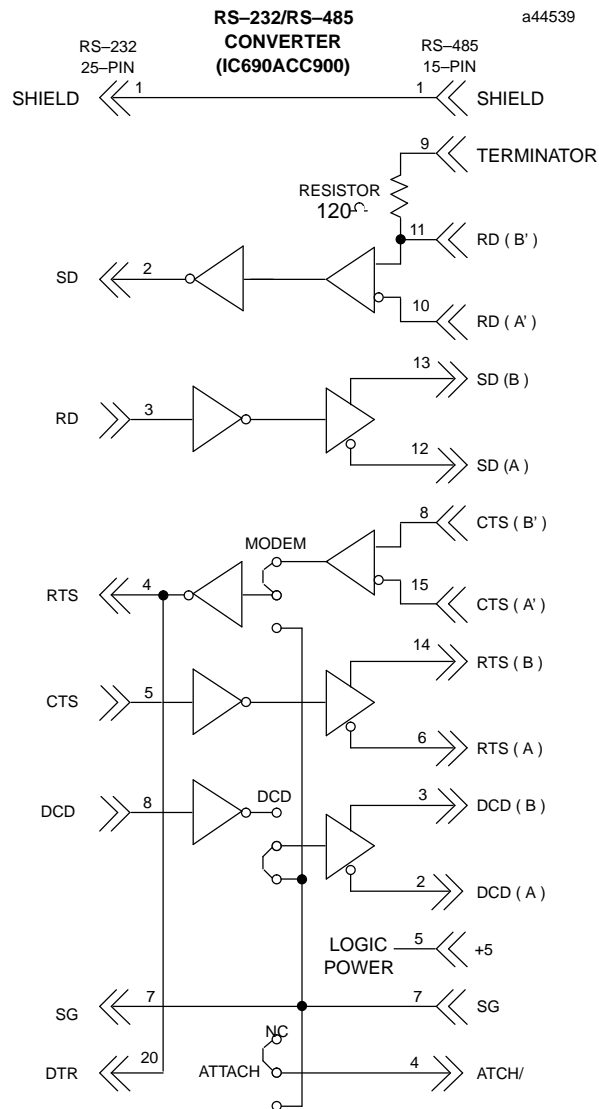
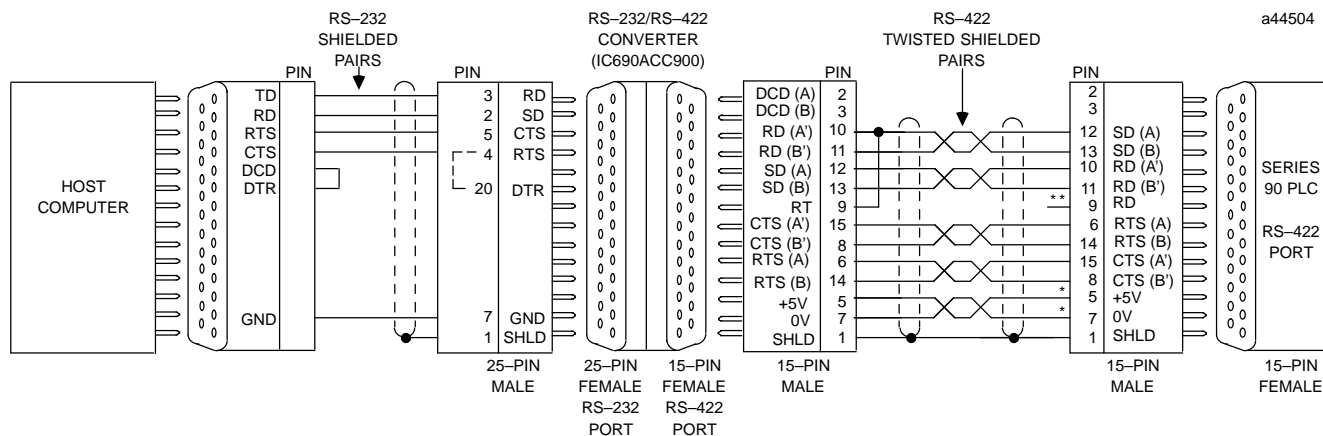


Figure A-4. RS-232 to RS-422 Converter Logic Diagram

### Note

Ground isolation is not a feature of this unit. For isolation refer to Figure A-6.





\* POWER SOURCE FOR POINT-TO-POINT CONNECTION 10 FEET (3 METERS) ONLY. CONVERTER POWER SOURCE BEYOND 10 FEET (3 METERS) AND FOR MULTIDROP CONNECTION MUST BE EXTERNAL SOURCE.

\*\* TERMINATION RESISTANCE FOR THE RECEIVE DATA (RD) SIGNAL NEEDS TO BE CONNECTED ONLY ON UNITS AT THE END OF THE LINES. THIS TERMINATION IS MADE ON THE SERIES 90 PLC PRODUCTS BY CONNECTING A JUMPER BETWEEN PIN 9 AND PIN 10 INSIDE THE 15-PIN D-SHELL WITH THE FOLLOWING EXCEPTION. FOR SERIES 90-70 PLCs, CATALOG NUMBERS IC697CPU731 AND IC697CPU771, THE TERMINATION FOR RD AT THE PLC IS IMPLEMENTED BY A JUMPER BETWEEN PIN 9 AND PIN 11.

Figure A-5. Example RS-232 to RS-422 Converter Connection

Table A-5. RS-232/RS-422 Converter Pin Assignment

RS-232 Port (25-pin connector)			RS-422 Port (15-pin connector)		
Pin No.	Signal	Description	Pin No.	Signal	Description
1	SHD	Shield	1	SHD	Shield
2	SD	Send Data	2	DCD(A)	Data Carrier Detect
3	RD	Receive Data	3	DCD(B)	Data Carrier Detect
4	RTS	Request to Send	4		NC
5	CTS	Clear to Send	5	+ 5V	Power Connection
6		NC	6	RTS(A)	Request to Send
7	GND	Signal Ground	7	0V	Ground Connection
8	DCD	NC	8	CTS(B')	Clear to Send
9-19		NC	9	RT	Terminating Resistor
20	DTR	JMP (See Figure A-4)	10	RD(A')	Receive Data
21-25		NC	11	RD(B')	Receive Data
			12	SD(A)	Send Data
			13	SD(B)	Send Data
			14	RTS(B)	Request to Send
			15	CTS(A)	Clear to Send

NC = No Connection

SD (Send Data) and RD (Receive Data) are the same as TXD and RXD (used in the Series Six PLC).

(A) and (B) are the same as – and +. A and B denote outputs, and A' and B' denote inputs.

Refer to the cable diagrams in the following pages for specific device RS-232/RS-422 converter connections.

---

## *Section 2: The Isolated Repeater/Converter (IC655CCM590)*

This section describes how to use the Isolated Repeater/Converter. The section covers the following topics:

- Description of the Isolated Repeater/Converter
- System Configurations
- Cable Diagrams

This unit can be purchased from GE Intelligent Platforms. Please contact any GE Intelligent Platforms sales office or field service representative.

### **Note**

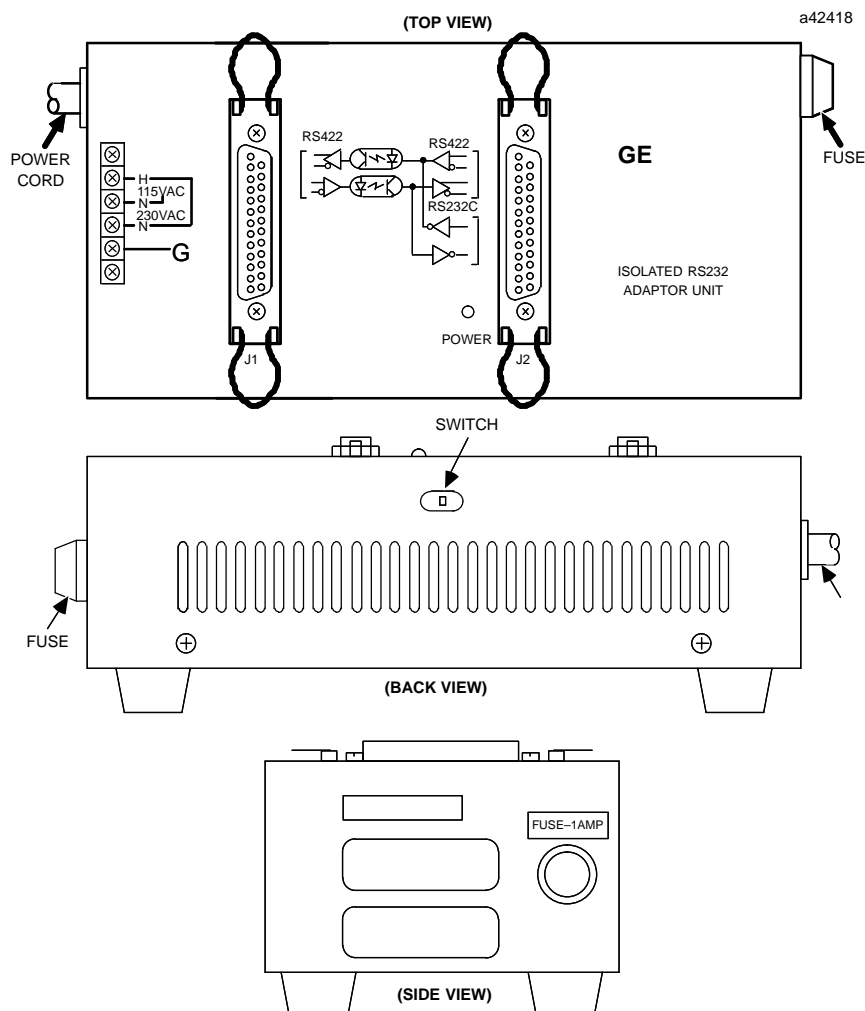
The catalog number for the Isolated Repeater/Converter was previously IC630CCM390.

## **Description of the Isolated Repeater/Converter**

The Isolated Repeater/Converter (IC655CCM590) can be used for the following purposes:

- To provide ground isolation where a common ground cannot be established between components.
- To boost RS-422 signals for greater distance and more drops.
- To convert signals from RS-232 to RS-422 or RS-422 to RS-232.

The figure below shows the appearance of the unit and the location of the user elements:



**Figure A-6. The Isolated/Repeater Converter**

The user elements of the Isolated Repeater/Converter are described below:

- Two 25-pin female D-type connectors.  
(Two 25-pin male, D-type connectors (solder pot), are included for user cabling.)
- 115/230Vac power connection (internal) 4-position terminal block.
- Fused 1 Amp power protection.
- Power ON (green) indicator LED.
- Three-position toggle switch, recessed in the back of the unit, is set according to "System Configurations" later in this section.

## Logic Diagram of the Isolated Repeater/Converter

The figure below provides a functional look at the unit. Note the 3-position switch for controlling the J1 port transmitters. This switch will be discussed in “System Configurations” later in this section.

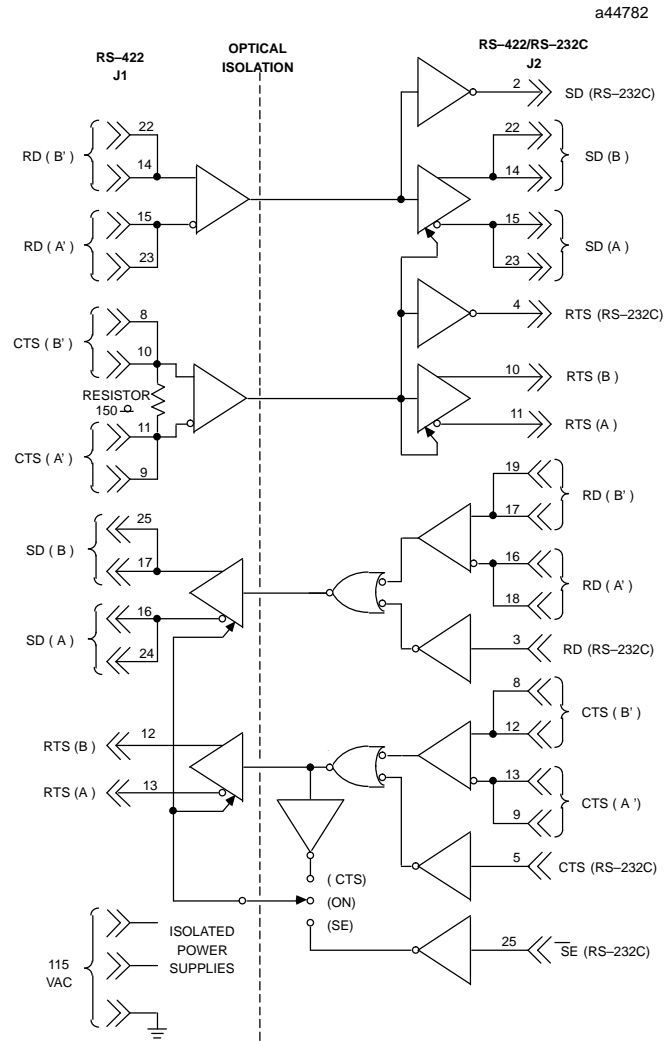


Figure A-7. RS-422 Isolated Repeater/RS-232 Converter Logic Diagram

### Note

All inputs on the unit are biased to the inactive state. Inputs left unconnected will produce a binary 1 (OFF) state on the corresponding output.

## Pin Assignments for the Isolated Repeater/Converter

J1 RS-422 Port (25-pin female connector)			J2 RS-422/RS-232 Port (25-pin female connector)		
1		NC	1		NC
2		NC	2	SD	Send Data (RS-232)
3		NC	3	RD	Receive Data (RS-232)
4		NC	4	RTS	Request to Send (RS-232)
5		NC	5	CTS	Clear to Send (RS-232)
6		NC	6		NC
7	0V	Ground Connection	7	0V	Ground Connection
8	CTS(B')	Clear to Send (Optional Termination)	8	CTS(B')	Clear to Send (Optional Termination)
9	CTS(A)	Clear to Send (Optional Termination)	9	CTS(A)	Clear to Send (Optional Termination)
10	CTS(B')	Clear to Send	10	RTS(B)	Request to Send
11	CTS(A)	Clear to Send	11	RTS(A)	Request to Send
12	RTS(B)	Request to Send	12	CTS(B')	Clear to Send
13	RTS(A)	Request to Send	13	CTS(A)	Clear to Send
14	RD(B')	Receive Data	14	SD(B)	Send Data
15	RD(A')	Receive Data	15	SD(A)	Send Data
16	SD(A)	Send Data	16	RD(A')	Receive Data
17	SD(B)	Send Data	17	RD(B')	Receive Data
18		NC	18	RD(A')	Receive Data (Optional Termination)
19		NC	19	RD(B')	Receive Data (Optional Termination)
20		NC	20		NC
21		NC	21		NC
22	RD(B')	Receive Data (Optional Termination)	22	SD(B)	Send Data
23	RD(A')	Receive Data (Optional Termination)	23	SD(A)	Send Data
24	SD(A)	Send Data	24		NC
25	SD(B)	Send Data	25	SE	Enable(RS-232C) Not Used

NC=No Connection

SD (Send Data) and RD (Receive Data) are the same as TXD and RXD (used in the Series Six™ PLC).

(A) and (B) are the same as – and +. A and B denote outputs, and A' and B' denote inputs.

**Caution**

The signal ground connections (pin 7 on each connector) must be made between the Isolated Repeater/Converter and the PLC for J1, and the Isolated Repeater/Converter and the host computer for J2.

Pin 7 of the J1 port is connected to the metal shell of the J1 connector. Pin 7 of the J2 port is connected to the metal shell of the J2 connector. These two signal ground connections are isolated from each other and are isolated from the power system ground (green wire on the terminal block). To maintain proper isolation, these signal grounds cannot be tied together.

## System Configurations

The figures below show various ways you can connect the Isolated Repeater/Converter to convert signals, expand the number of drops, and obtain greater distance. Any system configuration can be reduced to a minimum number of cables each covering a part of the overall system configuration. The example system configurations below refer to these cables as Cables A-D shown in “Cable Diagrams” later in this section.

**Downstream and Upstream Contention.** In this section, simple multidrop configurations are those where a single Isolated Repeater/Converter is used. Complex multidrop configurations contain one or more multidrop sections where an Isolated Repeater/Converter is included as one of the drops. In both simple and complex multidrop configurations, the transmitters directed downstream from the master can be on at all times. There will be no contention for the communication line because only one device (the master) transmits downstream.

In simple multidrop configurations, there will be no contention when transmitting upstream as long as devices tri-state their drivers when idle and turn them on only when they have something to transmit. This is the case for the Series 90-70 and Series 90-30 CMMs.

In complex multidrop configurations, however, special steps must be taken to switch the upstream transmitters of the Isolated Repeater/Converter.

**Switching Upstream Transmitters.** For the RS-422 drivers to be active at the J2 port of the Isolated Repeater/Converter, the RTS input at J1 must be true. The state of the RS-422 drivers at the J1 port depends on the position of the switch on the unit. When the switch is in the center position, the J1 transmitters will always be turned on. When the switch is in the CTS position, (toward the power cable), then either the RS-232 or RS-422 CTS signal must be true to turn on the J1 drivers.

**Note**

Note the position of the switch on the Isolated/Repeater Converter in the system configurations below.

## Simple Multidrop Configuration

This configuration shows how to connect a single Isolated Repeater/Converter for signal conversion or greater distance.

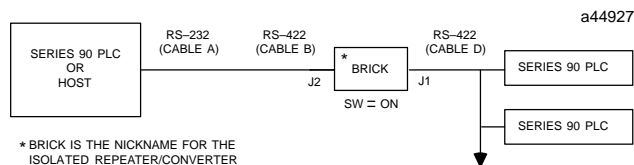


Figure A-8. Simple System Configuration Using the Isolated Repeater/Converter

## Complex Multidrop Configuration

This configuration shows how to connect multiple Isolated Repeater/Converters for signal conversion, greater distance, and more drops.

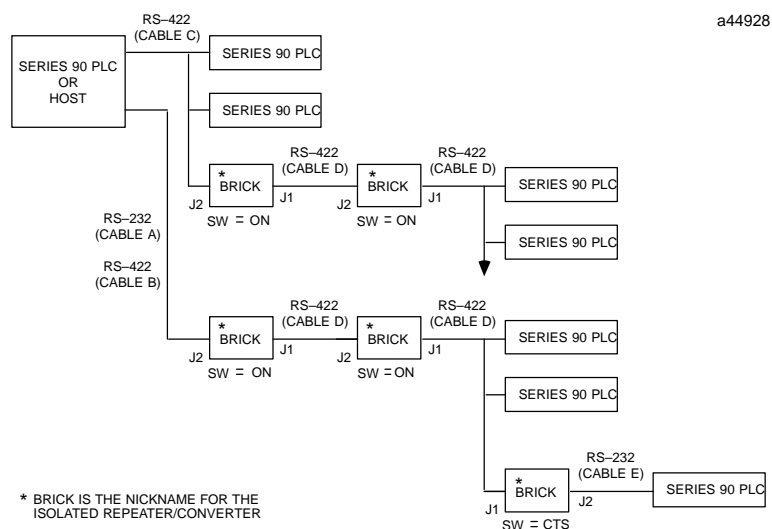


Figure A-9. Complex System Configuration Using the Isolated Repeater/Converter

## Rules for Using Repeater/Converters in Complex Networks

When designing a complex multidrop network including PLCs and RS-422 repeater/converters (bricks), the following rules apply:

**Rule 1:** When using a brick as a repeater, port J2 should always be directed toward the host device, and Port J1 should always be directed away from the host device. The switch located on the side of the brick should always be in the center position (ON). The only case in which Port J1 is directed toward the host is when the brick is used as a converter (RS-232) at the slave. The switch is in the right position (CTS).

**Rule 2:** If a Series 90 CMM slave device is located downstream of a brick, set the configuration of the CMM serial port to NONE flow control with a 10 ms Modem Turnaround Delay (Applies to CCM, SNP, and SNP-X protocols only).

**Rule 3:** Do not place more than 3 bricks in a single communication path between the host and the slave devices.

## Cable Diagrams

The cable diagrams below are referred to as Cables A-E from the system configurations in the previous figures. These diagrams show the principles for constructing your own cables and can be modified to fit your specific application.

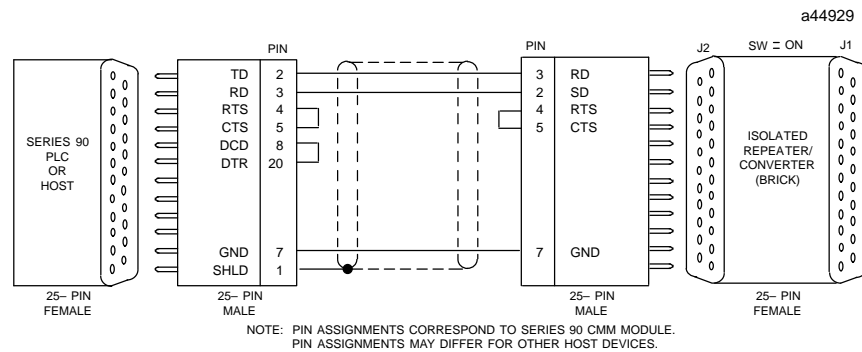


Figure A-10. Cable A; RS-232 Device To Converter

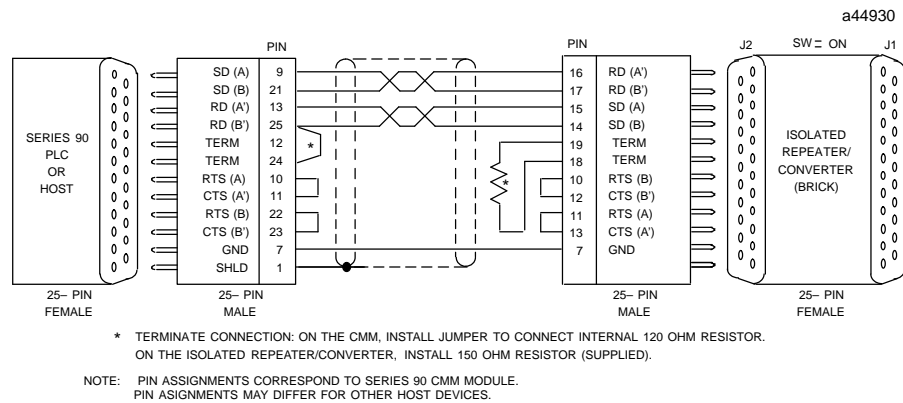


Figure A-11. Cable B; RS-422 Device to Converter



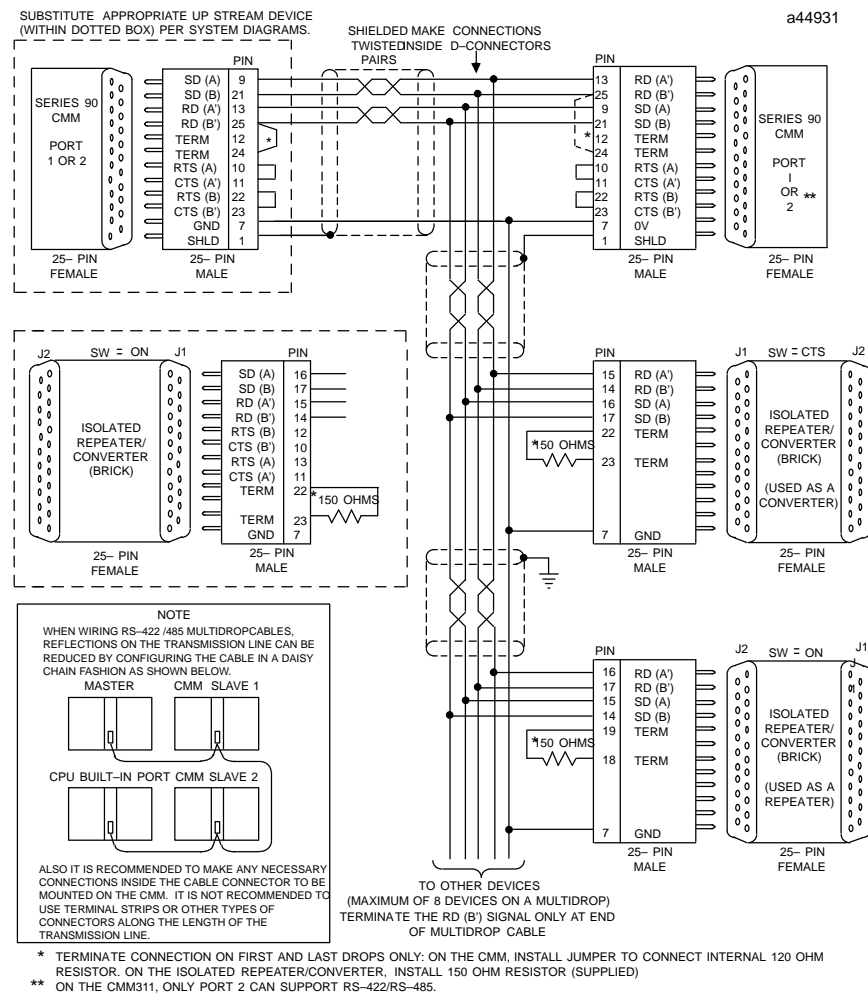


Figure A-12. Cable C; RS422 Twisted Pair

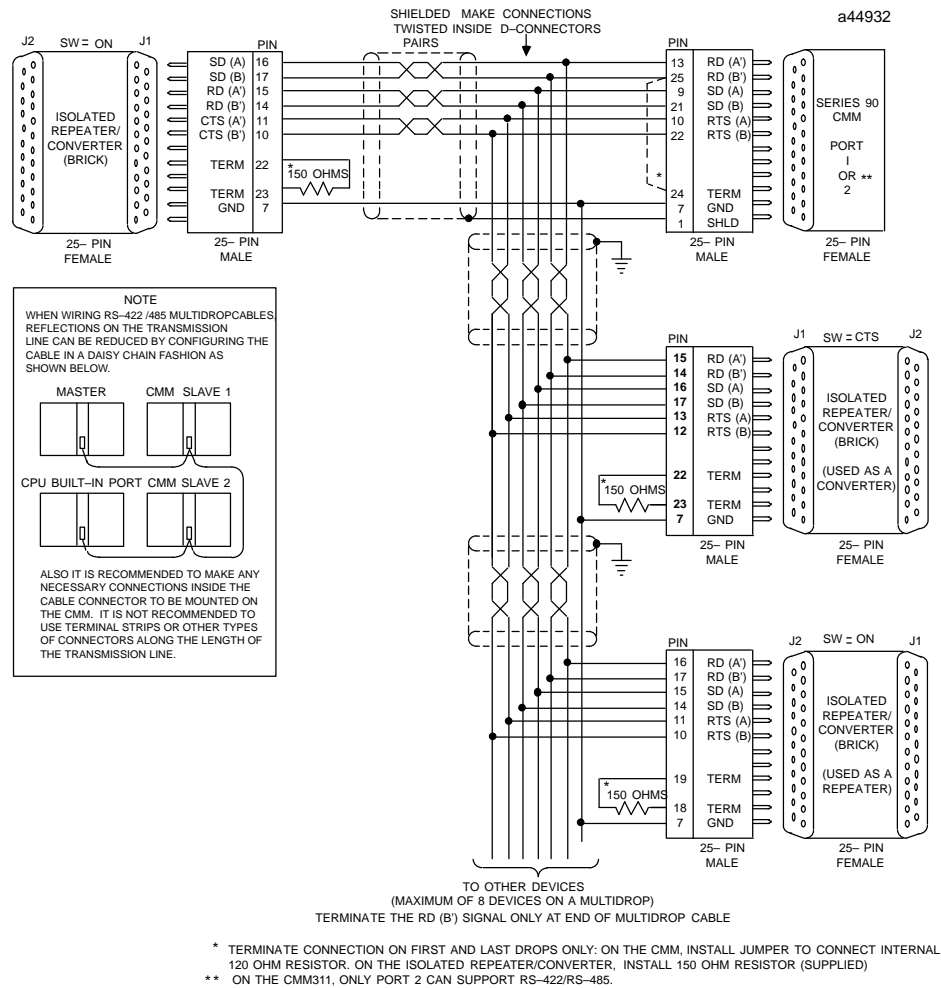


Figure A-13. Cable D; RS422 Twisted Pair

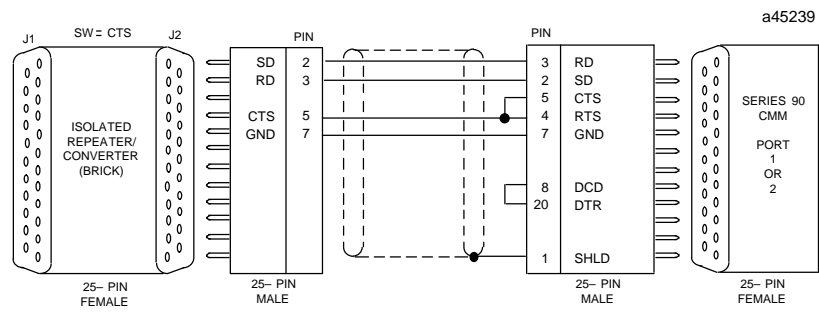


Figure A-14. Cable E; RS-232 Converter to CMM

## Serial Cable Diagrams

This section describes only a few of the many and various Point-to-Point, and Multidrop serial port connections for Series 90 PLCs.

In the point-to-point configuration only two devices can be connected to the same communication line. The communication line can be directly connected using RS-232 (50 feet, 15 meters maximum) or RS-422 (4000 feet, 1200 meters maximum). Modems can be used for longer distances.

### Note

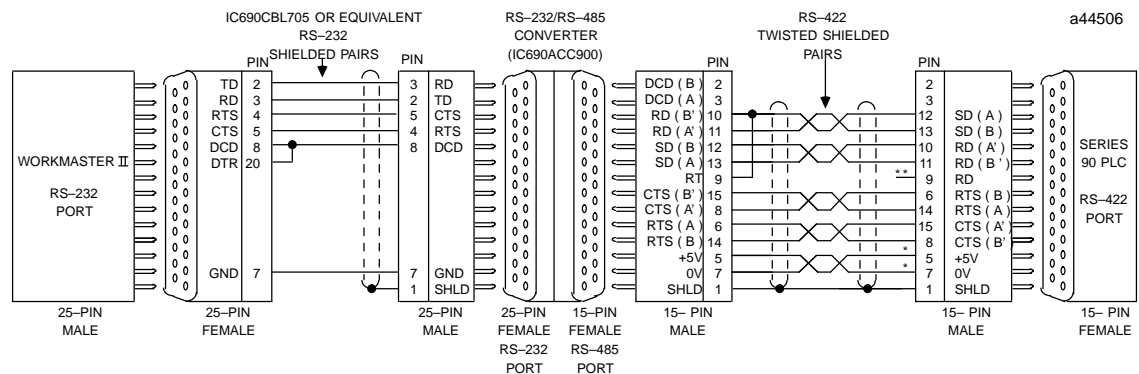
The cable connector for the Series 90-70 and Series 90-30 PLCs serial port must be a right angle connector in order for the hinged door on the module to close properly. Refer to Table A-1 Connector/Cable Specification.

### Caution

**In configurations where ground potentials may exist between components, ground isolation must be provided. Ground potential differences between non-isolated components will result in errors in transmission and/or damage to components.**

## RS-232 Point-to-Point Connections

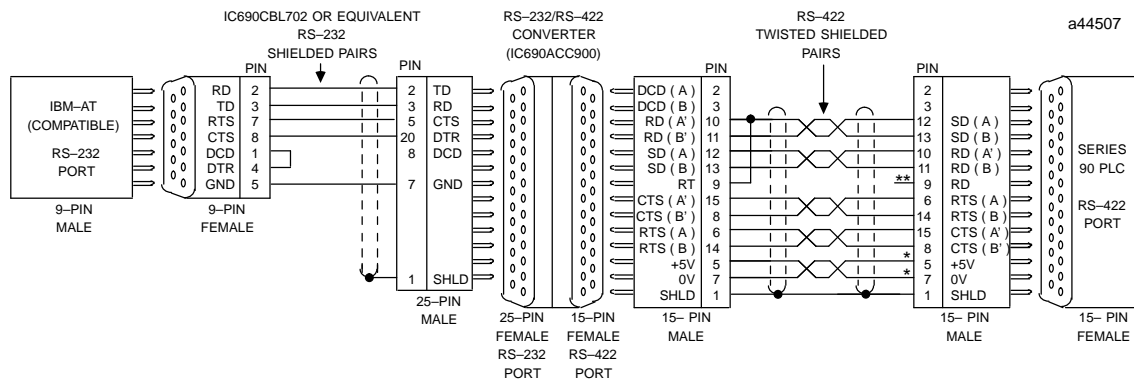
The next three figures illustrate typical RS-232 point-to-point connection to Series 90 PLCs.



\* POWER SOURCE FOR POINT-TO-POINT CONNECTION 10 FEET (3 METERS) ONLY. CONVERTER POWER SOURCE BEYOND 10 FEET (3 METERS) AND FOR MULTIDROP CONNECTION MUST BE EXTERNAL SOURCE.

\*\* TERMINATION RESISTANCE FOR THE RECEIVE DATA (RD) SIGNAL NEEDS TO BE CONNECTED ONLY ON UNITS AT THE END OF THE LINES. THIS TERMINATION IS MADE ON THE SERIES 90 PLC PRODUCTS BY CONNECTING A JUMPER BETWEEN PIN 9 AND PIN 10 INSIDE THE 15-PIN D-SHELL WITH THE FOLLOWING EXCEPTION. FOR SERIES 90-70 PLCs, CATALOG NUMBERS IC697CPU731 AND IC697CPU771, THE TERMINATION FOR RD AT THE PLC IS IMPLEMENTED BY A JUMPER BETWEEN PIN 9 AND PIN 11.

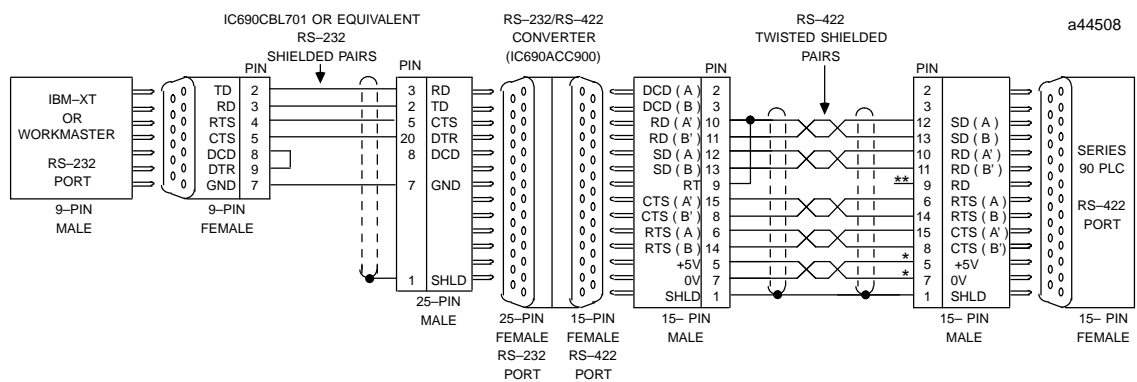
Figure A-15. Workmaster II (25-pin Serial Connection) to Series 90 PLCs



\* POWER SOURCE FOR POINT-TO-POINT CONNECTION 10 FEET (3 METERS) ONLY. CONVERTER POWER SOURCE BEYOND 10 FEET (3 METERS) AND FOR MULTIDROP CONNECTION MUST BE EXTERNAL SOURCE.

\*\* TERMINATION RESISTANCE FOR THE RECEIVE DATA (RD) SIGNAL NEEDS TO BE CONNECTED ONLY ON UNITS AT THE END OF THE LINES. THIS TERMINATION IS MADE ON THE SERIES 90 PLC PRODUCTS BY CONNECTING A JUMPER BETWEEN PIN 9 AND PIN 10 INSIDE THE 15-PIN D-SHELL WITH THE FOLLOWING EXCEPTION. FOR SERIES 90-70 PLCs, CATALOG NUMBERS IC697CPU731 AND IC697CPU771, THE TERMINATION FOR RD AT THE PLC IS IMPLEMENTED BY A JUMPER BETWEEN PIN 9 AND PIN 11.

Figure A-16. IBM-AT (compatibles) Personal Computer to Series 90 PLCs



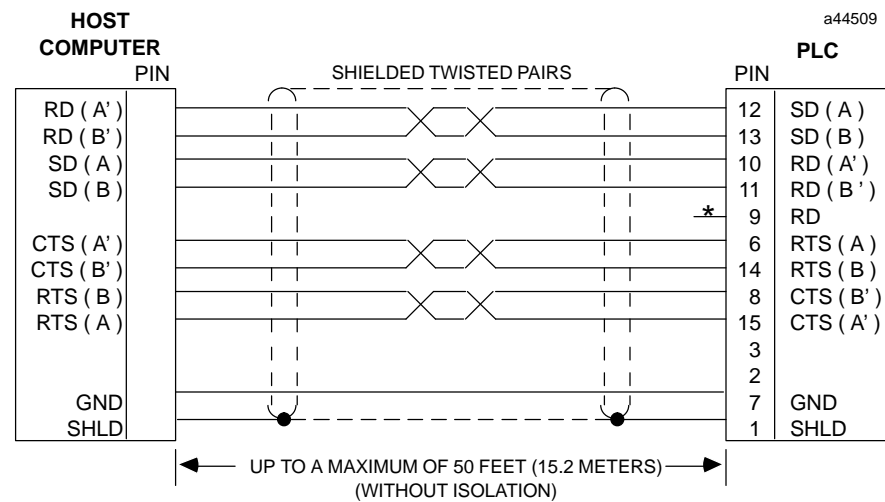
\* POWER SOURCE FOR POINT-TO-POINT CONNECTION 10 FEET (3 METERS) ONLY. CONVERTER POWER SOURCE BEYOND 10 FEET (3 METERS) AND FOR MULTIDROP CONNECTION MUST BE EXTERNAL SOURCE.

\*\* TERMINATION RESISTANCE FOR THE RECEIVE DATA (RD) SIGNAL NEEDS TO BE CONNECTED ONLY ON UNITS AT THE END OF THE LINES. THIS TERMINATION IS MADE ON THE SERIES 90 PLC PRODUCTS BY CONNECTING A JUMPER BETWEEN PIN 9 AND PIN 10 INSIDE THE 15-PIN D-SHELL WITH THE FOLLOWING EXCEPTION. FOR SERIES 90-70 PLCs, CATALOG NUMBERS IC697CPU731 AND IC697CPU771, THE TERMINATION FOR RD AT THE PLC IS IMPLEMENTED BY A JUMPER BETWEEN PIN 9 AND PIN 11.

Figure A-17. Workmaster or IBM-XT (compatibles) Personal Computer to Series 90 PLCs

RS-422 Point-to-Point Connection

If your host device is equipped with a RS-422 card you can connect directly to Series 90 PLCs as illustrated in Figure A-11.



\* TERMINATION RESISTANCE FOR THE RECEIVE DATA (RD) SIGNAL NEEDS TO BE CONNECTED ONLY ON UNITS AT THE END OF THE LINES. THIS TERMINATION IS MADE ON THE SERIES 90 PLC PRODUCTS BY CONNECTING A JUMPER BETWEEN PIN 9 AND PIN 10 INSIDE THE 15-PIN D-SHELL WITH THE FOLLOWING EXCEPTION. FOR SERIES 90-70 PLCs, CATALOG NUMBERS IC697CPU731 AND IC697CPU771, THE TERMINATION FOR RD AT THE PLC IS IMPLEMENTED BY A JUMPER BETWEEN PIN 9 AND PIN 11.

Figure A-18. Typical RS-422, Host to PLC Connection, with Handshaking

# Appendix *B*

## *Diskette Files*

---

---

This appendix contains a listing of the files, and a description of each of the files, that are supplied on the SNP Master Driver diskettes.

### SNP Master Driver Object Diskette Contents

#### Executable Files

SNPEXE	TSR SNP Driver executable file. On execution of this file, the driver is installed as a Terminate Stay Resident program.
SNPLIB	Linkable Library file containing the SNP Driver.
DEMOTSR.EXE	Demonstration executable file. GE Intelligent Platforms has provided a small program to demonstrate the capabilities of the TSR implementation of the SNP Master Driver.
DEMOLIB.EXE	Demonstration executable file. GE Intelligent Platforms has provided a small program to demonstrate the capabilities of the Library implementation of the SNP Master Driver.

#### C Header Files

Three C header files are provided on the diskette. These header files are used in programs compiled with Microsoft C (R) compiler, but should be useable with other C compilers (a few minor changes may be necessary).

The first two header files listed are the most important and will be of the most use to the application programmer. They contain constants and structures that would otherwise have to be set up by the application programmer. The application programmer should study what these files have to offer. If the application is not programmed in C, these files can be converted to the language being used (i.e., Pascal, Fortran, Assembler, etc.).

SNPTYPES.H	Contains useful constant and structure definitions for the SNP functions. <ol style="list-style-type: none"><li>1 Contains an enumeration of the SNP functions; the mnemonics in the enumeration can be used when setting up the AX register in preparation for invoking the SNP Driver interrupt.</li><li>2. Contains a constant declaration for the SNP Driver interrupt.</li><li>3. Contains constant declarations for various parameters for use with the applicable SNP function.</li><li>4. Contains structure definitions for Time Stamps, PLC Status Word, Piggy-back Status, Controller TYPE and ID information, and Fault Tables.</li></ol>
------------	---

<b>STATUS.H</b>	Contains the Major and Minor Error Status codes that are returned if the SNP Function request did not succeed.
<b>ACCESS.H</b>	Contains a table of segment selectors along with their access type (BIT versus BYTE versus WORD). The user may or may not find this file useful. Its use is applicable to reading and writing system memory.

## C Source File

<b>EXAMPLE.C</b>	Example application source file which makes use of the SNP Master Driver. It provides simple, easy-to-understand examples of each available SNP Driver function. It also makes recommendations on proper setup of parameters, driver interrupt invocation, and error checking.
------------------	--

If you purchase the SNP Driver software source files, additional files appear on the SNP Driver diskette along with those listed in Appendix B of GFK-0585 SNP Communications Driver User's Manual.

The following is a list and description of the SNP Driver Source files.

## SNP Master Driver Source Diskette Contents

### Assembly Files:

<b>SNPINST.ASM</b>	Contains the code that installs the SNP Driver. It contains the entry point <code>_Main</code> (TSR version) and the entry point <code>_init_snp_driver</code> (Library version). We check to see if the driver has already been installed so we do not install twice, and we ensure that the DOS version is 2.0 or greater. To install the driver, the SNP Function Interrupt is hooked into, and the routine that changes the clock frequency on the 8253 counter/timer chip is called in order to achieve a finer resolution for the SNP timers. If the TSR implementation is being used, configuration data is updated, and the install routine exists as a Terminate-Stay-Resident (TSR) program.
<b>SNPSOFT.ASM</b>	Contains the entry point to the SNP Driver when the SNP Driver Interrupt occurs. On installation, the address of <code>_Ssoft_int</code> was placed in the vector table at the interrupt for the SNP Driver. All SNP Driver Function calls funnel through <code>_Ssoft_int</code> . On entry, the AX register contains the SNP Driver Function Number.
<b>SNPTIMER.ASM</b>	Contains the routine that changes the clock frequency on the 8253 counter/timer chip in order to achieve a finer resolution for the SNP timers, and a routine that restores the clock frequency to its previous value. This module also contains the routines that service the SNP timers.
<b>SNPHARD.ASM</b>	Contains the routines which interface to the serial port, including the programming of the 8250 UART and the 8259A Programmable Interrupt Controller.

### C Source File:

<b>SNP.C</b>	Contains the Master implementation of the SNP Protocol. It builds requests to the PLC in the form expected by the PLC, and handles all PLC responses. The public entry points in this module for the various SNP Driver Remote Function requests are called from the interrupt routines <code>_Ssoft_int</code> contained in the module <code>SNPSOFT.ASM</code> .
--------------	--



---

### **C Header Files:**

- SNP\_INT.H**            Contains the external declarations for the SNP Driver's public data, and the prototypes for the public routines.
- SNP\_PROT.H**          Contains constants and structure definitions for the SNP protocol.

### **Assembly Include File:**

- SNP.INC**            Contains constant declarations for default timer values, interrupt constants, serial communication constants, and Driver Function Number constants. It also contains extra declarations for the SNP Driver's public data, and prototypes for the public routines.

### **Other Files:**

- MAKEFILE.**           Contains the compilation and link switches to build the SNP DOS TSR and Library. The  
**SNPTSR.RES**          file dependency lists are also included.  
**SNPLNK.RES**

# Appendix C

## Application Notes

---

This appendix contains useful information notes for the application programmer.

### Application Note for the TSR SNP Driver

1. One of the first things the application programmer should do is verify that the SNP TSR Driver is indeed loaded. The GET VERSION SNP Function request should be used for this purpose.
  - If the SNP TSR Driver is loaded, AX returns set to REQUEST\_OK, the Major software version number is returned in the BX register and the Minor software version number is returned in the CX register.
  - If the SNP TSR Driver has not been loaded, then AX will most likely return unchanged (assuming that no other program have been loaded at the SNP TSR Driver vector).

### Application Notes for the Library SNP Driver

1. The application programmer who uses the Library implementation of the SNP Master Driver must make a call to the function:  

```
Init_snp_driver
```

in order to install the driver before invoking any SNP Driver function request.
2. Any changes to the SNP configuration data must be made via the individual configuration function requests (e.g., Change T1 timer, Change T2 timer, Change CPU ID, etc.) or via the function request Process Configuration File.
3. On exit from the application program using the library implementation on the SNP Master Driver, the program MUST invoke the De-install driver function request, in order to restore the hardware clock frequency, to un-hook from the hardware timer interrupt, and restore the vector used by the SNP Driver to its original value.

## Application Notes for Either TSR or Library Implementation

1. The application program should always invoke the OPEN PORT SNP function on entry in order to ensure that the serial port used in the SNP communication is properly set up.
2. On exit from the application program, the program should always invoke the CLOSE PORT SNP function.
3. The SNP Timer T4 is configurable. However, in non-modem situations, T4 should be set to 50 milliseconds, and with modems T4 should be set to 600 milliseconds. GE recommends the 600 milliseconds T4 time with modems based on the testing it has done with two modems: DATARACE Action 24 modem and the FASTCOMM Communications FDX 9696 modem. Other modems may require more time or they may require less time.

## Symbols

%AI, 3-12  
%AQ, 3-12  
%G, 1-8 , 3-12  
%I, 1-8 , 3-12  
%M, 1-8 , 3-12  
    Bit Mode, 3-14  
%Q, 1-8 , 3-12  
    Bit Mode, 3-15  
%R, 3-12  
%S, 1-8 , 3-12  
%SA, 1-8 , 3-12  
%SB, 1-8 , 3-12  
%SC, 1-8 , 3-12  
%T, 1-8 , 3-12

## A

Abbreviations, 1-8  
Access  
    Bit Access, 3-13  
    Byte Access, 3-13  
    Word Access, 3-13  
Acronyms, 1-8  
Application Notes, C-1

## B

Basic Terms, 1-8  
Baud Rate, 2-7

## C

C Header Files, B-1  
C Program, 3-22  
C Source File, B-2  
Cable  
    Length, A-1  
    Multidrop, A-14  
    Point-to-Point, A-19  
    Specifications, A-3

Cable Diagrams for Isolated Repeater/  
    Converter, A-15

Clear Fault, 3-21  
Code Types System Memory, 3-11  
Common Mode, A-2  
CONFIG.SNP, 2-7  
Configuration Parameters, 2-6  
Connection  
    Modem, 1-6  
    Multidrop, 1-6  
    Point-to-Point, 1-4  
    RS-232/RS-422Converter, 1-4  
Connection Definition of Terms, 1-8  
Connection Modem, 1-6  
Connection Multidrop, 1-6  
Connector Specifications, A-3  
Controller  
    ID Request, 3-10  
    Type, 3-10  
Controller Destination Buffer, 3-10  
Converter, A-9

## D

Data  
    Length, 3-13  
    Offset, 3-13  
    Read, 3-14  
    Write, 3-15  
Date, 3-16  
Default Timers, 2-5  
Definition of Terms, 1-8  
Destination Buffer, 3-11  
Diskette Files, B-1  
Driver, RS-422, A-1

## E

Electrical Standards, A-1 , A-2  
Error Status Codes, 3-23  
Establish Session, 3-7  
Example Program, 3-22  
Executable Files, B-1

## F

- Fault Entry Format, 3-18
- Fault Table
  - Fault Action, 3-20
  - Fault Group Codes, 3-20
  - I/OFault, 3-19
  - PLC Fault, 3-18
- Fault Table Clear Fault, 3-21
- Files
  - Diskette, B-1
  - Executable, B-1
  - Header, B-1
  - Source, B-2
- Files Application Notes, C-1
- Function Calls, 3-1

## G

- Get Version, 3-7

## I

- I/OFault Entry, 3-19
- Isolated Repeater/Converter, A-9
  - Cable Diagrams, A-15
  - System Configurations, A-13

## L

- License Agreement, 2-1
- Logon, Series 90 PLC, 3-9

## M

- Modem, 1-5
- Modem Turn-Around Time, 2-4
- Multidrop, 1-6
- Multidrop Configuration, A-15

## P

- PLC CPU Code Types, 3-11
- PLC Fault Entry, 3-18

- Point-to-Point, 1-4
  - RS-232, A-19
  - RS-422, A-21
- Programmer Logon, 3-9

## R

- Receiver, A-1
- Repeater/Converter, A-9
  - RS-232, A-5
  - RS-422, A-2
  - RS-485, A-2

## S

- Segment Selector, 3-13
- Serial Port, 2-4
  - Cable Diagrams, A-19
  - IBM-AT/XTA-6
  - Series 90 PLC, A-4
  - Workmaster, A-5
- Serial Port Converter, A-7
- Service Requests, 3-1
- Set Date, 3-16
- Set Time, 3-16
- SMEM, 1-8
- SNP Data Size, 2-4
- SNP Function Requests
  - Login Requirements, 3-1
  - Privilege Level, 3-1
- Software
  - DOS Environment, 2-1
  - Installation, 2-2
  - License Agreement, 2-1
- Software Serial Port, 2-4
- Specifications
  - Cable, A-3
  - Connector, A-3
- Standards
  - RS-422, A-2
  - RS-485, A-2
- Status Codes, 3-23
- System Memory, 3-11

## T

- Terms, 1-8

- Time, 3-16
- Timers, 2-5
  - T1 Time, 2-5
  - T2 Time, 2-5
  - T3 Time, 2-5
  - T3p Time, 2-5
  - T4 Time, 2-5
- Timers Default Timers, 2-5