PDF ELECTRIC & SUPPLY COMPANY
A GLOBAL SUPPLIER OF AUTOMATION

GFK-0529

Buy GE Fanuc Series 90-30 NOW!

GE Fanuc Manual Series 90-30

SNP Communications

1-800-360-6802
sales@pdfsupply.com

Copyright 2013 PDFsupply.com All Rights Resevered

GE Fanuc Automation

Programmable Control Products

Series 90™ PLC
SNP Communications

User's Manual

GFK-0529C *September 1998*

Warnings, Cautions, and Notes as Used in this Publication

Warning

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

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

Caution

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

Note

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

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

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

The following are trademarks of GE Fanuc Automation North America, Inc.

Alarm Master	Field Control	Modelmaster	Series 90
CIMPLICITY	GEnet	Motion Mate	Series One
CIMPLICITY Control	Genius	PowerMotion	Series Six
CIMPLICITY PowerTRAC	Genius PowerTRAC	ProLoop	Series Three
CIMPLICITY 90–ADS	Helpmate	PROMACRO	VuMaster
CIMSTAR	Logicmaster	Series Five	Workmaster

©Copyright 1998 GE Fanuc Automation North America, Inc.
All Rights Reserved.

Content of This Manual

This manual provides an introduction to the Series 90 Protocol (SNP) and provides a very detailed description of the SNP Protocol exchange. This manual should be used by individuals implementing the master side of the SNP Protocol.

Chapter 1. **Introduction:** Provides an overview of the SNP protocol and introduces the communication services and configuration.

Chapter 2. **Configuration:** Provides an in-depth discussion of the SNP serial configuration and setup parameters.

Chapter 3. **SNP Protocol:** Describes the header, trailer, and data that make up the SNP packets, along with the SNP message types.

Chapter 4. **Integrity Checking and Error Recovery:** Describes the ACK/NAK message, error checking, sequences control and retries.

Chapter 5. **SNP Parameter Select Message:** Describes the capability of adjusting parameters in the communications link.

Chapter 6. **Service Requests:** Explains the SNP service requests and includes programming examples and explanations.

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

At GE Fanuc Automation, we strive to produce quality technical documentation. After you have used this manual, please take a few moments to complete and return the Reader's Comment Card located on the next page.

Contents

Contents

Chapter 1 Introduction

The Series 90 Protocol (SNP) is a serial protocol used in the Series 90 PLC family to communicate between a host device and the Programmable Logic Controller (PLC) via a serial port on the PLC. A protocol is a set of rules that define an orderly transmission of data. In the case of SNP, it is a set of rules that establishes and maintains a serial communications link between a Master (host device running the master implementation of the SNP protocol) and a Slave (PLC). SNP is a Master/Slave protocol where the master initiates all communications and the slave responds to the master's requests. There is no peer-to-peer capability with the SNP protocol.

The purpose of this manual is to describe the master side of the SNP protocol so that you can write a master implementation to run on a host device (e.g., a Workmaster, an IBM-PC, a Macintosh™, etc), enabling you to communicate with the PLC. SNP is a half-duplex protocol that uses the RS-485 (enhanced version of RS-422) electrical interface. Several system configurations are possible. There may be just one PLC on the serial link (direct, point-to-point port 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. SNP also supports communication over modems.

The following figures illustrate some of the possible master/slave device connections:

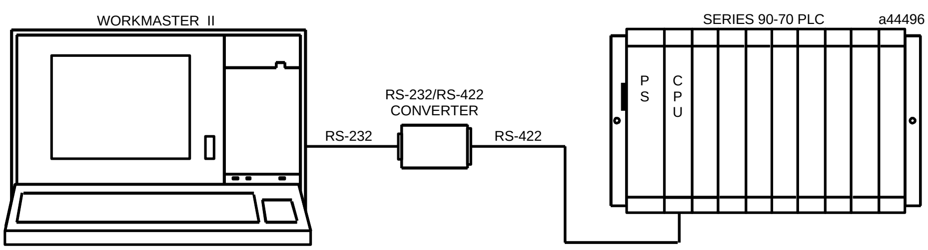


Figure 1-1. RS-232/RS-422 Point-to-Point Serial Connection to Series 90-70 PLC

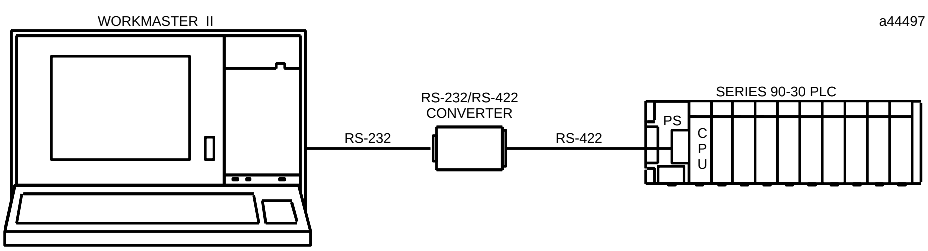


Figure 1-2. RS-232/RS-422 Point-to-Point Serial Connection to Series 90-30 PLC

RS-422 Point-to-Point Connection

If you have a host device equipped with a RS-422 interface the RS-232/RS-422 converter is not required; you can connect directly to the PLC.

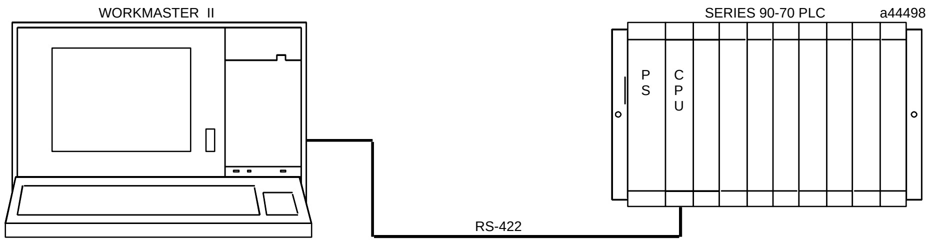


Figure 1-3. RS-422 Point-to-Point Serial Connection to Series 90-70 PLC

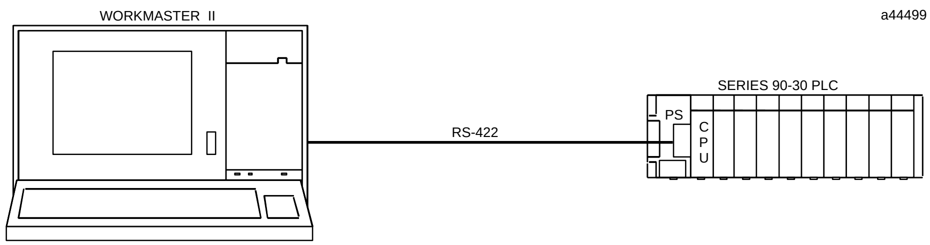


Figure 1-4. RS-422 Point-to-Point Serial Connection to Series 90-30 PLC

Series 90 PLC Modem Serial Link

The serial communications link between devices may be through a modem for longer distances.

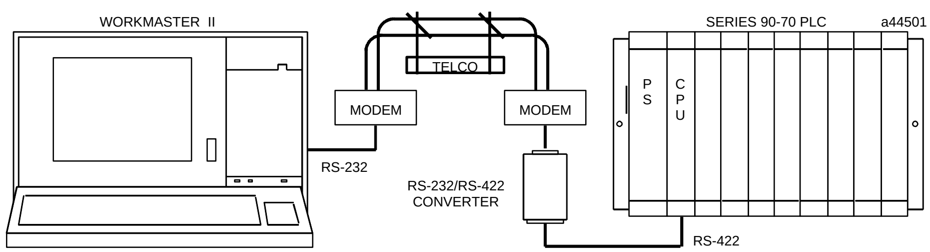


Figure 1-5. Point-to-Point Modem Serial Link Connection for Series 90 PLCs

Multidrop Series 90 PLC Serial Connection

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.

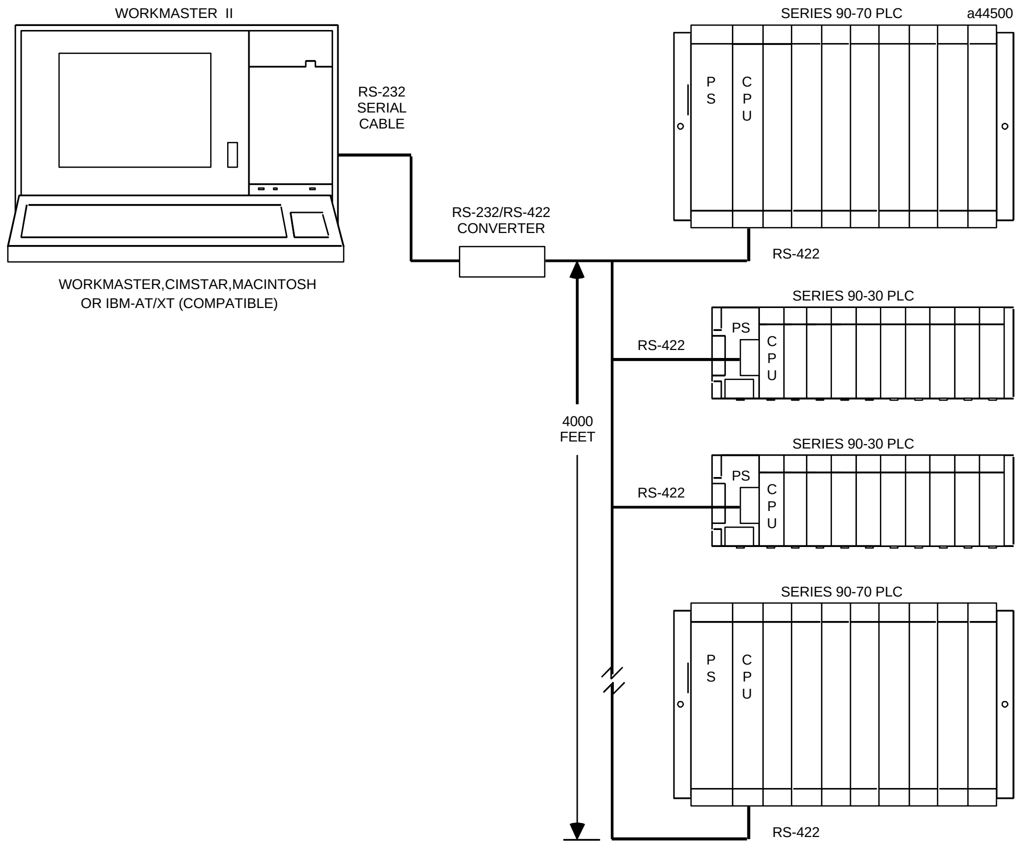


Figure 1-6. Multidrop Serial Link Connection for Series 90 PLCs

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 Fanuc 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 illustrates a full multidrop system using RS-422 isolated repeaters.

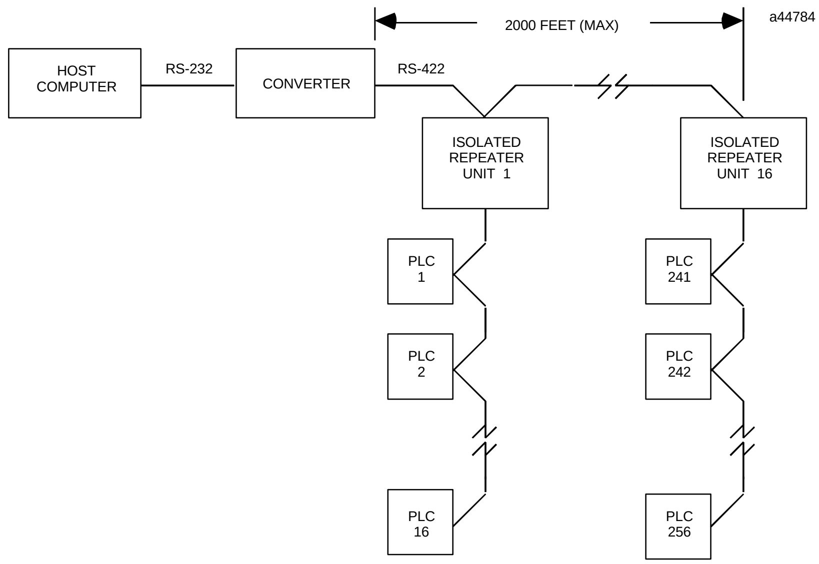


Figure 1-7. Multidrop System Using RS-422 Isolated Repeater/RS-232 Converter

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

Using this Manual

This chapter provides an introduction to the SNP serial protocol, touching briefly on the main topics that are discussed in later chapters.

The main topics of general discussion in this chapter include:

- Communications Architecture
- SNP Service Requests
- GE Fanuc Master SNP Driver
- Introduction to SNP Protocol
- Definition of Terms
- Documentation Conventions

Before you use this manual you should become familiar with the special terms, acronyms, and abbreviations that are used throughout. Also, you should become familiar with the common documentation notations.

For an explanation of these terms and conventions, refer to the last portion of this chapter: page 1-12 for a Definition of Terms, and page 1-13 for the Documentation Conventions.

Communications Architecture

It is useful to think of the SNP environment as divided into three parts: a Configuration Layer, a communication Session Layer, and an Application Layer.

Configuration Layer

The Configuration Layer consists of serial setup data (baud rate, parity, etc) and protocol timers. This layer is described in detail in Chapter 2.

Session Layer

The communication Session Layer deals with the format of SNP messages, the SNP message types, and establishing and maintaining a communication session between the master device and a Series 90 PLC. This layer is described in detail in Chapters 3, 4, and 5.

Application Layer

The Application Layer consists of the requests, called Service Requests, which the master issues to the PLC to read or write data within the PLC; load program logic from the PLC and store program logic to the PLC, or issue commands to the PLC such as starting or stopping the PLC. The application layer is described in detail in Chapter 6.

The remaining sections in this chapter briefly describe the Service Requests available to the SNP user and the SNP protocol.

Service Requests Available to the SNP User

Once a SNP serial communication session is established between the master and a slave via a defined attach sequence, a variety of requests (called service requests) can be initiated by the master to read or write data within the PLC, load program logic from the PLC and store program logic to the PLC, or issue commands to the PLC such as starting or stopping the PLC.

All the service requests have a minimum privilege level which the master must be at in order to make the request. In addition to the privilege level requirement, some service requests, such as those involved with loading and storing logic programs, require the master to be “logged on as a programmer”.

The word “logon” (or programmer logon) should not be confused with attach: the master must always be attached to a PLC before you can issue any service request; you may or may not be logged on as a programmer depending on which service requests you want to invoke. The logon request is also useful for associating a control program task name with the master SNP device. A table containing the privilege level and logon requirements for each service request available to the SNP user can be found in Chapter 6.

The service requests available to the SNP user are listed here, followed by a brief description for each. Each of the service requests are described in detail in Chapter 6.

Change Privilege Level
Read and Write PLC Memory
PLC Short Status
Return Control Program Name
Return Controller TYPE and ID
Return PLC Time/Date
Return Fault Table
Clear Fault Table
Programmer Logon
Set Control ID
Set PLC Time/Date
Set PLC State
Toggle Force System Memory
Storing and Loading Logic Programs
Datagrams

Change Privilege Level

When the master establishes a communication session with a PLC, the PLC assigns a default privilege level to the master. There are five privilege levels in Series 90-70 PLCs (0 through 4), and four privilege levels in Series 90-30 PLCs (1 through 4). The default privilege level assigned to the master is: 0 for Series 90-70 PLCs, and 1 for Series 90-30 PLCs.

Each service request and memory type within the PLC has an access privilege level associated with it. If a service request is issued by the master, and the master does not have the proper privilege level access for the request, the PLC rejects the request. The Change Privilege Level service request enables the master to change its access privilege level to the PLC CPU (provided that you know the password(s) of the level(s) you are changing).

Read and Write PLC Memory

The master can read and write all reference data (%I, %Q, %T, %M, %S, %G, %R, %AI, %AQ, %P, and %L) via the Read/Write System Memory service requests and the Read/Write Program Block (Series 90-70 PLCs only) service requests. A single contiguous block of data can be accessed from one memory type via these Read/Write service requests.

PLC Short Status

The Short Status request allows the master to retrieve information on control program tasks, which control program tasks have programmers currently logged into them, and the state of the programmer window (i.e., open or closed). This last piece of information is applicable to Series 90-70 PLCs only, since the programmer window on Series 90-30 PLCs cannot be closed.)

Return Control Program Name

This request returns the number of control program tasks in the PLC CPU, and the name of each program task. There is always one control program in Series 90-30 PLCs (it may be just the default empty program). Series 90-70 PLCs may have none or just one in today's current implementation.

Return Controller TYPE and ID

This service request returns the PLC CPU controller ID, the Major type and Minor type of the PLC CPU, the number of control program tasks, and the main control program task name. The control program task name is used in the programmer logon service request (either Null string or 8-byte NUL-terminated ASCII string).

Return PLC Time/Date

This service request returns the current time and date stored in the PLC CPU to the master.

Return Fault Table

The Return Fault Table service request returns fault data from either the PLC fault table or the I/O fault table to the master. 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

This service request clears all fault table information in either the PLC fault table or I/O fault table.

Programmer Logon

This service request allows the master to login and logout as a programmer attachment. Some service requests require that the master be logged in as a programmer attachment, such as Set Control ID.

Set Control ID (CPU ID)

This service request allows the master to specify a PLC CPU controller ID for a given PLC CPU. The following rules apply to valid controller IDs:

Series 90-70 PLC:	Maximum of seven ASCII characters followed by a NUL.
Series 90-30 PLC:	Maximum of six ASCII characters followed by a NUL; furthermore, the valid characters are restricted to the ASCII characters '0' through '9', and 'A' through 'F' inclusive.

Set PLC Time/Date

This service request allows the master to set the internal time and date of the PLC CPU.

Set PLC State

The Set PLC State service request allows the master to change the execution state of the PLC. The 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

Toggle Force System Memory

This service request allows the master to change the bit state of status, override, and transition bit memory to their opposite state.

Storing and Loading Logic Programs To and From the PLC

Logic programs on Series 90-30 PLCs consist of logic blocks and declaration blocks; on Series 90-70 PLCs, logic programs consist of logic blocks, declaration blocks, and data blocks. There are Program Load and Program Store service requests on Series 90-30 PLCs, and Program Block Load and Program Block Store requests on Series 90-70 PLCs.

Datagrams

The Datagram service requests enable the master to obtain reference data values for one or more PLC memory types via a single service request. An area of memory which will reside within the PLC CPU called a Datagram Connection Area must first be established (Establish Datagram request) and defined (Write Datagram request). Once this is done, the values of the reference data from different memory types defined in the Datagram connection area may be accessed via one service request (Update Datagram request). Multiple Datagram connections may be created, where the maximum number is limited by the amount of memory available within the PLC CPU to store the Datagram information.

Brief Introduction to the SNP Protocol

Note

The SNP Protocol has been revised to support Break-Free SNP. Break independent operation improves serial communications using modems, as certain modems alter the timing of the break, or interpret it as a modem command. Firmware for Series 90 CPUs is being updated to incorporate Break-Free SNP. If your SNP master implementation is based on break-free operation, verify that all SNP slaves have been updated to support Break-Free SNP.

The master transmits a long break as the first step in establishing a communication session with a given PLC*. A communication session can only be established with one PLC at a time. The long break generates a hardware interrupt on all the CPUs on the serial link. In response to the long break, all CPUs initialize and set up their communication hardware to wait for an Attach message from the master*. The Attach message specifies which CPU the master wishes to establish the communication session with. Only that CPU will respond to the master with the Attach Response message, and all other CPUs on the serial link go back to an idle state.

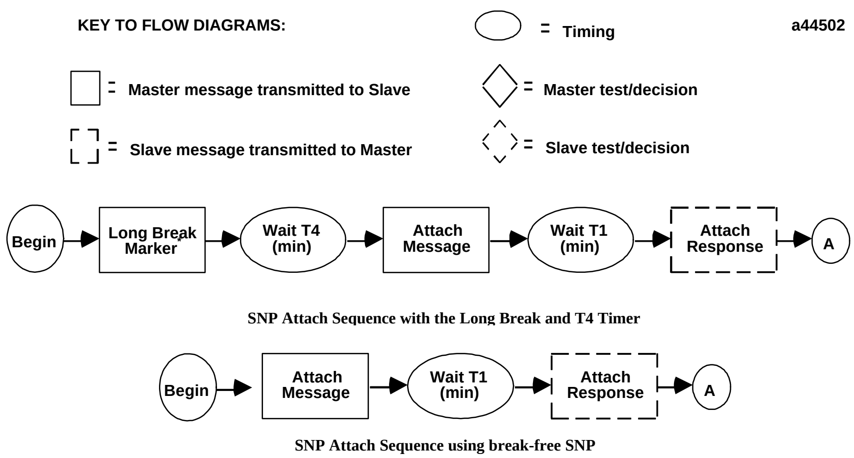


Figure 1-8. SNP Attach Sequence with a Series 90 PLC

Once a communication session is established between the master and a PLC, the master can issue service requests to the PLC via Mailbox messages. Mailbox messages contain a field which specifies a service request code that identifies the desired request, along with any necessary parameter fields for that service request. Both the master and slave do error checking on all incoming messages. Each message received by a given side must be ACKed or NAKed (transmission errors detected). Continuing with the flow diagram above, we have the following:

* The long break is not required for Break-Free SNP operation

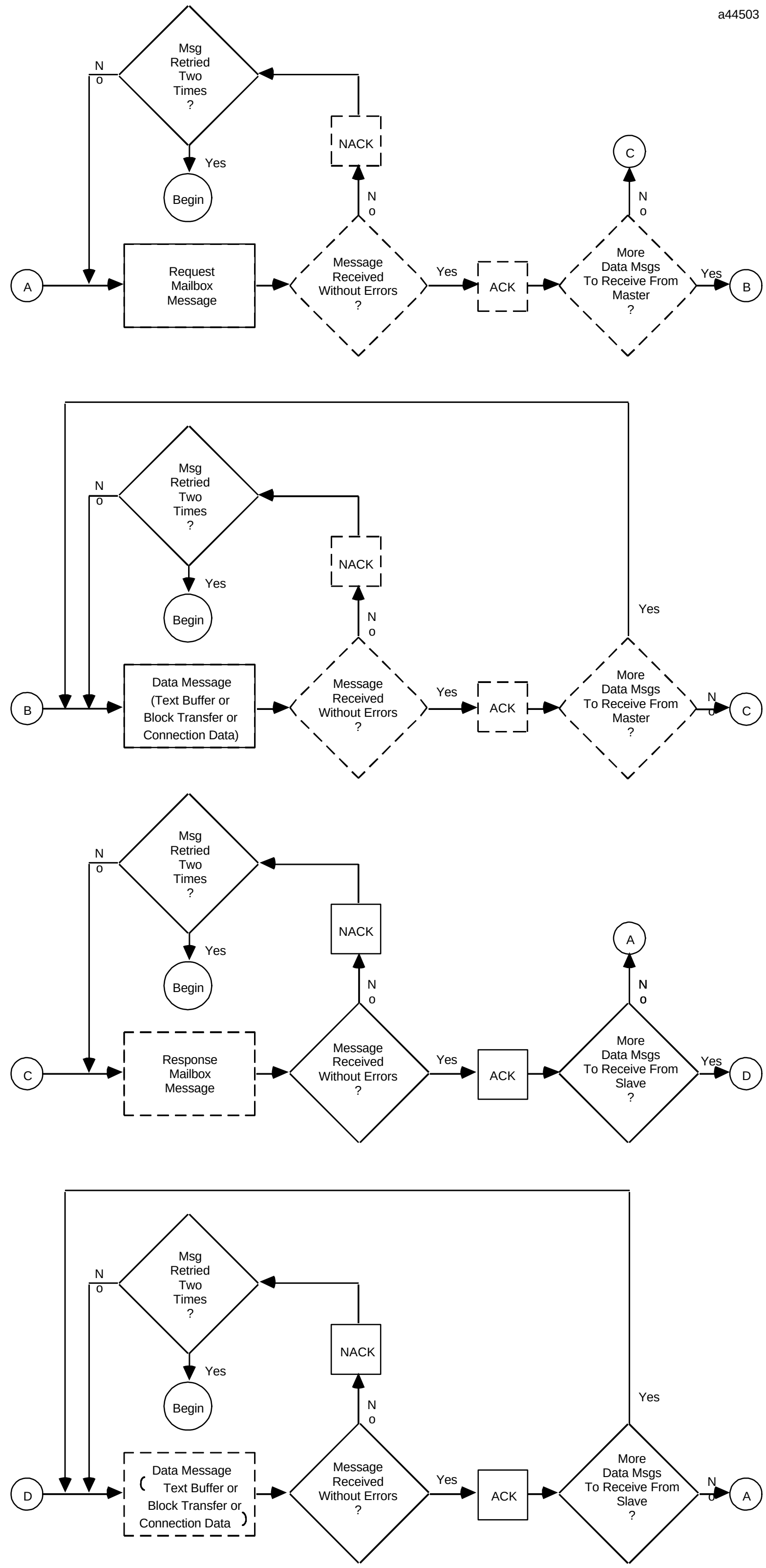


Figure 1-9. SNP Service Request to the PLC via Mailbox Message

In order to give you a quick idea of the protocol, a brief exchange of the messages is shown here for two cases: one that reads three words of register data, followed by one that writes three words of register data. The details of the messages within each exchange is explained in other sections within this document.

Read Register Data

Read three words of Register data:

MASTER	SLAVE
Long Break _*	
Attach Message_	
	_Attach Response Message
Read System Memory MB_	
	_ACK
	_Read System Memory Response MB
ACK _	

Write Register Data

Write three words of Register data:

MASTER	SLAVE
Long Break_*	
Attach Message_	
	_Attach Response Message
Change Privilege Level MB_ (Change to Level 2 in order to write system memory)	
	_ACK
	_Change Privilege Level Response MB
ACK_	
Write System Memory MB_	
	_ACK
	_Write System Memory Response MB
ACK_	

* The long break is not required for Break-Free SNP operation

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	American Standard Code for Information Interchange. An eight-bit (7 bits plus 1 parity bit) code used for data.
ACK	The ASCII ACKnowledgment character (06h - hexadecimal).
BCC	Block Check Code.
BCD	Binary Coded Decimal.
CPU	Central Processing Unit; usage in this document refers to a Series 90 family PLC CPU.
CRC	Cyclic Redundancy Check.
ESC	The ASCII ESCape character (1bh - hexadecimal).
ETB	The ASCII End of Block character (17h - hexadecimal).
I/O	Input/Output
K	1024
LM90	LogicMaster 90: GE Fanuc 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.
MB	Mailbox message.
MS	Millisecond.
NAK	The ASCII Negative ACKnowledgment character (15h - hexadecimal).
NUL	The ASCII NUL character (00h - hexadecimal).
PBMEM	Program Block MEMory (%L); Series 90-70 PLCs only.
PLC	Programmable Logic Controller.
SMEM	System MEMory (%I, %Q, %T, %M, %SA, %SB, %SC, %S, %G).
SNP	Series Ninety Protocol.
SNP-X	Series Ninety Protocol – Enhanced
TMEM	Task MEMory (%P); Series 90-70 PLCs only.
UART	Universal Asynchronous Receiver/Transmitter; samples communications bit-stream input data.
WSIB/WS9	Work Station Interface, revision B (WSIB) and Work Station Interface, Series 90 (WS9) boards; high speed data interfaces between PC-XT/AT (WSIB) and Micro Channel® (WS9) host machines for Series 90 Configurer/Programmer software and the Series 90 PLC family of CPUs.
xx	A notation used in the SNP service request examples and Mailbox Message descriptions within this document. This notation represents those bytes whose values are either meaningless to the master SNP user, or are dependent on the request being made.

Micro Channel® - Registered trademark of International Business Machines Corporation

Basic Terms

Break-Free SNP	An updated version of the SNP protocol providing break-free operation. When normal breaks are received, SNP/SNP-X communication is identical to previous versions of the protocol requiring the long break. However, this version of the protocol can also recognize an Attach or X-Attach message without a preceding break.
Error Nack	A Mailbox error message returned to the master whose mailbox type is the character D1h (hexadecimal).
Hexadecimal	A numbering system, having 16 as a base, represented by the digits '0' through '9', and 'A' through 'F'.
Long Break	Holding the transmit signal in the Mark State for three or more character times. (A character time is dependent on the baud rate being used.)
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 loading and storing logic programs.
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.

Conventions

1. Byte Ordering within a Word Field: each SNP message field that is defined as a word field is transmitted as the least significant byte first, followed by the most significant byte. This convention is very important and must be understood when studying the SNP protocol examples, and when implementing your master SNP Driver.

   As an example; the word field Next Message Length within the trailer of a SNP message is transmitted as least significant byte first followed by most significant byte. Therefore, the two hexadecimal bytes:

   3A 01 = 013Ah = 314, which in this case, equals a Next Message Length of 314 bytes.

2. Hexadecimal numbers in the main text of this document have a lower-case "h" appended to them:

   `53h or E45Fh`

   However, all data within the SNP protocol examples in the tables are coded as hexadecimal values, but do not have the lower case "h" appended. It will be obvious that the examples are hexadecimal.

3. Indicators used to avoid confusion between ASCII characters and the numeric codes used to represent them are:

   Individual printable ASCII characters are identified by single quotes (for example: 'A').

   Strings of printable ASCII characters are set off by double quotes (for example: "33101A").

Non-printable ASCII characters are referred to by their two or three letter ASCII names (for example: NUL, ACK, and NAK).

Chapter 2 Configuration

Series 90 Protocol (SNP) is a half-duplex protocol that uses the RS-485 (enhanced version of RS-422) interface. It supports a number of different baud rates, parity, and stop bits in order to support a broad number of applications. One PLC may be on the serial link to the programming device (point-to-point) or many different PLCs within the Series 90 PLC family can coexist on a serial link to the programming device (multi-drop). The communications link may be over varying amounts of distance or even over a telephone line. Serial setup parameters and timing parameters are configurable by the user in order that optimum SNP performance may be achieved.

Serial Setup Parameters

Serial Setup parameters within the PLC CPU configuration data include baud rate, data length, parity, stop bits, maximum link idle time, and modem turn-around time. The valid values or range of values are:

Baud Rate:	300	600	1200	2400	4800	9600	19.2K
Data Length:	8 bits						
Parity:	odd	even	none				
Stop Bits:	1	2					
Max Link Idle Time:	1 to 60 seconds, in 1 second increments						
Modem Turn Around Time:	0 to 2550 milliseconds, in 10 ms increments						

Baud Rate, Data Length, Parity, and **Stop Bits** are self explanatory.

Maximum Link Idle Time: the time a device currently waits to receive a message before deciding that the connection has been lost and returning to its initial state.

Modem Turn-Around Time: a time 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 1 (1 = 10 ms).

The factory defaults are as follows:

	Series 90-30 PLCs	**Series 90-70 PLCs**
Baud Rate:	19.2K	19.2K
Data Length:	8 bits	8 bits
Parity:	odd	odd
Stop Bits:	1	1
Max Link Idle Time:	10 seconds	5 seconds
Modem Turn Around Time:	0 ms	0 ms

If you wish to use values other than the factory defaults, you must set up the new values through some means external to SNP before the SNP protocol can be invoked. On Series 90-70 PLCs this may be accomplished via the LM90/WSIB link, or on Series 90-30 PLCs via the Hand-Held Programmer or LM90/WSIB link.

Timers

Adjustable timers are provided because different PLCs within the Series 90 PLC family can coexist on a given SNP serial link, each with its potentially different processor and different processing capability. The SNP protocol 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 five SNP 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 PLCs on the serial link (must be at least 50 ms; 600 ms with modems).

T1 - T1 is the amount of time that must elapse between the reception of the last character or transmission 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.

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.

T3 - T3 is the maximum link idle time. It is used only by the master to determine when to force a message over the link in order to prevent the slave from timing out due to lack of activity on the serial link.

T3' - T3' 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 T3' time. If one device is waiting for a message from the other device, and T3' time is exceeded before any part of the message is received (i.e. no character was received within T3' time), a loss of communication is assumed and the device goes back to its “start” state. Once any part of a message is received, T3' 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 T4 timer is not required for Break-Free SNP operation

The following table shows the defaults for all five timers for Series 90-30 PLCs, Series 90-70 PLCs, and Logicmaster 90. (LM90: master device used to communicate with the Series 90 PLCs via SNP).

Table 2-1. Series 90 PLC Timer Defaults

	Logicmaster 90	Series 90-30 PLC	Series 90-70 PLC
T1	10 ms (+ modem turnaround)	5 ms (+ modem turnaround)	5 ms (+ modem turnaround)
T2	3 seconds	1 second	500 ms
T3	10 seconds	N/A	N/A
T3'	T3 + 80 character times (based on baud rate)	10 seconds	5 seconds (currently disabled on Series 90-70 PLCs)
T4*	50 ms (+ 500 ms when modem turnaround time is not equal to zero)	N/A	N/A

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. The SNP data size does not include the message header and trailer. (see Chapter 3, entitled "SNP Protocol" for SNP message structure and SNP message types.)

You may specify the maximum amount of data allowed in SNP messages via the SNP Parameter Select message (see Chapter 5, entitled "SNP Parameter Select Message"). The default (and also the maximum allowed) for the Series 90 PLC family is:

Series 90-70 PLC: SNP data size = 8K bytes*

Series 90-30 PLC: SNP data size = 1000 bytes

You can also 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.

* 8K bytes is the maximum used for Block Transfers (transferring Program Logic Blocks); 2K bytes is the maximum SNP data size that should be used when reading and writing System, Task, and Program Block memory within the PLC CPU.

* The T4 timer is not required for Break-Free SNP operation

Chapter 3 SNP Protocol

The Series 90 Protocol (SNP) message is made up of three parts: a header, a trailer, and data between the header and trailer. The following table shows the format of SNP messages:

Table 3-1. Series 90 Protocol (SNP) Message Format

Byte Number	Size in Bytes	Description
1	1	Start of message character (ESC = 1bh).
2	1	Message type character.
3 to N + 2	N	Data bytes of the message; Length N and content depend on the message type.
N + 3	1	End of block character (17h) marks the beginning of the message trailer.
N + 4	1	Next message type.
N + 5	1	Least significant byte of the next message length.
N + 6	1	Most significant byte of the next message length.
N + 7	1	Status byte or most significant ASCII hexadecimal digit of Block Check Code in Attach/Attach Response message.
N + 8	1	Block Check Code or least significant ASCII hexadecimal digit of Block Check Code in Attach/Attach Response message.

SNP Message Header

Two bytes make up the header of the SNP message and are defined as follows:

ESC Character (1bh)

Start of message: every SNP message begins with this character.

Message Type

One byte message type whose valid values are:

41h	(A)	for Attach message.
52h	(R)	for attach Response message.
4Dh	(M)	for Mailbox message.
54h	(T)	for Text buffer message.
42h	(B)	for Block transfer message.
43h	(C)	for Connection data message.
55h	(U)	for Update real-time connect message.
49h	(I)	for Inquiry message.

SNP Message Trailer

The SNP message trailer is made up of six bytes which is defined as follows:

ETB Character (17h)

Every SNP message's trailer begins with the end of block character.

Next Message Type

This field is one byte and equals the type of message that the transmitting device will send next. When a given side (either the master or the slave) has completed a given sequence of messages, this byte must be set to zero to signal the receiver that this is the last message of the sequence and the "link should be turned around". In other words, it is now the other side's turn to transmit data.

Next Message Length

This field is a *word*, where the least significant byte is transmitted first, followed by the most significant byte. If the "Next Message Type" byte is non-zero, then this word must contain the number of bytes contained in the message that will follow this one. This value includes ALL bytes in the next message: header, trailer, and data bytes. If you are "turning the link around" (i.e., Next Message Type is zero), then you must set this word to zero.

Status Byte

This byte is currently used only in the Attach and Attach Response messages. Except for the Attach/Attach Response messages, this field is ignored and is typically set to zero by all Series 90 SNP implementations. If the message is an Attach or Attach Response, this byte will be the most significant ASCII hexadecimal digit of the BCC. The Block Check Code (see next item in trailer) for these two messages are encoded in ASCII hexadecimal and therefore take two bytes, whereas all other messages' Block Check Codes are binary values and use only one byte.

Block Check Code (BCC)

The BCC is used as an integrity check of the data received in the message. The BCC is calculated over the range of bytes as follows:

Attach/Attach Response Message:	First byte in the message up to, and including, the "length of next message" field in the trailer. The BCC is encoded in ASCII hexadecimal and follows the "length of next message" field in the trailer as most significant ASCII hexadecimal digit followed by least significant ASCII hexadecimal digit.
All other SNP message types:	First byte in the message up to, and including, the status byte field in the trailer.

The Block Check Code (BCC) is calculated as follows:

```
    ; assume:
    ;    msg_length = Number of bytes in message over which the BCC
                      is calculated.
    ;    msg_ptr    = Pointer to the first byte (start of message char) of
    ;                 the message.
VAR
  i        : integer;  snp_bcc  : byte;
BODY
  i       = 0;
  snp_bcc = 0;
  while ( i < msg_length)
    snp_bcc = snp_bcc XOR (byte contents at (msg_ptr + i))
    rotate snp_bcc left by 1; 8-bit rotate (ROTATE HIGH BIT TO LOW BIT)
    i = i+1
  end while

END BODY
```

Data Bytes

The content and number of data bytes is dependent on the message type specified in the header. Each of the SNP message types is described here with an example. Notice that all examples are coded as hexadecimal values.

Attach Message

The Attach message is a master-only message and must be the first message sent after the long break*. The master must wait T4 time after the end of the long break before sending the Attach message in order to give all PLC CPUs on the serial link time to recognize the long break, and to set up to receive the Attach message*.

The Attach message contains the CPU ID of the slave (PLC CPU) that the master wishes to establish a communication session with, and it also contains the master's T1 time. Only the slave whose ID matches that in the Attach message will respond with an Attach Response message. If the master passes a CPU ID of all NULs (point-to-point connection), there must be one and only one PLCs on the serial link, as all slaves will respond to a point-to-point Attach message. The Attach message is 24 bytes long and has the following format:

* The long break is not required for Break-Free SNP operation

Table 3-2. Attach Message (Master-Only) Format

Byte Number	Size in Bytes	Description
1	1	Start of message character. (ESC = 1bh)
2	1	Message type character. (41h for Attach)
3-10	8	PLC CPU ID in ASCII (NUL terminated).
11-14	4	The Master's SNP timer T1 value in milliseconds encoded as ASCII hexadecimal.
15	1	Reserved; set to ASCII zero (30h).
16-18	3	Reserved; set to ASCII space (20h).
19	1	End of block character (17h) marks the beginning of the message trailer.
20	1	Next message type.
21-22	2	Next message length.
23-24	2	BCC coded as ASCII hexadecimal values; most significant byte first.

Attach CPU ID Field

The CPU ID field is 8 bytes long. For 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 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).

Attach Master T1 Field

The T1 time is a word value defined in milliseconds and encoded in ASCII hexadecimal format, using four bytes within the Attach message, with the two-character codes representing the least significant byte of the word value occurring first, followed by the two-character codes representing the most significant byte. For instance, if T1 equals 10 milliseconds, the four bytes in ASCII hexadecimal are:

```
30 41 30 30  ==  "0A00"  ==  000A (hexadecimal value) ==  10 ms
```

Attach Message Example

In this example, the master wishes to connect to the PLC CPU whose ID is “33101A”, and the master's T1 time is 10 ms. The master sends the following series of bytes over the serial link:

<u>1b</u> <u>41</u> <u>33 33 31 30 31 41 00 00</u> <u>30 41 30 30</u> <u>30</u> <u>20 20 20</u> <u>17</u> <u>00</u> <u>00 00</u> <u>38 35</u>

where the bytes break down as follows:

Table 3-3. Attach Message Example

Byte Number	Value	Description
1	1b	Start of message character (ESC).
2	41	Message type character. (41h for Attach)
3-10	33 33 31 30 31 41 00 00	Eight bytes of ASCII PLC CPU ID.
11-14	30 41 30 30	Four bytes of ASCII hexadecimal encoded Master T1; in this example, the characters "0A00" encode the value 000Ah which equals 10 ms.
15	30	Reserved; set to ASCII zero (30h).
16-18	20 20 20	Reserved; set to ASCII space (20h).
19	17	End of block character (17h) marks the beginning of the message trailer.
20	00	Next Message Type: value of zero turns the link around; the master is telling the slave that it is waiting for information from the slave: namely, an Attach Response.
21-22	00 00	Next Message Length: equals zero since next Message Type byte is zero.
23-24	38 35	BCC - special two byte ASCII-encoded BCC for the Attach message; calculated over the bytes starting with the start of message character (ESC) through and including the Next Message Length.

Attach Response

The Attach Response message is a slave-only message that is sent to the master in response to the master's Attach message. The slave must wait at least T1 time before starting to transmit the Attach Response message in order to give the master time to transition from the transmitting state to the receiving state.

The Attach Response message contains the responding slave's CPU ID (ASCII null string if it doesn't have one), and a T1 time. The Attach Response message is 24 bytes long and has the following format:

Table 3-4. Attach Response (Slave-Only) Message Format

Byte Number	Size in Bytes	Description
1	1	Start of message character. (ESC = 1bh)
2	1	Message type character (52h for Attach Response).
3-10	8	PLC CPU ID in ASCII (NUL terminated).
11-14	4	SNP timer T1 value in milliseconds encoded as ASCII hexadecimal; larger of the master's T1 from the Attach message and the slave's T1.
15-18	4	Don't care: ignore.
19	1	End of block character (17h) marks beginning of message trailer.
20	1	Next message type.
21-22	2	Next message length.
23-24	2	BCC coded as ASCII hexadecimal values; most significant byte first.

Attach Response CPU ID Field

The responding PLC CPU returns it's CPU ID to the master via this field. In the case of a multi-drop connection; this field is identical to that sent in the Attach message. In the direct point-to-point connection where the master sent all NULs as the CPU ID; the return CPU ID in the Attach Response message is the CPU ID of the only PLC CPU on the serial link. (ASCII null string if the CPU has no ID.) The same rules concerning valid characters apply for the CPU ID field in the Attach Response message as in the Attach message.

Attach Response T1 Field

The responding slave compares the master T1 time sent in the Attach message to its own T1 time, and passes the larger of the two back to the master in the Attach Response message. The format of T1 in the Attach Response is the same as that in the Attach message. The master must use the T1 value passed back in the Attach Response as the T1 time for this connection session.

Attach Response Example

In this example, the PLC CPU whose ID is equal to the CPU ID sent in the Attach message example responds (CPU ID = 33101A). The PLC CPUs T1 value is less than the master's 10 millisecond T1, so it returns 10 ms. The slave sends the following series of bytes over the serial link:

1b 52 33 33 31 30 31 41 00 00 30 41 30 30 30 20 20 20 17 00 00 00 45 37

where the bytes break down as follows:

Table 3-5. Attach Response Example

Byte Number	Size in Bytes	Description
1	1b	Start of message character (ESC).
2	52	Message type byte; 52h = `R' for Attach Response.
3-10	33 33 31 30 31 41 00 00	Eight bytes of CPU ID.
11-14	30 41 30 30	Four bytes of ASCII hexadecimal encoded T1; slave PLC CPU compared the master's T1 passed in the Attach message to its own T1, and passed the greater of the two back to the master in the Attach Response message.
15	30	Don't care.
16-18	20 20 20	Don't care.
19	17	End of block character (17h) marks the beginning of the message trailer.
20	00	Next Message Type: value of zero turns the link around; the slave is now waiting for a Mailbox message from the master.
21-22	00 00	Next Message Length.
23-24	45 37	BCC - special two-byte ASCII-encoded BCC for the Attach Response message; calculated over the bytes starting with the start of character (ESC) through and including Next Message Length.

Mailbox Message

The Mailbox message is the main means through which the master issues requests to the slave and receives the slave's responses. Once the communication session is established, a Mailbox message must precede all other message types (Text Buffer messages, Block Transfer messages, and Connection Data messages, which collectively are referred to as SNP Data Messages).

All Mailbox messages have 40 bytes (8 bytes of header and trailer, 32 bytes of data) and take one of two forms, depending on whether or not the amount of data being passed requires the use of a Text Buffer. SNP utilizes five principal Mailbox message types:

Initial Request Mailbox message	C0h
Initial Request Mailbox message with Text Buffer	80h
Completion ACK Mailbox message	D4h
Completion ACK Mailbox message with Text Buffer	94h
Error Nack Mailbox message	D1h

The Program Block Load and Program Block Store service requests for Series 90-70 PLCs utilize additional mailbox types. For details, refer to the section in Chapter 6, Service Requests.

A quick description of key fields common to all five types is included here so that you will be familiar with the common fields that apply to both the Series 90-70 and Series 90-30 PLCs.

Time Stamp: Three byte field containing the current reading of the time of day in SS:MM:HH BCD format. This field is optional, and may or may not be filled in by either side. Each Series 90 PLC CPU model that has a time of day clock fills in this field in Completion ACK and Error Nack Mailbox messages returned to the master. The master SNP implementation may or may not fill this in as seen fit by the implementor.

Sequence Number: One byte field that the master fills in. Valid values range from 00 through ffh inclusive. This value is copied into the sequence number field of the reply mailbox message: either the Completion ACK or Error Nack Mailbox message.

Mailbox Type: One byte code that specifies the type: valid values used by master SNP implementations are C0h and 80h. Valid values used by slave SNP implementations are D4h, 94h, and D1h.

Mailbox Source ID and Destination ID fields: Each ID field is four bytes in length and describes the task where the Mailbox message originates (source) and where the Mailbox message is destined (destination). Quite simply, there are three possible IDs:

Device	Four-byte ID
Master SNP device	10 3a 00 00
Slave Service Request task	10 0a 00 00
Slave SNP task	10 3e 00 00

In the Initial Request Mailbox messages (both with and without Text Buffers), the Source ID field is always set to that of the master SNP device (10 3a 00 00). The Destination ID field is always set to the slave Service Request task (10 0a 00 00), except in the one special case of the SNP Parameter Select Message. The SNP Parameter Select message contains information for the slave SNP task only, and therefore has a Destination ID of slave SNP task (10 3e 00 00). Refer to Chapter 5, SNP Parameter Select Message for details.

In the Completion ACK mailbox messages (both with and without Text Buffers), the Destination ID field always becomes the Master SNP device (10 3a 00 00). The Source ID field becomes the Service Request task (10 0a 00 00), except in the previously mentioned case of the SNP Parameter Select message where the Source ID field would become that of the slave SNP task (10 3e 00 00). In other words, what was the Source ID in the Initial Request Mailbox message becomes the Destination ID in the Completion ACK Mailbox message, and what was the Destination ID in the Initial Request Mailbox message becomes the Source ID in the Completion ACK Mailbox message.

Packet Number

Total Packets: Both these fields have no real meaning within SNP messages and should always be set to one (1).

The following sections describe the format of the five Mailbox message types.

Initial Request Mailbox Messages

The master uses Initial Request Mailbox messages containing PLC service requests to send commands or data to the PLC and to request data from the PLC. If the service request parameter data is thirteen bytes or less, an Initial Request Mailbox message without a Text Buffer (Mailbox Type C0h) is sufficient. If, however, the data in the master's service request requires more than thirteen bytes, then an Initial Request Mailbox message with Text Buffer (Mailbox Type 80h) must be used, and one or more Text Buffer messages must follow the Initial Request Mailbox message. The following two tables describe the format for the two different types of Initial Request Mailbox message. The values of the bytes are listed when they are known and fixed; values are marked “xx” for values dependent on the service request being made.

Caution

Fields that are either reserved or not used in the Initial Request Mailbox messages must be set to zero. Failure on the user's part to comply with this rule may cause unexpected results in current or future revisions of the PLC CPU firmware as later features are implemented.

The Initial Request Mailbox message (Mailbox Type C0h) is shown in the following table:

Table 3-6. Initial Request Mailbox Message

Byte Number	Size in Bytes	Value	Description
1	1	1b	Start of message character.
2	1	4d	Message type character (4dh for Mailbox message).
3-4	2	00 00	Reserved; must be set to zero.
5-7	3	xx xx xx	Time Stamp (optional, may be zeroes).
8	1	00	Reserved; must be set to zero.
9	1	xx	Sequence Number: 00 to ffh inclusive.
10	1	c0	Mailbox Type: Initial Request.
11-14	4	10 3a 00 00	Mailbox Source ID (master SNP device).
15-18	4	10 0a 00 00	Mailbox Destination ID (PLC service request task).
19	1	01	Packet Number, always one (1).
20	1	01	Total Packets, always one (1).
21	1	xx	Request code; dependent on the service request.
22-34	13	xx xx xx xx xx xx	Service request data:
		xx xx xx xx xx xx xx	Dependent on service request
35	1	17	End of block character.
36	1	00	Next message type.
37-38	2	00 00	Next message length.
39	1	00	Status byte.
40	1	xx	Block Check Code (BCC).

The Initial Request Mailbox message with Text Buffer (80h) is shown in the following table:

Table 3-7. Initial Request Mailbox Message, With Text Buffer

Byte Number	Size in Bytes	Value	Description
1	1	1b	Start of message character.
2	1	4d	Message type character (4dh for Mailbox message).
3-4	2	00 00	Reserved; must be set to zero.
5-7	3	xx xx xx	Time Stamp (optional, may be zeroes).
8	1	00	Reserved; must be set to zero.
9	1	xx	Sequence Number: 00 to ffh inclusive.
10	1	80	Mailbox Type: Initial Request with Text Buffer.
11-14	4	10 3a 00 00	Mailbox Source ID (master SNP device).
15-18	4	10 0a 00 00	Mailbox Destination ID (PLC service request task).
19	1	00	Reserved; must be set to zero.
20	1	01	Must be set to one (1).
21-22	2	xx xx	Total length of data in bytes.
23-26	4	00 00 00 00	Reserved; must be set to zero.
27	1	01	Packet Number, always one (1).
28	1	01	Total Packets, always one (1).
29	1	xx	Request code; dependent on the service request.
30-34	5	xx xx xx xx xx	Service request data: dependent on service request.
35	1	17	End of block character.
36	1	xx	Next message type: must be 54h, 42h, or 43h for T, B, or C respectively.
37-38	2	xx xx	Next message length.
39	1	00	Status byte.
40	1	xx	Block Check Code (BCC).

Completion ACK Mailbox Messages

The PLC CPU (slave) responds to Initial Request Mailbox messages with Completion ACK Mailbox messages. If the response data is six bytes or less, a Completion ACK Mailbox message without a Text Buffer (Mailbox Type D4h) is sent to the master.

If, however, the data in the response requires more than six bytes, then a Completion ACK Mailbox message with Text Buffer (Mailbox Type 94h) is sent to the master, followed by one or more Text Buffer messages.

Piggy-Back Status Information

The last six bytes in the data portion of all Completion ACK Mailbox messages contain "piggy-back" status information, which relays the following information to the master:

Byte 29	Control Program Number
Byte 30	Current Privilege Level
Bytes 31-32	Last Sweep Time
Bytes 33-34	PLC Status Word

Control Program Number

This value represents the number of the control program task the SNP master is currently logged into. The valid values in today's implementations are -1 and 0 as follows:

-1 SNP master is not logged into a control program task.

-0 SNP master is logged into control program task 0.

Privilege Level

Current privilege level of the SNP master device. Valid values are 0 through 4 for Series 90-70 PLCs, and 1 through 4 for Series 90-30 PLCs.

Last Sweep Time

This value is equal to the time taken by the last complete sweep for the main control program task. The value is in 100 microsecond increments and is measured from Start of Sweep (X-1) to Start of Sweep (X).

PLC Status Word

The bits in this word are defined in the following table, where bit 0 is the least significant bit, and bit 15 is the most significant bit.

Recalling the convention of byte ordering within a word, byte 33 is the least significant byte (containing bits 0 through 7) and byte 34 is the most significant byte (containing bits 8 through 15):

PLC Status Word	Byte 33	Byte 34
	7-0	15-8

Table 3-8. Bits of the PLC Status Word

Bit Number	Description
Bit 0	Oversweep flag; meaningful only when constant sweep mode is active. 1 = Constant Sweep value exceeded. 0 = No oversweep condition exists.
Bit 1	Constant Sweep Mode. 1 = Constant Sweep Mode active. 0 = Constant Sweep Mode is not active.
Bit 2	PLC Fault Entry since last read. 1 = PLC fault table has changed since last read by this device. 0 = PLC fault table unchanged since last read.
Bit 3	I/O Fault Entry since last read. 1 = I/O fault table has changed since last read by this device. 0 = I/O fault table unchanged since last read.
Bit 4	PLC Fault Entry Present. 1 = One or more fault entries in PLC fault table. 0 = PLC fault table is empty.
Bit 5	I/O Fault Entry Present. 1 = One or more fault entries in I/O fault table. 0 = I/O fault table is empty.
Bit 6	Programmer attachment flag. 1 = Programmer attachment found. 0 = No programmer attachment found.
Bit 7	Front panel ENABLE/DISABLE switch setting. 1 = Outputs disabled. 0 = Outputs enabled.
Bit 8	Front panel RUN/STOP switch setting. 1 = RUN, 0 = STOP
Bit 9	OEM protected bit. 1 = OEM protection in effect. 0 = No OEM protection.
Bit 10	Not used.
Bit 11	Not used.
Bits 12-15	PLC State: 0 = Run I/O enabled. 1 = Run I/O disabled. 2 = Stop I/O disabled. 3 = CPU stop faulted. 4 = CPU halted. 5 = CPU suspended. 6 = Stop I/O enabled.

The following two tables describe the format for the two different types of Completion ACK Mailbox messages. The values of the bytes are listed when they are known and fixed; values are marked "xx" where we don't care, or the values are dependent on the response to the original service request made by the master.

The Completion ACK Mailbox message (Mailbox Type D4h) is shown in the following table:

Table 3-9. Completion ACKnowledge Mailbox Message

Byte Number	Size in Bytes	Value	Description
1	1	1b	Start of message character.
2	1	4d	Message type character (4dh for Mailbox).
3-4	2	xx xx	Don't care: ignore.
5-7	3	xx xx xx	Time Stamp.
8	1	xx	Don't care: ignore.
9	1	xx	Sequence Number: 00 to ffh inclusive; equal to the sequence number passed in the Initial Request.
10	1	d4	Mailbox Type: Completion ACK.
11-14	4	10 0a 00 00	Mailbox Source ID (PLC service request task).
15-18	4	10 3a 00 00	Mailbox Destination ID (master SNP device).
19	1	01	Packet Number.
20	1	01	Total Packets.
21	1	xx	Status code.
22	1	xx	Status data.
23-28	6	xx xx xx xx xx xx	Service request response data.
29	1	xx	Control program number.
30	1	xx	Current privilege level.
31-32	2	xx xx	Last sweep time.
33-34	2	xx xx	PLC status word.
35	1	17	End of block character.
36	1	00	Next message type.
37-38	2	00 00	Next message length.
39	1	00	Status byte.
40	1	xx	Block Check Code (BCC).

The Completion ACK Mailbox message with Text Buffer (Mailbox Type 94h) is shown in the following table:

Table 3-10. Completion ACKnowledge Mailbox Message, With Text Buffer

Byte Number	Size in Bytes	Value	Description
1	1	1b	Start of message character.
2	1	4d	Message type character (4dh for Mailbox).
3-4	2	xx xx	Don't care: ignore.
5-7	3	xx xx xx	Time Stamp.
8	1	xx	Don't care: ignore.
9	1	xx	Sequence Number: 00 to ffh inclusive; equal to the sequence number passed in the Initial Request.
10	1	94	Mailbox Type: Completion ACK with Text Buffer.
11-14	4	10 0a 00 00	Mailbox Source ID (PLC service request task).
15-18	4	10 3a 00 00	Mailbox Destination ID (master SNP device).
19	1	xx	Don't care: ignore.
20	1	01	Always one (1).
21-22	2	xx xx	Text buffer data length in bytes.
23-26	4	xx xx xx xx	Don't care: ignore.
27	1	01	Packet Number.
28	1	01	Total Packets.
29	1	xx	Control program number.
30	1	xx	Current privilege level.
31-32	2	xx xx	Last sweep time.
33-34	2	xx xx	PLC status word.
35	1	17	End of block character.
36	1	xx	Next message type: must be 54h, 42h, or 43h for T, B, or C respectively.
37-38	2	xx xx	Next message length.
39	1	00	Status byte.
40	1	xx	Block Check Code (BCC).

Error Nack Mailbox Message

If the PLC CPU is unable to comply with the service request made in the Initial Request Mailbox message (perhaps due to bad parameters, insufficient privilege, etc.), an Error Nack Mailbox message is sent back to the master. An error code is included in the Error Nack Mailbox message to inform the master why the Initial Request Mailbox message has been rejected. The following table shows the format of the Error Nack Mailbox message:

The Error Nack Mailbox message (Mailbox Type D1h) is shown in the following table:

Table 3-11. Error Nack Mailbox Message (D1h) Format

Byte Number	Size in Bytes	Value	Description
1	1	1b	Start of message character.
2	1	4d	Message type character (4dh for Mailbox).
3-4	2	xx xx	Don't care: ignore.
5-7	3	xx xx xx	Time Stamp.
8	1	xx	Don't care: ignore.
9	1	xx	Sequence Number: 00 to ffh inclusive; equal to the sequence number passed in the Initial Request Mailbox.
10	1	d1	Mailbox Type: Error Nack.
11-14	4	10 0a 00 00	Mailbox Source ID (PLC service request task).
15-18	4	10 3a 00 00	Mailbox Destination ID (master SNP device).
19	1	01	Packet Number.
20	1	01	Total Packets.
21	1	xx	Major Error Status (See Table 3-12).
22	1	xx	Minor Error Status (See Tables 3-13 and 3-14).
23-34	12	xx xx xx xx xx xx xx xx xx xx xx xx	Don't care.
35	1	17	End of block character.
36	1	00	Next message type.
37-38	2	00 00	Next message length.
39	1	00	Status byte.
40	1	xx	Block Check Code (BCC).

The key fields in the Error Nack mailbox are bytes 21 (Major Error Status) and 22 (Minor Error Status). The minor error status only has meaning for certain major error status codes. The following tables define the error status codes. Unless otherwise stated, the minor error status code is undefined. The minor error status codes are broken into two tables: one for general errors, and the second for errors due specifically to program load and store requests.

Table of Major Error Status Codes

Table 3-12. Major Error Status Codes

Error Status	Description
01h	Illegal Service Request: either not defined or not supported.
02h	Insufficient Privilege: the minor status field contains the privilege level required for the service request.
04h	Protocol Sequence Error: the CPU has received a message that is out of order.
05h	Service Request Error: Minor status field contains the request specific error code. See table of Minor Error Status Codes below.
06h	Illegal Mailbox Type: Service request mailbox type is either undefined or unexpected.
07h	The PLC CPU's Service Request Queue is full. The master should retry later. It is recommended that the master wait a minimum of 10 ms before sending another service request.

Table of Minor Error Status Codes

Table 3-13. Minor Error Status Codes: General

Decimal	Hex	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.

Table 3-13. Minor Error Status Codes: General - continued

Decimal	Hex	Description
-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.
-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.
-64	0C0	Bad OMF record checksum in store.
-65	0BF	Illegal OMF record type/data contents.

Table 3-13. Minor Error Status Codes: General - continued

Decimal	Hex	Description
-66	0BE	Bad Block Type given in Load/Store.
-67	0BD	Block Set (subblock name) not found.
-68	0BC	Block Type (e.g., data) not found.
-69	0BB	Maximum length of a partial store exceeded.
-70	0BA	Block Set already exists, cannot create.
-71	0B9	Executable flag in TYPDEF record not set.
-72	0B8	Size of the Segment Selector Table in TYPDEF record is not correct.
-73	0B7	Segment length in Verify not equal to the segment length of block in the PLC.
-74	0B6	Cyclic Redundancy Check (CRC) checksum comparison in Verify failed.
-75	0B5	Additive checksum comparison in Verify failed.
-76	0B4	Attempt to alter interrupt list in MAIN DECL BLOCK during RUN MODE.
-77	0B3	Length limit exceeded; includes read past end of transferred data, writes past end of program block.
-78	0B2	Program block already exists and cannot be replaced.

Table 3-14. Minor Error Status Codes: Program Load and Store Requests

Decimal	Hex	Description
80	50	Problem with sending mail to the slave Service Request task. (Series 90-70 PLC CPUs only).
81	51	Problem with getting mail from the slave Service Request task. (Series 90-70 PLC CPUs only).
85	55	Slave SNP task timed out before receiving SRP response. (Series 90-70 PLC CPUs only).
86	56	Slave SNP task could not find the requested Datagram connection. (Series 90-70 PLC CPUs only).
87	57	Slave SNP task encountered an error in trying to write the Datagram. (Series 90-70 PLC CPUs only).
88	58	Slave SNP task encountered an error in trying to update the Datagram. (Series 90-70 PLC CPUs only).

Text Buffer Message

A Text Buffer Message is the main means through which data is passed back and forth between the master and the PLC. Register memory and I/O data are passed back and forth via Text Buffer Messages (assuming that the amount of data to be passed is greater than what will fit into the accompanying Mailbox message). Multiple Text Buffer messages may be necessary to transfer all the data.

The number of Text Buffer messages required to transfer some number of bytes depends on the SNP Data Size allowed in a SNP Data message. For example, suppose the SNP Data Size has been set to 200 bytes (via a SNP Parameter Select message), and the master reads 800 bytes of discrete data. The slave responds to the master's request with a Completion ACK Mailbox message with Text Buffer (94h), followed by four Text Buffer messages. Suppose 850 bytes of discrete data are read: then five Text Buffer messages are required to transfer all the data bytes.

The Mailbox message that precedes a Text Buffer message(s) has a word field that contains the total number of data bytes to be transferred. The Next Message Type field in the Mailbox message's trailer is equal to 54h for `**T**', and the Next Message Length field in the Mailbox message's trailer equals the number of data bytes contained in the next message (the Text Buffer message). After this Mailbox message is transmitted, one or more Text Buffer messages are transmitted until all data bytes have been transferred.

The following table shows the format of the Text Buffer message, where N is the number of data bytes contained in the message.

Table 3-15. Text Buffer Message Format

Byte Number	Bytes	Description
1	1	Start of message character (ESC = 1bh).
2	1	Message type character (54h for Text Buffer).
3-N+2	xx xx xx ...	Data text bytes.
N+3	1	End of block character (17h) marks the beginning of the packet trailer.
N+4	1	Next message type (0 or 54h) where:
		0 means no more Text Buffer messages (link turnaround).
		54h means another Text Buffer message is going to be transmitted which contains more data bytes.
N+5	2	Next message length.
N+6	1	Status byte.
N+8	1	Block Check Code (BCC).

Block Transfer Message

A Block Transfer message is used to pass PLC logic programs back and forth between the master and the PLC. Multiple Block Transfer messages may be necessary to transfer all the program data associated with a PLC logic program. The number of Block Transfer messages required to transfer some number of bytes depends on the SNP Data Size allowed in a SNP Data message.

The Mailbox Type and format of the Mailbox message that precedes a Block Transfer message(s) differs between the Series 90-30 PLC CPUs and the Series 90-70 PLC CPUs:

For Series 90-70 PLC CPUs, the Mailbox type is either C0h (Initial Request Mailbox message) or D4h (Completion ACK Mailbox message), depending on which direction the program data is being passed.

For Series 90-30 PLC CPUs, the Mailbox type is either 80h (Initial Request Mailbox with Text Buffer message) or 94h (Completion ACK Mailbox with Text Buffer message), depending on which direction the program data is being passed.

For both Series 90-30 PLCs and Series 90-70 PLCs, the Next Message Type field in the Mailbox message's trailer is equal to 42h for `B', and the Next Message Length field in the Mailbox message's trailer equals the number program data bytes contained in the next message (the Block Transfer message). After this Mailbox message is transmitted, one or more Block Transfer messages are transmitted until all program data bytes have been transferred (Series 90-70 has some further differences in this respect).

The following is the format of the Block Transfer message, where N is the number of program bytes contained in the message:

Table 3-16. Block Transfer Message Format

Byte Number	Size in Bytes	Description
1	1	Start of message character (ESC = 1bh).
2	1	Message type character (42h for Block Transfer).
3-N+2	xx xx xx ...	Data text bytes.
N+3	1	End of block character (17h) marks the beginning of the packet trailer.
N+4	1	Next message type (0 or 42h) where:
		0 means no more Block Transfer messages (link turnaround).
		42h means another Block Transfer message is going to be transmitted which contains more data bytes.
N+5	2	Next message length.
N+6	1	Status byte.
N+8	1	Block Check Code (BCC).

If the next message type field in the trailer of the Block Transfer message is 0, then the transmission of all the program data is complete; it fit into one text buffer. If, however, this field is not 0, it must be a 42h for **B**, and the next message length must have the length of the next SNP packet to be transferred: another Block Transfer message. Block Transfer messages will be transmitted from the sender to the receiver until all program data bytes (whose total number was specified in the introductory Mailbox message) have been transferred.

Connection Data Message

A Connection Data message is the means through which Datagram information is passed back and forth between the programming device and the PLC. Datagrams enable the user to obtain reference data values for one or more PLC memory types via a single service request once a Datagram Connection Area has been established and defined. Please see the section `Datagram Service Requests' in Chapter 6 for Datagram details.

Multiple Connection Data messages may be necessary to transfer all the data associated with a Datagram Connection Area. The number of Connection Data messages required to transfer some number of bytes depends on the SNP Data Size allowed in a SNP Data message.

The Mailbox Type and format of the Mailbox message that precedes a Connection Data message(s) differs between the Series 90-30 PLC CPUs and the Series 90-70 PLC CPUs:

For Series 90-70 PLC CPUs, the Mailbox type is either C0h (Initial Request Mailbox message) or D4h (Completion ACK Mailbox message), depending on which direction the connection data is being passed.

For Series 90-30 PLC CPUs, the Mailbox type is either 80h (Initial Request Mailbox with Text Buffer message) or 94h (Completion ACK Mailbox with Text Buffer message), depending on which direction the connection data is being passed.

For both Series 90-30 PLCs and Series 90-70 PLCs, the Next Message Type field in the Mailbox message's trailer is equal to 43h for `**C**', and the Next Message Length field in the Mailbox message's trailer equals the number of connection data bytes contained in the next message (the Connection Data message). After this Mailbox message is transmitted, one or more Connection Data messages are transmitted until all connection data bytes have been transferred.

The following is the format of the Connection Data message, where N is the number of connection data bytes contained in the message:

Table 3-17. Connection Data Message Format

Byte Number	Size in Bytes	Description
1	1	Start of message character (ESC = 1bh).
2	1	Message type character (**43h** for **C**onnection Data).
3-N+2	xx xx xx ...	Data text bytes.
N+3	1	End of block character (17h) marks the beginning of the packet trailer.
N+4	1	Next message type (0 or 43h) where:
		0 means no more Connection Data messages (link turnaround).
		43h means another Connection Data message is going to be transmitted which contains more data bytes.
N+5	2	Next message length.
N+6	1	Status byte.
N+8	1	Block Check Code (BCC).

Inquiry Message

Whenever there has been no activity on the serial link in **T3** time, the master may send the Inquiry message to the slave in order to keep the communication session active. Otherwise, if there is no activity on the serial link causing the slave's **T3'** timer to time out, the slave will assume a loss of communication and return to its initial state. The slave echoes the Inquiry message, byte for byte, back to the master. The Inquiry message is 40 bytes long and has the following format:

Table 3-18Inquiry Message Format

Byte Number	Size in Bytes	Description
1	1	Start of message character (ESC = 1bh).
2	1	Message type character (**49h** for **I**nquiry).
3-34	32	Always zero.
35	1	End of block character (17h) marks the beginning of the packet trailer.
36	1	Next message type.
37-38	2	Next message length.
39	1	Status byte.
40	1	Block Check Code (BCC).

Inquiry Message Example

In this example, the master's **T3** timer has expired, which means that there has been no activity over the serial link in that time, and a message must be sent to the slave in order to keep the established communication session active. Otherwise, the slave's **T3'** timer will time out and the slave will assume loss of communication and go back to an idle state, waiting for a long break* or an attach message*. To issue an Inquiry message to the slave, the master sends the following series of bytes over the serial link:

1b-49-

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

17-00-00 00-00-3d

where the bytes break down as shown in Table 3-18.

The slave's response message to the Inquiry message is the same as the master's Inquiry message. The slave just echoes it back to the master.

* The long break is not required for Break-Free SNP operation

Chapter 4

Integrity Checking and Error Recovery

As in other serial communication protocols, the Series 90 Protocol (SNP) follows a communication sequence between the source device and the target device. The source device (master) initiates the sequence and the target device (slave) responds.

Acknowledgment and Negative Acknowledgment Messages

Two special messages are defined in the SNP protocol as a means of providing each side a way to tell the other side that:

Yes, I got your message okay.

or

No, I did not get your message, please re-transmit the message if possible.

Either an Acknowledgment or a Negative Acknowledgment message must be sent after every SNP packet is received by each SNP device (master or slave), in order to let the transmitting device know that its message was either received successfully, or received with some error.

Note

The exceptions to this rule are the Attach message and the Attach Response message: there is no acknowledgment for the Attach message or the Attach Response. If the Attach message did not make it over the link successfully, the slave will never answer with an Attach Response, and the master should retry the attach sequence: transmit a long break* followed by an Attach message. If the master receives an Attach Response message with errors, it should also retry the attach sequence.

Both the acknowledgement (ACK) and negative acknowledgment (NAK) are two-byte messages with the following formats:

Table 4-1. Acknowledge (ACK) Message Format

Byte Number	Size in Bytes	Code	Description
1	1	06	Acknowledge.
2	1	00	Reserved for future use.

* The long break is not required for Break-Free SNP operation

Table 4-2. Negative Acknowledge (NAK) Message Format

Byte Number	Size in Bytes	Code	Description
1	1	15h	Negative Acknowledge.
2	1	xx	Error code:
			00 = BCC error or Parity error. 01 = Overrun or Framing error. 02 = Sequence error. 03 = Bad Next Message Length. (Next Message Length exceeds size allowed with the currently defined SNP Maximum Data Size.)

Block Check Code (BCC)

As already described in previous sections, a BCC is included in every SNP packet to provide an integrity check on the data in the packet. Each side, as it receives a SNP packet should calculate the BCC for the packet and compare the calculated BCC to that passed in the packet. The two values must match. If they do not match, a NAK message with the error code set to zero must be sent by the device that received the packet to the transmitting device. Otherwise, if the block check codes do match, and no other errors are encountered (i.e., Sequence, Parity, Overrun, or Framing errors), then an ACK message must be sent to the device that transmitted the packet. (The Attach/Attach Response packets are the exception to the ACK/NAK rule, as previously noted.)

Parity, Overrun, or Framing Errors

The PLC CPU (slave) checks the status of the UART each time it receives an SNP packet. If the status indicates a UART error (parity, overrun, or framing error), a NAK message is sent to the master with the error code set to zero (0) for parity error, or set to one (1) for overrun or framing error.

Likewise, if the programming device (master) has the capability to interrogate the status of its UART, it should do so upon receipt of each SNP packet. If the programming device detects a UART error, then a NAK message is sent to the slave with the error code set appropriately.

If the UART status is okay, and no other errors with the packet are detected (i.e. sequence error or BCC error), then an ACK message is sent to the transmitting device. (The Attach/Attach Response packets are the exception to the ACK/NAK rule, as previously noted.)

Sequence Errors

A sequence error occurs whenever a message type is received that was not expected: either

`the message type in the current packet does not match the Next Message Type from the trailer of the previous message`

`or`

`after a SNP device has turned the link around, (next message type in the trailer of the last packet equals zero), the next message type received from the other device was not a Mailbox message.`

For example, suppose the slave receives a Mailbox message from the master whose next message type field in the trailer is set to 'T' (54h). The slave expects the next SNP packet from the master to have a message type of 'T'. But the next SNP packet from the master has a message type of 'M' (4Dh). This constitutes a sequence error, and instead of receiving an ACK message, the master will receive a NAK message with the error code set to two (2). Sequence Errors are fatal to the current connection session. When either SNP device detects a sequence error, it transmits the NAK message, and then returns to its start state.

SNP Retries

As previously stated, sequence errors are fatal, and no retries are attempted. If, however, the NAK message is due to a BCC error or data transmission errors (error code in NAK message equals 0 or 1), then the transmission of the message should be attempted again. If either side receives a NAK to a SNP packet it has just transmitted, with an error code of 0 or 1, then that side must re-transmit the SNP packet. There are two scenarios from the master point of view:

1. Master has received a SNP packet from a slave, and has detected some error (for example, the master's calculated BCC for the SNP packet doesn't match that passed in the packet). The master sends a two-byte NAK message to the slave device, and waits for the slave to re-transmit the SNP packet. If the packet still is not received successfully after two retries, the master will have to give up, and return to its start state.

2. The master receives a NAK message from the slave in response to a SNP packet that the master sent to the slave device. The slave device is now in a state where it is waiting for the master to re-transmit the same SNP packet. The master must do so. As in the case above, if the master SNP packet is NAKed by the slave after two retries, the master will have to give up and return to its start state.

Chapter 5 SNP Parameter Select Message

The Series 90 Protocol (SNP) Parameter Select Message provides you the capability to fine tune your SNP communication link by adjusting the following SNP parameters: Timers **T2** and **T3'**, and Maximum Data Size. There is another field called Queue Depth, which must always be set to one (1) in today's implementations. This will always hold true for Series 90-30 PLCs, but may change in the future on Series 90-70 PLCs to allow the queuing of two or more requests.

This is the only Mailbox message whose Destination ID is the Slave SNP driver as opposed to the Slave Service Request task. The reply to this message is a Mailbox message in the same format as the request, but with the Mailbox Source ID and Destination ID transposed, with a response code of 'D4' instead of a request code of 'C0', and with possibly different **T2**, **T3'**, and maximum data size values than those sent. This Mailbox message has the following format:

Table 5-1. SNP Parameter Select Message Format

Byte Number	Size in Bytes	Description
1	1	Start of message character (ESC = 1bh).
2	1	Message type character (**4dh** for **M**ailbox).
3-4	2	Reserved for internal use.
5-7	3	Time Stamp (not used by PLC CPU slaves).
8	1	Reserved for internal use.
9	1	Sequence Number.
10	1	Mailbox Type Code.
11-14	4	Mailbox Source ID.
15-18	4	Mailbox Destination ID.
19	1	Packet Number.
20	1	Total Packets.
21	1	Request code.
22	1	Reserved for internal use.
23-24	2	T2
25-26	2	T3'
27-28	2	Maximum Data Size.
29-30	2	Queue Depth.
31-34	4	Reserved for internal use.
35	1	End of block character (17h) marks the beginning of the packet trailer.
36	1	Next message type.
37-38	2	Next message length.
39	1	Status byte.
40	1	Block Check Code (BCC).

T2 Field

The master passes a T2 value in milliseconds to the slave. The slave PLC CPU compares this value to its own T2 value, and passes the greater of the two back to the master. Whatever value is returned to the master in the T2 field must be the T2 time used throughout this connection session.

T3' Field

The master passes a T3' value in milliseconds to the slave. The slave PLC CPU compares this value to its own T3' value, and passes the greater of the two back to the master. Whatever value is returned to the master in the T3' field must be the T3' time used throughout this connection session.

Maximum Data Size

The master passes a packet data size, defined in bytes, to the slave. This value reflects the maximum bytes allowed in the data portion of a SNP packet (i.e., does not include the header and trailer bytes). The slave PLC CPU compares this value to its own Maximum Data Size allowed, and passes the lesser of the two back to the master. Whatever value is returned to the master in the Maximum Data Size field must be the Maximum Data Size value used throughout this connection session.

Maximum Queue Depth

The value of the Maximum Queue Depth for Series 90-30 PLCs is always one (1).

For Series 90-70 PLCs, the current implementation is only one (1). Future enhancements may allow Queue Depth to be increased, but in today's implementation, it must be set to one.

The SNP protocol provides you the capability to disable the **T2** (wait for ACK timeout) and **T3'** (link idle timeout) timers. If the master wants to disable these timeouts, the master sends a SNP Parameter Select Message to the slave with either or both the T2 and T3' fields set to zero. One timer cannot be disabled independently of the other; both are disabled if either time field in the SNP Parameter Select Message is zero.

Parameter Select Example

Suppose the master has attached to a Series 90-30 331 CPU, and wants to set up the following:

```
T2                 = 5 seconds  = 5000 ms
T3'                = 25 seconds = 25000 ms
Maximum Data size = 1000 bytes
Queue Depth        = 1 (Must be 1)
```

The master sends the following series of bytes over the serial link:

1b 4d

00 00 00 00 00 00 03 c0 10 3a 00 00 10 3e 00 00 01 01 00 00 88 13 a8 61 e8 03

01 00 00 00 00 00

17 00 00 00 00 29

where the bytes break down as follows:

Table 5-2. SNP Parameter Select Message Example

Byte Number	Value	Description
1	1b	Start of message character (ESC).
2	4d	Message type byte; 4dh = M for Mailbox.
3-4	00 00	Reserved for internal use.
5-7	00 00 00	Time Stamp.
8	00	Reserved for internal use.
9	03	Sequence number.
10	c0	Mailbox type.
11-14	10 3a 00 00	Mailbox Source ID.
15-18	10 3e 00 00	Mailbox Destination ID.
19	01	Packet number.
20	01	Total packets.
21	00	Request code.
22	00	Reserved for internal use.
23-24	88 13	T2 time in ms = 1388h = 5000 ms = 5 secs.
25-26	a8 61	T3' time in ms = 61a8h = 25000 ms = 25 secs.
27-28	e8 03	Maximum Data Size = 3e8h = 1000 bytes.
29-30	01 00	Queue depth = 1.
31-34	00 00 00 00	Reserved for internal use.
35	17	End Of Block character (ETB).
36	00	Next Message Type: value of zero turns the link around.
37-38	00 00	Next Message Length.
39	00	Status byte.
40	--	Block Check Code (BCC).

Chapter 6

Service Requests

Service requests made via Initial Request Mailbox messages are the means through which the master transfers data to the PLC CPU, and reads data from the PLC CPU, and issues commands to the PLC CPU.

Some of the service requests require a minimum privilege level before the request can be honored, and some require that the master device be logged in to the Programmer Communication Window. The following table lists the service requests available to a user who is writing a master SNP implementation, and the minimum privilege level and login requirements for each.

Table 6-1. SNP Service Requests

Service Request	Privilege Level (min)		Logged In
	Series 90-30 PLC	Series 90-70 PLC	
Change Privilege Level	1	0	NO
Read System Memory	1	1	NO
Read Task Memory	n/a	1	NO
Read Program Block Memory	n/a	1	NO
Write System Memory	2	2	NO
Write Task Memory	n/a	2	NO
Write Program Block Memory	n/a	2	NO
Toggle Force System Memory	2	2	NO
Programmer Logon/Logoff	1	0	NO
Return Fault Table	1	1	NO
Clear Fault Table	2	2	YES
Return Control Program Name	1	0	NO
Return Controller Type and ID	1	1	NO
Return PLC Time/Date	1	1	NO
PLC Short Status	1	0	NO
Set Controller ID	3 *	3	YES
Set PLC Time/Date	2 **	2	NO
Set PLC State	2	2	YES
Establish Datagram	1	1	NO
Write Datagram	1	1	NO
Update Datagram	1	1	NO
Cancel Datagram	1	1	NO
Program Load	1	n/a	YES
Program Store	3	n/a	YES
Program Block Store	n/a	3	YES
Program Block Load	n/a	1	YES

*For series 90-30 PLCs with Release 1.x CPU firmware, privilege level 4 is required.

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

The sections that follow describe each of the service requests in the table above. Examples along with explanations are included for each.

In the service request tables that follow, you will notice that all examples are in hexadecimal, and that several abbreviations are used (e.g., BCC, MB, xx). Refer to Chapter 1 for a complete listing of terms and acronyms.

```
BCC = Block Check Code
MB  = Mailbox
xx  = Don't Care:  Meaningless data and should be ignored.
```

Caution

Fields that are either reserved or not used in the Initial Request Mailbox messages <u>must</u> **be set to zero, as shown in all of the following examples. Failure on the user's part to comply with this rule may cause unexpected results in current or future revisions of the PLC CPU firmware as later features are implemented.**

Change Privilege Level

The Change Privilege Level request enables the master to change its access privilege level to the PLC CPU so long as the proper password for the requested level is provided (none needed if no password is active for that level.) For Series 90-70 PLCs, there are five privilege levels (0 through 4), for Series 90-30 PLCs, there are four privilege levels (1 through 4).

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	21h	Change PLC CPU Privilege Level.
Change Level (byte 22):	-1	Change to the highest level belonging to the specified password.
	0	Level 0 (Series 90-70 only).
	1	Level 1
	2	Level 2
	3	Level 3
	4	Level 4
Password (bytes 23-30):		NUL terminated ASCII string.

Please note that if a user knows a password, but not the level associated with the password, the request can be issued with a Change Level of -1 followed by the 8-byte ASCII encoded password. The PLC CPU will change the requestor's privilege level to the level associated with the password. If the password is used for more than one level, then the highest level with that password is reached.

Response Mailbox Message:

	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox Message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response to a privilege request are:

- Invalid password.
- No password provided when one is required.

Examples with full explanations follow.

Example of Change Privilege Level with No Password

This example changes the master's privilege level to four (4). The example assumes that the master has already attached to the slave device, and that there are no active passwords. (See the next example for a case where there are passwords active, and the correct password must be provided in order to change the level). The master sends the Initial Request Mailbox message, with the service request code for Change Privilege Level (21h) along with the desired level to the slave, and the slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 04 c0 10 3a 00 00 10 0a 00 00	
01 01 21 04 00 00 00 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 04 d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 04 00 00 5c 20
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Change Privilege Level With No Password

Table 6-2. Change Privilege Level With No Password

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	04	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	21	Request code: Change Privilege level.
	22	04	Privilege Level requested: in this case, 4.
	23-30	00 00 00 00 00 00 00 00	Password: Null
	31-34	00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	04	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	04	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Change Privilege Level With Password

The following example assumes that the master has already attached to the slave device, and knows the password to be "AB1". The master sends the Initial Request Mailbox message to the slave, with the service request code for Change Privilege Level (21h), change level field set to -1, and the password field set to "AB1" terminated with a NUL. The slave responds with a Completion ACK Mailbox message which includes the piggy-back status information. It can be noted from the piggy-back status that the privilege level associated with the password "AB1" must be level three (3).

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 52 c0 10 3a 00 00 10 0a 00 00	
01 01 21 ff 41 42 31 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 52 d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 03 00 00 5c 20
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Change Privilege Level With Password

Table 6-3. Change Privilege Level With Password

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	52	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	21	Request code: Change Privilege level.
	22	ff	Privilege Level requested: -1 (find the level).
	23-30	41 42 31 00 00 00 00 00	Password: "AB1"
	31-34	00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	52	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	03	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Read and Write PLC Memory

This section describes service requests that allow you to access memory areas within the PLC:

- Read System Memory request
- Write System Memory request
- Read Task Memory
- Write Task Memory
- Read Program Block Memory request
- Write Program Block Memory request

Simply stated, these requests allow the user to read and write reference data (i.e., %I, %Q, %T, %R, %L, etc).

Discrete data may be accessed as bit data or byte data. Registers, Analog I/O, Local Subblock Data and Program Block Data may be accessed as word data only.

Each possible access mode (bit, byte, or word) for all reference memory types is assigned a distinct "segment selector". (You may think of segment selector, as a type of addressing scheme). Table 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 the memory types within this table require a privilege level of 1 or higher, and write requests require a privilege level of 2 or higher. It should be noted that the %S Discrete are READ ONLY.

Table 6-4. Reference Data Segment Selectors

Memory Type*	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 Discrete	78	4E	-bit
	24	18	-byte
%SB Discrete	80	50	-bit
	26	1A	-byte
%SC Discrete	82	52	-bit
	28	1C	-byte
%S Discrete	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
Local Subblock Data (%L) (90-70 only)	00	00	-word
Program Block Data (%P) (90-70 only)	04	04	-word

*The maximum addressable ranges for each memory type depends on the model of CPU and memory configuration.

Except for Program Block Data (%P) and Local Subblock Data (%L), all memory types within this table are accessed via the service requests **Read System Memory** and **Write System Memory**.

The %P and %L memory types exist only in Series 90-70 PLCs. The %P memory is global data accessible to all program blocks within a control program task. One task's %P memory is not accessible to any other task. The %L memory type is data local to a particular program block within a control program task.

You can read and write %P data with either the **Read Task Memory** and **Write Task Memory** service requests, or the **Read Program Block Memory** and **Read Program Block Memory** service requests. Only a task name is required to determine the correct %P memory to reference. Therefore, the Read/Write Task Memory service requests are preferred since they reduce the number of messages needed to access the data.

To read and write %L data, you must use the **Read Program Block Memory** and **Write Program Block Memory** service requests, since a task name and program block name must be provided in order to determine the correct %L memory area to reference.

There are three fields in the Initial Request Mailbox message common to these six service requests that require detailed explanation:

Segment Selector:

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

Data Offset:

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.

Data Length:

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)
(3)	%Q497 to %Q499 in bit mode:		
	-Segment Selector	= 48	(%Q memory in bit mode)
	-Data Offset	= F0 01	(= 01F0h = 496 = bit index to %Q497)
	-Data Length	= 03 00	(= 0003h = 3 bits)
(4)	%Q497 to %Q512 in bit mode:		
	-Segment Selector	= 48	(%Q memory in bit mode)
	-Data Offset	= F0 01	(= 01F0h = 496 = bit index to %Q497)
	-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	= 08	(%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)
(3)	%L7 in word mode:		
	Segment Selector	= 00	(%L memory in word mode)
	Data Offset	= 06 00	(= 0006h = word index to %L7)
	Data Length	= 01 00	(= 0001h = 1 word)

Read System Memory

The Read System Memory (Read SMEM) request allows the master to access a single contiguous block of data from any PLC System Memory Type.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	04h	Read System Memory.
Segment Selector (byte 22):		See Segment Selector Table (Table).
Data Offset (bytes 23-24):		Zero-based offset of data.
Data Length (bytes 25-26):		Length of data to read.
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	94h	Completion ACK Mailbox message with Text Buffer.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) or Completion ACK Mailbox message with Text Buffer (94h) is returned to the master. The type of Mailbox message returned depends on the amount of data to be passed back to the master. If the number of data bytes being read is six bytes or less, the slave responds with a Completion ACK Mailbox message which contains the data read. If the number of data bytes being read is greater than six bytes, the slave responds with a Completion ACK Mailbox message with Text Buffer, and the data read is returned to the master in one or more Text Buffer messages.

Completion ACK Mailbox message (D4h):		
Requested Data (bytes 23-28):		
Completion ACK Mailbox message with Text Buffer (94h):		
Total length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Next message type (byte 36):	54h	\`T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the next Text Buffer message.

If the CPU is unable to comply with the request, it is rejected. In this case, an Error Nack Mailbox message (D1h) is returned along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message is:

- Insufficient privilege (Series 90-70 PLCs only; level 1 or higher).
- Invalid input parameter in the request mailbox message.

Response Text Buffer Message:

If the slave returns a Completion ACK Mailbox message with Text Buffer (94h), the data returns to the master in one or more Text Buffer messages. The number of Text Buffer messages returned depends on the amount of data being read and the maximum data size allowed in a SNP Data message.

Examples with full explanations follow.

Example of Read SMEM No Text Buffer: %M in Bit Mode

This example reads the eleven discrete internal bits %M99 through %M109, using the %M bit mode segment selector and a data offset and data length 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

The master sends the Initial Request Mailbox message to the slave, with the service request code for Read System Memory (04h) along with the segment selector, data offset, and data length. The slave responds with a Completion ACK Mailbox message which contains the data requested and includes the piggy-back status information. It should be noted that all bits **not** requested within a byte are returned as zero, regardless of their true state.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 02 c0 10 3a 00 00 10 0a 00 00	
01 01 04 4c 62 00 0b 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	Xx xx xx xx xx xx 02 d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx 90 13 xx xx xx xx 00 04 00 00 5c 20
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Read SMEM no Text Buffer: %M in Bit Mode

Table 6-5. Read SMEM no Text Buffer: %M in Bit Mode

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	02	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	04	Request code: Read System Memory.
	22	4c	Segment Selector: %M data in bit mode.
	23-24	62 00	Data offset: 0062h = 98 (zero-based) = %M99.
	25-26	0b 00	Data length: 000bh = 11 bits.
	27-34	00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	02	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22	xx	Don't care.
	23-24	90 13	Data returned: %M99 through %M109.
	25-28	xx xx xx xx	Don't care.
	29	00	Control program number.
	30	04	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Read SMEM with Text Buffer: %R Word Mode

This example reads the eleven words of register data starting at %R3 up to and including %R13, using the %R segment selector with a data offset and data length whose units are word. The example assumes that the master has already attached to the slave device, and %R3 through %R13 have the following values:

```
%R3 = 3489h  %R5 = 9071h  %R7 = 2627h  %R9 = 8431h  %R11 = 1234h
%R4 = 1217h  %R6 = 2853h  %R8 = 0917h  %R10 = 4172h %R12 = 5678h  %R13 = 0987h
```

The master sends the Initial Request Mailbox message to the slave, with the service request code equal to Read System Memory (04h), along with Segment Selector, Data offset, and Data Length. The slave responds with a Completion Ack Mailbox message with Text Buffer which includes piggy-back status information. After the master acknowledges the Mailbox message, the slave sends the Text Buffer message. The Text Buffer message has 22 bytes of data which reflect the 11 words of register data %R3 through %R13.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 62 c0 10 3a 00 00 10 0a 00 00	
01 01 04 08 02 00 0b 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message with Text Buffer]
	1b 4d
	xx xx xx xx xx xx 62 94 10 0a 00 00 10 3a 00 00
	xx 01 16 00 xx xx xx xx 01 01 00 01 00 00 4c 20
	17 54 1e 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	
	(wait **T1** time)
	5: ←[Response Text Buffer message]
	1b 54
	89 34 17 12 71 90 53 28 27 26 17 09 31 84
	72 41 34 12 78 56 87 09
	17 00 00 00 00 BCC
(wait **T1** time)	
6: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Read SMEM with Text Buffer: %R Word Mode

Table 6-6. Read SMEM with Text Buffer: %R Word Mode

Packet Number	Byte Number(s)	Hex Value	Description
1:→	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	62	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	04	Request code: Read System Memory.
	22	08	Segment Selector: %R data, word mode.
	23-24	02 00	Data offset: 0002h = word offset for %R3.
	25-26	0b 00	Data length: 000bh = 11 words.
	27-34	00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	62	Sequence number.
	10	94	Mailbox type: Completion ACK with Text Buffer.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	xx xx	Reserved: don't care.
	21-22	16 00	Total length of data in bytes = 0016h = 22
	23-26	xx xx xx xx	Don't care.
	27-28	01 01	Packet number, total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	4c 20	PLC status word.
	35	17	End of block character.
	36	54	Next message type =`T' for Text Buffer.
	37-38	1e 00	Next message length in bytes = 001eh = 30
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-24	89 34 17 12 71 90 53 28 27 26 17 09	%R3 through %R13, where each word is returned
	25	31 84 72 41 34 12 78 56 87 09	as least significant byte 1st, most significant byte2nd.
	26	17	End of block character.
	27-28	00	Next message type.
	29	00 00	Next message length.
	30	00	Status byte.
		BCC	Block Check Code.

Write System Memory

The Write System Memory (Write SMEM) request allows the master to write a single contiguous block of data to any PLC System Memory Type.

Key fields within the request and response messages:

Request Mailbox Message without Text Buffer:

If the amount of data to be written is eight bytes or less, then an Initial Request Mailbox message with the data contained in the Mailbox message itself is all that is needed.

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	07h	Write System Memory.
Segment Selector (byte 22):		See Segment Selector Table (Table).
Data Offset (bytes 23-24):		Zero-based offset of data.
Data Length (bytes 25-26):		Length of data to write.
Write Data (bytes 27-34):		Maximum of 8 bytes to write.

Request Mailbox Message with Text Buffer:

If the amount of data to be written is greater than eight bytes, then an Initial Request Mailbox message with Text Buffer is required.

Mailbox Type (byte 10):	80h	Initial Request Mailbox message with Text Buffer.
Total Length of Data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Service Request code (byte 29):	07h	Write System Memory.
Segment Selector (byte 30):		See Segment Selector Table (Table).
Data Offset (bytes 31-32):		Zero-based offset of data.
Data Length (bytes 33-34):		Length of data to write.
Next Message Type (byte 36):	54h	"T" for Text Buffer.
Next Message Length (bytes 37-38):		Number of bytes in the next Text Buffer message.

Request Text Buffer Message:

If the master sends an Initial Request Mailbox with Text Buffer (80h), the data to be written to the PLC system memory is transmitted to the slave in one or more Text Buffer messages. The number of Text Buffer messages required depends on the amount of data the master is writing to the PLC system memory, and the maximum data size allowed in a SNP Data message.

Response Mailbox Message:

Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU is unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message is:

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

Examples with full explanations follow.

Example of Write SMEM no Text Buffer: %Q in Bit Mode

This example writes to the discrete outputs %Q19 through %Q41 inclusive, using the %Q bit mode segment selector and a data offset and data length 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** written to within a byte 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

The master sends the Initial Request Mailbox message to the slave, with the service request code for Write System Memory (07h) along with the segment selector, data offset, and data length. The slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 04 c0 10 3a 00 00 10 0a 00 00	
01 01 07 48 12 00 17 00 e2 34 57 ff 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 04 d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 02 00 00 5c 20
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Write SMEM no Text Buffer: %Q in Bit Mode

Table 6-7. Write SMEM no Text Buffer: %Q in Bit Mode

Packet Number	Byte Number(s)	Hex Value	Description
1 →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	04	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	07	Request code: Write System Memory.
	22	48	Segment Selector: %Q data in bit mode.
	23-24	12 00	Data offset: 0012h = 18 (zero-based) = %Q19.
	25-26	17 00	Data length: 0017h = 23 bits.
	27-30	e2 34 57 ff	Data to write.
	31-34	00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	04	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	04	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Write SMEM with Text Buffer: %T in Byte Mode

This example writes 16 bytes of temporary discrete data starting at the discrete point %T81 up to and including the discrete point %T208, using the %T byte mode segment selector with a data offset and data length whose units are byte. The example assumes that the master has already attached to the slave device, and has set the privilege level to two (2). The discrete points %T81 through %T208 inclusive are currently set to zero.

When the write request is complete, bytes %T81 through %T201 should contain the follows values:

%T81 = 23h	**%T113 =39h**	**%T145 = 87h**	**%T177 = 34h**
%T89 = 89h	**%T121 =10h**	**%T153 = 90h**	**%T185 = 12h**
%T97 = 76h	**%T129 =23h**	**%T161 = 72h**	**%T193 = 78h**
%T105 = 46h	**%T137 =45h**	**%T169 = 41h**	**%T201 = 56h**

The master sends the Initial Request Mailbox message to the slave, with the service request code equal to Read System Memory (04h), along with Segment Selector, Data offset, and Data Length. After the slave successfully receives the mailbox message, the master sends the Text Buffer message containing the data to write. The Text Buffer message contains 16 bytes of data to be stored in %T81 through %T208. The slave responds with a Completion Ack Mailbox message which contains piggy-back status information.

MASTER	**SLAVE**
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message with Text Buffer] 1b 4d 00 00 00 00 00 00 98 80 10 3a 00 00 10 0a 00 00 00 01 10 00 00 00 00 00 01 01 07 14 0a 00 10 00 17 54 18 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
(wait **T1** time) 3: →[Request Text Buffer message] 1b 54 23 89 76 46 39 10 23 45 87 90 72 41 34 12 78 56 17 00 00 00 00 BCC	
	(wait **T1** time) 4: ←[ACK] 06 00 (wait **T1** time) 5: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 98 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 02 00 00 5c 20 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Write SMEM with Text Buffer: %T in Byte Mode

Table 6-8. Write with Text Buffer: %T in Byte Mode

Packet Number	Byte Numbers(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	98	Sequence number.
	10	80	Mailbox type: Initial Request with Text Buffer.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	00 01	Reserved; first byte zero, second byte one.
	21-22	10 00	Total length of data in bytes = 0010h = 16
	23-26	00 00 00 00	Reserved; must be set to zero.
	27-28	01 01	Packet number, total packets.
	29	07	Request code: Write System Memory.
	30	14	Segment Selector: %T data, byte mode.
	31-32	0a 00	Data offset: 000ah = byte offset for %T81.
	33-34	10 00	Data length: 0010h = 16 bytes.
	35	17	End of block character.
	36	54	Next message type.
	37-38	18 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: →	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-18	23 89 76 46 39 10 23 45 87 90 72 41 34 12 78 56	Byte data to store in %T81 - %T208.
	19	17	End of block character.
	20	00	Next message type.
	21-22	00 00	Next message length.
	23	00	Status byte.
	24	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	98	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	02	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Read Task Memory

The Read Task Memory (Read TMEM) request allows the master to access a single contiguous block of data from a PLC Program Task Memory Type. It is a Series 90-70 PLC only request, and should be used by the SNP user to access data within a Main Control Program Task Data selector; namely, %P memory.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	05h	Read Task Memory.
Segment Selector (byte 22):	04h	Segment Selector = 4 for %P data.
Data Offset (bytes 23-24):		Zero-based offset of data.
Data Length (bytes 25-26):		Length of data to read in words.
Program task name (bytes 27-34):		Main program task name (8-byte NUL terminated ASCII string).

Response Mailbox Message:

Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	94h	Completion ACK Mailbox message with Text Buffer.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) or Completion ACK Mailbox message with Text Buffer (94h) is returned to the master. The type of Mailbox message returned depends on the amount of data to be passed back to the master. If the number of data bytes being read is six bytes or less, the slave responds with a Completion ACK Mailbox message which contains the data read. If the number of data bytes being read is greater than six bytes, the slave responds with a Completion ACK Mailbox message with Text Buffer, and the data read is returned to the master in one or more Text Buffer messages.

Completion ACK Mailbox message (D4h):

Requested Data (bytes 23-28)

Completion ACK Mailbox message with Text Buffer (94h):

Total length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Next message type (byte 36):	54h	`T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the next Text Buffer message.

If the CPU is unable to comply with the request, it must reject the request. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message is:

- Insufficient privilege (Series 90-70 PLCs only: level 1 or higher).
- Invalid input parameter in the request mailbox message.

Response Text Buffer Message:

If the slave returns a Completion ACK Mailbox message with Text Buffer (94h), the main program task data returns to the master in one or more Text Buffer messages. The number of Text Buffer messages required depends on the amount of data being read and the maximum data size allowed in a SNP Data message.

An example with full explanation follows.

Example of Read Task Memory

This example reads the five words of %P data at %P39 through %P43 belonging to the main control program task "STAT_1". The current values of %P39 through %P43 are:

%P39 = 4A9Bh   %P40 = 3463h   %P41 = 1ACBh   %P42 = 91D4h   %P43 = 8BC3h

The example assumes that the master has attached to a Series 90-70 PLC slave device and has set the privilege level to one (1). The master sends the Initial Request Mailbox message to the slave, with the service request code for Read Task Memory (05h) along with the segment selector, data offset, data length, and the main control program task name. The slave responds with a Completion ACK Mailbox message containing the piggy-back status information, followed by a Text Buffer message containing the requested %P data..

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message with Text Buffer]	
1b 4d	
00 00 00 00 00 00 f1 c0 10 3a 00 00 10 0a 00 00	
01 01 05 04 26 00 05 00	
53 54 41 54 5f 31 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx f1 94 10 0a 00 00 10 3a 00 00
	xx 01 0a 00 xx xx xx xx 01 01
	xx xx xx xx xx xx
	17 54 12 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	
	(wait **T1** time)
	5: ←[Response Text Buffer message]
	1b 54
	9b 4a 63 34 cb 1a d4 91 c3 8b
	17 00 00 00 00 BCC
(wait **T1** time)	
6: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Read Task Memory

Table 6-9. Read Task Memory

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	f1	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	05	Request code: Read Program BLock Memory.
	22	04	Segment Selector: %P memory.
	23-24	26 00	Data offset: 26h = 38 = word offset for %P39.
	25-26	05 00	Data length: 5 words --> %P39 - %P43.
	27-34	53 54 41 54 5F 31 00 00	Main Program Task Name: "STAT_1".
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	f1	Sequence number.
	10	94	Mailbox type: Completion ACK with Text Buffer.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	xx xx	Reserved: don't care.
	21-22	0a 00	Total length of data in bytes = 00h = 10
	23-26	xx xx xx xx	Don't care.
	27-28	01 01	Packet number, total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	54	Next message type: `T' for Text Buffer
	37-38	12 00	Next message length in bytes: 0012h = 18
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-12	9b 4a 63 34 cb 1a d4 91 c3 8b	Requested data: %P39 - %P43.
	13	17	End of block character.
	14	00	Next message type.
	15-16	00 00	Next message length.
	17	00	Status byte.
	18	BCC	Block Check Code.

Write Task Memory

The Task Memory (Write TMEM) request allows the master to write to a single contiguous block of data from a PLC Program Task Type. It is a Series 90-70 PLC only request, and should be used by the SNP user to access data within a Main Control Program Task segment; namely, %P memory.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	80h	Initial Request Mailbox message with Text Buffer.
Total length of data (bytes 21-22):		The total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Service Request code (byte 29):	08h	Write Program Block Memory.
Segment Selector (byte 30):	04h	Segment Selector = 4 for %P data.
Data Offset (bytes 31-32):		Zero-based offset of data.
Data Length (bytes 33-34):		Length of data to write.
Next message type (byte 36):	54h	`T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the Text Buffer message.
Request Text Buffer Message:		
Program task name (bytes 1-8):		Main program task name (8-byte NUL terminated ASCII string).
Write data:		Data bytes to be written; number of bytes depends on how much %P data the user wishes to write.
Response Mailbox Message Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU is unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message is:

- Insufficient privilege (must be level 2 or higher).
- Invalid input parameter in the request.

An example with full explanation follows.

Example of Write Task Memory

This example writes five words of data to %P39 through %P43. The %P data in this example belongs to the Main Control program task "STAT_1". After the write service request, %P39 through %P43 are equal to:

%P39 = 4A9Bh %P40 = 3463h %P41 = 1ACBh %P42 = 91D4h %P43 = 8BC3h

This example assumes that the master has attached to a Series 90-70 PLC slave device and has set the privilege level to two (2). The master sends the Initial Request Mailbox message to the slave, with the service request code for Write Task Memory (08h) along with the segment selector, data offset, and data length. Once this Mailbox message is acknowledged by the slave, the master sends the Text Buffer message containing the main control program name, and the data to be written. The slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message with Text Buffer] 1b 4d 00 00 00 00 00 00 f2 80 10 3a 00 00 10 0a 00 00 00 01 12 00 00 00 00 00 01 01 08 04 26 00 05 00 17 54 1a 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
(wait **T1** time) 3: →[Request Text Buffer message] 1b 54 53 54 41 54 5f 31 00 00 9b 4a 63 34 cb 1a d4 91 c3 8b 17 00 00 00 00 BCC	
	(wait **T1** time) 4: ←[ACK] 06 00 (wait **T1** time) 5: ←[Completion ACK MB message] 1b 4d Xx xx xx xx xx xx f2 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 02 00 00 7c 31 17 00 00 00 00 BCC
(wait **T1** time) 4: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Write Task Memory

Table 6-10. Write Task Memory

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	f2	Sequence number.
	10	80	Mailbox type: Initial Request with Text Buffer.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	00 01	Reserved; first byte zero, second byte one.
	21-22	12 00	Total length of data in bytes = 0012h = 18
	23-26	00 00 00 00	Reserved; must be set to zero.
	27-28	01 01	Packet number, total packets.
	29	08	Request code: Write Task Memory.
	30	04	Segment Selector: %P memory.
	31-32	26 00	Data offset: 26h = 38 = word offset for %P39.
	33-34	05 00	Data length: 5 words.
	35	17	End of block character.
	36	54	Next message type: `T' for Text Buffer.
	37-38	1a 00	Next message length in bytes: 001ah = 26.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: →	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-10	53 54 41 54 5f 31 00 00	Control Program name: "STAT_1"
	11-20	9b 4a 63 34 cb 1a d4 91 c3 8b	Data to write to %P39 - %P43
	21	17	End of block character.
	22	00	Next message type.
	23-24	00 00	Next message length.
	25	00	Status byte.
	26	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	f2	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	02	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Read Program Block Memory

The Read Program Block Memory (Read PBMEM) request allows the master to access a single contiguous block of data from a PLC Program Block Memory Type. It is a Series 90-70 PLC only request, and should be used by the SNP user to access data within a Local Subblock Data segment; namely, %L memory.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	80h	Initial Request Mailbox message with Text Buffer.
Total Length of Data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Service Request code (byte 29):	06h	Read Program Block Memory.
Segment Selector (byte 30):	00h	Segment Selector = 0 for %L data.
Data Offset (bytes 31-32):		Zero-based offset of data.
Data Length (bytes 33-34):		Length of data to read in words.
Next message type (byte 36):	54h	'T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the next Text Buffer message.
Request Text Buffer Message:		
Program task name:		Main program task name (8-byte NUL terminated ASCII string).
Program block name:		Name of the program block to which the %L data belongs; (8-byte NUL terminated ASCII string).
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	94h	Completion ACK Mailbox message with Text Buffer.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) or Completion ACK Mailbox message with Text Buffer (94h) is returned to the master. The type of Mailbox message returned depends on the amount of data to be passed back to the master. If the number of data bytes being read is six bytes or less, the slave responds with a Completion ACK Mailbox message which contains the data read. If the number of data bytes being read is greater than six bytes, the slave responds with a Completion ACK Mailbox message with Text Buffer, and the data read is returned to the master in one or more Text Buffer messages.

Completion ACK Mailbox message (D4h):

Requested Data (bytes 23-28)

Completion ACK Mailbox message with Text Buffer (94h):

Total length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Next message type (byte 36):	54h	\`T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the next Text Buffer message.

If the CPU is unable to comply with the request, it must reject the request. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message is:

- Insufficient privilege (Series 90-70 PLCs only: level 1 or higher).
- Invalid input parameter in the request mailbox message.

Response Text Buffer Message:

If the slave returns a Completion ACK Mailbox message with Text Buffer (94h), the program block data returns to the master in one or more Text Buffer messages. The number of Text Buffer messages required depends on the amount of data being read and the maximum data size allowed in a SNP Data message.

An example with full explanation follows.

Example of Read Program Block Memory

This example reads the two words of %L data at %L9 and %L10 belonging to the program block "SUB1". The main control program name is "STAT_1". The current values of %L9 and %L10 are:

`%L9 = 87ACh` **`%L10 = 4BCDh`**

The example assumes that the master has attached to a Series 90-70 PLC slave device and has set the privilege level to one (1). The master sends the Initial Request Mailbox message with Text Buffer to the slave, with the service request code for Read Program Block Memory (06h) along with the segment selector, data offset, and data length. Once this Mailbox message is acknowledged by the slave, the master sends the Text Buffer message containing the main control program name and the program block name. The slave responds with a Completion ACK Mailbox message containing the requested %L data and the piggy-back status information.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message with Text Buffer] 1b 4d 00 00 00 00 00 00 f1 80 10 3a 00 00 10 0a 00 00 00 01 10 00 00 00 00 00 01 01 06 00 08 00 02 00 17 54 18 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
(wait **T1** time) 3: →[Request Text Buffer message] 1b 54 53 54 41 54 5f 31 00 00 53 55 42 31 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 4: ←[ACK] 06 00 (wait **T1** time) 5: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx f1 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx ac 87 cd 4b xx xx 00 01 00 00 7c 31 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Read Program Block Memory

Table 6-11. Read Program Block Memory

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	f1	Sequence number.
	10	80	Mailbox type: Initial Request with Text Buffer.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	00 01	Reserved; first byte zero, second byte one.
	21-22	10 00	Total length of data in bytes = 0010h = 16
	23-26	00 00 00 00	Reserved; must be set to zero.
	27-28	01 01	Packet number, total packets.
	29	06	Request code: Read Program BLock Memory.
	30	00	Segment Selector: %L memory.
	31-32	08 00	Data offset: 8 = word offset for %L9.
	33-34	02 00	Data length: 2 words --> %L9 - %L10.
	35	17	End of block character.
	36	54	Next message type: `T' for Text Buffer.
	37-38	18 00	Next message length in bytes: 0018h = 24
	39	00	Status byte.
	40	BCC	Block Check Code.
3:→	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-10	53 54 41 54 5f 31 00 00	Control Program name: "STAT_1"
	11-18	53 55 42 31 00 00 00 00	Program Block name: "SUB1"
	19	17	End of block character.
	20	00	Next message type.
	21-22	00 00	Next message length.
	23	00	Status byte.
	24	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	f1	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22	xx	Don't care.
	23-24	ac 87	%L9 data.
	25-26	cd 4b	%L10 data.
	27-28	xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Write Program Block Memory

The Write Program Block Memory (Write PBMEM) request allows the master to write to a single contiguous block of data from a PLC Program Block Memory Type. It is a Series 90-70 PLC only request, and should be used by the SNP user to access data within a Local Subblock Data segment; namely, %L memory.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	80h	Initial Request Mailbox message with Text Buffer.
Total length of data (bytes 21-22):		The total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Service Request code (byte 29):	09h	Write Program Block Memory.
Segment Selector (byte 30):	00h	Segment Selector = 0 for %L data.
Data Offset (bytes 31-32):		Zero-based offset of data.
Data Length (bytes 33-34):		Length of data to write.
Next message type (byte 36):	54h	`T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the Text Buffer message.
Request Text Buffer Message:		
Program task name:		Main program task name (8-byte NUL terminated ASCII string).
Program block name:		Name of the program block to which the %L data belongs; (8-byte NUL terminated ASCII string).
Write data:		Data bytes to be written; number of bytes depends on how much %L data the user wishes to write.
Response Mailbox Message Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU is unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message is:

- Insufficient privilege (must be level 2 or higher).
- Invalid input parameter in the request.

An example with full explanation follows.

Example of Write Program Block Memory

This example writes one word of data to %L7. The %L data in this example belongs to the program block "SUB1". The main control program name is "STAT_1". The word of data written to %L7 is 84A2h. The example assumes that the master has attached to a Series 90-70 PLC slave device and has set the privilege level to two (2).

The master sends the Initial Request Mailbox message to the slave, with the service request code for Write Program Block Memory (09h) along with the segment selector, data offset, and data length. Once this Mailbox message is acknowledged by the slave, the master sends the Text Buffer message containing the main control program name, the program block name, and the data to be written. The slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message with Text Buffer] 1b 4d 00 00 00 00 00 00 f2 80 10 3a 00 00 10 0a 00 00 00 01 12 00 00 00 00 00 01 01 09 00 06 00 01 00 17 54 1a 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
(wait **T1** time) 3: →[Request Text Buffer message] 1b 54 53 54 41 54 5f 31 00 00 53 55 42 31 00 00 00 00 a2 84 17 00 00 00 00 BCC	
	(wait **T1** time) 4: ←[ACK] 06 00 (wait **T1** time) 5: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx f2 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 02 00 00 7c 31 17 00 00 00 00 BCC
(wait **T1** time) 4: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Write Program Block Memory

Table 6-12. Write Program Block Memory

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	f2	Sequence number.
	10	80	Mailbox type: Initial Request with Text Buffer.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	00 01	Reserved; first byte zero, second byte one.
	21-22	12 00	Total length of data in bytes = 0012h = 18
	23-26	00 00 00 00	Reserved; must be set to zero.
	27-28	01 01	Packet number, total packets.
	29	09	Request code: Write Program BLock Memory.
	30	00	Segment Selector: %L memory.
	31-32	06 00	Data offset: 6 = word offset for %L7.
	33-34	01 00	Data length: 1 word.
	35	17	End of block character.
	36	54	Next message type: `T' for Text Buffer.
	37-38	1a 00	Next message length in bytes: 001ah = 26
	39	00	Status byte.
	40	BCC	Block Check Code.
3: →	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-10	53 54 41 54 5f 31 00 00	Control Program name: "STAT_1"
	11-18	53 55 42 31 00 00 00 00	Program Block name: "SUB1"
	19-20	a2 84	Data to write to %L7 --> = 84a2h
	21	17	End of block character.
	22	00	Next message type.
	23-24	00 00	Next message length.
	25	00	Status byte.
	26	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	f2	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	02	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

PLC Short Status Request

The PLC Short Status request from the master to the PLC CPU requests the status information that is piggy-backed onto all mailbox responses, plus some additional status information on control programs, logins, and the programmer window.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	00h	Short Status request.
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
Number of Control Programs (byte 23):	0	No program tasks currently defined.
	1	One program task defined for the PLC CPU.
Programmer Flags (byte 24):		Set of boolean flags indicating which control program tasks have programmers currently attached to them. Each bit position corresponds to a control program task (0 = no programmer attached; 1 = programmer attached), where bit 0 corresponds to control program 0. Since at most one control program is currently allowed, this byte is either zero or one.
Login Types (bytes 25-26):		Set of 2-bit codes indicating the type of login associated with each control program. The two-bit field beginning at bit 0 corresponds to control program 0. Since there can be at most one control program in the current implementation, only bit 0 and bit 1 have meaning. See table below for definition. (Series 90-70 PLC only)
Auxiliary Status (byte 27):		Only the highest bit is defined, and only for the Series 90-70. Bit 7 ==> Programmer window status: -= 0 = Programmer window closed. -= 1 = Programmer window open.

Table 6-13. Two-bit Login Type Codes

Value	Description
03h	Parallel Work Station Interface (WSI) attached.
02h	Serial device attached at PLC CPU.
01h	Non-dedicated programmer attached.
00h	No programmer attached.

An example with full explanation follows.

Example of PLC Short Status Request

The following example issues a Short Status Request to the PLC CPU. The example assumes that the master has already attached to a Series 90-30 PLC slave device. The master sends the Initial Request Mailbox message to the slave, with the service request code for Short Status request (00h), and the slave responds with a Completion ACK Mailbox message which includes the piggy-back status information and other status information mentioned above.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: → [Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 6c c0 10 3a 00 00 10 0a 00 00	
01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 6c d4 10 0a 00 00 10 3a 00 00
	01 01 00 00 01 01 xx xx xx xx 00 02 00 00 54 20
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Short Status Request

Table 6-14. Short Status Request

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6c	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Request code: Short Status request.
	22-34	00 00 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6c	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Major status code: 0 = okay.
	22	00	Minor status code: 0 = okay.
	23	01	Number of control programs: one.
	24	01	Programmer flags: 1 = programmer attached.
	25-28	xx xx xx xx	Don't care.
	29	00	Control program number.
	30	02	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	54 20	PLC status word: 2054 = -Stop I/O disabled. -Programmer attached. -PLC fault since last clear. -PLC fault since last read.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Return Control Program Name(s)

The Return Control Program Name(s) request returns the number of control program tasks in the PLC CPU, and the name of each program task. For Series 90-30 CPUs, there can be at most one program name. Currently, Series 90-70 PLCs also support only one program name. Future enhancements to Series 90-70 PLCs may allow for more than one control program, but with the existing implementation, only one is allowed.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	03h	Return Control Program Name(s).
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message (Series 90-70 PLCs only).
	94h	Completion ACK Mailbox message with Text Buffer.
Total length of data (bytes 21-22):		Number of **data bytes** in all subsequent Text Buffer messages (only applicable with MB type 94h).
Next message type (byte 36):	54h	`T' for Text Buffer (only applicable with MB type 94h).
Next message length (bytes 37-38):		Number of bytes in the next Text Buffer message (only applicable with MB type 94h).

A Series 90-30 PLC CPU always has one and only one control program. Therefore, a Completion ACK Mailbox message with Text Buffer is always the response. Within the Text Buffer message, the `Number of Control Programs' field is always one (1), followed by the Program Name (ASCII Null string if no program has been entered).

Such is not the case on Series 90-70 PLCs: there may be no control program task, in which case, a Completion ACK Mailbox message with no Text Buffer is the response; otherwise, if there are one or more (maximum of 8) control program tasks, a Completion ACK Mailbox message with Text Buffer is returned with the necessary information.

Response Text Buffer Message:

Number of Control Programs (bytes 3-4).

Program Name (bytes 5-12 ==> max 8 byte name).

An example with full explanation follows.

Example of Return Control Program Name

This example issues a Return Control Program Name request to a slave device. The slave device is a Series 90-30 331 CPU, and has one program named "ESS331". The example assumes that the master has already attached to the slave device. The master sends the Initial Request Mailbox message to the slave, with the service request code equal to Return Control Program Name (03h), and the slave responds with a Completion ACK Mailbox message which includes piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits the Text Buffer message. The Text Buffer message has 10 bytes of data: 2 bytes = number of control programs, and 8 bytes for the program name.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 66 c0 10 3a 00 00 10 0a 00 00 01 01 03 00 00 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00 (wait **T1** time) 3: ←[Completion ACK MB message with Text Buffer] 1b 4d xx xx xx xx xx xx 66 94 10 0a 00 00 10 3a 00 00 xx 01 0a 00 xx xx xx xx 01 01 00 01 00 00 4c 20 17 54 12 00 00 BCC
(wait **T1** time) 4: →> [ACK] 06 00\	
	(wait **T1** time) 5: ←[Response Text Buffer message] 1b 54 01 00 45 53 53 33 33 31 00 00 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Return Control Program Name

Table 6-15. Return Control Program Name

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	66	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	03	Request code: Return Control Program Name.
	22-34	00 00 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	66	Sequence number.
	10	94	Mailbox type: Completion ACK with Text Buffer.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	xx	Reserved: don't care.
	20	01	
	21-22	0a 00	Total length of data in bytes.
	23-26	xx xx xx xx	Don't care.
	27	01	Packet number.
	28	01	Total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	4c 20	PLC status word.
	35	17	End of block character.
	36	54	Next message type = `T' for Text Buffer.
	37-38	12 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-4	01 00	Number of programs: 0001
	5-12	45 53 53 33 33 31 00 00	Program name: "ESS331"
	13	17	End of block character.
	14	00	Next message type.
	15-16	00 00	Next message length.
	17	00	Status byte.
	18	BCC	Block Check Code.

Return Controller Type and ID Information

The PLC Return Controller Type and ID Information Request allows the master to obtain information that may be necessary to the logon sequence, such as the PLC CPU controller ID and the Major and Minor type of the PLC CPU.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	43h	Return Controller Type and ID information.

Response Mailbox Message:

Mailbox Type (byte 10):	94h	Completion ACK Mailbox message with Text Buffer.
Total length of data (byte 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Next message type (byte 36):	54h	`T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the next Text Buffer message.

Response Text Buffer Message:

CPU controller ID	Bytes 3-10
PLC Series 90 Major Type	Byte 11 (see table below)
PLC Series 90 Minor Type	Byte 12 (see table below)
Number of Control programs	Byte 13
Reserved (spare) byte	Byte 14
Main Control Program Name	Bytes 15-22
Number of Program Blocks	Bytes 23-24
Total Length of Program Blocks	Bytes 25-28
Sum of Program blocks Additive checksums	Bytes 29-30
Sum of Program blocks CRC checksums	Bytes 31-34
Length of Configuration records	Byte 35-36
Sum of Configuration records Additive checksums	Bytes 37-38
Sum of Configuration records CRC checksums	Bytes 39-42

Table 6-16. PLC Major/Minor Code Types

Dec	Hex	Description	Dec	Hex	Description
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

Table 6-16. PLC Major/Minor Code Types – Continued

Dec	Hex	Description	Dec	Hex	Description
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

An example with full explanation follows.

Example of Return Controller Type and ID Information

The following assumes that the master has already attached to the slave device. The slave device is a Series 90-30 331 CPU with a controller ID of "33101A" and contains a program named "ESS331". The master sends the Initial Request Mailbox message to the slave, with the service request code for Return Controller TYPE and ID information (43h), and the slave responds with a Completion ACK Mailbox message which includes piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits the Text Buffer message which contains the controller Type and ID information.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 65 c0 10 3a 00 00 10 0a 00 00 01 01 43 00 00 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
	(wait **T1** time) 3: ←[Completion ACK MB message with Text Buffer] 1b 4d xx xx xx xx xx xx 65 94 10 0a 00 00 10 3a 00 00 xx 01 28 00 xx xx xx xx 01 01 00 01 00 00 4c 20 17 54 30 00 00 BCC
(wait **T1** time) 4: →[ACK] 06 00	
	(wait **T1** time) 5: ←[Response Text Buffer message] 1b 54 33 33 31 30 31 41 00 00 10 23 01 00 45 53 53 33 33 31 00 00 01 00 a3 07 00 00 9a 01 9b cd 00 00 d6 06 22 01 c8 eb 00 00 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Return Controller Type and ID

Table 6-17. Return Control Type and ID

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	65	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	43	Request code: Return control Type and ID.
	22-34	00 00 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	65	Sequence number.
	10	94	Mailbox type: Completion ACK with Text Buffer.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	xx	Reserved: don't care.
	20	01	
	21-22	28 00	Total length of data in bytes.
	23-26	xx xx xx xx	Don't care.
	27	01	Packet number.
	28	01	Total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	4c 20	PLC status word.
	35	17	End of block character.
	36	54	Next message type = `T' for Text Buffer.
	37-38	30 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Table 6-17. Return Control Type and ID - continued

Packet Number	Byte Number(s)	Hex Value	Description
5: ←	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-10	33 33 31 30 31 41 00 00	PLC CPU controller ID: "33101A"
	11	10	PLC CPU Major Type: 10 hex = Series 90-30 PLC
	12	23	PLC CPU Minor Type: 23 hex = 331 CPU
	13	01	Number of programs: 1
	14	00	Reserved; don't care.
	15-22	45 53 53 33 33 31 00 00	Program name: "ESS331"
	23-24	01 00	Number of programs: 0001
	25-28	a3 07 00 00	Total length of program blocks: 07a3h
	29-30	9a 01	Sum of Program blocks additive checksums. 019ah = Sum of program block additive checksums.
	31-34	9b cd 00 00	Sum of Program blocks CRC checksums. 0cd9bh = Sum of program block CRC checksums.
	35-36	d6 06	Length of Configuration records: 6d6h
	37-38	22 01	Sum of Configuration records Additive checksums. 0122h = Sum of configuration additive checksums.
	39-42	c8 eb 00 00	Sum of Configuration records CRC checksums. 0ebc8 = Sum of configuration CRC checksums.
	43	17	End of block character.
	44	00	Next message type.
	45-46	00 00	Next message length.
	47	00	Status byte.
	48	BCC	Block Check Code.

Return PLC Time/Date

The Return PLC Time request enables the master to obtain the current time and date stored in the PLC CPU.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	25h	Return PLC Time/Date.
Response Mailbox Message:		
Mailbox Type (byte 10):	94h	Completion ACK Mailbox message with Text Buffer.
Total length of data (bytes 21-22):		Number of **data bytes** in all subsequent Text Buffer messages.
Next message type (byte 36):	54h	`T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the next Text Buffer message.

Response Text Buffer Message:

Seconds (byte 3)	Current seconds in packed BCD format.
Minutes (byte 4)	Current minutes in packed BCD format.
Hour (byte 5)	Current hour in packed BCD format.
Day (byte 6)	Current day of month in packed BCD format.
Month (byte 7)	Current month in packed BCD format.
Year (byte 8)	Current year in packed BCD format.
Day of Week (byte 9)	Day of week; valid range 1 - 7; where Sunday = 1, ..., Saturday = 7.

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

An example with full explanation follows.

Example of Return PLC Time/Date

The following example assumes that the master has already attached to the slave device, and has set the privilege level to one (1). The current time in the PLC CPU is 10:48:59, and the date is Friday May 4, 1990. The master sends the Initial Request Mailbox message to the slave, with the service request code for Return PLC Time/Date (25h), and the slave responds with a Completion ACK Mailbox message which includes piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits the Text Buffer message containing the PLC CPU's current time and date.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 1f c0 10 3a 00 00 10 0a 00 00 01 01 25 00 00 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: →[ACK] 06 00 (wait **T1** time) 3: ←[Completion ACK MB message with Text Buffer] 1b 4d xx xx xx xx xx xx 1f 94 10 0a 00 00 10 3a 00 00 xx 01 08 00 xx xx xx xx 01 01 00 01 00 00 4c 20 17 54 10 00 00 BCC
(wait **T1** time) 4: →[ACK] 06 00	
	(wait **T1** time) 5: ←[Response Text Buffer message] 1b 54 59 48 10 04 05 90 06 xx 17 00 00 00 00 BCC
(wait **T1** time) 6: → [ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Return PLC Time/Date

Table 6-18. Return PLC Time/Date

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	1f	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	25	Request code: Return PLC Time/Date.
	22-34	00 00 00 00 00 00	Not used.
		00 00 00 00 00 00 00	
	35	17	End of block character.
	36	00	Next Message type.
	37-38	00 00	Next Message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	1f	Sequence number.
	10	94	Mailbox type: Completion ACK with Text Buffer.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	xx	Reserved: don't care.
	20	01	
	21-22	08 00	Total length of data in bytes.
	23-26	xx xx xx xx	Don't care.
	27-28	01 01	Packet number, total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	4c 20	PLC status word.
	35	17	End of block character.
	36	54	Next Message type.
	37-38	10 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-5	59 48 10	Time: 10:48:59
	6-8	04 05 90	Date: May 4, 1990
	9	06	Day of week: Friday
	10	xx	Spare
	11	17	End of block character.
	12	00	Next message type.
	13-14	00 00	Next message length.
	15	00	Status byte.
	16	BCC	Block Check Code.

Return Fault Table

The Return Fault Table request returns fault data from either the PLC Fault Table or the I/O Fault Table to the master. The Fault Table Type is specified in the Initial Request Mailbox message. The PLC fault table has a maximum of 16 faults and the I/O fault table has a maximum of 32 faults.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	38h	Return Fault Table.
Fault Table Type (byte 22):	1	I/O Fault Table.
	2	PLC Fault Table.
Fault Table Index (bytes 23-24):	0 through 15	Index into PLC Fault table.
	0 through 31	Index into I/O Fault table.
Number of Faults to Return (bytes 25-26):		Number of fault entries the user wants to be returned within this service request.

Response Mailbox Message:

Mailbox Type (byte 10):	94h	Completion ACK Mailbox message with Text Buffer.
	D1h	Error Nack mailbox message.
Total length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Text Buffer messages.
Next message type (byte 36):	54h	`T' for Text Buffer.
Next message length (bytes 37-38):		Number of bytes in the next Text Buffer message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response to a Return Fault Table request are:

- D-Insufficient privilege (Series 90-70 PLC: must be level 1 or higher).
- D-Invalid Fault Table Type (1 = I/O, 2 = PLC, all others invalid).

Response Text Buffer Message:

The first twelve (12) bytes of the response Text Buffer message is a fault header, which has the following format:

Time Stamp of last clear	Bytes 3-8
Faults since last clear	Bytes 9-10
Faults in the table	Bytes 11-12
Faults entries in this response	Bytes 13-14

The remainder of the Text Buffer message(s) is the actual fault data for each fault requested. Every fault takes 42 bytes. Therefore, where `N' is the number of fault entries returned in the response, the total size of fault data returned from this request is as follows: 12 (fault header bytes) + (42 * N).

The format of the 42 bytes differs between the two tables. Picking up the byte numbering after the header, we have the following two formats:

Table 6-19. Format for the PLC and I/O Faults

PLC Fault (42 bytes)		I/O Fault (42 bytes)	
		Spare byte	Byte 15
Spare bytes	Bytes 15-18	Reference address	Bytes 16-18
Rack number	Byte 19	Rack number	Byte 19
Slot number	Byte 20	Slot number	Byte 20
Unit number	Byte 21	I/O bus number	Byte 21
Spare byte	Byte 22	Bus addresses	Byte 22
Fault group	Byte 23	Point address	Word 23-24
Fault action	Byte 24	Fault group	Byte 25
Fault error code	Bytes 25-26	Fault action	Byte 26
Spare bytes	Bytes 27-50	Fault category	Byte 27
Time error logged	Bytes 51-56	Fault type	Byte 28
		Fault description	Byte 29
		Spare bytes	Bytes 30-50
		Time error logged	Bytes 51-56

Groups of 42 bytes will continue in the above manner until all requested faults have been returned. If more than one text buffer is required to return all fault data, please take note that the fault header is only sent in the first text buffer; all subsequent text buffers contain 42-byte fault data entries only.

The following two tables define the fault group and fault action fields.

Table 6-20. Fault Action Codes

Fault	Description
1	Informational
2	Diagnostic
3	Fatal

Table 6-21. Fault Group Codes

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

Examples with full explanations follows.

Example of Return PLC Fault Table

This example retrieves all faults in the PLC fault table. The example assumes that the master has already attached to the slave device, and has changed the privilege level to one (1). The master sends the Initial Request Mailbox message to the slave, with the service request code equal to Return Fault Table (38h), and the Fault Table Type field specified as PLC fault table. The slave responds with a Completion ACK Mailbox message which includes piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits the Text Buffer message containing the PLC fault table data.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 6e c0 10 3a 00 00 10 0a 00 00 01 01 38 02 00 00 10 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
	(wait **T1** time) 3: ←[Completion ACK MB message with Text Buffer] 1b 4d xx xx xx xx xx xx 6e 94 10 0a 00 00 10 3a 00 00 xx 01 60 00 xx xx xx xx 01 01 00 01 00 00 58 20 17 54 68 00 00 BCC
(wait **T1** time) 4: → [ACK] 06 00	
	(wait **T1** time) 5: ←[Response Text Buffer message] 1b 54 37 19 11 07 06 90 02 00 02 00 02 00 xx xx xx xx 00 01 00 03 12 02 00 00 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx 53 19 11 07 06 90 xx xx xx xx 00 01 1d xx 84 02 00 00 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx 36 20 11 07 06 90 17 00 00 00 00 BCC
(wait **T1** time) 6: → [ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Return PLC Fault Table

Table 6-22. Return PLC Fault Table

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6e	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	38	Request code: Return Fault Table.
	22	02	Fault Table Type: in this case, PLC.
	23-24	00 00	Fault table index: start with first fault.
	25-26	10 00	Number of Entries to Return: in this case, entire table = 0010h = 16.
	27-34	00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6e	Sequence number.
	10	94	Mailbox type: Completion ACK with Text Buffer.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	xx	Reserved: don't care.
	20	01	Number of destinations: always one.
	21-22	60 00	Total length of data in bytes = 0060h = 96.
	23-26	xx xx xx xx	Don't care.
	27	01	Packet number.
	28	01	Total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	58 20	PLC status word.
	35	17	End of block character.
	36	54	Next message type = `T' for Text Buffer.
	37-38	68 00	Next message length in bytes = 0068h = 104
	39	00	Status byte.
	40	BCC	Block Check Code.

Table 6-22. Return PLC Fault Table - Continued

Packet Number	Byte Number(s)	Hex Value	Description
5: ←	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-8	37 19 11 07 06 90	Time Stamp for last time PLC fault table was cleared: 11:19:37, on Jun 7, 1990.
	9-10	02 00	Faults since last clear: 0002.
	11-12	02 00	Total faults in table: 0002.
	13-14	02 00	Faults returned in this response: 0002.
			Start of Fault Entry Number 1:
	15-18	xx xx xx xx	-Spare bytes: don't care.
	19	00	-Rack number.
	20	01	-Slot number.
	21	00	-Unit number (task # reporting the error).
	22	xx	-Don't care.
	23	12	-Fault Group: Low Battery.
	24	02	-Fault Action: Diagnostic.
	25-26	00 00	-Fault Error Code.
	27-50	xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx	-Don't care.
	51-56	53 19 11 07 06 90	-Time error logged: 11:19:53, Jun 7, 1990.
			Start of Fault Entry Number 2:
	57-60	xx xx xx xx	-Spare bytes: don't care.
	61	00	-Rack number.
	62	01	-Slot number.
	63	1d	-Unit number.
	64	xx	-Don't care.
	65	84	-Fault Group: Password Access Failure.
	66	02	-Fault Action: Diagnostic.
	67-68	00 00	-Fault Error Code.
	69-92	xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx	-Don't care.
	93-98	36 20 11 07 06 90	-Time error logged: 11:20:36, Jun 7, 1990.
	99	17	End of block character.
	100	00	Next message type.
	101-102	00 00	Next message length.
	103	00	Status byte.
	104	BCC	Block Check Code.

Example of Return I/O Fault Table

This example retrieves all faults in the I/O fault table. The example assumes that the master has already attached to the slave device, and has changed the privilege level to one (1). The master sends the Initial Request Mailbox message to the slave, with the service request code equal to Return Fault Table (38h), and the Fault Table Type field specified as I/O fault table. The slave responds with a Completion ACK Mailbox message which includes piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits the Text Buffer message containing the PLC fault table data.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 1c c0 10 3a 00 00 10 0a 00 00 01 01 38 01 00 00 20 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
	(wait **T1** time) 3: ←[Completion ACK MB message with Text Buffer] 1b 4d xx xx xx xx xx xx 1c 94 10 0a 00 00 10 3a 00 00 xx 01 8a 00 xx xx xx xx 01 01 00 01 00 00 74 20 17 54 92 00 00 BCC
(wait **T1** time) 4: →[ACK] 06 00	
	(wait **T1** time) 5: ←[Response Text Buffer message] 1b 54 53 20 12 07 06 90 03 00 03 00 03 00 xx ff 00 00 00 03 7f 7f ff 7f 03 02 0e 00 00 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx 21 55 12 07 06 90 xx ff 00 00 00 06 7f 7f ff 7f 07 02 0f 00 00 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx 21 55 12 07 06 90 xx ff 00 00 01 03 7f 7f ff 7f 03 02 0e 00 00 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx 21 55 12 07 06 90 17 00 00 00 00 06
(wait **T1** time) 6: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Return I/O Fault Table

Table 6-23. Return I/O Fault Table

Packet Number	Byte Number(s)	Hex Value	Description
1:→	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	1c	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source.
	15-18	10 0a 00 00	Mailbox destination.
	19	01	Packet number.
	20	01	Total Packets.
	21	38	Request code: Return Fault Table.
	22	01	Fault Table Type: in this case, IO.
	23-24	00 00	Fault table index: start with first fault.
	25-26	20 00	No. of Entries to Return: in this case, entire table = 0020h = 32.
	27-34	00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	1c	Sequence number.
	10	94	Mailbox type: Completion ACK with Text Buffer.
	11-14	10 0a 00 00	Mailbox source.
	15-18	10 3a 00 00	Mailbox destination.
	19	xx	Reserved; don't care.
	20	01	Number of destinations: always one.
	21-22	8a 00	Total length of data in bytes = 008ah = 138.
	23-26	xx xx xx xx	Don't care.
	27	01	Packet number.
	28	01	Total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	74 30	PLC status word.
	35	17	End of block character.
	36	54	Next message type = `T' for Text Buffer.
	37-38	92 00	Next message length in bytes = 0092h = 146.
	39	00	Status byte.
	40	BCC	Block Check Code.

Table 6-23. Return I/O Fault Table - continued

Packet Number	Byte Number(s)	Hex Value	Description
5:←	1	1b	Start of message character.
	2	54	Message type = `T' for Text Buffer.
	3-8	53 20 12 07 06 90	Time Stamp for last time I/O fault. table was cleared: 12:20:53, on Jun 7, 1990.
	9-10	03 00	Faults since last clear: 0003.
	11-12	03 00	Total faults in table: 0003.
	13-14	03 00	Faults returned in this response: 0003.
			Start of Fault Entry Number 1:
	15	xx	-Spare bytes: don't care.
	16-18	ff 00 00	-Reference address.
	19	00	-Rack number.
	20	03	-Slot number.
	21	7f	-I/O Bus Number.
	22	7f	-Bus address.
	23-24	ff 7f	-Point address.
	25	03	-Fault Group: Loss of or Missing I/O Module.
	26	02	-Fault Action: Diagnostic.
	27	0e	-Fault Category.
	28	00	-Fault Type.
	29	00	-Fault description.
	30-50	xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx	-Don't care.
	51-56	21 55 12 07 06 90	-Time error logged: 12:55:21, Jun 7, 1990.
			Start of Fault Entry Number 2:
	57	xx	-Spare bytes: don't care.
	58-60	ff 00 00	-Reference address.
	61	00	-Rack number.
	62	06	-Slot number.
	63	7f	-I/O Bus Number.
	64	7f	-Bus address.
	65-66	ff 7f	-Point address.
	67	07	-Fault Group: Addition of or Extra I/O module.
	68	02	-Fault Action: Diagnostic.
	69	0f	-Fault Category.
	70	00	-Fault Type.
	71	00	-Fault description.
	72-92	xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx	-Don't care.
	93-98	21 55 12 07 06 90	-Time error logged: 12:55:21, Jun 7, 1990.
			Start of Fault Entry Number 3:
	99	xx	-Spare bytes: don't care.
	100-102	ff 00 00	-Reference address.
	103	01	-Rack number.
	104	03	-Slot number.
	105	7f	-I/O Bus Number.
	106	7f	-Bus address.
	107-108	ff 7f	-Point address.
	109	03	-Fault Group: Loss of or Missing I/O module.
	110	02	-Fault Action: Diagnostic.
	111	0e	-Fault Category.
	112	00	-Fault Type.
	113	00	-Fault description.
	114-134	xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx	-Don't care.
	135-140	21 55 12 07 06 90	-Time error logged: 12:55:21, Jun 7, 1990.
	141	17	End of block character.
	142	00	Next message type.
	143-144	00 00	Next message length.
	145	00	Status byte.
	146	BCC	Block Check Code.

Clear Fault Table

The request Clear Fault Table enables the master to clear the fault table information. There are two fault tables: PLC fault table and I/O fault table. The Fault Table Type to be cleared is specified in a parameter field within the Initial Request Mailbox message.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	39h	Clear Fault Table.
Fault Table Type (byte 22):	1	I/O Fault Table.
	2	PLC Fault Table.
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response to a Clear Fault Table are:

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

An example with full explanation follows.

Example of Clear PLC Fault Table

This example clears the PLC fault table. The example assumes that the master has already attached to the slave device, has set the privilege level to two (2), and has logged in as a programmer attachment. The master sends the Initial Request Mailbox message to the slave, with the service request code for Clear Fault Table (39h) and the Fault Table type set to indicate the PLC fault table. The slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →Initial Request MB message] 1b 4d 00 00 00 00 00 00 f7 c0 10 3a 00 00 10 0a 00 00 01 01 39 02 00 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK] 06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx f7 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 04 00 00 5c 20 17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Clear PLC Fault Table

Table 6-24. Clear PLC Fault Table

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	f7	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	39	Request code: Clear Fault Table.
	22	02	Fault table to clear: in this case, PLC.
	23-34	00 00 00 00 00 00	Not used.
		00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	f7	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	04	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Programmer Logon

The PLC Programmer Logon Request allows a device to **login** and **logout** as a programmer attachment. Some service requests require that the master be logged in as a programmer attachment, such as Set Control ID and Set PLC State.

When the master attaches to a Series 90-70 PLC CPU, the master is automatically logged in to the Null task. Therefore, in order to issue a Set Control ID or Set PLC State service request, no explicit logon request is required. If, however, the master wishes to Load and Store Program Blocks, the master must issue a logon request with a valid program task name before making a Load or Store Program Block service request.

There is no automatic logon when the master attaches to a Series 90-30 PLC CPU. Therefore, the master must make an explicit logon request (either to the Null task or program task) before any service request requiring programmer attachment login can be made.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	20h	Logon/Logoff request code.
Log mode (byte 22):	2	Login Serial Mode.
	3	Logout mode (used to logout from a program task).
	4	Logout bye mode (used when connection is going to be broken).
Program Task Name (bytes 23-30):		8-byte NUL-terminated ASCII string or 8 NULs if logon to Null task.
Block Transfer Buffer Size (bytes 31-32):		**Series 90-70 PLC only:** Used to specify the maximum communication buffer size (in bytes) during program load and store that the master SNP device can support. Must contain a non-zero, positive number.

On Series 90-30 PLCs, there is no difference between "logout mode" and "logout bye mode", so you can use either value when logging out of the programmer window.

On Series 90-70 PLCs, you should use "logout mode" to log out of a given control program task. "Logout bye mode" is used if it is known that the communication session is going to be broken (i.e. a long break* is going to be issued), in order to do housekeeping within Series 90-70 PLCs.)

The Block Transfer Buffer Size is a Series 90-70 PLC field only. The GE Fanuc LM90/WSIB programmer implementation uses a communications buffer size of 16K in its logon request. Since the current implementation of the Series 90-70 PLC CPU family allows a maximum 8K program Block Transfer buffer, the Series 90-70 PLC returns 8K back to LM90 in the response Mailbox message, and 8K is the agreed upon Block Transfer Buffer Size between the two devices when loading and storing programs. It doesn't hurt anything to logon with a buffer size larger than what the CPU can currently handle, since the CPU will pass back its size in this case. This allows for future enhancement where the CPU's buffer size may increase, with no need to change the logon request.

* The long break is not required for Break-Free SNP Operation

If you are not going to be loading and storing program data, this value has no meaning. A value must be entered, however, to fulfill the logon service request requirements, and might just as well be set to 16K. If you need to do program loads and stores, then refer to the section on Program LOAD and Program STORE for the Series 90-70 PLC for a complete definition of this field, and determination of a value applicable to the User's SNP environment.

Response Mailbox Message:

Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack mailbox message.
Block Transfer buffer Size (bytes 23-24):		**Series 90-70 PLC only:** Master should use this value in determining the maximum Block Transfer buffer size during program loads and stores on the Series 90-70 PLCs.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response to a logon request are:

- Invalid task name.
- No communications buffer size (Series 90-70 PLC only).

Examples of programmer logons to both a Series 90-30 CPU and Series 90-70 CPU follow.

Example of Programmer Logon for Series 90-30 PLC

The following assumes that the master has already attached to a 90-30 CPU device and wishes to logon as a programmer attachment. The master sends the Initial Request Mailbox message to the slave, with the service request code for Programmer Logon (20h), the log mode byte set for "Login Serial Mode" (02h), along with the task name (ASCII null string in this example) and the slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 22 c0 10 3a 00 00 10 0a 00 00	
01 01 20 02 00 00 00 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 22 d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 01 00 00 5c 20
	17 00 00 00 00 BCC
(wait **T1** time)	
4: → [ACK]	
06 00	

A full explanation of the SNP message used in the example follows.

Explanation of Programmer Logon for Series 90-30 PLC

Table 6-25. Series 90-30 PLC Programmer Logon

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	22	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	20	Request code: Programmer logon.
	22	02	Logon mode: in this case logon serial.
	23-30	00 00 00 00 00 00 00 00	Task name: NULL.
	31-34	00 00 00 00	Not used in 90-30.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	22	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Programmer Logon for Series 90-70 PLC

The following assumes that the master has already attached to a 90-70 CPU device and wishes to logon as a programmer attachment. The control program task within the PLC is called "STAT_1". The master SNP device has the storage capability to handle a communications buffer size of up to 16K bytes.

The master sends the Initial Request Mailbox message to the slave with the service request code for Programmer Logon (20h), the log mode byte set for "Login Serial Mode" (02h), the task name "STAT_1"), and a communications buffer size of 16K. The slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 5b c0 10 3a 00 00 10 0a 00 00	
01 01 20 02 53 54 41 54 5f 31 00 00 00 40 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 5b d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx 00 20 xx xx xx xx 00 00 00 00 7c 31
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Programmer Logon for Series 90-70 PLC

Table 6-26. Series 90-70 PLC Programmer Logon

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	5b	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	20	Request code: Programmer logon.
	22	02	Logon mode: in this case logon serial.
	23-30	53 54 41 54 5f 31 00 00	Task name: "STAT_1"
	31-32	00 40	Communication buffer size: 4000h=16384 bytes=16K.
	33-34	00 00	Reserved; must be set to zero.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	5b	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Major status code: 0 = okay.
	22	xx	Don't care.
	23-24	00 20	Communication buffer size = 2000h=8192 bytes=8K.
	25-28	xx xx xx xx	Don't care.
	29	00	Control program number.
	30	00	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Set Control ID (CPU ID)

The Set Control ID request allows the master to specify a PLC CPU controller ID for a given PLC CPU. The following rules apply to valid controller IDs:

Series 90-70 PLC: Maximum of seven ASCII characters followed by a NUL.

Series 90-30 PLC: Maximum of six ASCII characters followed by a NUL; furthermore, the valid characters are restricted to the ASCII characters `0' through `9' and `A' through `F' inclusive.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	22h	Set PLC CPU Controller ID.
Controller ID (bytes 22-29):		ASCII string with NUL terminator.
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status and if applicable, a minor error status. Common reasons for an Error Nack Mailbox message response are:

- Insufficient privilege (must be level 3 or higher). *
- Not logged in as a programmer (Series 90-30 PLCs only).
- Invalid length (six characters for Series 90-30 PLCs, seven for Series 90-70 PLCs; NUL terminated).
- Invalid characters (Series 90-30 PLCs are restricted to `0' through `9' and `A through `F' inclusive).

An example with full explanation follows.

*Note: For Series 90-30 PLCs with Release 1.x CPU firmware, privilege level 4 is required.

Example of Set Control ID

This example sets the CPU Controller ID to "33101A". The example assumes that the master has already attached to the slave device, has set the privilege level to three (3), and has logged in as the programmer. The master sends the Initial Request Mailbox message to the slave, with the service request code for Set CPU Control ID (22h), and the slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 bc c0 10 3a 00 00 10 0a 00 00	
01 01 22 33 33 31 30 31 41 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx bc d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 04 00 00 4c 20
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Set Control ID

Table 6-27. Set Control ID

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	bc	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	22	Request code: Set CPU Controller ID.
	22-29	33 33 31 30 31 41 00 00	Controller ID: "33101A"
	30-34	00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	bc	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	04	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	4c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Set PLC Time/Date

The Set PLC Time/Date request allows the master to set the internal time and date of the PLC CPU.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	24h	Set PLC Time.
Time-Date mode (byte 22):	1	Set both time and date.
	2	Set time only.
	3	Set date only.
Seconds (byte 23)		Current seconds in packed BCD format.
Minutes (byte 24)		Current minutes in packed BCD format.
Hour (byte 25)		Current hour in packed BCD format.
Day (byte 26)		Current day of month in packed BCD format.
Month (byte 27)		Current month in packed BCD format.
Year (byte 28)		Current year in packed BCD format.
Day of Week (byte 29)		Day of week: Valid range 1 - 7 where Sunday = 1, ..., Saturday = 7 (ignored in Series 90-30 PLCs since the 90-30 always computes its own).

Packed BCD format: tens digit in bits 4 through 7, ones digit in bits 0 through 3.

Response Mailbox Message:

Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message response is:

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

An example with full explanation follows.

Example of Set Time/Date

The following example assumes that the master has already attached to the slave device, has set the privilege level to two (2), and wishes to set the time to 15:27:37 (3:27:37 pm) and set the date to Thursday May 10, 1990. The master sends the Initial Request Mailbox message to the slave with the service request code for Set PLC Time/Date (24h), and the slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 04 c0 10 3a 00 00 10 0a 00 00 01 01 24 01 37 27 15 10 05 90 05 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK] 06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 04 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 04 00 00 4c 20 17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Set Time/Date

Table 6-28. Set Time/Date

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	04	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	24	Request code: Set PLC Time/Date.
	22	01	Time-Data mode: 1 = set both.
	23	37	Seconds in packed BCD.
	24	27	Minutes in packed BCD.
	25	15	Hour in packed BCD.
	26	10	Day in packed BCD.
	27	05	Month in packed BCD.
	28	90	Year in packed BCD.
	29	05	Day of week.
	30-34	00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	04	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	04	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	4c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Set PLC State (Run versus Stop)

The Set PLC State request allows the master to change the execution state of the PLC. The 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.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	23h	Set PLC State.
New State (bytes 22):	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.
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response are:

- icient privilege (must be level 2 or higher).
- ot logged in as a programmer (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 (valid values: 0, 1, 2, or 6).

Two examples with full explanations follow.

Example of Set PLC State to Run I/O Enabled

This example sets the PLC state to Run mode I/O enabled. The example assumes that the master has already attached to the slave device, has set the privilege level to two (2), and has logged in as the programmer. The master sends the Initial Request Mailbox message to the slave with the service request code for Set PLC State (23h), and the new state field set Run I/O Enabled (00h). The slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 05 c0 10 3a 00 00 10 0a 00 00	
01 01 23 00 00 00 00 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 05 d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 02 xx xx 4c xx
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Set PLC State to Run I/O Enabled

Table 6-29. Set PLC State to Run I/O Enabled

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	05	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	23	Request code: Set PLC State.
	22	00	New PLC state: Run I/O enabled.
	23-34	00 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	05	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	02	Current privilege level.
	31-32	xx xx	Last sweep time.
	33-34	4c xx	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Set PLC State to Stop I/O Enabled

This example sets the PLC state to Stop mode I/O enabled. The example assumes that the master has already attached to the slave device, has set the privilege level to two (2), and has logged in as the programmer. The master sends the Initial Request Mailbox message to the slave with the service request code for Set PLC State (23h), and the slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 09 c0 10 3a 00 00 10 0a 00 00	
01 01 23 06 00 00 00 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 09 d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 02 xx xx 4c xx
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Set PLC State to Stop I/O Enabled

Table 6-30. Set PLC to Stop I/O Enabled

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	09	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	23	Request code: Set PLC State.
	22	06	New PLC state: Stop I/O enabled.
	23-34	00 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	09	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	02	Current privilege level.
	31-32	xx xx	Last sweep time.
	33-34	4c xx	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Toggle Force System Memory

The Toggle Force System Memory request allows the master to change the bit state of status, override, and transition bit memory to their opposite state.

For example, if you wanted to toggle the state of a given input status bit, and that input happened to have an override active, then you would use this request to "override the override", if you will, in order to change the state of the input, while leaving the override active: the state of the input is toggled (set opposite to what it currently is), and the override is still active.

You can also use this request to toggle the state of transition and override bit memories.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	44h	Toggle Force System Memory.
Segment Selector (byte 22):		Memory Type Segment Selector. (See table below)
Offset (bytes 23-24):		Zero-based offset of bit within the memory type specified by the Segment Selector.
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response are:

- Insufficient privilege (must be level 2 or higher, must be level 3 or greater for override table).
- Invalid segment selector: must be BIT mode type (see table below).

The following table lists those memory types that are valid with the Toggle Force System Memory request. All others are invalid.

Table 6-31. Valid Toggle Force SMEM Memory Types

Segment Selectors		
Decimal	**Hexadecimal**	**Memory Type**
-70	046	I memory in bit mode.
-72	048	Q memory in bit mode.
-74	04A	T memory in bit mode.
-76	04C	M memory in bit mode.
-78	04E	SA memory in bit mode.
-80	050	SB memory in bit mode.
-82	052	SC memory in bit mode.
-84	054	S memory in bit mode.
-86	056	G memory in bit mode.
150	096	I override table in bit mode.
152	098	Q override table in bit mode.
154	09A	T override table in bit mode.
156	09C	M override table in bit mode.
158	09E	SA override table in bit mode.
160	0A0	SB override table in bit mode.
162	0A2	SC override table in bit mode.
164	0A4	S override table in bit mode.
166	0A6	G override table in bit mode.
168	0A8	I transition table in bit mode.
170	0AA	Q transition table in bit mode.
172	0AC	T transition table in bit mode.
174	0AE	M transition table in bit mode.
176	0B0	SA transition table in bit mode.
178	0B2	SB transition table in bit mode.
180	0B4	SC transition table in bit mode.
182	0B6	S transition table in bit mode.
184	0B8	G transition table in bit mode.

An example with full explanation follows.

Example of Toggle Force System Memory

This example toggles the state of discrete input bit 29 (%I29), which is currently overridden. It assumes that the master has already attached to the slave device and has set the privilege level to two (2). The master sends the Initial Request Mailbox message to the slave with the service request code for Toggle Force System Memory (44h), and the slave responds with a Completion ACK Mailbox message which includes the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 32 c0 10 3a 00 00 10 0a 00 00 01 01 44 46 1c 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK] 06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 32 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 02 00 00 54 20 17 00 00 00 00 BCC
(wait **T1** time)	
4: →> [ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Toggle Force System Memory

Table 6-32. Toggle Force System Memory

Packet Number	Byte Number(s)	Hex Value	Description
1:→	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	32	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	44	Request code: Toggle Force SMEM.
	22	46	Segment Selector: %I memory in Bit Mode.
	23-24	1c 00	Bit offset: 001c = 28 = %I 29.
	25-34	00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	32	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	02	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	54 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Datagram Service Requests

The Datagram service requests enable the user to obtain reference data values for one or more PLC memory types via a single service request. An area of memory within the PLC CPU called a Datagram Connection Area must first be established and defined.

The Datagram Connection Area in the PLC CPU is made up of three parts: a fixed length header which is twelve (12) bytes long; the list of reference data to monitor, where each reference data access is defined by a four (4) byte point format record; and the data area, where all the current reference data accessed by each point format record is stored before being sent to the SNP master. A maximum of 256 point formats can be defined for each Datagram.

The fixed header portion contains three fields:

Program Block Name: Applicable to Series 90-70 PLCs only; eight-byte NUL terminated string which specifies the name of the program subblock in the main control program which is assumed by any Local Subblock Data segment selectors (%L) specified within the Datagram point formats. If no Local Subblock Data segment selectors are specified in the point formats, then this 8-byte field should be the ASCII Null string. For the Series 90-30 PLCs, this field will always be the ASCII Null string. If data from two or more subblocks is required, separate Datagrams must be established and defined for each one.

Program Block Segment: Word value; should always be set to zero.

Number of point Formats: Word value; total number of point formats defined in this Datagram connection area.

The size of each point format definition is four bytes which breaks down as follows:

Segment selector	Byte field; see segment selector table in Read and Write System Memory section for values.
Point offset	Zero-based offset into memory area defined by the segment selector (word field, least significant byte followed by most significant byte); the unit of offset is determined by the segment selector specified: each segment selector has an implicit access type (either bit, byte, or word).
Point length	Length of data to be accessed (byte field; the units of length is determined by the segment selector specified: each segment selector has an implicit access type (either bit, byte, or word). Each word access takes up two bytes within the connection area, and each byte and bit access takes up one byte within the connection area.

One special note needs to be made concerning bit data: bit mode Datagrams are supported only by Series 90-30 PLCs. The maximum point length for bit mode point formats is eight (8). Furthermore, all bits within the point format (from point offset to point offset plus point length, inclusive) must lie within a single byte. For example, bits %I1 through %I8 are in the same byte. Byte boundaries start at %I1, %I9, %I17, ... %I(8n + 1), ... , where n is any positive integer. The corresponding bit mode point offsets are 0, 8, 16, ..., 8n, ... A bit mode point format with a point offset of eight (8) could have a point length of 1 to 8, but a point offset of nineteen (19) could only have a point length in the range of 1 to 5.

The size of bit mode data within the Datagram connection area is always one byte regardless of the point length.

Once a Datagram Connection Area is established and defined, the values of reference data from different memory types defined in the Datagram Connection Area may be accessed via one service request. Multiple Datagram connections may be created; the maximum number is limited by the amount of memory available within the PLC CPU to store the Datagram information.

There are four PLC CPU Datagram service requests:

Establish Datagram Request

This request informs the PLC CPU the size in bytes required for a Datagram Connection Area for this Datagram, including the size of all the data which will be returned by future Update Datagram requests. The PLC CPU reserves the needed space, assigns the Datagram connection area a unique ID, and passes this ID back to the master. This ID is used in all subsequent Datagram service requests that deal with this particular Datagram connection area

Write Datagram Request

This request defines the Datagram header and point formats that go into the Datagram connection area.

Update Datagram Request

Once a Datagram has been established and defined (via Establish Datagram and Write Datagram requests, respectively, the user may issue the Update Datagram request as needed to retrieve the values of the reference data defined in the Datagram connection area

Cancel Datagram Request

This request cancels a given Datagram connection area. When a Datagram is no longer needed, it should be cancelled in order to free memory within the PLC CPU for other uses.

When the master wishes to use a Datagram connection, the Datagram must first be "Established" and then "Written (defined)", "Updated" to retrieve the current reference data values, and finally "Cancelled" when no longer required. Please see the specific sections for each Datagram service request for details. Two examples are given for the three requests to Establish, Write, and Update a Datagram: a Datagram connection with one point format and another Datagram connection with multiple point formats. Having two examples for the Establish Datagram request may seem redundant at first, but it was done to give the reader a better understanding of the computation of the Datagram size field (a field within the Establish Datagram Initial Request Mailbox message). The Cancel Datagram request has only one example since two examples would indeed be redundant in this case.

Establish Datagram Request

The Establish Datagram request, together with the Write Datagram request, enables the master to specify a fixed set of reference data to be accessed as a group via subsequent Update Datagram requests. This allows the user to pick out key reference data (not all necessarily of the same memory type) and to access the data in a single service request (Update Datagram request). This set or group of reference data is referred to as a Datagram. The Establish Datagram request specifies the Datagram area size; the actual reference data to be included within this Datagram is defined in the Write Datagram request.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	15h	Establish Datagram.
Datagram Type (byte 22):	01h	Normal (Update on Request) 81h = Permanent (Update on Request)
Datagram Size (bytes 23-24):		Required Datagram area size in bytes; consists of: -12 bytes (fixed header size). -+ 4 * N bytes (N = number of point formats). -+ number of data bytes accessed.

Response Mailbox Message:

Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.
Datagram ID (byte 23):		Identification number for this Datagram area; this ID should be used in the Write, Update, and Cancel Datagram requests that pertain to this Datagram area.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message response to an Establish Datagram request is:

- Insufficient privilege (Series 90-70 PLCs only: must be level 1 or higher).

An example with full explanation follows.

Example of Establish Datagram: One Point Format

The master wants to set up a Datagram area that accesses the discrete inputs %I1 to %I112. This group of inputs spans 14 bytes. The point format data takes four bytes (segment selector, point offset, and point length), and the fixed portion of the Datagram area takes 12 bytes (program block name, program block segment, and number of point formats). The Datagram Size required is 12 plus 4 plus 14, or 30 bytes.

The master sends the Initial Request Mailbox message to the slave with the service request code for Establish Datagram (15h), the Datagram Type field set to Update on Request (01), and the Datagram Size set for 30 (1eh) bytes. The slave responds with a Completion ACK Mailbox message which contains a Datagram ID of five (5) along with the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 6d c0 10 3a 00 00 10 0a 00 00 01 01 15 01 1e 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK] 06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 6d d4 10 0a 00 00 10 3a 00 00 01 01 00 xx 05 xx xx xx xx xx 00 01 00 00 5c 20 17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Establish Datagram: One Point Format

Table 6-33. Establish Datagram: One Point Format

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6d	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	15	Request code: Establish Datagram.
	22	01	Datagram type: 01 = Update on Request.
	23-24	1e 00	Datagram size in bytes: 001eh = 30
	25-34	00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6d	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22	xx	Don't care.
	23	05	Datagram Connection ID.
	24-28	xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Establish Datagram: Multiple Point Formats

The master wants to set up a Datagram area that accesses the following groups of reference data:

```
%R05 - %R09 (five words)    %M03 - %M05 (one byte)  %M33 - %M64 (four bytes)
%T17 - %T48 (four bytes)    %I82 - %I87 (one byte)
```

This group of reference data spans 20 bytes. The point format data takes 20 bytes (five point formats at four bytes each), and the fixed header of the Datagram area takes 12 bytes (program block name, program block segment, and number of point formats). The Datagram Size required is 20 plus 20 plus plus 12, or 52 bytes.

The master sends the Initial Request Mailbox message to the slave with the service request code for Establish Datagram (15h), the Datagram Type field set to Update on Request (01), and the Datagram Size set for 52 (34h) bytes. The slave responds with a Completion ACK Mailbox message which contains a Datagram ID of six (6) along with the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 6d c0 10 3a 00 00 10 0a 00 00 01 01 15 01 34 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK] 06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 6d d4 10 0a 00 00 10 3a 00 00 01 01 00 xx 06 xx xx xx xx xx 00 01 00 00 5c 20 17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Establish Datagram: Multiple Point Formats

Table 6-34. Etablish Datagram: Multiple Point Formats

Packet Number	Byte Number(s)	Hex Value	Description
1:→	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6d	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	15	Request code: Establish Datagram.
	22	01	Datagram type: 01 = Update on Request.
	23-24	34 00	Datagram size in bytes: 0034h = 52
	25-34	00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6d	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22	xx	Don't care.
	23	06	Datagram Connection ID.
	24-28	xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Write Datagram Request

The Write Datagram request allows the master to specify the reference data to be included within a given Datagram area. This data includes the fixed header portion of the Datagram area (program block name, program block segment, and the number of point formats), along with `N' number of point formats (where each format is made up of a segment selector, point offset, and point length). There is a key difference in the mailbox type of the request mailbox between the Series 90-30 and Series 90-70 PLCs: the Series 90-30's request Mailbox Type is 80h, and the Series 90-70's request Mailbox Type is is C0h. In both cases, the Mailbox message trailer specifies a Next Message Type `C' for Connection Data. After the slave acknowledges the Mailbox message, the master transmits one or more Connection Data messages containing the definition of the Datagram area. Please see the description directly below.

Key fields within the request and response messages:

Request Mailbox Message for Series 90-30 PLCs:

Mailbox Type (byte 10):	80h	Initial Request Mailbox with Connection Data.
Total length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Connection Data messages.
Service Request code (byte 29):	48h	Write Datagram.
Datagram ID (byte 30):		Datagram connection ID whose point definition we want to define. The Datagram ID is the ID passed back to the master in the Establish Datagram request.
Datagram Type (byte 31):	01h	Normal (Update on Request) 81H Permanent (Update on Request)
Next message type (byte 36):	43h	`C' for Connection data.
Next message length (bytes 37-38):		Number of bytes in the next Connection Data message.

Request Mailbox Message for Series 90-70 PLCs:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox.
Service Request code (byte 21):	48h	Write Datagram.
Datagram ID (byte 22):		Datagram connection ID whose point definition we want to define. The Datagram ID is the ID passed back to the master in the Establish Datagram request.
Datagram Type (byte 23):	01h	Normal (Update on Request) 81H Permanent (Update on Request)
Next message type (byte 36):	43h	`C' for Connection data.
Next message length (bytes 37-38):		Number of bytes in the next Connection Data message.

Request Connection Data Message:

Datagram ID (byte 3):	ID that was passed back in Establish Datagram request; it identifies the Datagram area for which we are defining the reference data. (Note that this is the same Datagram ID that is specified in the Request Mailbox message.)
Point definition size (bytes 7-8):	Number of bytes in point definition; consists of:
	-12 bytes (fixed header size).
	-+ 4 * N bytes (N = number of point formats).
Program block name (bytes 9-16):	Should be set to Null unless your Datagram accesses the Local Subblock Data segment or Main Program Block Data segment (%L or %P respectively for Series 90-70 PLCs only).
Program block segment (bytes 17-18):	Should be set to zero.
Number of point formats (bytes 19-20):	Word field which specifies the number of point formats contained in the Datagram connection area (least significant byte followed by most significant byte).
First point's Segment selector (byte 21):	Byte field that specifies the segment selector of the memory type to access.
First point's offset (bytes 22-23):	Word field that specifies the zero-based offset into the memory type to access (least significant byte followed by the most significant byte).
First point's length (byte 24):	Byte field that specifies the length of data to access.

•

•

•

The last four bytes are repeated as necessary until all point formats are defined.

Response Mailbox Message:

Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response to an Write Datagram request is:

- Insufficient privilege (Series 90-70 PLCs only: must be level 1 or higher).
- Unable to locate specified Datagram ID.

Examples with full explanations follow.

Example of Write Datagram for the Series 90-30 PLCs: One Point Format

This example defines the reference data for Datagram ID five (5). The example assumes that the master has already issued the Establish Datagram request, and that the PLC returned a Datagram Connection ID of five (5). The definition specifies one point format: inputs %I1 to %I112. This group of inputs spans 14 bytes. The segment selector used is the %I byte mode; the point offset is zero, and the point length is 14 bytes.

The master sends the Initial Request Mailbox message with Connection Data to the slave with the service request code for Write Datagram (48h). After the slave acknowledges the Mailbox message, the master transmits a Connection Data message containing all the parameter data. The slave responds with a Completion ACK Mailbox message which contains the piggy-back status information.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message with Connection Data] 1b 4d 00 00 00 00 00 00 6e 80 10 3a 00 00 10 0a 00 00 00 01 16 00 00 00 00 00 01 01 48 05 01 00 00 00 17 43 1e 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
(wait **T1** time) 3: →[Request Connection Data message] 1b 43 05 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 01 00 10 00 00 0e 17 00 00 00 00 BCC	
	(wait **T1** time) 4: ←[ACK] 06 00 (wait **T1** time) 5: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 6e d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 01 00 00 5c 20 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Write Datagram for Series 90-30 PLCs: One Point Format

Table 6-35. Write Datagram for Series 90-30 PLCs: One Point Format

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6e	Sequence number.
	10	80	Mailbox type: Initial Request with Connection Data.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	00 01	Reserved; first byte zero, second byte one.
	21-22	16 00	Connection Data size in bytes = 0016h = 22
	23-26	00 00 00 00	Reserved; must be set to zero.
	27-28	01 01	Packet number, total packets.
	29	48	Request code: Write Datagram.
	30	05	Datagram connection ID.
	31	01	Normal Datagram
	32-34	00 00 00	Not used; set to zero.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	1e 00	Next message length in bytes: 001eh = 30
	39	00	Status byte.
	40	BCC	Block Check Code.
3: →	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	05	Datagram Connection ID.
	4-6	00 00 00	Reserved; must be set to zero.
	7-8	10 00	Point definition size in bytes: 0010h = 16
	9-16	00 00 00 00 00 00 00 00	Program block name.
	17-18	00 00	Program block segment.
	19-20	01 00	Number of point formats= 0001
	21	10	%I memory in byte mode.
	22-23	00 00	Point offset: 0 -->%I1
	24	0e	Point length = 0eh = 14 bytes.
	25	17	End of block character.
	26	00	Next message type.
	27-28	00 00	Next message length.
	29	00	Status byte.
	30	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6e	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Write Datagram for Series 90-30 PLCs: Multiple Point Formats

This example defines the reference data for Datagram ID six (6). We assume the master has already issued the Establish Datagram request and was passed back a Datagram Connection ID of six. This example defines the following reference data to be included in Datagram area six:

```
%R05 - %R09 (five words)    %M03 - %M05 (one byte)  %M33 - %M64 (four bytes)
%T17 - %T48 (four bytes)    %I82 - %I87 (one byte)
```

The master sends the Initial Request Mailbox message with Connection Data to the slave, with the service request code for Write Datagram (48h). After the slave acknowledges the Mailbox message, the master transmits a Connection Data message containing all the parameter data. The slave responds with a Completion ACK Mailbox message which contains the piggy-back status information.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message with Connection Data] 1b 4d 00 00 00 00 00 00 6e 80 10 3a 00 00 10 0a 00 00 00 01 26 00 00 00 00 00 01 01 48 06 01 00 00 00 17 43 2e 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
(wait **T1** time) 3: →> [Request Connection Data message] 1b 43 06 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00 05 00 08 04 00 05 14 02 00 04 4c 02 00 03 46 51 00 06 16 04 00 04 17 00 00 00 00 BCC	
	(wait **T1** time) 4: ←[ACK] 06 00 (wait **T1** time) 5: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 6e d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 01 00 00 5c 20 17 00 00 00 00 BCC
(wait **T1 time)** 6: →ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Write Datagram for Series 90-30 PLCs: Multiple Point Formats

Table 6-36. Write Datagram for Series 90-30 PLCs: Multiple Point Formats

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6e	Sequence number.
	10	80	Mailbox type: Initial Request with Connection Data.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	00 01	Reserved; first byte zero, second byte one.
	21-22	26 00	Connection Data size in bytes = 0026h = 38
	23-26	00 00 00 00	Reserved; must be set to zero.
	27-28	01 01	Packet number, total packets.
	29	48	Request code: Write Datagram.
	30	06	Datagram Connection ID.
	31	01	Normal Datagram
	32-34	00 00 00	not used; set to zero.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	2e 00	Next message length in bytes: 002eh = 46
	39	00	Status byte.
	40	BCC	Block Check Code.
3: →	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	06	Datagram connection ID.
	4-6	00 00 00	Reserved; must be set to zero.
	7-8	20 00	Point definition size in bytes: 0020h = 32
	9-16	00 00 00 00 00 00 00 00	Program block name.
	17-18	00 00	Program block segment.
	19-20	05 00	Number of point formats = 0005
	21	08	%R word memory.
	22-23	04 00	Point offset: in words = 0004 -->%R5
	24	05	Point length = 5 words: %R5 - %R9
	25	14	%T byte memory.
	26-27	02 00	Point offset: in bytes = 0002 -->%T17
	28	04	Point length: 4 bytes: %T17 - %T48
	29	4c	%M bit memory.
	30-31	02 00	Point offset: in bits = 0002 -->%M3
	32	03	Point length: 3 bits: %M3 - %M5
	33	46	%I bit memory.
	34-35	51 00	Point offset: in bits = 0051h = 81 -->%I82

Table 6-36. Write Datagram for Series 90-30 PLCs: Multiple Point Formats - continued

Packet Number	Byte Number(s)	Hex Value	Description
	36	06	Point length: 6 bits: %I82 - %I87
	37	16	%M byte memory.
	38-39	04 00	Point offset: in bytes = 0004h -->%M33
	40	04	Point length: 4 bytes.
	41	17	End of block character.
	42	00	Next message type.
	43-44	00 00	Next message length.
	45	00	Status byte.
	46	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6e	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Write Datagram for Series 90-70 PLCs: One Point Format

This example defines the reference data for Datagram ID one (1). We assume the master has already issued the Establish Datagram request and the PLC returned a Datagram Connection ID of one. The definition specifies one point format: inputs %I1 to %I112. This group of inputs spans 14 bytes. The segment selector used is %I byte mode; the point offset is zero, and the point length is 14 bytes.

The master sends the Initial Request Mailbox message to the slave, with the service request code for Write Datagram (48h) and the Datagram Connection ID. After the slave acknowledges the Mailbox message, the master sends the Connection Data message containing all the parameter data. The slave responds with a Completion ACK Mailbox message which contains the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 6e c0 10 3a 00 00 10 0a 00 00	
01 01 48 01 01 00 00 00 00 00 00 00 00 00 00 00	
17 43 1e 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
(wait **T1** time)	
3: →[Request Connection Data message]	
1b 43	
01 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00	
01 00 10 00 00 0e	
17 00 00 00 00 BCC	
	(wait **T1** time)
	4: ← [ACK]
	06 00
	(wait **T1** time)
	5: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 6e d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 01 00 00 7c 31
	17 00 00 00 00 BCC
(wait **T1** time)	
6: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Write Datagram for Series 90-70 PLCs: One Point Format

Table 6-37. Write Datagram for Series 90-70 PLCs: One Point Format

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type= `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6e	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	48	Request code: Write Datagram.
	22	01	Datagram Connection ID.
	23	01	Normal Datagram
	24-34	01 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	1e 00	Next message length in bytes: 001eh = 30
	39	00	Status byte.
	40	BCC	Block Check Code.
3: →	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	01	Datagram connection ID.
	4-6	00 00 00	Reserved; must be set to zero.
	7-8	10 00	Point definition size in bytes: 0010h = 16
	9-16	00 00 00 00 00 00 00 00	Program block name.
	17-18	00 00	Program block segment.
	19-20	01 00	Number of point formats= 0001
	21	10	%I memory in byte mode.
	22-23	00 00	Point offset: 0 -->%I1
	24	0e	Point length = 0eh = 14 bytes.
	25	17	End of block character.
	26	00	Next message type.
	27-28	00 00	Next message length.
	29	00	Status byte.
	30	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6e	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Example of Write Datagram for the Series 90-70: Multiple Point Formats

This example defines the reference data for Datagram ID two (2). We assume the master has already issued the Establish Datagram request and the PLC returned a Datagram Connection ID of two. This example defines the following reference data to be included in Datagram area two:

```
%R05 - %R09  (five words)           %R41 - %R43  (three words)
%T17 - %T48  (four bytes)           %M33 - %M64  (four bytes)
```

The master sends the Initial Request Mailbox message to the slave, with the service request code for Write Datagram (48h) and a Datagram Connection ID of two. After the slave acknowledges the Mailbox message, the master sends the Connection Data message containing all the parameter data. The slave responds with a Completion ACK Mailbox message which contains the piggy-back status information.

Master	Slave
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 6e c0 10 3a 00 00 10 0a 00 00 01 01 48 02 01 00 00 00 00 00 00 00 00 00 00 00 17 43 2a 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
(wait **T1** time) 3: →[Request Connection Data message] 1b 43 02 00 00 00 1c 00 00 00 00 00 00 00 00 00 00 00 04 00 08 04 00 05 08 28 00 03 14 02 00 04 16 04 00 04 17 00 00 00 00 BCC	
	(wait **T1** time) 4: ←[ACK] 06 00
	(wait **T1** time) 5: ← [Completion ACK MB message] 1b 4d xx xx xx xx xx xx 6e d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 01 00 00 7c 31 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Write Datagram for Series 90-70 PLCs: Multiple Point Formats

Table 6-38. Write Datagram for Series 90-70 PLCs: Multiple Point Formats

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6e	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	48	Request code: Write Datagram.
	22	02	Datagram connection ID.
	23	01	Normal Datagram
	24-34	01 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	2a 00	Next message length in bytes: 002ah = 42
	39	00	Status byte.
	40	BCC	Block Check Code.
3: →	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	02	Datagram connection ID.
	4-6	00 00 00	Reserved; must be set to zero.
	7-8	1c 00	Point definition size in bytes: 001ch = 28
	9-16	00 00 00 00 00 00 00 00	Program block name.
	17-18	00 00	Program block segment.
	19-20	04 00	Number of point formats = 0004
	21	08	%R word memory.
	22-23	04 00	Point offset: in words = 0004 -->%R5
	24	05	Point length = 5 words: %R5 - %R9
	25	08	%R word memory.
	26-27	28 00	point offset: in words = 0028h = 40 -->%R41
	28	03	Point length = 3 words: %R41 - %R43
	29	14	%T byte memory.
	30-31	02 00	Point offset: in bytes = 0002 -->%T17
	32	04	Point length: 4 bytes: %T17 - %T48
	33	16	%M byte memory.
	34-35	04 00	Point offset: in bytes = 0004 -->%M33
	36	04	Point length: 4 bytes.
	37	17	End of block character.
	38	00	Next message type.
	39-40	00 00	Next message length.
	41	00	Status byte.
	42	BCC	Block Check Code.

Table 6-38. Write Datagram for Series 90-70 PLCs: Multiple Point Formats - continued

Packet Number	Byte Number(s)	Hex Value	Description
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6e	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Update Datagram

The Update Datagram request is used to access the data specified within a given Datagram Connection area. The current values for all point formats defined within the given Datagram area are returned to the master. There is a key difference in the mailbox type of the response Mailbox message between Series 90-30 PLCs and Series 90-70 PLCs. The Series 90-30 PLC's response Mailbox Type is 94h, and the Series 90-70 PLC's response Mailbox Type is D4h. In both cases, the Mailbox message trailer specifies a Next Message Type of `43h' for Connection Data. After the master acknowledges the Mailbox message, the slave transmits one or more Connection Data messages containing the current values for all points formats. Refer to the description directly below.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	16h	Update Datagram.
Datagram ID (byte 22):		Datagram connection ID: Datagram area to update.
Datagram Type (byte 23):	01h	Normal (Update on Request) 81h Permanent (Update on Request)

Response Mailbox Message for Series 90-30 PLCs:

Mailbox Type (byte 10):	94h	Completion ACK Mailbox message withConnection Data.
	D1h	Error Nack Mailbox message.
Total length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Connection Data messages.
Next message type (byte 36):	43h	`C' for Connection Data.
Next message length (bytes 37-38):		Number of bytes in the next Connection Data message.

Response Mailbox Message for Series 90-70 PLCs:

Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.
Next message type (byte 36):	43h	`C' for Connection Data.
Next message length (bytes 37-38)		Number of bytes in the next Connection Data message.

Both the Series 90-30 and 90-70 PLCs:

If the CPU is unable to comply with the request, it is rejected. In this case, a Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message is:

- Insufficient privilege (Series 90-70 PLCs only: must be level 1 or higher).
- Unable to locate specified Datagram ID.

Response Connection Data Message:

The point data is returned to the master via a Connection Data message. The number of Connection Data messages returned to the master depends on the amount of point data being accessed and the maximum data size allowed in a SNP Data message.

-Datagram Connection ID (byte 3):	ID of Datagram area being returned.
-Number of reference data bytes (bytes 7-8):	Number of reference data bytes that follow.
-Reference Data (9-N):	The remaining data bytes contain the values of the reference data for all point formats within the Datagram area. The number of bytes that go with a given point format depends on the layout specified in the Write Datagram request.

Examples with full explanations follow.

Example of Update Datagram Request for the Series 90-30 PLCs: One Point Format

This example requests an update of reference data defined in Datagram ID five (5). Datagram ID five has already been established and defined as containing one point format that spans %I1 to %I112. The reference data %I1 through %I112 is currently set as follows:

%I01 = 87h	**%I25 = 98h**	**%I49 = 86h**	**%I73 = 22h**	**%I97 = 71h**
%I09 = 23h	**%I33 = 21h**	**%I57 = 09h**	**%I81 = 05h**	**%I105 = 43h**
%I17 = 23h	**%I41 = 34h**	**%I65 = 19h**	**%I89 = 12h**	

The master sends the Initial Request Mailbox message to the slave with the service request code for Update Datagram (16h) along with the Datagram Connection ID. The slave responds with a Completion ACK Mailbox message with Connection Data which contains the piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits the Connection Data message containing the values of the reference data.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 71 c0 10 3a 00 00 10 0a 00 00 01 01 16 05 01 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
	(wait **T1** time) 3: ←[Completion ACK MB message with Connection Data] 1b 4d xx xx xx xx xx xx 71 94 10 0a 00 00 10 3a 00 00 xx 01 14 00 xx xx xx xx 01 01 00 01 00 00 5c 20 17 43 1c 00 00 BCC
(wait **T1** time) 4: →[ACK] 06 00	
	(wait **T1** time) 5: ←[Response Connection Data message] 1b 43 05 xx xx xx 0e 00 87 23 23 98 21 34 86 09 19 22 05 12 71 43 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Update Datagram for Series 90-30 PLCs: One Point Format

Table 6-39. Update Datagram for Series 90-30 PLCs: One Point Format

Packet Number	Byte Number(s)	Hex Value	Description
1:→	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	71	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	16	Request code: Update Datagram.
	22	05	Datagram Connection ID.
	23	01	Normal Datagram
	24-34	01 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	71	Sequence number.
	10	94	Mailbox type: Completion ACK with Connection Data.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	xx xx	Reserved: don't care.
	21-22	1c 00	Connection Data size in bytes.
	23-26	xx xx xx xx	Don't care.
	27-28	01 01	Packet number, total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	24 00	Next message length in bytes: 0024h = 36
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	05	Datagram Connection ID.
	4-6	xx xx xx	Don't care.
	7-8	0e 00	Point definition size.
	9-22	87 23 23 98 21 34 86 09 19	
	11	17	End of block character.
	12	00	Next message type.
	13-14	00 00	Next message length.
	15	00	Status byte.
	16	BCC	Block Check Code.

Example of Update Datagram Request for Series 90-30 PLCs: Multiple Point Formats

This example requests an update of reference data defined in Datagram ID six (6). Datagram ID six has already been established and defined as containing five point formats (see example of Write Datagram for multiple points). The reference data accessed by the five point formats is currently set as follows:

%R05 = 2219h %R06 = 1205h %R07 = 4371h %R08 = 2198h %R09 = 7065h %T17 = 2465h %T33 = 3674h
%M01 = 003fh
%I81 = 4613h
%M33 = 1290h %M49 = 6739h

The master sends the Initial Request Mailbox message to the slave, with the service request code for Update Datagram (16h) along with the Datagram Connection ID. The slave responds with a Completion ACK Mailbox message which contains the piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits the Connection Data message.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 71 c0 10 3a 00 00 10 0a 00 00 01 01 16 06 01 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
	(wait **T1** time) 3: ←[Completion ACK MB message with Connection Data] 1b 4d xx xx xx xx xx xx 71 94 10 0a 00 00 10 3a 00 00 xx 01 1a 00 xx xx xx xx 01 01 00 01 00 00 5c 20 17 43 22 00 00 BCC
(wait **T1** time) 4: →[ACK] 06 00	
	(wait **T1** time) 5: ←[Response Connection Data message] 1b 43 06 xx xx xx 14 00 19 22 05 12 71 43 98 21 65 70 65 24 74 36 1c 12 90 12 39 67 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

Explanation of Update Datagram for the Series 90-30 PLCs: Multiple Point Formats

Table 6-40. Update Datagram for the Series 90-30 PLCs: Multiple Point Formats

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	71	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	16	Request code: Update Datagram.
	22	06	Datagram Connection ID.
	23	01	Normal Datagram
	24-34	01 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	71	Sequence number.
	10	94	Mailbox type: Completion ACK with Connection Data.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	xx xx	Reserved: don't care.
	21-22	1a 00	Connection Data size in bytes.
	23-26	xx xx xx xx	Don't care.
	27-28	01 01	Packet number, total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	22 00	Next message length in bytes: 0022h = 34
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	06	Datagram Connection ID.
	4-6	xx xx xx	Don't care.
	7-8	14 00	Point definition size.
	9-18	19 22 05 12 71 43 98 21 65 70	%R5 - %R9;
	19-22	65 24 74 36	%T17 - %T48
	23	1c	%M3 - %M5
	24	12	%I82 - %I87
	25-28	90 12 39 67	%M33 - %M64
	29	17	End of block character.
	30	00	Next message type.
	31-32	00 00	Next message length.
	33	00	Status byte.
	34	BCC	Block Check Code.

Example of Update Datagram for Series 90-70 PLCs: One Point Format

This example requests an update of reference data defined in Datagram ID one (1). Datagram ID one has already been established and defined as containing one point format that spans %I1 to %I112. The reference data %I1 through %I112 is currently set as follows:

```
%I01 = 87h    %I25 = 98h    %I49 = 86h    %I73 = 22h    %I97 = 71h
%I09 = 23h    %I33 = 21h    %I57 = 09h    %I81 = 05h    %I105 = 43h
%I17 = 23h    %I41 = 34h    %I65 = 19h    %I89 = 12h
```

The master sends the Initial Request Mailbox message to the slave, with the service request code for Update Datagram (16h) along with the Datagram Connection ID. The slave responds with a Completion ACK Mailbox message which contains the piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits a Connection Data message containing the values of the reference data.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 71 c0 10 3a 00 00 10 0a 00 00	
01 01 16 01 01 00 00 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 71 d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 01 00 00 7c 31
	17 43 1c 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	
	(wait **T1** time)
	5: ←[Response Connection Data message]
	1b 43
	01 xx xx xx 0e 00 87 23 23 98 21 34 86 09 19 22
	05 12 71 43
	17 00 00 00 00 BCC
wait **T1** time)	
6: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Update Datagram for Series 90-70 PLCs: One Point Format

Table 6-41. Update Datagram for Series 90-70 PLCs: One Point Format

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	71	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	16	Request code: Update Datagram.
	22	01	Datagram Connection ID.
	23	01	Normal Datagram
	24-34	01 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	71	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	1c 00	Next message length in bytes: 001ch = 28
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	01	Datagram Connection ID.
	4-6	xx xx xx	Don't care.
	7-8	0e 00	Reference data size = 000eh = 14 bytes.
	9-22	87 23 23 98 21 34 86 09 19 22 05 12 71 43	%I1 through %I112
	11	17	End of block character.
	12	00	Next message type.
	13-14	00 00	Next message length.
	15	00	Status byte.
	16	BCC	Block Check Code.

Example of Update Datagram for Series 90-70 PLCs: Multiple Point Formats

This example request an update of reference data defined in Datagram ID two (2). Datagram ID two has already been established and defined as containing four point formats (see example of Write Datagram for Series 90-70 PLCs with multiple points). The reference data accessed by the four point formats is currently set as follows (shows 16 bits worth):

```
%R05 = 2219h   %R06 = 1205h    %R07 = 4371h   %R08 = 2198h   %R09 = 7065h
%R41 = 1297h   %R42 = 3112h    %R43 = 1acdh
%T17 = 2465h   %T33 = 3674h
%M33 = 1290h   %M49 = 6739h
```

The master sends the Initial Request Mailbox message to the slave, with the service request code for Update Datagram (16h) along with the Datagram Connection ID. The slave responds with a Completion ACK Mailbox message which contains the piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits a Connection Data message containing the values of the reference data.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time) 1: → [Initial Request MB message] 1b 4d 00 00 00 00 00 00 71 c0 10 3a 00 00 10 0a 00 00 01 01 16 02 01 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
	(wait **T1** time) 3: ← [Completion ACK MB message] 1b 4d xx xx xx xx xx xx 71 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 01 00 00 7c 31 17 43 26 00 00 BCC
(wait **T1** time) 4: → [ACK] 06 00	
	(wait **T1** time) 5: ← [Response Connection Data message] 1b 43 02 xx xx xx 18 00 19 22 05 12 71 43 98 21 65 70 97 12 12 31 cd 1a 65 24 74 36 90 12 39 67 17 00 00 00 00 BCC
(wait **T1** time) 6: → [ACK] 06 00	

A full explanation of the SNP messages used in the example follows

Explanation of Update Datagram for Series 90-70 PLCs: Multiple Point Formats

Table 6-42. Update Datagram for Series 90-70 PLCs: Multiple Point Formats

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	71	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	16	Request code: Update Datagram.
	22	02	Datagram Connection ID.
	23	01	Normal Datagram
	24-34	01 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	71	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Major status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	7c 31	PLC status word.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	26 00	Next message length in bytes: 0026h = 38
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	02	Datagram Connection ID.
	4-6	xx xx xx	Don't care.
	7-8	18 00	Reference data size = 0018h = 24 bytes.
	9-18	19 22 05 12 71 43 98 21 65 70	%R5 - %R9;
	19-24	97 12 12 31 cd 1a	%R41 - %R43
	25-28	65 24 74 36	%T17 - %T48
	29-32	90 12 39 67	%M33 - %M64
	33	17	End of block char.
	34	00	Next message type.
	35-36	00 00	Next message length.
	37	00	Status byte.
	38	BCC	Block Check Code.

Cancel Datagram

The Cancel Datagram request terminates a specified Datagram ID. Each Datagram ID should be cancelled when it is no longer needed in order to free memory within the PLC.

Key fields within the request and response messages:

Request Mailbox Message:		
Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	17h	Cancel Datagram.
Datagram ID (byte 22):		Datagram Connection ID: Datagram area to terminate. (FFh = All Datagrams)
Datagram Type (byte 23):	01h	Normal Datagrams 81h Permanent Datagrams
Response Mailbox Message:		
Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response to a Cancel Datagram request is:

- Insufficient privilege (Series 90-70 PLCs only: must be level 1 or higher).
- Unable to locate specified Datagram connection ID.

An example with full explanation follows.

Example of Cancel Datagram

This example cancels the Datagram Connection ID five (5). The master sends the Initial Request Mailbox message to the slave, with the service request code for Cancel Datagram (17h) along with the Datagram ID. The slave responds with a Completion ACK Mailbox message which contains the the piggy-back status information.

MASTER	SLAVE
•	•
•	•
•	•
(wait **T1** time)	
1: →[Initial Request MB message]	
1b 4d	
00 00 00 00 00 00 6f c0 10 3a 00 00 10 0a 00 00	
01 01 17 05 00 00 00 00 00 00 00 00 00 00 00 00	
17 00 00 00 00 BCC	
	(wait **T1** time)
	2: ←[ACK]
	06 00
	(wait **T1** time)
	3: ←[Completion ACK MB message]
	1b 4d
	xx xx xx xx xx xx 6f d4 10 0a 00 00 10 3a 00 00
	01 01 00 xx xx xx xx xx xx xx 00 01 00 00 5c 20
	17 00 00 00 00 BCC
(wait **T1** time)	
4: →[ACK]	
06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Cancel Datagram

Table 6-43. Cancel Datagram

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	6e	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19	01	Packet number.
	20	01	Total packets.
	21	17	Request code: Cancel Datagram.
	22	05	Datagram ID.
	23	01	Normal Datagram
	24-34	01 00 00 00 00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	6f	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	01	Packet number.
	20	01	Total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Update Real-Time Datagram

The Update Real-Time Datagram message is used to perform a combined Attach and Update Datagram request. This message is a master-only message and like the Attach message is sent after the long break*. The master must wait T4* time after the end of the long break* before sending the Update Real-Time Datagram message in order to give all PLC CPUs on the serial link time to recognize the long break*, and to set up to receive the Update Real-Time Datagram message.

The Update Real-Time Datagram request is used to access the data specified within a given Datagram Connection area. The current values for all point formats defined within the given Datagram area are returned to the master. There is a key difference in the mailbox type of the response Mailbox message between Series 90-30 PLCs and Series 90-70 PLCs. The Series 90-30 PLC's response Mailbox Type is 94h, and the Series 90-70 PLC's response Mailbox Type is D4h. In both cases, the Mailbox message trailer specifies a Next Message Type of 43h for Connection Data. After the master acknowledges the Mailbox message, the slave transmits one or more Connection Data messages containing the current values for all point formats.

The Update Real-Time Datagram message contains the CPU ID of the slave (PLC CPU) where the specified Datagram Connection area resides. Refer to the description of the CPU ID field used in the Attach message explained earlier in this manual.

The Update Real-Time Datagram message also contains the Datagram ID of the datagram connection area. The Datagram ID specifies a particular "permanent" datagram connection area in the PLC CPU. The "permanent" datagram must have been established in the PLC CPU prior to the sending of the Update Real-Time Datagram message.

* The long break and T4 timer are not required for Break-Free SNP implementations

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 2):	55h	Update Real-Time Datagram message.
CPU ID (bytes 3-10):		CPU ID (8 bytes maximum)
Datagram ID (byte 11):		Datagram ID. Must be established as a permanent datagram.
Block Check Code (bytes 23-24):		Block Check Code - coded in 2-byte ASCII hex as in Attach message.

Response Mailbox Message for Series 90-30 PLCs:

Mailbox Type (byte 2):	94h	Completion ACK Mailbox message with Connection Data.
	D1h	Error NACK Mailbox message.
Total length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Connection Data messages.
Next message type (byte 36):	43h	`C' for Connection Data
Next message length (bytes 37-38):		Number of bytes in the next Connection Data message

Response Mailbox Message for Series 90-70 PLCs:

Mailbox Type (byte 2):	D4h	Completion ACK Mailbox message
	D1h	Error NACK Mailbox message.
Next message type (byte 36):	43h	'C' for Connection Data
Next message length (bytes 37-38):		Number of bytes in the next Connection Data message

Both the Series 90-30 and 90-70 PLCs:

If the CPU is unable to comply with the request, it is rejected. In this case, a Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reason for an Error Nack Mailbox message is:

- Insufficient privilege (Series 90-70 PLCs only; must be level 1 or higher).
- Unable to locate specified Datagram ID.

Response Connection Data Message:

The point data is returned to the master via a Connection Data message. The number of Connection Data messages returned to the master depends on the amount of point data accessed and the maximum data size allowed in a SNP Data message.

Datagram Connection ID (byte 3):	ID of datagram being returned.
Number of reference data bytes (bytes 7-8):	Number of reference data bytes that follow.
Reference Data (bytes 9-N):	The remaining data bytes contain the values of the reference data for all point formats within the Datagram area. The number of bytes that go with a given point format depends on the layout specified in the Write Datagram request.

Example of Update Real-Time Datagram for Series 90-30 PLCs: One Point Format

This example requests an update of reference data defined in Datagram ID five (5). Datagram ID five has already been established as a permanent datagram and defined as containing one point format that spans %I1 to %I112. The reference data %I1 through %I112 is currently set as follows:

%I01 = 87h	**%I25 = 98h**	**%I49 = 86h**	**%I73 = 22h**	**%I97 = 71h**
%I09 = 23h	**%I33 = 21h**	**%I57 = 09h**	**%I81 = 05h**	**%I105 = 43h**
%I17 = 23h	**%I41 = 34h**	**%I65 = 19h**	**%I89 = 12h**	

The master sends the long break*, waits T4*, and sends the Update Real-Time Datagram request to the slave. The slave whose CPU ID matches that specified in the request waits T1 and responds with an ACK. The slave processes the Update Real-Time Datagram request, waits T1 and sends a Completion ACK Mailbox message with Connection Data which contains the piggy-back status information. The master waits T1 and responds with an ACK. The slave waits T1 and transmits the Connection Data message containing the values of the reference data. The master responds with the final ACK and the logical attachment between the two devices is terminated.

MASTER	SLAVE
0:→[Long break*]	
(wait **T4*** time)	
1:→[Update Real-Time Datagram message] 1b 55 00 00 00 00 00 00 00 00 05 00 00 00 00 00 00 00 17 00 00 00 BCC2	
	(wait **T1** time)
	2:←[ACK] 06 00
	(wait **T1** time)
	3:←[Completion ACK MB message with Connection Data] 1b 4d xx xx xx xx xx xx xx 94 10 0e 00 00 10 3a 00 00 xx xx 0e 00 xx xx xx xx 01 01 00 01 00 00 52 20 17 43 1c 00 00 BCC
(wait **T1** time)	
4:→[ACK] 06 00	
	(wait **T1** time)
	5:←[Response Connection Data Message] 1b 43 05 xx xx xx 0e 00 87 23 23 98 21 34 86 09 19 22 05 12 71 43 17 00 00 00 00 BCC
(wait **T1** time)	
6:→[ACK] 06 00	

* The long break and T4 timer are not required for Break-Free SNP implementations

Explanation of Update Real-Time Datagram for Series 90-30 PLCs: One Point Format

Table 6-44. Update Real-Time Datagram for Series 90-30 PLCs: One Point Format

PacketNumber	Byte Number(s)	Hex Value	Description
1:→	1	1b	Start of message character.
	2	55	Message type = `U' for Update Real-Time Datagram.
	3-10	00 00 00 00 00 00 00 00	CPU ID of destination PLC CPU.
	11	05	Datagram ID.
	12-18	00 00 00 00 00 00 00	Not used.
	19	17	End of block character.
	20-22	00 00 00	Not used.
	23-24	BCC2	Block Check Code - special two byte ASCII-encoded BCC for this message calculated over bytes 1-22. Same BCC used in Attach message.
3:←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	00	Sequence number.
	10	94	Mailbox type: Completion ACK with Connection Data.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	xx xx	Reserved: don't care.
	21-22	0e 00	Connection Data size in bytes.
	23-26	xx xx xx xx	Don't care.
	27-28	01 01	Packet number, total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	52 20	PLC status word.
	35	17	End of block character.
	36	43	Next message type: `C' for Connection Data.
	37-38	1c 00	Next message length in bytes: 001ch = 28.
	39	00	Status byte.
	40	BCC	Block Check Code.
5:←	1	1b	Start of message character.
	2	43	Message type = `C' for Connection Data.
	3	05	Datagram Connection ID.
	4-6	xx xx xx	Don't care.
	7-8	0e 00	Point definition size.
	9-22	87 23 23 98 21 34 86 09 19 22 05 12 71 43	14 bytes of data
	23	17	End of block character.
	24	00	Next message type.
	25-26	00 00	Next message length.
	27	00	Status byte.
	28	BCC	Block Check Code.

Loading and Storing User Programs to Series 90-30 PLCs

This section describes loading user programs from the PLC CPU and storing user programs to the PLC CPU. It is assumed that the program you are trying to store to the PLC CPU is a program that was originally created via LM90 or with the Hand-Held Programmer, and was previously loaded from a PLC. This section makes no attempt to define or describe the bytes within the user program.

Program Load: Series 90-30 PLCs

The Program Load Request allows the master to retrieve the user program from a Series 90-30 PLC. A user program on Series 90-30 PLCs consists of two parts: the Logic Block, and the Declaration Block. The master must retrieve both blocks in order to obtain a complete user program. Therefore, two Program Load service requests are necessary: one to load the Logic Block from the PLC to the master, and one to load the Declaration Block from the PLC to the master.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	C0h	Initial Request Mailbox message.
Service Request code (byte 21):	40h	Program Load.
Block Type (byte 22):	0	Logic Block.
	5	Declaration Block.
Block Offset (bytes 23-24):		Should always be set to zero: tells the PLC to start with the first byte of the block.
Block Length (bytes 25-26):		Should always be set to zero: a length of zero tells the PLC to return the entire block being asked for.

Response Mailbox Message:

Mailbox Type (byte 10):	94h	Completion ACK Mailbox message with Block Transfer.
	D1h	Error Nack Mailbox message.
Total length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Block Transfer messages.
Next message type (byte 36):	42h	`B' for Block Transfer.
Next message length (bytes 37-38):		Number of bytes in the next Block Transfer message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response to a program load request are:

- Insufficient privilege (Series 90-70 PLCs only: must be level 1 or higher).
- -Not logged in as a programmer.

Response Block Transfer Message:

All user program data (Logic or Declaration data) returns to the master in one or more Block Transfer messages. The number of Block Transfer messages returned depends on the amount of user program data to transfer and the maximum data size allowed in a SNP Data message.

An example with full explanation follows.

Example of Program Load for Series 90-30 PLCs

This example loads the Logic Block and Declaration Block from a Series 90-30 PLC to the master SNP device. The example assumes that the master has already attached to a Series 90-30 PLC slave device and has logged in as a programmer attachment. The master sends the Initial Request Mailbox message to the slave, with the service request code for Program Load (40h) and the Block Type field set for Logic. The slave responds with a Completion ACK Mailbox message with Block Transfer message which includes piggy-back status information. After the master acknowledges the Mailbox message, the slave transmits the Block Transfer message which contains the Logic Block bytes. The same sequence is repeated for the Declaration Block. You will note that the Logic Block size and the Declaration Block size are returned to the master in the Completion ACK Mailbox messages with Block Transfer (131 bytes and 618 bytes respectively). In an actual master SNP implementation, these sizes must be recorded because they will be needed in any Program Store service requests of a user program (see Program Store service request).

Please note that the actual Logic and Declaration bytes returned are shown as "Don't care" bytes. You only need to have the means to retrieve the user program, and should not be concerned with the value of the bytes within the Logic and Declaration blocks or their meaning.

MASTER	**SLAVE**
•	•
•	•
•	•
(wait **T1** time) 1: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 2b c0 10 3a 00 00 10 0a 00 00 01 01 40 00 00 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
	(wait **T1** time) 3: ←[Completion ACK MB message with Block Transfer] 1b 4d xx xx xx xx xx xx 2b 94 10 0a 00 00 10 3a 00 00 xx 01 83 00 xx xx xx xx 01 01 00 01 00 00 4c 20 17 42 8b 00 00 BCC
(wait **T1** time) 4: →[ACK] 06 00	
	(wait **T1** time) 5: ←[Response Block Transfer message] 1b 42 xx * 131 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 0600	

MASTER	SLAVE
(wait **T1** time) 7: →[Initial Request MB message] 1b 4d 00 00 00 00 00 00 2c c0 10 3a 00 00 10 0a 00 00 01 01 40 05 00 00 00 00 00 00 00 00 00 00 00 00 17 00 00 00 00 BCC	
	(wait **T1** time) 8: ←[ACK] 06 00
	(wait **T1** time) 9: ←[Completion ACK MB message with Block Transfer] 1b 4d xx xx xx xx xx xx 2c 94 10 0a 00 00 10 3a 00 00 xx 01 6a 02 xx xx xx xx 01 01 00 01 00 00 4c 20 17 42 72 02 00 BCC
(wait **T1** time) 10: →[ACK] 06 00	
	(wait **T1** time) 11: ←[Response Block Transfer message] 1b 42 xx * 618 17 00 00 00 00 BCC
(wait **T1** time) 12: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Program Load for Series 90-30 PLCs

Table 6-45. Program Load for Series 90-30 PLCs

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	2b	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	40	Request code: Program Load.
	22	00	Block Type: 0 = Logic Block.
	23-24	00 00	Block Offset: 0 = start with first byte.
	25-26	00 00	Block Length: 0 = return entire block to host.
	27-34	00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	2b	Sequence number.
	10	94	Mailbox type: Completion ACK with Block
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	xx	Reserved: don't care.
	20	01	
	21-22	83 00	Block Transfer data size in bytes.
	23-26	xx xx xx xx	Don't care.
	27	01	Packet number.
	28	01	Total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	4c 20	PLC status word.
	35	17	End of block character.
	36	42	Next message type = `B' for Block Transfer.
	37-38	8b 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	42	Message type = `B' for Block Transfer.
	3-133	xx * 131	Logic block bytes: 131 (decimal) bytes.
	134	17	End of block character.
	135	00	Next message type.
	136-137	00 00	Next message length.
	138	00	Status byte.
	139	BCC	Block Check Code.

Table 6-45. Program Load for Series 90-30 PLCs - continued

Packet Number	Byte Number(s)	Hex Value	Description
7: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	2c	Sequence number.
	10	c0	Mailbox type: Initial Request.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	01 01	Packet number, total packets.
	21	40	Request code: Program Load.
	22	05	Block Type: 5 = Declaration Block.
	23-24	00 00	Block Offset: 0 = start with first byte.
	25-26	00 00	Block Length: 0 = return entire block to host.
	27-34	00 00 00 00 00 00 00 00	Not used.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
9: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	2c	Sequence number.
	10	94	Mailbox type: Completion ACK with Block
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19	xx	Reserved: don't care.
	20	01	
	21-22	6a 02	Block Transfer data size in bytes.
	23-26	xx xx xx xx	Don't care.
	27	01	Packet number.
	28	01	Total packets.
	29	00	Control program number.
	30	01	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	4c 20	PLC status word.
	35	17	End of block character.
	36	42	Next message type = `B' for Block Transfer.
	37-38	72 02	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.
11: ←	1	1b	Start of message character.
	2	42	Message type = `B' for Block Transfer.
	3-620	xx * 618	Declaration block bytes: 618 (decimal) bytes.
	621	17	End of block character.
	622	00	Next message type.
	623-624	00 00	Next message length.
	625	00	Status byte.
	626	BCC	Block Check Code.

Program Store: Series 90-30 PLCs

The Program Store Request allows the master to store a user program from the master SNP device to a Series 90-30 PLC. A user program on Series 90-30 PLCs consists of two parts: the Logic Block and the Declaration Block. The master must store **both** blocks in order to have a complete user program stored in the PLC memory. Therefore, two Program Store service requests are required: one to **store** the Logic Block **from** the master **to** the slave, and one to **store** the Declaration Block **from** the master **to** the slave.

Key fields within the request and response messages:

Request Mailbox Message:

Mailbox Type (byte 10):	80h	Initial Request Mailbox message with Block Transfer.
Total Length of data (bytes 21-22):		Total number of **data bytes** that are going to be transmitted in all subsequent Block Transfer messages.
Service Request code (byte 29):	3fh	Program Store.
Block Type (byte 30):	0	Logic Block.
	5	Declaration Block.
Block Offset (bytes 31-32):		Should always be set to zero.
Block Length (bytes 33-34):		Length in bytes of the block being stored; The master knows what this value is by having done a Program Load for the block type.
Next message type (byte 36):	42h	'B' for Block Transfer.
Next message Length (bytes 37-38):		Number of bytes in the next Block Transfer message.

Request Block Transfer Message:

All user program data (Logic or Declaration data) is transmitted to the slave in one or more Block Transfer messages. The number of Block Transfer messages required depends on the amount of user program data to store in the PLC and the maximum data size allowed in a SNP Data message.

Response Mailbox Message:

Mailbox Type (byte 10):	D4h	Completion ACK Mailbox message.
	D1h	Error Nack Mailbox message.

If the request is completed successfully, a Completion ACK Mailbox message (D4h) is returned to the master.

Otherwise, the CPU was unable to comply with the request and had to reject it. In this case, an Error Nack Mailbox message (D1h) is returned, along with a major error status, and if applicable, a minor error status. The most common reasons for an Error Nack Mailbox message response to a program store request are:

- Insufficient privilege (must be level 3 or higher).
- Not logged in as a programmer.

An example with full explanation follows.

Example of Program Store for Series 90-30 PLCs

This example stores a Logic Block and a Declaration Block from a master SNP device to a Series 90-30 PLC . The Logic Block size is 131 bytes and the Declaration BLock size is 618 bytes. The example assumes that the master has already attached to a Series 90-30 PLC slave device, has set the privilege level to three (3), and has logged in as a programmer attachment. The master sends the Initial Request Mailbox message with Block Transfer to the slave, with the service request code for Program Store (3fh) and the Block Type field set for Logic. After the slave acknowledges the Mailbox message, the master transmits the Block Transfer message which contains the Logic Block bytes. The slave responds with a Completion ACK Mailbox message which includes piggy-back status information. The same sequence is repeated for the Declaration Block.

Please note that the actual Logic and Declaration bytes transmitted are shown as "Don't care" bytes. You only need to have the means to store the user program, and should not be concerned with the value of the bytes within the Logic and Declaration blocks or their meaning.

MASTER	SLAVE
• • •	• • •
(wait **T1** time) 1: →[Initial Request MB message with Block Transfer] 1b 4d 00 00 00 00 00 00 98 80 10 3a 00 00 10 0a 00 00 00 01 83 00 00 00 00 00 01 01 3f 00 00 00 83 00 17 42 8b 00 00 BCC	
	(wait **T1** time) 2: ←[ACK] 06 00
(wait **T1** time) 3: →[Request Block Transfer message] 1b 42 xx * 131 17 00 00 00 00 BCC	
	(wait **T1** time) 4: ←[ACK] 06 00 (wait **T1** time) 5: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 98 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 03 00 00 5c 20 17 00 00 00 00 BCC
(wait **T1** time) 6: →[ACK] 06 00	

MASTER	SLAVE
(wait **T1** time) 7: →[Initial Request MB message with Block Transfer] 1b 4d 00 00 00 00 00 00 99 80 10 3a 00 00 10 0a 00 00 00 01 6a 02 00 00 00 00 01 01 3f 05 00 00 6a 02 17 42 72 02 00 BCC	
	(wait **T1** time) 8: ←[ACK] 06 00
(wait **T1** time) 9: →[Request Block Transfer message] 1b 42 xx * 618 17 00 00 00 00 BCC	
	(wait **T1** time) 10: ←[ACK] 06 00 (wait **T1** time) 11: ←[Completion ACK MB message] 1b 4d xx xx xx xx xx xx 99 d4 10 0a 00 00 10 3a 00 00 01 01 00 xx xx xx xx xx xx xx 00 03 00 00 5c 20 17 00 00 00 00 BCC
(wait **T1** time) 12: →[ACK] 06 00	

A full explanation of the SNP messages used in the example follows.

Explanation of Program Store for Series 90-30 PLCs

Table 6-46. Program Store for Series 90-30 PLCs

Packet Number	Byte Number(s)	Hex Value	Description
1: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	98	Sequence number.
	10	80	Mailbox type: Initial Request with Block Transfer.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	00 01	Reserved; first byte zero, second byte one.
	21-22	83 00	Total length of data in bytes = 0083h = 131
	23-26	00 00 00 00	Reserved; must be set to zero.
	27-28	01 01	Packet number, total packets.
	29	3f	Request code: Program Store.
	30	00	Block Type: 0 = Logic Block.
	31-32	00 00	Block Offset: MUST be set to ZERO.
	33-34	83 00	Block Length: 0083h = 131 bytes.
	35	17	End of block character.
	36	42	Next message type: `B' for Block Transfer.
	37-38	8b 00	Next message length: 139 bytes.
	39	00	Status byte.
	40	BCC	Block Check Code.
3: →	1	1b	Start of message character.
	2	42	Message type = `B' for Block Transfer.
	3-133	xx * 131	Logic block bytes: 131 (decimal) bytes.
	134	17	End of block character.
	135	00	Next message type.
	136-137	00 00	Next message length.
	138	00	Status byte.
	139	BCC	Block Check Code.
5: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	98	Sequence number
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	03	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Table 6-46. Program Store for Series 90-30 PLCs - continued

Packet Number	Byte Number(s)	Hex Value	Description
7: →	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	00 00	Reserved; must be set to zero.
	5-7	00 00 00	Time Stamp.
	8	00	Reserved; must be set to zero.
	9	99	Sequence number.
	10	80	Mailbox type: Initial Request with Block Transfer.
	11-14	10 3a 00 00	Mailbox source: Master SNP device.
	15-18	10 0a 00 00	Mailbox destination: PLC service request task.
	19-20	00 01	Reserved; first byte zero, second byte one.
	21-22	6a 02	Total length of data in bytes = 026ah = 618
	23-26	00 00 00 00	Reserved; must be set to zero.
	27-28	01 01	Packet number, total packets.
	29	3f	Request code: Program Store.
	30	05	Block Type: 5 = Declaration Block.
	31-32	00 00	Block Offset: MUST be set to ZERO.
	33-34	6a 02	Block Length: 026ah = 618 bytes.
	35	17	End of block character.
	36	42	Next message type: `B' for Block Transfer.
	37-38	72 02	Next message length: 626 bytes.
	39	00	Status byte.
	40	BCC	Block Check Code.
9: →	1	1b	Start of message character.
	2	42	Message type = `B' for Block Transfer.
	3-620	xx * 618	Declaration block bytes: 618 (decimal) bytes.
	621	17	End of block character.
	622	00	Next message type.
	623-624	00 00	Next message length.
	625	00	Status byte.
	626	BCC	Block Check Code.
11: ←	1	1b	Start of message character.
	2	4d	Message type = `M' for Mailbox.
	3-4	xx xx	Reserved: don't care.
	5-7	xx xx xx	Time Stamp.
	8	xx	Reserved: don't care.
	9	99	Sequence number.
	10	d4	Mailbox type: Completion ACK.
	11-14	10 0a 00 00	Mailbox source: PLC service request task.
	15-18	10 3a 00 00	Mailbox destination: Master SNP device.
	19-20	01 01	Packet number, total packets.
	21	00	Status code: 0 = okay.
	22-28	xx xx xx xx xx xx xx	Don't care.
	29	00	Control program number.
	30	03	Current privilege level.
	31-32	00 00	Last sweep time.
	33-34	5c 20	PLC status word.
	35	17	End of block character.
	36	00	Next message type.
	37-38	00 00	Next message length.
	39	00	Status byte.
	40	BCC	Block Check Code.

Appendix A Serial Port and Cables

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:	**Series 90 PLC**: 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
	Workmaster II: 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)
	Workmaster: 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)
	IBM-AT/XT: 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)
	RS-232/RS-422 Converter: 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-task resulting from the mismatching will affect the performance of the communications system. When routing communication cables outdoors, transient suppression devices can be used to reduce the possibility of damage due to lightning or static discharge. *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.*

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.

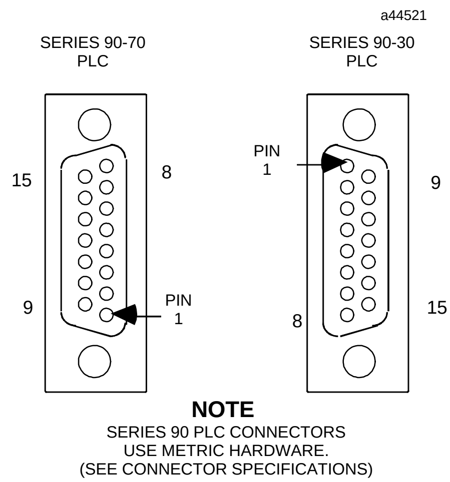


Figure A-1. Series 90 PLC, RS-422 Serial Port Connector Configuration

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.

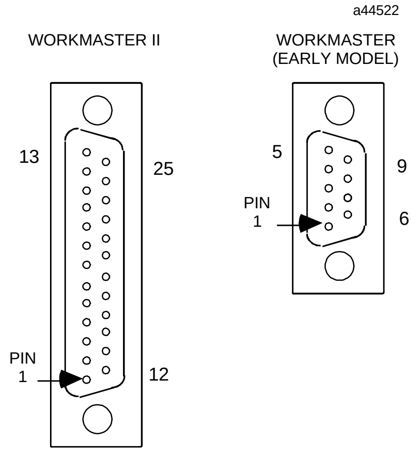


Figure A-2. Workmaster RS-232 Serial Port Connector Configuration

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

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

Workmaster (9-pin connector)		
Pin No.	**Signal**	**Description**
1		NC
2	TD	Transmit Data
3	RD	Receive Data
4	RTS	Request to Send
5	CTS	Clear to Send
6		NC
7	GND	Signal Ground, 0V
8	DCD	Data Carrier Detect
9	DTR	Data Terminal Ready

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.

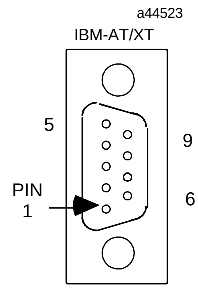


Figure A-3. IBM-AT/XT Serial Port

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

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

IBM-XT Pin No.	Signal	Description
1		NC
2	TD	Transmit Data
3	RD	Receive Data
4	RTS	Request to Send
5	CTS	Clear to Send
6		NC
7	GND	Signal Ground
8	DCD	Data Carrier Detect
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 Fanuc Automation. Please contact any GE Fanuc Automation sales office or field service representative.

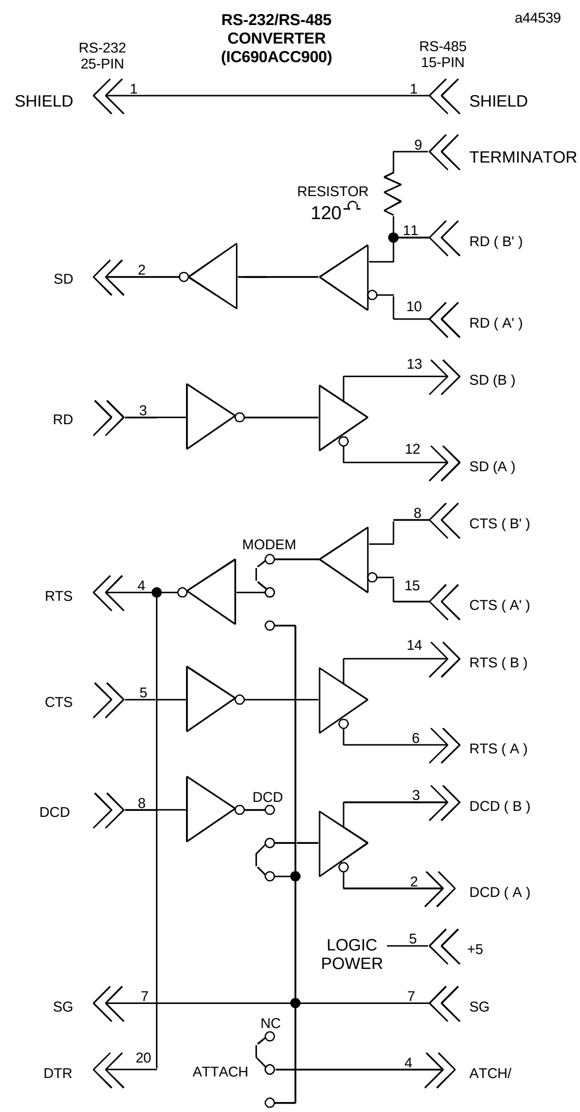
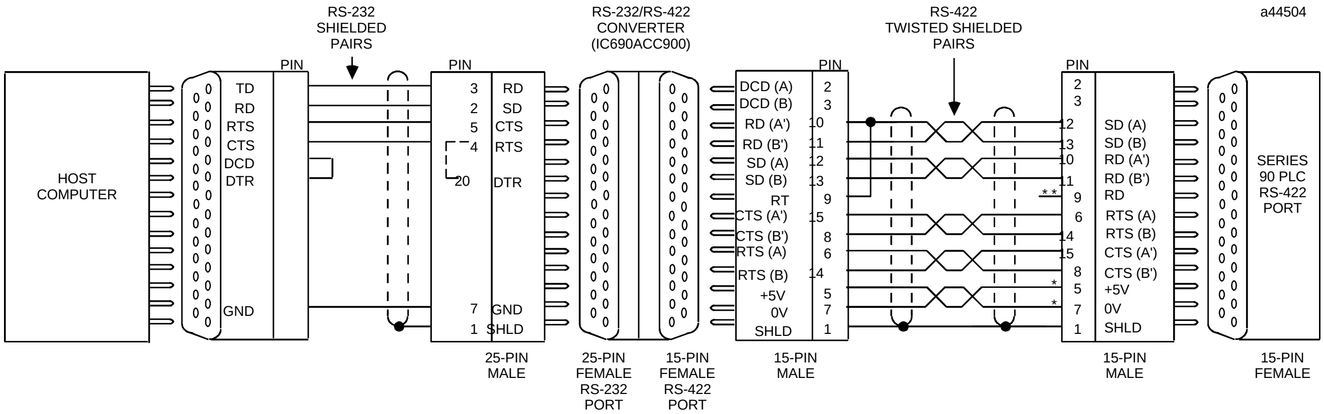


Figure A-4. RS-232 to RS-422 Converter Logic Diagram

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)		
Pin No.	**Signal**	**Description**
-1	SHD	Shield
-2	SD	Send Data
-3	RD	Receive Data
-4	RTS	Request to Send
-5	CTS	Clear to Send
-6		NC
-7	GND	Signal Ground
-8	DCD	NC
9-19		NC
20	DTR	JMP (See Figure A-4)
21-25		NC

RS-422 Port (15-pin connector)		
Pin No.	**Signal**	**Description**
-1	SHD	Shield
-2	DCD(A)	Data Carrier Detect
-3	DCD(B)	Data Carrier Detect
-4		NC
-5	+ 5V	Power Connection
-6	RTS(A)	Request to Send
-7	0V	Ground Connection
-8	CTS(B')	Clear to Send
-9	RT	Terminating Resistor
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

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 Fanuc Automation. Please contact any GE Fanuc Automation 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:

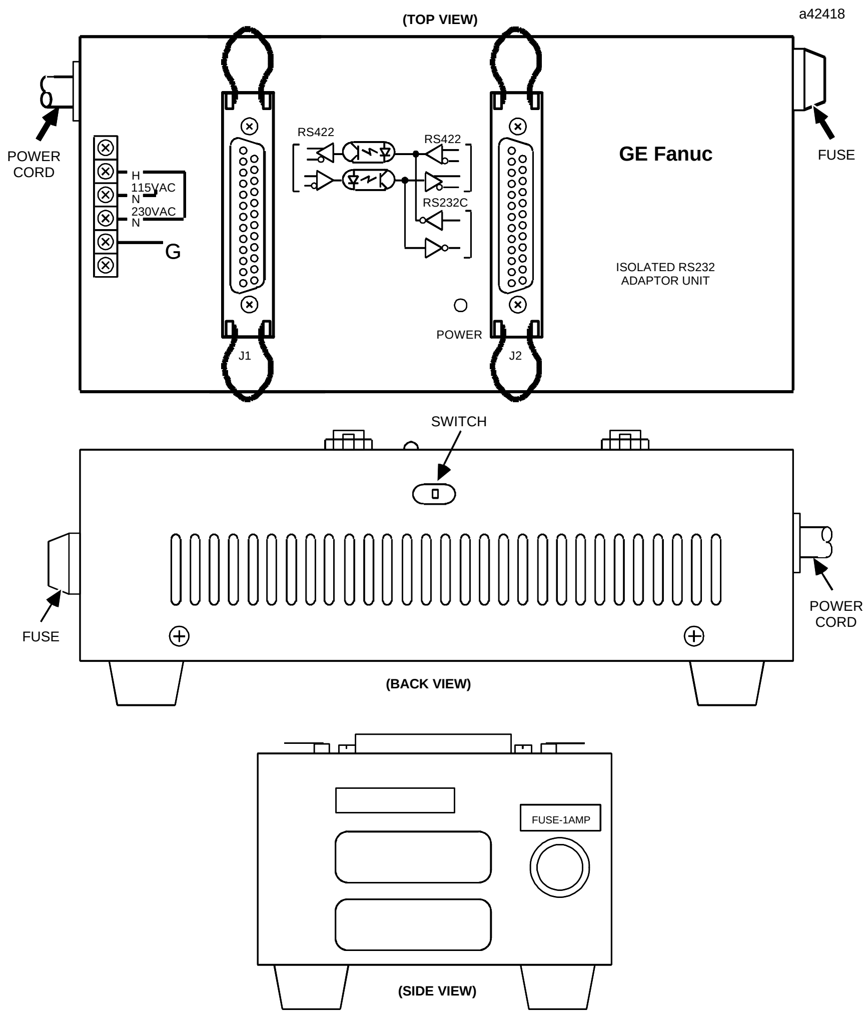


Figure A-6. The Isolated/Repeater Converter

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.)
- 115/230 VAC 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.

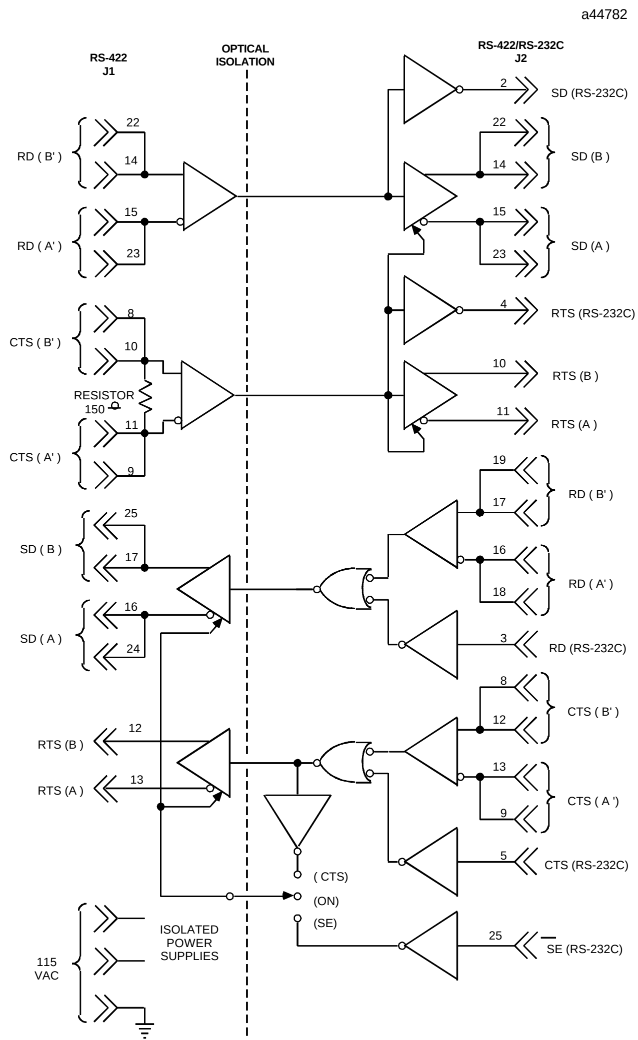


Figure A-7. RS-422 IsolatedRepeater/RS-232 Converter Logic Diagram

Note

All inputs on the unit are biased to the inactive state. Inputs left Figure A-7. RS-422 Isolated RS-422unconnected 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)		
1		NC
2		NC
3		NC
4		NC
5		NC
6		NC
7	0V	Ground Connection
8	CTS(B')	Clear to Send (Optional Termination
9	CTS(A')	Clear to Send (Optional Termination)
10	CTS(B')	Clear to Send
11	CTS(A')	Clear to Send
12	RTS(B)	Request to Send
13	RTS(A)	Request to Send
14	RD(B')	Receive Data
15	RD(A')	Receive Data
16	SD(A)	Send Data
17	SD(B)	Send Data
18		NC
19		NC
20		NC
21		NC
22	RD(B')	Receive Data (Optional Termination)
23	RD(A')	Receive Data (Optional Termination)
24	SD(A)	Send Data
25	SD(B)	Send Data

J2 RS-422/RS-232 Port (25-pin female connector)		
1		NC
2	SD	Send Data (RS-232)
3	RD	Receive Data (RS-232)
4	RTS	Request to Send (RS-232)
5	CTS	Clear to Send (RS-232)
6		NC
7	0V	Ground Connection
8	CTS(B')	Clear to Send (Optional Termination)
9	CTS(A')	Clear to Send (Optional Termination)
10	RTS(B)	Request to Send
11	RTS(A)	Request to Send
12	CTS(B')	Clear to Send
13	CTS(A')	Clear to Send
14	SD(B)	Send Data
15	SD(A)	Send Data
16	RD(A')	Receive Data
17	RD(B')	Receive Data
18	RD(A')	Receive Data (Optional Termination)
19	RD(B')	Receive Data (Optional Termination)
20		NC
21		NC
22	SD(B)	Send Data
23	SD(A)	Send Data
24		NC
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.

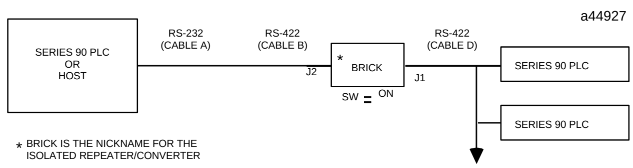


Figure A-8. Simple System Configuration Using the Isolated Repeater/Converter

Complex Multidrop Configuration

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

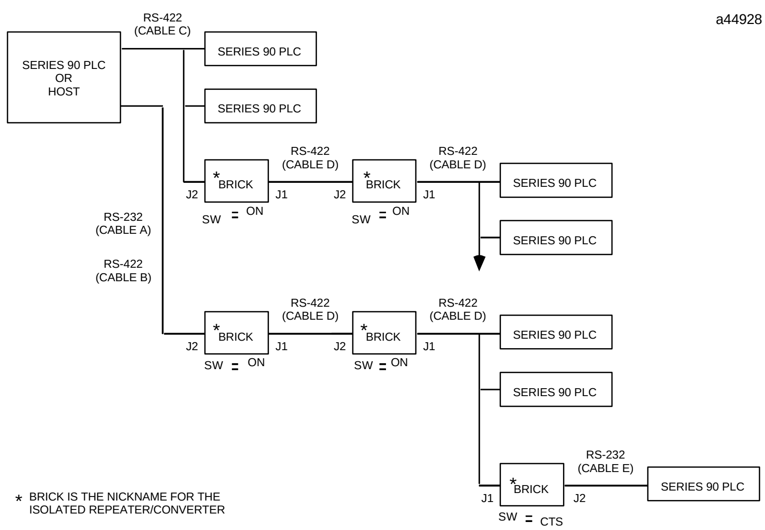


Figure A-9. Complex System Configuration Using the Isolated Repeater/Converter

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.

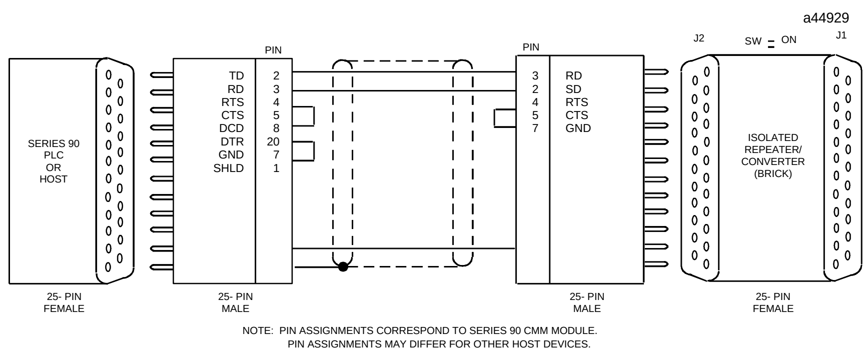


Figure A-10. Cable A; RS-232 Device To Converter

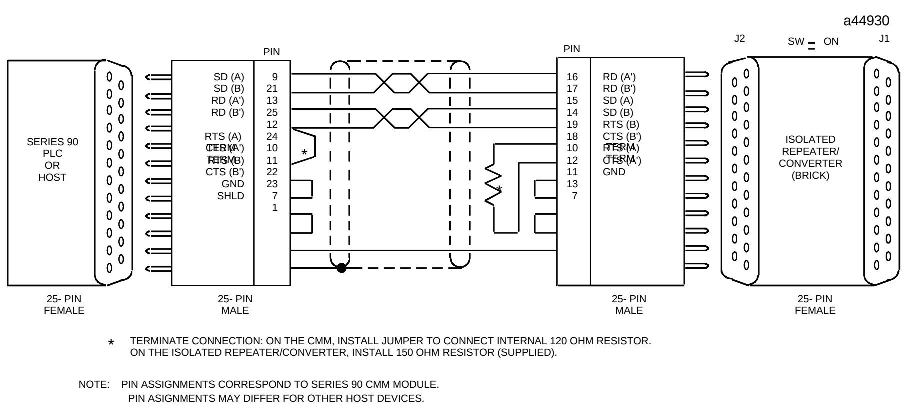


Figure A-11. Cable B; RS-422 Device to Converter

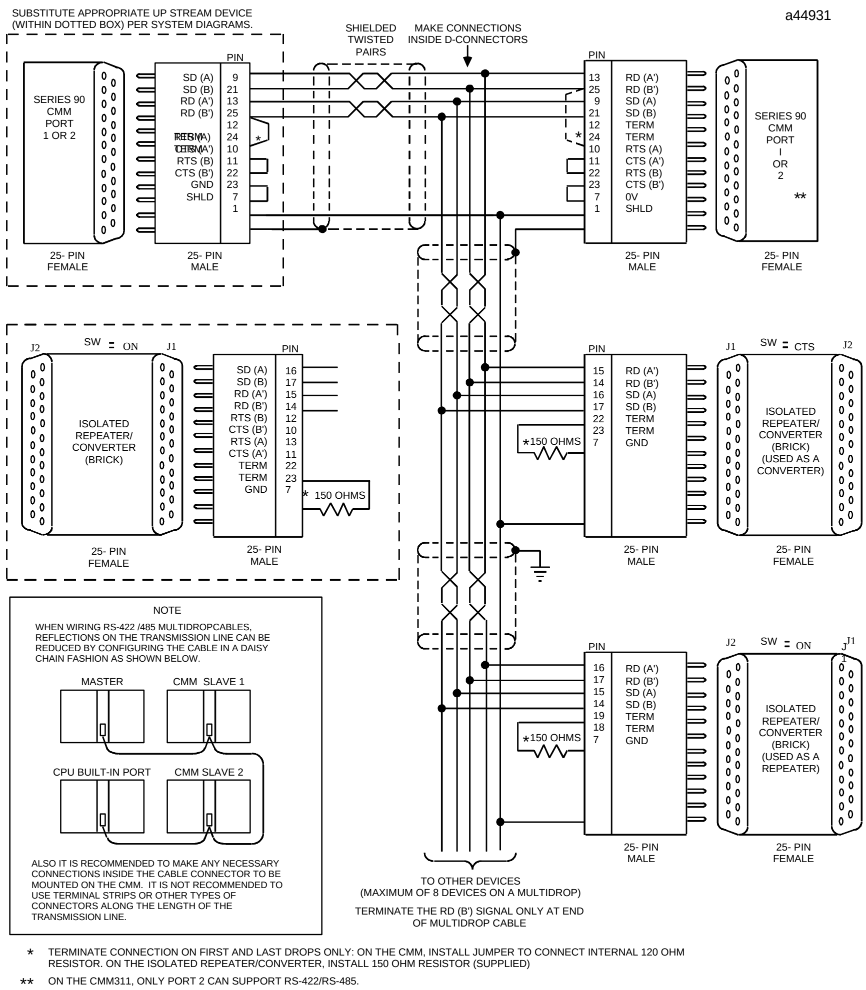


Figure A-12. Cable C; RS422 Twisted Pair

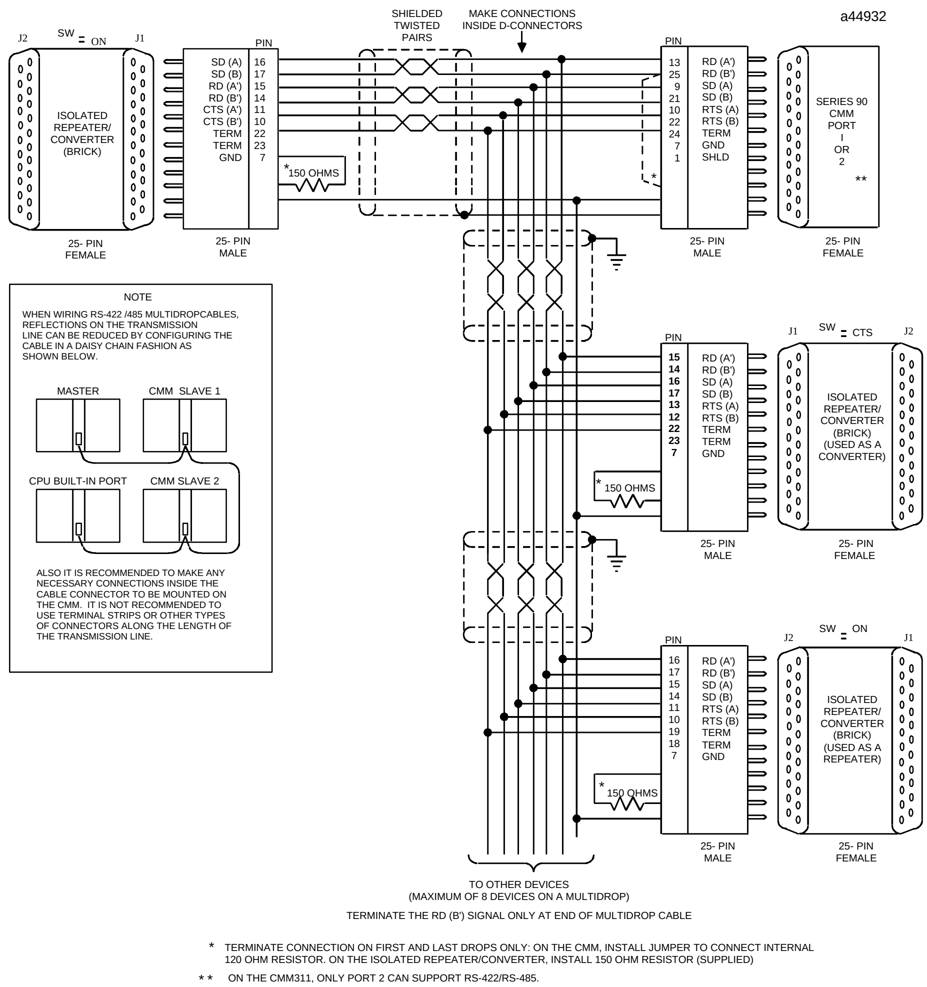


Figure A-13. Cable D; RS422 Twisted Pair

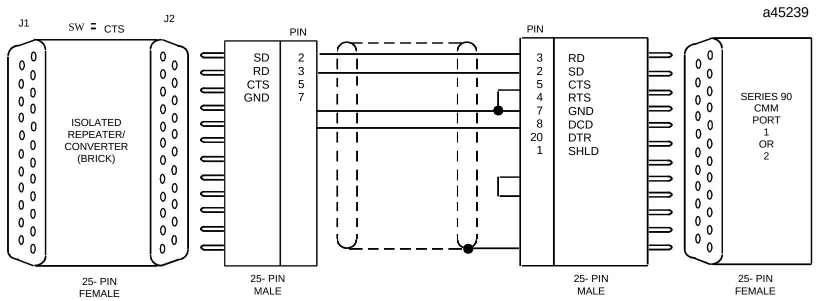


Figure A-14. Cable E; RS-232 Converter to CMM

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

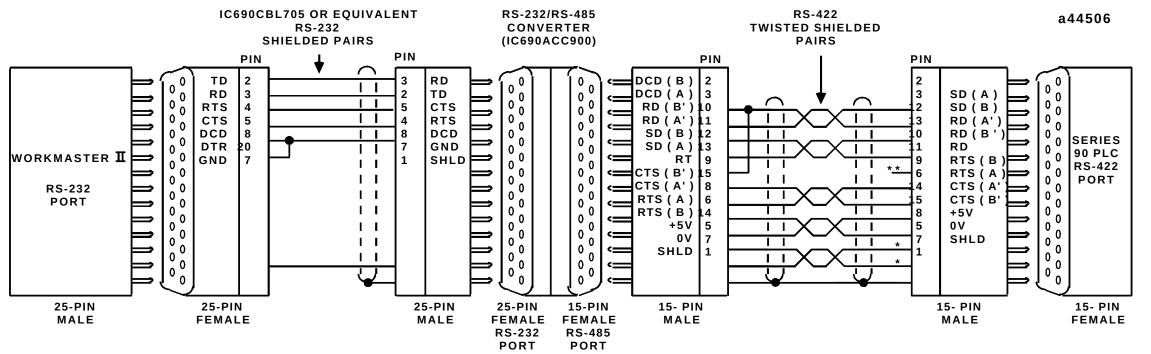
The cable connector for the Series 90-70 and Series 90-30 PLCs serial port must be a right angle connector in order for the hinged door on the module to close properly. Refer to Table A-1 Connector/Cable Specification.

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

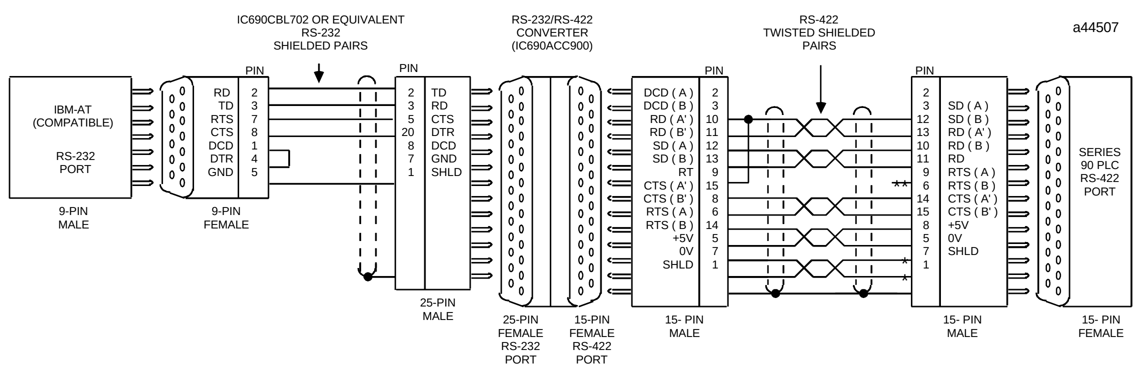
The next three figures illustrate typical RS-232 point-to-point connection to Series 90 PLCs.



* 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 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 PLC CATALOG NUMBERS IC697CPU731 AND IC697CPU771, THE TERMINATION FOR RD AT THE PLC IS IMPLEMENTED BY A JUMPER BETWEEN PIN 9 AND PIN 11.

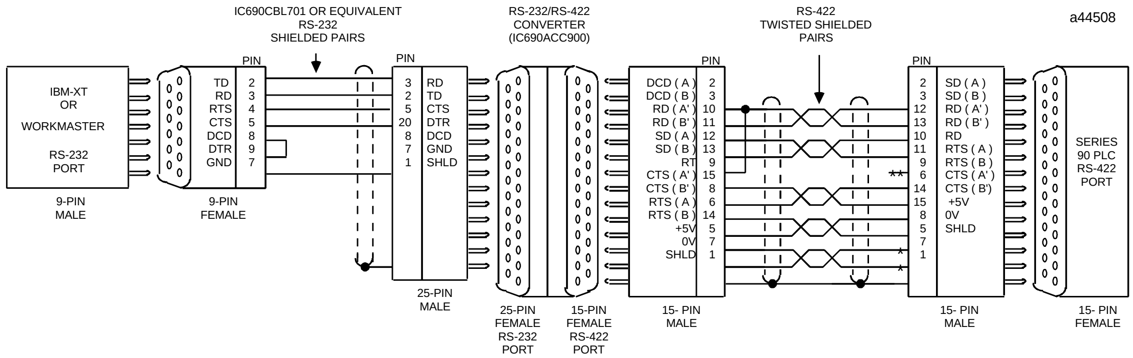
Figure A-15. Workmaster II (25-pin Serial Connection to Series 90 PLCs)



* 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



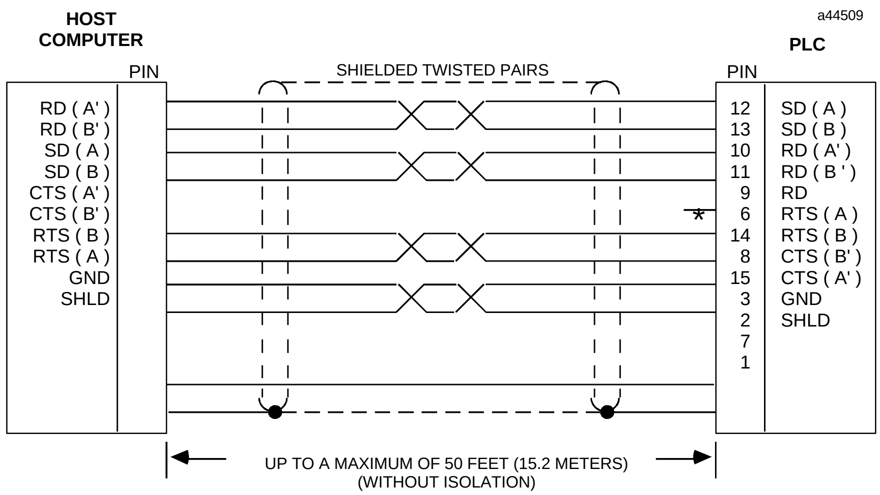
* POWER SOURCE FOR POINT-TO-POINT CONNECTION 10 FEET (3 METERS) ONLY. CONVERTER POWER SOURCE BEYOND 10 FEET (3 METERS) AND FOR MULTIDROP CONNECTION MUST BE EXTERNAL SOURCE.

** 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

Index