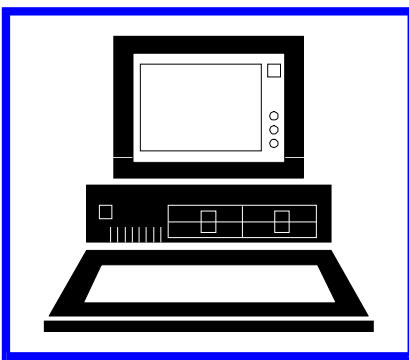
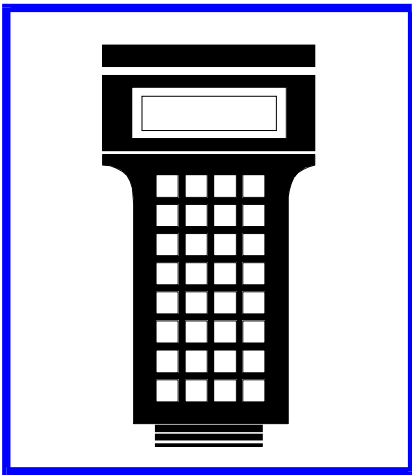
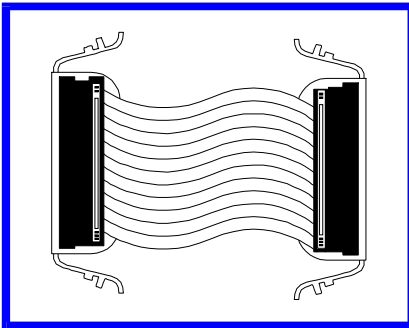
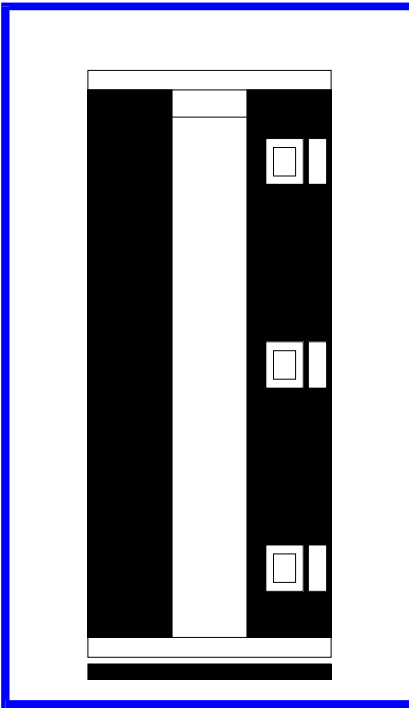
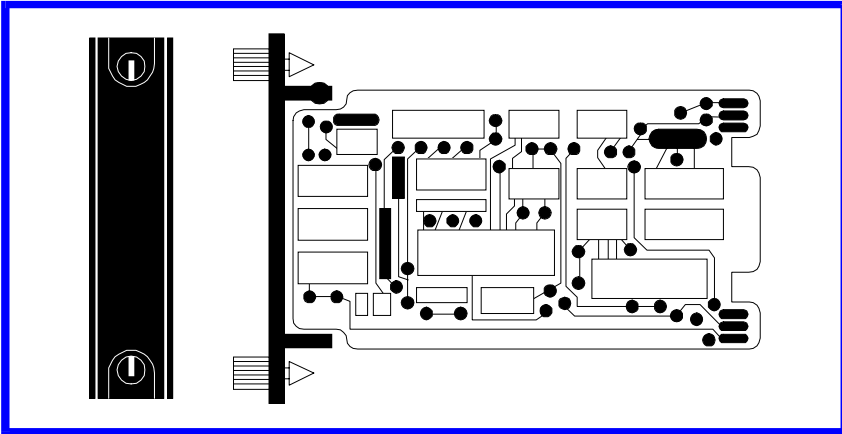
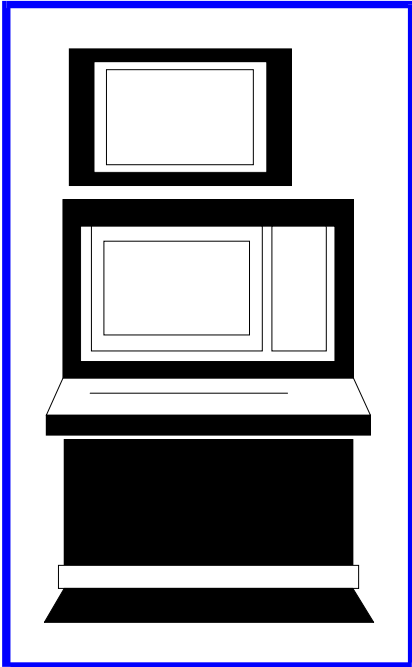


E97-811-20

Instruction

LAN-90[®] Process Control View (PCV[®]) Harmony/Modbus Protocol Specific Driver (PSD) (Software Release 5.2)



WARNING notices as used in this instruction apply to hazards or unsafe practices that could result in personal injury or death.

CAUTION notices apply to hazards or unsafe practices that could result in property damage.

NOTES highlight procedures and contain information that assists the operator in understanding the information contained in this instruction.

WARNING

INSTRUCTION MANUALS

DO NOT INSTALL, MAINTAIN, OR OPERATE THIS EQUIPMENT WITHOUT READING, UNDERSTANDING, AND FOLLOWING THE PROPER **Elsag Bailey** INSTRUCTIONS AND MANUALS; OTHERWISE, INJURY OR DAMAGE MAY RESULT.

RADIO FREQUENCY INTERFERENCE

MOST ELECTRONIC EQUIPMENT IS INFLUENCED BY RADIO FREQUENCY INTERFERENCE (RFI). CAUTION SHOULD BE EXERCISED WITH REGARD TO THE USE OF PORTABLE COMMUNICATIONS EQUIPMENT IN THE AREA AROUND SUCH EQUIPMENT. PRUDENT PRACTICE DICTATES THAT SIGNS SHOULD BE POSTED IN THE VICINITY OF THE EQUIPMENT CAUTIONING AGAINST THE USE OF PORTABLE COMMUNICATIONS EQUIPMENT.

POSSIBLE PROCESS UPSETS

MAINTENANCE MUST BE PERFORMED ONLY BY QUALIFIED PERSONNEL AND ONLY AFTER SECURING EQUIPMENT CONTROLLED BY THIS PRODUCT. ADJUSTING OR REMOVING THIS PRODUCT WHILE IT IS IN THE SYSTEM MAY UPSET THE PROCESS BEING CONTROLLED. SOME PROCESS UPSETS MAY CAUSE INJURY OR DAMAGE.

NOTICE

The information contained in this document is subject to change without notice.

Elsag Bailey, its affiliates, employees, and agents, and the authors and contributors to this publication specifically disclaim all liabilities and warranties, express and implied (including warranties of merchantability and fitness for a particular purpose), for the accuracy, currency, completeness, and/or reliability of the information contained herein and/or for the fitness for any particular use and/or for the performance of any material and/or equipment selected in whole or part with the user of/or in reliance upon information contained herein. Selection of materials and/or equipment is at the sole risk of the user of this publication.

This document contains proprietary information of Elsag Bailey, Elsag Bailey Process Automation, and is issued in strict confidence. Its use, or reproduction for use, for the reverse engineering, development or manufacture of hardware or software described herein is prohibited. No part of this document may be photocopied or reproduced without the prior written consent of Elsag Bailey.

Preface

This manual provides general information and guidance with respect to:

- The Harmony Modbus Protocol Specific Driver (PSD).
- Hardware requirement and cable connections.
- PSD start-up requirements.
- Device and data addressing during configuration of the interface.

List of Effective Pages

Total number of pages in this instruction is 26, consisting of the following:

Page No.	Change Date
Preface	Original
List of Effective Pages	Original
iii through v	Original
1-1 through 1-3	Original
2-1	Original
3-1 through 3-5	Original
4-1 through 4-4	Original
A-1 through A-5	Original
B-1 through B-2	Original
C-1	Original

When an update is received, insert the latest changed pages and dispose of the superseded pages.

NOTE: On an update page, the changed text or table is indicated by a vertical bar in the outer margin of the page adjacent to the changed area. A changed figure is indicated by a vertical bar in the outer margin next to the figure caption. The date the update was prepared will appear beside the page number.

Table of Contents

	<i>Page</i>
SECTION 1 - INTRODUCTION	1-1
OVERVIEW	1-1
REQUIREMENTS	1-2
Hardware Requirements	1-2
Software Requirements	1-2
GLOSSARY	1-2
REFERENCES	1-3
SECTION 2 - INSTALLATION	2-1
MODBUS PSD INSTALLATION	2-1
PSD ASSOCIATED FILES DIRECTORY ASSIGNMENTS	2-1
SECTION 3 - CONFIGURATION	3-1
THE PSD START-UP FILE	3-1
MODBUS MESSAGE DATA ASSOCIATION WITH HARMONY DATA TYPES	3-2
Harmony Data Types 0, Unsigned Short, and 1, Signed Short	3-3
Harmony Data Types 2, Unsigned Long, and 3, Signed Long	3-3
Harmony Data Type 4, Float	3-4
Harmony Data Type 5, Double	3-4
Harmony Data Type 8, Byte Oriented Digital	3-4
Harmony Data Types 10 and 11, BCD and BCD-12	3-4
Harmony Data Types Not Supported	3-5
SECTION 4 - OPERATION	4-1
OVERVIEW	4-1
PSD AND EXECUTIVE INTERDEPENDENCE	4-1
THE PSD START-UP SEQUENCE	4-1
PSD PRIMARY FUNCTIONS	4-2
SINGLE PORT VS MULTIPLE PORTS	4-3
PROTOCOL/DEVICE SPECIFIC ADDRESSING VALIDATION	4-3
APPENDIX A - MODBUS FUNCTIONS	A-1
MODBUS FUNCTIONS SUPPORTED BY THE MODBUS PSD	A-1
APPENDIX B - ERROR MESSAGE TABLE	B-1
ERROR MESSAGE TABLE	B-1
APPENDIX C - MODBUS CABLE CONNECTIONS	C-1
MODBUS CABLE CONNECTIONS	C-1

List of Figures

<i>No.</i>	<i>Title</i>	<i>Page</i>
C-1.	Modbus Cable Connections	C-1

List of Tables

<i>No.</i>	<i>Title</i>	<i>Page</i>
3-1.	Harmony Data Types	3-2
3-2.	Modbus Data Types	3-3
4-1.	Group Size, Generic Versus Modbus Data Type	4-3
A-1.	Read Internal Coils Query Message (1)	A-1
A-2.	Read Internal Coils Reply Message	A-1
A-3.	Read Discrete Inputs Query Message (2)	A-1
A-4.	Read Discrete Inputs Reply Message	A-2
A-5.	Read Holding Registers Query Message (3)	A-2
A-6.	Read Holding Registers Reply Message	A-2
A-7.	Read Input Registers Query Message (4)	A-2
A-8.	Read Input Registers Reply Message	A-3
A-9.	Force Single Coil Query Message (5)	A-3
A-10.	Preset Single Register Query Message (6)	A-3
A-11.	Preset Multiple Registers Query Message (16)	A-4
A-12.	Preset Multiple Registers Reply Message	A-4
A-13.	Diagnostics - Return Query Data (8)	A-4
A-14.	Diagnostics - Return Query Data Reply Message	A-5
B-1.	Error Message Table	B-1

Trademarks and Registrations

Registrations and trademarks in this document include:

® Elsag Bailey	Registered trademark of Elsag Bailey Process Automation.
® INFI-NET	Registered trademark of Elsag Bailey Process Automation.
® Intel	Registered trademark of Intel Corporation.
® LAN-90	Registered trademark of Elsag Bailey Process Automation.
® Modicon	Registered trademark of Schneider Automation, Inc.
® PCV	Registered trademark of Elsag Bailey Process Automation.
® QNX	Registered trademark of QNX Software Systems.

SECTION 1 - INTRODUCTION

OVERVIEW

This document describes the LAN-90 PCV Harmony 90 Protocol Specific Driver (PSD) for Modicon® Modbus data communication.

This PSD contains all Modbus-specific descriptives for addressing Modbus devices and Modbus data communication transactions.

The PSD initiates all communication transactions for the interface. It prompts the Harmony Executive for requests to perform any and all data transactions with Modbus devices.

Specifically, the PSD supports Modbus functions codes 1, 2, 3, 4, 5, 6, and 16 for receiving and sending data.

The PSD processes Harmony Executive requests to:

- Parse and evaluate text strings entered via the Harmony Editor for the device specific parameters: Modbus device type and address, data point address, group data point range.
- Construct protocol specific transaction command messages.
- Initiate communication with devices, monitor completion of transactions, send responses back to the Harmony Executive.
- Return status information regarding itself and device communication to the Harmony Executive.

The PSD converts incoming data from the Modbus data type to the Harmony data type associated with the data group and converts all outgoing data from the specified Harmony data type to the associated Modbus data type. Tag parameters inherit the Harmony data type of the associated data group. The Modbus data type associated with a data group is determined from the group Modbus address (coil, discrete input, input register or holding register).

For maximum throughput, the user should assign non-reply-blocked message exchange with the Harmony Executive. (The executive then replies immediately and will not block driver execution.) The driver sleeps for approximately the sleep time interval in millisecond whenever it has performed all its internal tasks and has received a Harmony Executive

reply indicating no action is required. The sleep time interval is defined in the start-up file.

REQUIREMENTS

Hardware Requirements

The communication medium is RS-232-C. The PSD needs at least one of the serial communication ports available on the LAN-90 PCV. For some applications, it may utilize more than one of the available ports and may require a 4 or 8 port CTI card. For each port used, a cable is required to connect to the device network equipment. Depending on the physical environment at each port connection, modems may be required for DC common isolation or to connect to devices more than 50 feet away from the LAN-90 PCV hardware. The PSD can support up to 8 ports.

When multiple devices must be connected to the same port (multidropped), other hardware is required to interconnect the devices to the RS-232-C port (e.g., RS-232-C/RS485 converters or multidrop modems).

Software Requirements

The PSD needs the Harmony Executive and Editor to form a Harmony interface for LAN-90 PCV. To make the Harmony Executive available, the Harmony option must be enabled under the LAN-90 PCV System Options menu.

GLOSSARY

CIU (Computer Interface Unit)	A device used to connect a host computer system to the Elsag Bailey INFI-NET [®] data highway.
LAN (Local Area Network)	A network whose nodes are connected in relatively close proximity to each other.
LAN-90 PCV (LAN-90 Process Control View)	The software package comprising the operator consoles, data acquisition and local area network software developed and manufactured by Elsag Bailey running under the QNX [®] operating system.
RTDB (Real-Time Data Base)	The LAN-90 PCV database comprising a set of console tags whose values, states and other information are maintained in real-time.
Modbus	The Modicon Gould communications system protocol for industrial distributed control systems. It is a single master/multiple slave protocol.

PSD (<i>Protocol Specific Driver</i>)	This acronym refers to device and protocol specific interfaces developed for the LAN-90 PCV Harmony optional software.
Harmony 90	The LAN-90 PCV optional foreign device interface.
Coil	A digital data variable (bit) in a Modbus device. A coil can be read or written.
Discrete Input	A digital data variable (bit) in a Modbus device. A discrete input can be read only.
Holding Register	An analog data variable in a Modbus device. A holding register can be read or written. It typically represents a signed or unsigned 16 bit (2 byte) data word.
Input Register	An analog data variable in a Modbus device. An input register can be read only. It typically represents a signed or unsigned 16 bit (2 byte) data word.

REFERENCES

- LAN-90 PCV Software Release 5.2 Installation manual, **I-E97-811-1**.
- LAN-90 PCV Software Release 5.2 Configuration and Operation manuals, **I-E97-811-2.1** and **I-E97-811-2**.
- PCV Subsystem Interface Communications Driver API Reference manual, **I-E97-807-16A**, Rev. 1.1.
- Modicon Modbus Protocol Reference Guide, PI-MBUS Rev. B.

SECTION 2 - INSTALLATION

MODBUS PSD INSTALLATION

The Modbus PSD must be installed after the LAN-90 PCV System Option Harmony has been enabled. This enables an additional System Option to install the PSD for Modbus communication. The PSD requires three support files which are described in the following section. The support files are installed in the same step as the PSD.

PSD ASSOCIATED FILES DIRECTORY ASSIGNMENTS

The Modbus PSD task file name is "modbus". The file is stored in directory:

"/bci/pcv/bin/"

Support files for the Modbus PSD task reside in directory:

"/bci/pcv/etc/win/psd/modbus/"

Support files include:

"error.msg"	This file contains error messages used by the Harmony Editor for the protocol specific configuration data fields.
".ttl"	Help title file. This file contains the single text line in upper case: <i>MODBUS PSD HELP FILE</i> at the beginning of the file.

The help file for the Modbus PSD is stored in directory:

"/bci/pcv/etc/win/help/config/psd/modbus/"

The help file name is:

"modbus.help"	This file is used by the Harmony Editor for help with respect to the Modbus protocol/device specific configuration data fields. It contains at the beginning of the file the title text of the ".ttl" file in the form: <ESC> TITLE "MODBUS PSD HELP FILE".
---------------	---

SECTION 3 - CONFIGURATION

THE PSD START-UP FILE

The PSD task is started by the Harmony Executive which assigns a task ID and priority and passes a start-up file name.

The PSD task will read the start-up file parameters only at start-up or restart.

The start-up file is an ASCII text file in which formatted text lines contain the start-up arguments as follows:

p:XXXXXXXXXX i:X b:BBBB d:N s:N a:XXX

p: Specifies a port identification name, *XXXXXXXXXX*, in ASCII string. e.g., "/dev/ser1"

i: Specifies the port redundancy mode as:

N non-redundant.

P primary port.

S secondary port for the primary port on the preceding line (currently not supported for Modbus).

b: Specifies the port baud rate, *BBBB*. e.g. "9600".

d: Specifies the number of data bits, *N*. Typically 8 for Modbus.

s: Specifies the number of stop bits, *N*. Typically 1 for Modbus.

a: Specifies parity, *XXX*; ODD, EVEN or NONE.

The port and all its parameters are specified on one line. Up to eight (8) ports may be specified on consecutive lines.

r: N

r: Specifies the Executive message exchange mode to be Reply Blocked, 1, or Not Reply Blocked, 0. When this parameter is not specified in the start-up parameters, the default Reply Blocked, 1, will be implemented.

NOTE: Regarding Reply Blocking (*r*):

For single port applications, the one transaction at a time Modbus master/slave protocol communication works well in the Reply Blocked mode. When the same PSD serves more than one port, the Reply Not Blocked mode should be invoked to allow the message exchange between the PSD and the Executive for requests for data on different ports to overlap.

w:NNN

w: Specifies delay time *NNN* when the Executive message exchange mode is Not Reply Blocked. When Reply Blocked is specified, the delay time is ignored. The delay (sleep) time is invoked when the Harmony Executive requires no action from the PSD.

v: N

v: Specifies the number of retries, *N*, that should be performed when a port communication transaction fails.

u: NNNN

u: Specifies the time interval, *NNNN*, within which a port communication transaction reply must be received and after which the transaction is set complete with time-out. Timed out transactions on a port will be retried per the retry specification when the current status of the port is normal.

start-up file example:
(Note - the double quotes precede a comment line)

p:dev/ser1 i:N b:9600 d:8 s:1 a:EVEN No redundant port

"p:dev/ser1 i:P b:9600 d:8 s:1 a:EVEN Primary port

r: N Not reply blocked

w: 100 No action sleep time

v: 2 Number of retries

u: 1000 Modbus transaction timeout(msec)

MODBUS MESSAGE DATA ASSOCIATION WITH HARMONY DATA TYPES

Table 3-1. Harmony Data Types

Data Type	Data Type Descriptor
0	Unsigned short integer (Intel [®] format - low byte first).
1	Signed short integer (Intel format - low byte first).
2	Unsigned long integer (Intel format).
3	Signed long integer (Intel format).
4	IEEE float (real 4 - Intel format).
5	IEEE double (real 8 - Intel format).
8	Byte oriented digital data (data bits packed 8 to a byte).
10	BCD (binary coded decimal) unsigned short integer, 16 bits to 4 digits.
11	BCD - 12 bits to 3 digits - unsigned short integer.

Table 3-2 describes the Modbus data types (or files) referenced by the Modicon commands.

Table 3-2. Modbus Data Types

Modbus Function Code	Data Type	Read/Write Access
1	Coils	Read
2	Discrete inputs	Read
3	Holding registers	Read
4	Input registers	Read
5	Coil	Write
6	Holding register	Write
8	Event registers	Read
16	Holding registers	Write

The Harmony data type is assigned to a data group from which individual data cells are assigned to tag attributes.

Harmony data types 0, 1, 2, 3, 4, 5 and 10 are always associated with either Modbus holding registers or input registers. In the case of Harmony data types 2, 3, 4 and 5, which are wider than 2 bytes, multiple consecutive Modbus registers are associated with each Harmony data value.

Harmony Data Types 0, Unsigned Short, and 1, Signed Short

Harmony data type 0 (unsigned short) or type 1 (signed short) is always associated with either a single holding register or input register. As 16 bit integers, they have the following data range:

Signed short -32768 to +32767

Unsigned short 0 to 65535

NOTE: These Harmony data types refer to single registers. For data types 0 and 1, the individual tag assignments reference registers associated with the data group.

Harmony Data Types 2, Unsigned Long, and 3, Signed Long

A Harmony data type 2 (unsigned long integer) or type 3 (signed long integer), each 32 bits wide (four bytes), is typically associated with two consecutive registers of which the lower order register represents the two highest order bytes of the long integer and the next register the lower order two bytes.

The data range is as follows:

Signed short -2147483648 to +2147483647

Unsigned short 0 to 4294967295

NOTE: Since these Harmony data types refer to two consecutive registers, the second of the two registers cannot be referenced independently. For data types 2 and 3, the individual tag assignments must correspond to the first, third, fifth, etc., register associated with the data group.

Harmony Data Type 4, Float

Harmony data type 4 (IEEE real 4 (float)) is also associated with two consecutive registers (holding or input registers) similar to Harmony types 2 and 3.

The range is 10 to the power ± 38 .

NOTE: Since this Harmony data type also refers to two consecutive registers, the second of the two registers cannot be referenced independently. For data type 4, the individual tag assignments must correspond to the first, third, fifth, etc., register associated with the data group.

Harmony Data Type 5, Double

Harmony data type 5 (IEEE real 8 (double)) is like Harmony data type 4 except that the mantissa has 16 significant decimal digits.

The range is 10 to the power ± 38 .

NOTE: This Harmony data type refers to four consecutive registers. The second, third or fourth of the four registers cannot be referenced independently. For data type 5, the individual tag assignments must correspond to the first, fifth, ninth, etc., register associated with the data group.

Harmony Data Type 8, Byte Oriented Digital

Harmony data type 8 (byte oriented digital data) is always associated with coil or discrete input data.

NOTE: This Harmony data type refers to individual data bits which are directly associated with PLC coils or discrete inputs.

Harmony Data Types 10 and 11, BCD and BCD-12

A Harmony type 10 or 11 (BCD or BCD-12 binary coded decimal in unsigned short) is always associated with either a single holding register or input register.

As four and three decimal digit unsigned integers, they have the following data range:

BCD 0 to 9999

BCD -12 0 to 999

NOTE: These Harmony data types refer to single registers. The Harmony data type is assigned to a data group from which individual data cells are assigned to tag attributes. For BCD-12 type data, the high order 4 bits of the 16 bit unsigned integer are masked out and ignored.

Harmony Data Types Not Supported

The Modbus PSD does not support Harmony data types 6, 7, 9 or 12.

SECTION 4 - OPERATION

OVERVIEW

The Modbus PSD software works in tandem with the Harmony Executive to provide an interface between LAN-90 PCV and the Modbus controllers.

PSD AND EXECUTIVE INTERDEPENDENCE

The PSD supplies protocol specific device configuration reference information and communication. The configuration reference information is used by the PSD to evaluate the device data group and data point definitions entered via the Harmony Editor. These definitions are entered as text strings that are parsed and evaluated by the PSD before being accepted into the device configuration file. If this data is incorrect, a message will be printed in the error log.

The PSD task performs communication transactions with programmable controllers that are accessible via one or more of the LAN-90 PCV serial communication ports specified in the start-up file. For each transaction, the PSD performs the following:

- Selects the appropriate port.
- Sends a Modbus command message to a specified controller.
- Waits for a reply to the command message.
- Processes the reply message.

Each Modbus communication data transaction is performed in direct response to a request from the Harmony Executive.

THE PSD START-UP SEQUENCE

After the PSD has been started by the Executive and has accepted the start-up parameters it initializes its communication ports. Next it initiates a message exchange with the Executive to obtain all data group definitions in order of group number to record the protocol or device specific components of each group's definition. This serves to associate group numbers with group definitions which include port identification and device addresses. It further serves to track the status of each port, device and data group.

PSD PRIMARY FUNCTIONS

The primary role of the PSD is to provide protocol specific communication for exchanging data between the LAN-90 PCV Harmony Executive and devices, generally programmable controllers, addressed via the specified ports. This communication allows the Harmony Executive to request the PSD task to read data from or write data to the programmable controllers.

The Harmony Executive and Editor contain no external device or communication protocol specific information. This information is supplied by the PSD which contains all protocol and device specific definitions and algorithms. It includes logic to:

- Evaluate device and protocol specific configuration parameters entered via the Harmony editor.
- Build protocol messages for supported transactions.
- Recognize data addressing conventions and restrictions with respect to the supported devices.
- Adhere to the protocol specific transaction handling sequences for both normal and error replies.
- Perform data format conversion for all supported device data types.
- Perform appropriate extraction/insertion of data from/to transaction messages with respect to device data type.

The PSD task polls the Harmony Executive by sending `WORK_QUERY` request messages. With each `WORK_QUERY` message it opens the intertask communication (message) link to receive the next action required by the Harmony Executive. In general, the PSD task receives a request from the Harmony Executive for each data transaction that it performs. Once a request has been received, the PSD performs the action independently and on completion returns a reply message to the Harmony Executive.

The PSD monitors completion of each device transaction that it initiates. If the device does not respond within a preset time interval, the transaction is retried a number of times. After retries have been exhausted, the PSD task returns the device transaction status to the Harmony executive in the request response message. The reply (status) codes and description are listed in [Appendix B](#). The preset time interval (time-out) and the number of retries may be defined in the PSD start-up file with the u: and v: start-up parameters.

SINGLE PORT VS MULTIPLE PORTS

If the PSD is serving a single port (per the PSD start-up parameters), then Reply Blocked message exchange should be selected for polling the executive.

If a PSD is serving multiple ports, then the Not Reply Blocked message exchange should be selected to allow Executive requests to overlap. Request actions may sometimes have to be queued while a designated port is not available. When the port becomes available the suspended request at the head of the queue is automatically made active next.

The PSD initializes (opens with the specified RS-232-C protocol) all ports (primary and secondary) at start-up. During normal run time, it re-initializes failed ports and where designated, assigns redundant port active and standby status. To the extent possible, the PSD determines success or failure of port initialization and reports to the Executive in this respect.

PROTOCOL/DEVICE SPECIFIC ADDRESSING VALIDATION

The Modbus PSD evaluates all user entered port, device, group and PLC point address parameters.

All ports serviced by the PSD are defined in the start-up file. The port referenced in a data group definition must be one of the primary ports defined in the start-up file. The user cannot reference a data group to a secondary port.

Groups created for a specific Harmony data type, must associate with (map to) an appropriate Modbus data type (Table 4-1).

Harmony limits the data group size by Harmony data type. The Modbus PSD further limits data group size by related Modbus data type (Table 4-1).

Table 4-1. Group Size, Generic Versus Modbus Data Type

Harmony Data Type	Modbus Data Type	Data Type Descriptor	Maximum Harmony Group Size	Maximum Modbus Group Size
0	Holding or input register	Unsigned short integer (Intel format - low byte first)	256	125
1	Holding or input register	Signed short integer (Intel format - low byte first)	256	125
2	Holding or input register	Unsigned long integer (Intel format)	128	62
3	Holding or input register	Signed long integer (Intel format)	128	62

Table 4-1. Group Size, Generic Versus Modbus Data Type (continued)

Harmony Data Type	Modbus Data Type	Data Type Descriptor	Maximum Harmony Group Size	Maximum Modbus Group Size
4	Holding or input register	IEEE float (real 4 - Intel format)	128	62
5	Holding or input register	IEEE double (real 8 - Intel format)	64	31
8	Coil or discrete input	Byte oriented digital data	512	512
10	Holding or input register	BCD (binary coded decimal)	256	125
11	Holding or input register	BCD - 12 bits to 3 digits	256	125

APPENDIX A - MODBUS FUNCTIONS

MODBUS FUNCTIONS SUPPORTED BY THE MODBUS PSD

See Modicon document PI-MBUS-300 for more detail.

Read Internal Coils Query Message (1)

Table A-1. Read Internal Coils Query Message (1)

Addr	Func	Data Start		# Of Data		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
11	01	01	F4	00	C8	XX	XX

Reads coils 500 through 699 from device 17. Values in the above table are expressed in hexadecimal. Thus address 17 (hex 11), data address 500 (hex 01F4) and # of data 200 (hex 00C8).

Read Internal Coils Reply Message

Table A-2. Read Internal Coils Reply Message

Addr	Func	Byte Count	Coils 500-507	Coils 508-515	Coils 692-699	CRC	
						High Byte	Low Byte
11	01	19	XX	XX	XX	XX	

Reply message contains data bits for coils 500 through 699 in 25 (hex 19) bytes of data.

Read Discrete Inputs Query Message (2)

Table A-3. Read Discrete Inputs Query Message (2)

Addr	Func	Data Start		# Of Data		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
11	02	01	F4	00	C8	XX	XX

Reads discrete inputs 500 through 699 from device 17. Values in the above table are expressed in hexadecimal. Thus address 17 (hex 11), data address 500 (hex 01 F4) and # of data 200 (hex 00 C8).

Read Discrete Inputs Reply Message

Table A-4. Read Discrete Inputs Reply Message

Addr	Func	Byte Count	Discretes 500-507	Discretes 508-515	Discretes 692-699	CRC	
						High Byte	Low Byte
11	01	19	XX	XX	XX	XX	XX

Reply message contains data bits for discrete inputs 500 through 699 in 25 (hex 19) bytes of data.

Read Holding Registers Query Message (3)

Table A-5. Read Holding Registers Query Message (3)

Addr	Func	Data Start		# Of Data		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
15	03	00	FA	00	32	XX	XX

Reads holding registers 250 through 299 from device 21. Values in the above table are expressed in hexadecimal. Thus address 21 (hex 15), data address 250 (hex 00 FA) and # of data 50 (hex 00 32).

Read Holding Registers Reply Message

Table A-6. Read Holding Registers Reply Message

Addr	Func	Byte Count	Register 250		Register 299		CRC	
			High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
15	03	64	XX	XX	XX	XX	XX	XX

Reply message contains data for holding registers 250 through 299 in 100 (hex 64) data bytes.

Read Input Registers Query Message (4)

Table A-7. Read Input Registers Query Message (4)

Addr	Func	Data Start		# Of Data		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
15	04	01	F4	00	C8	XX	XX

Reads input registers 250 through 299 from device 21. Values in the above table are expressed in hexadecimal. Thus address 21 (hex 15), data address 250 (hex 00 FA) and # of data 50 (hex 00 32).

Read Input Registers Reply Message

Table A-8. Read Input Registers Reply Message

Addr	Func	Byte Count	Register 250		Register 299		CRC	
			High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
15	04	64	XX	XX	XX	XX	XX	XX

Reply message contains data for input registers 250 through 299 in 100 (hex 64) data bytes.

Force Single Coil Query Message (5)

Table A-9. Force Single Coil Query Message (5)

Addr	Func	Coil Address		Force Indicator		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
02	05	05	DC	FF	00	XX	XX

Forces coil number 1500 in device 2. Values in the above table are expressed in hexadecimal. Thus data address 1500 (hex 05 DC). With the force indicator bytes equal to hex FF 00, the coil is forced on. To force the coil off the force indicator must be set to hex 00 00.

Force Single Coil Reply Message

The reply message is identical to the Force Single Coil Query Message when the transaction is successful.

Preset Single Register Query Message (6)

Table A-10. Preset Single Register Query Message (6)

ADDR	FUNC	Register Address		Force Value		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
02	06	05	DC	02	EE	XX	XX

Set register number 1500 in device 2 to value 725. Values in the above table are expressed in hexadecimal. Thus data address 1500 (hex 05 DC) and force value 725 (hex 02 EE).

Preset Single Register Reply Message

The reply message is identical to the Preset Single Register Query Message when the transaction is successful.

Preset Multiple Registers Query Message (16)

Table A-11. *Preset Multiple Registers Query Message (16)*

Addr	Func	Register Address		# Of Registers		Preset Value		Preset Value		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
02	10	04	B0	00	02	00	64	00	C8	XX	XX

Set registers number 1200 through 1201 in device 2 to the decimal values 100 and 200 respectively. Values in the above table are expressed in hexadecimal. Thus data address 1200 (hex 04 B0) and preset values 100 (hex 00 64) and 200 (hex 00 C8).

Preset Multiple Registers Reply Message

Table A-12. *Preset Multiple Registers Reply Message*

Addr	Func	Register Address		# Of Registers		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
02	06	04	B0	00	02	XX	XX

The reply message includes all except the preset value fields of the query message when the transaction is successful.

Diagnostics - Return Query Data (8)

Table A-13. *Diagnostics - Return Query Data (8)*

Addr	Func	Subfunction		Data		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
05	08	00	00	00	00	XX	XX

Sends query data message to device 5. The subfunction value 0 (hex 00 00) defines the Return Query Data (loop back) diagnostic function. The data field is arbitrarily set to 0 (hex 00 00). This command message will be used only to verify redundant standby ports.

Diagnostics - Return Query Data Reply Message

Table A-14. Diagnostics - Return Query Data Reply Message

Addr	Func	Subfunction		Data		CRC	
		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
05	08	00	00	00	00	XX	XX

Reply message content is identical to the query message if the transaction is successful.

APPENDIX B - ERROR MESSAGE TABLE

ERROR MESSAGE TABLE

Table B-1. Error Message Table

Number	Message
00	No error
01	Transaction failed to complete - miscellaneous reason
02	Time-out Error On Communication
03	Error reply from host device
04	Corrupt reply - unrecognizable reply
05	Communication error with port device
06	Address error
07	Mode error
08	Mode error
09	Data type error
10	Spare 10
11	Spare 11
12	Spare 12
13	Spare 13
14	Spare 14
15	Spare 15
16	Spare 16
17	Unknown message command
18	A request for this group is already outstanding
19	Unsupported request
20	Port selection invalid
21	Device address invalid
22	Group definition invalid
23	Illegal PLC data type selected for group
24	Illegal PLC file type selected
25	Illegal port
26	Port queue full
27	Too many devices
28	Illegal device address
29	Group too large
30	Modbus reply data field size error
31	Harmony/Driver internal error
32	Group defn changed, data rejected
33	Cannot open start-up file

Table B-1. Error Message Table (continued)

Number	Message
34	No port defined in start-up file
35	Too many ports specified in start-up file
36	Illegal baud rate specified
37	For Modbus, RS-232-C no. of data bits must be 8
38	For Modbus, RS-232-C no. of stop bits must be 1 or 2
39	For Modbus, RS-232-C parity must be NO, ODD or EVEN
40	Invalid port parameter specified
41	Port initialization error
42	Receive message CRC error
43	Start-up file read error
44	Screen entry format error
45	No write access allowed
46	Group point reference out of range
47	No device types defined in start-up file
48	Message too long

APPENDIX C - MODBUS CABLE CONNECTIONS

MODBUS CABLE CONNECTIONS

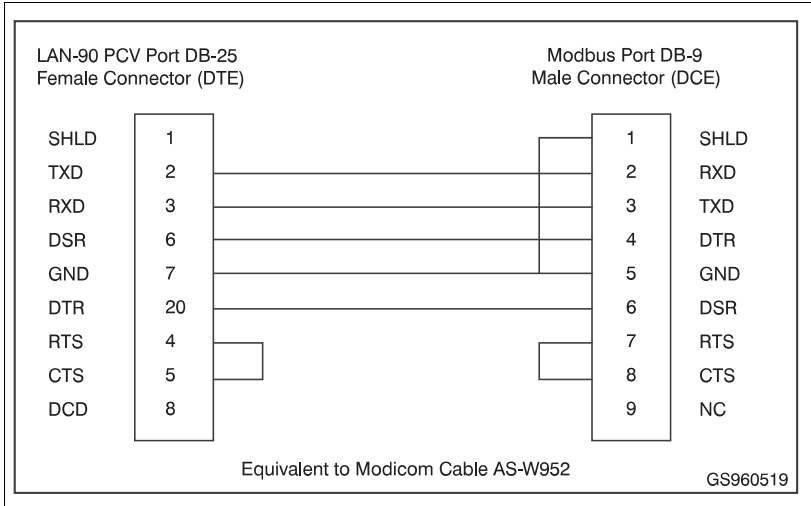


Figure C-1. Modbus Cable Connections

Visit Elsag Bailey on the World Wide Web at <http://www.ebpa.com>

Our worldwide staff of professionals is ready to meet *your* needs for process automation.
For the location nearest you, please contact the appropriate regional office.

AMERICAS

29801 Euclid Avenue
Wickliffe, Ohio USA 44092
Telephone 1-440-585-8500
Telefax 1-440-585-8756

ASIA/PACIFIC

152 Beach Road
Gateway East #20-04
Singapore 189721
Telephone 65-391-0800
Telefax 65-292-9011

EUROPE, AFRICA, MIDDLE EAST

Via Puccini 2
16154 Genoa, Italy
Telephone 39-10-6582-943
Telefax 39-10-6582-941

GERMANY

Graefstrasse 97
D-60487 Frankfurt Main
Germany
Telephone 49-69-799-0
Telefax 49-69-799-2406