INSTRUCTION MANUAL

700222J

COMMUNICATION SPECIFICATIONS FOR EN6021/EN6041 SERIES CONTROLS

FOR
MBTCP/RTU COMMUNICATION CARD 410383
EIP/MBTCP COMMUNICATION CARD 410384

MICROPROCESSOR BASED
Weld Sequence Controls
With
Solid State Thyristor Contactors

EN6021 Control Manual – Instruction Manual 700221
EN6041 Control Manual – Instruction Manual 700227

PCB 410384 – firmware version 1.4 and higher
EN6021 CPU – firmware version 2.00 and higher
EN6041 CPU – firmware version 2.00 and higher

ENTRON Controls, LLC.
1402 S. Batesville Road
Greer, South Carolina 29650
(864) 416-0190
FAX: (864) 416-0195
www.entroncontrols.com

RWMA® MEMBER

THIS PAGE IS INTENTIONALLY BLANK

ENTRON Controls, LLC.

MICROPROCESSOR BASED WELDING CONTROLS

OPERATION MANUAL FOR:
Model Series EN6021/EN6041
NEMA Type: ALL

! CAUTION !

READ THIS MANUAL COMPLETELY BEFORE ATTEMPTING TO INSTALL OR OPERATE THIS CONTROL

ENTRON Controls, LLC.
Greer, South Carolina 29650

REVISION CHANGES

For previous revisions of Communication Manual, contact ENTRON directly.

Rev A Changes (released 3/23/11)
Set Weld/No Weld function (Message ID 0x36) – added to Table 4-13 and Section 4.3.3.
Counter Value for Weld Log data – max changed to 60000 (Table 5-3A & Table 5-3B).
Counter Value for Error Log data – max changed to 60000 (Table 5-4A & Table 5-4B).
Table 5-10B Weld function – max for Parameter 1 changed to 100

Rev B Changes (released 5/17/11)
Time and Date Data Structure – added to Table 4-13, Section 4.3.3, and Table 5-2A; Section 5.5 and Table 5-5 were added; subsequent Sections and Tables renumbered.

Rev C Changes (released 3/22/12)
Added EIP/MBTCP Communication Card – Section 1.0 was revised; new Sections 3.0 and 5.0 were added.
Added Appendix A.

Section 1.0	Added “Conventions Used in Manual” Notice
Section 2.0	Renamed to “MBTCP/RTU Communication Card” Added Default IP address Section 2.2 was Section 3.0
Section 4.3.3	Message ID 0x22 – max of Weld Log index changed to 57599; min of Weld Log record count changed to 1 Message ID 0x24 – min of Error Log record count changed to 1
Section 6.0	Was Section 5.0 – all Sub-sections, Table numbers and Table references were renumbered
Table 6-2A	Weld Log index – max changed to 57600
Table 6-3A	Start Record Index & Index(1) – max changed to 57599
Table 6-3B	Index – max changed to 57599
Table 6-4A	Start Record Index, Index(1) & Index (2) – max changed to 999
Table 6-4B	Index – max changed to 999
Table 6-8	Byte Offset 24 – changed to “Water Saver Delay” RS Com Mode – changed descriptions for values 0-2; added value 4; changed max to 4

Rev D Changes (released 3/18/13)
Added EN6041 Series Control references throughout manual and control-specific parameters to Tables 6-1, 6-2A, 6-3B, 6-6, 6-7, 6-12, 6-15, and Appendix A.
Added LDT (Linear Displacement Transducer) option functions and parameters for EN6021 controls to Tables 6-2A, 6-2B, 6-3B, 6-6, and 6-15.
Added “For EN6021 only” for control-specific parameters to Tables 6-2A, 6-6, 6-8, 6-11B, 6-12, 6-15, and Appendix A.

Table 4-5	Added Sequencer flag 1 & 2 and Sequencer analog input/output 1 & 2
Table 4-9	Added Sequencer flag 1 & 2 and Sequencer analog output 1 & 2
Table 6-1	Control type – changed max to 255
Table 6-2A	Control type added
Table 6-2B	Several Error Flag descriptions corrected to match ENLINK/RPP2 labels
Table 6-3A	Weld Log data structure revised
Table 6-3B	Data Unit description changed and explanation added
Table 6-4A	Error Log data structure revised
Table 6-6	Valve Mode parameter changed
Table 6-7	Event Interval for EN6021 – values for Weld1 & 2nd Stage changed

REVISION CHANGES (cont.)

Rev D Changes (released 3/18/13) (cont.)

Table 6-11B	"Weld" function changed to "Spot-weld with Sch"
	Added Operation Code Values 29-32
Table 6-12	AC Port 33-36 – additional parameter information added
Table 6-15	Cal IPC & IPS (Byte Offset 36-50) – additional parameter information added

Rev E Changes (released 9/6/13)

Added Revision Changes listing and Revision Change indicators throughout manual.

Table 5-7	Attribute Code 0x6C Access Rule changed to "Get" & Data Structure changed to Table 6-1
	Added Attribute Code 0x75
Section 6.1	Added "FOR READ/GET FUNCTIONS" for clarification
Table 6-7	Event Delay[0] through [3] – max cycles changed to 98
Table 6-11A	Operation Code – max values of 32 for EN6021 & 30 for EN6041 added
Section 6.16	Added "FOR WRITE/SET FUNCTIONS" for clarification

Rev F Changes (released 2/26/14)

Table 6-2A	Added Log Memory Status
Table 6-8	Regulation Mode – added value 2
	Added Log Recording Mode
Table 6-15	Coil Sensitivity – changed ranges
	Max Current - changed minimum value

Rev G Changes (released 5/5/14)

Front Page	Address changed due to ENTRON Controls move.

Rev H Changes (released 7/15/14)

Front Page	Changed firmware version for EN6021 & EN6041 CPU (2.00).
Table 4-9	Corrected "Output port status of P3" to "Output port status of P2" (Address 911).
	Corrected "Output port status of P11" to "Output port status of P10" (Address 912).

Rev J Changes (released 10/24/14)

Front Page	Changed firmware version for PCB 410384 (1.4).
Table 3-1	"Flashing Amber" changed to "Flashing Red" due to part change.
Page 65	Added Firmware History page.

TABLE OF CONTENTS

FIGURES

TABLE OF CONTENTS (cont.)

TABLES

THIS PAGE IS INTENTIONALLY BLANK

1.0 INTRODUCTION

The EN6021 and EN6041 Controls offer a number of options for communication of data and status information to a remote device. The communication options use either EtherNet/IP protocol or Modbus© protocol over Ethernet, serial RS232 or serial RS485.

Either communication option requires an additional Communication Card to be installed on the EN6021/© EN6041 Control. Two types of cards are available:

MBTCP/RTU Communication Card (P/N 730014-007) – implements Modbus/TCP communication over Ethernet networks or using serial RS232 or RS485 connections.

EIP/MBTCP Communication Card (P/N 730014-013) – implements either EtherNet/IP or Modbus/TCP communication.

The type of communication interface is set using Communication Card (COM) parameter in **Configure© Menu**. When using EIP/MBTCP Communication Card, set this parameter to EIP+MB Ethernet mode. When using MBTCP/RTU Communication Card, parameter values include MB Ethernet, MB RS232 RTU, MB RS485 RTU, or Label Printing mode – set to appropriate value for implementation. See Instruction Manual 700221 (for EN6021) or 700227 (for EN6041) for complete programming details. The user can modify these settings through RPP2 Programming Pendant or a remote device which is communicating with control. New settings will take effect on control's next reboot.

Ethernet communication offers multiple connections between EN6021/EN6041 Controls and remote device. Typical Ethernet connection is shown in Figure 1-1.

Figure 1-1. *Typical Ethernet connection*

1.0 INTRODUCTION (cont.)

Serial RS232 connection provides one-to-one communication between EN6021/EN6041 Control and remote device. Typical RS232 connections are shown in Figure 1-2.

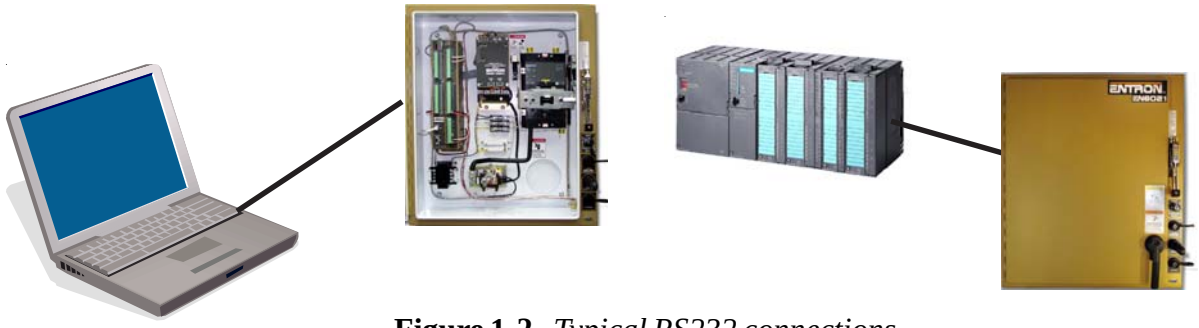


Figure 1-2. *Typical RS232 connections*

Serial RS485 connection provides multiple-connection communication between up to 99 EN6021/ EN6041 Controls and remote device. Typical RS485 connection is shown on Figure 1-3.

Figure 1-3. *Typical RS485 connection*

NOTICE ©

CONVENTIONS USED IN THIS MANUAL

To follow the CIP and Modbus documents, these terms are used to describe type/structure of data:

Byte	1-byte unsigned data
Word	2-byte unsigned data
DWORD	4-byte unsigned data
USINT	1-byte unsigned data
UINT	2-byte unsigned data
UDINT	4-byte unsigned data

2.0 MBTCP/RTU COMMUNICATION CARD ©

2.1 ETHERNET INTERFACE

The Ethernet port on MBTCP/RTU Communication Card allows connection of multiple controls with ENLINK, PLC and touch screens over Modbus-TCP/IP protocol.

2.1.1 SPECIFICATIONS

Ethernet: 10/100BASE-T
Speed: Autodetect 10/100 Mb
Connector: RJ45
Default IP address: 192.168.0.100 ©

2.1.2 CONNECTOR

The connector for Ethernet connection is P9 on the CPU. Signal definitions are shown in Table 2-1.

Table 2-1. *Ethernet interface signals pin out*

Signal Name	Direction	Contact	Primary Function
TX+	Out	1	Differential Ethernet transmit data +
TX–	Out	2	Differential Ethernet transmit data –
RX+	In	3	Differential Ethernet receive data +
RX–	In	6	Differential Ethernet receive data –
Not used		4	Terminated
Not used		5	Terminated
Not used		7	Terminated
Not used		8	Terminated
SHIELD			Chassis ground

There are two status LEDs on the Ethernet connector, shown in Figure 2-1. The LED functions are shown in Table 2-2.

Table 2-2. *Ethernet connector LED functions*

Color	Link LED (Left)	Activity LED (Right)
Off	No Link	No Activity
Amber	10 Mbps	Half Duplex
Green	100 Mbps	Full Duplex

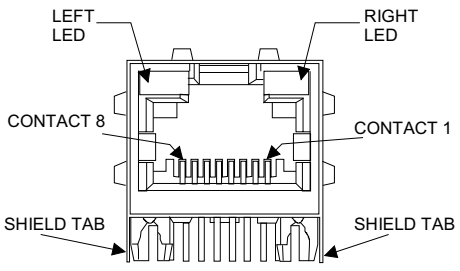


Figure 2-1.
Ethernet connector LEDs

2.2 RS232 AND RS485 SERIAL INTERFACE ©

The RS232 and RS485 serial ports on MBTCP/RTU Communication Card allow connection of single or multiple controls with PLC, touch screens or other Modbus masters over Modbus-RTU protocol.

2.2.1 SPECIFICATIONS

Control ID number:	1 through 99
Baud rate:	19200
Parity:	EVEN
Mode:	RTU
Coding system:	8-bit binary
Bits per byte:	1 start bit
	8 data bits, least significant bit sent first
	1 bit for parity completion
	1 stop bit

2.2.2 CONNECTOR

The connector for RS232 and RS485 connection is P8 on the CPU. Signal definition is shown in Table 2-3.

Table 2-3. *RS232 and RS485 interface signals pin out*

Signal Name	Direction	Contact	Primary Function
RS232 RXD	In	2	Receive pin for RS232
RS232TXD	Out	3	Transmit pin for RS232
RS485 A	In/Out	4	Pin A for RS485
RS485 B	In/Out	8	Pin B for RS485
COM GND	GND	5,9	Ground for communication

2.2.3 TERMINAL RESISTOR SETTING FOR RS485 COMMUNICATION

If control is connected at the end of RS485 communication line, a jumper block (P/N 331139) must be installed on JP2 of MBTCP/RTU Communication Card to enable termination resistor.

3.0 EIP/MBTCP COMMUNICATION CARD ©

3.1 ETHERNET INTERFACE

The Ethernet port on EIP/MBTCP Communication Card provides the ability to network multiple controls using PLC or other devices which use Common Industrial Protocol (CIP™). This card also provides Modbus communication for devices which use Modbus/TCP protocol.

3.1.1 SPECIFICATIONS

Ethernet:	10/100BASE-T
Speed:	Autodetect 10/100 Mb
Connector:	RJ45
Support cable:	Regular direct Ethernet cable and crossover Ethernet cable

3.1.2 CONNECTOR

The connector for Ethernet connection is P9 on the CPU. The Ethernet interface has same pin layout shown in Table 2-1.

There are two status indicator LEDs on the Ethernet connector, shown in Figure 3-1. The status indicator LED functions are described in Table 3-1.

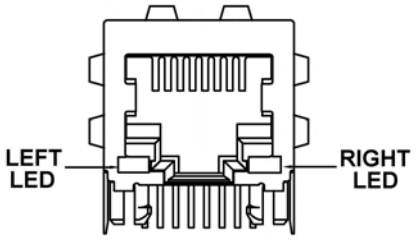


Figure 3-1.
Status Indicator LEDs

Table 3-1. *Status indicator LED functions* Ⓙ

Color/Status	Module Status (Left)	Network Status (Right)
Steady Off	No power	No power
Flashing Red *	Control selects incorrect communication mode; communication card will not work	N/A
Flashing Green	N/A	No CIP connections are established
Steady Green	Control selects correct communication mode	At least one CIP connection is established

* Flashing Amber on PCB 410384 Rev A and B

3.1.3 JUMPER FOR FACTORY DEFAULT SETTING

Jumper JP3 on EIP/MBTCP Communication Card is used to force use of default settings for Ethernet interface configuration (see Figure 3-2). The default Ethernet interface configuration includes:

Default IP address:	192.168.0.100
Default Network mask:	255.255.255.0
Default Gateway:	192.168.0.1

If Ethernet interface configuration has been set to an unknown value, Communication Card will not communicate with other devices. To solve this problem, place a jumper block on Jumper JP3 and reboot the system; card will now use default settings.

After a new Ethernet interface configuration setting is sent to the card, remove the jumper block on Jumper JP3. The new setting will take effect on next reboot.

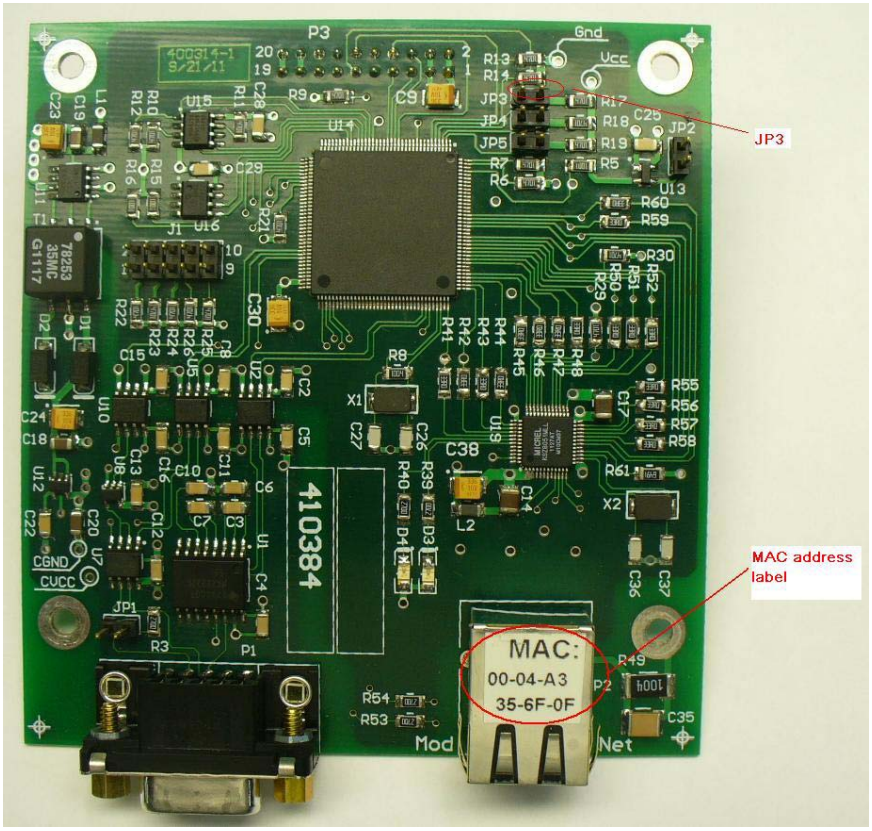


Figure 3-2.
Jumper JP3 and MAC address label

3.1.4 MAC ADDRESS LABEL

The MAC address label is found on the top side of Ethernet connector as Figure 3-2 shows.

3.2 RS232/RS485 INTERFACE

The EIP/MBTCP Communication Card has both RS232 and RS485 ports. These ports are not functional at present time.

4.0 MODBUS PROTOCOL

The EN6021/EN6041 implements RS232 and RS485 serial communication via Modbus RTU (Remote Terminal Unit) mode protocol and implements Ethernet communication via Modbus TCP/IP protocol. Full details may be obtained from www.modbus.org.

The Modbus protocol defines a simple Protocol Data Unit (PDU) to transfer data as Figure 4-1 shows.

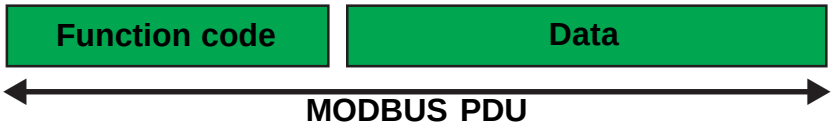


Figure 4-1. *Modbus Protocol Data Unit*

In Modbus PDU, function code indicates to server (EN6021/EN6041) what kind of action to perform. EN6021/EN6041 implements Modbus function code 4 (0x04), function code 16 (0x10) and function code 43 (0x2B), MEI value 128 (0x80).

The function code can be followed by a data field that contains request and response parameters.

4.1 MODBUS FRAME OVER SERIAL INTERFACE (RS232 & RS485)

A Modbus frame over serial interface includes three parts: Address field, Modbus PDU and Error Checking field (CRC) as shown in Figure 4-2.

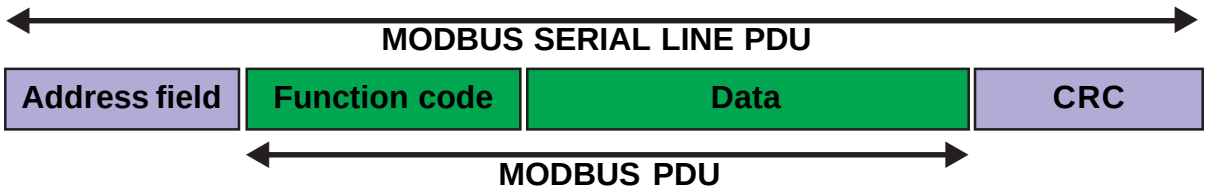


Figure 4-2. *Modbus frame over serial interface*

Address field – On Modbus serial line, address field only contains slave address of EN6021/EN6041 Control (ID number). Range of valid slave addresses for the EN6021/EN6041 are 1-99 decimal. Modbus master (PLC, touch screen or other Modbus device) addresses EN6021/EN6041 Control by placing slave address in address field of message. When the control returns its response, it places its own address in response address field to let the master know which slave is responding.

Modbus PDU – Includes function code and optional data as Figure 4-1 shows.

Error Checking field – Result of "Redundancy Checking" calculation that is performed on message contents. Cyclical Redundancy Checking (CRC) result is used in Error Checking field on EN6021/EN6041.

4.2 MODBUS ADU OVER TCP/IP INTERFACE

Modbus Application Data Unit (ADU) frame over TCP/IP interface includes two parts: MBAP header and Modbus PDU as Figure 4-3 shows.

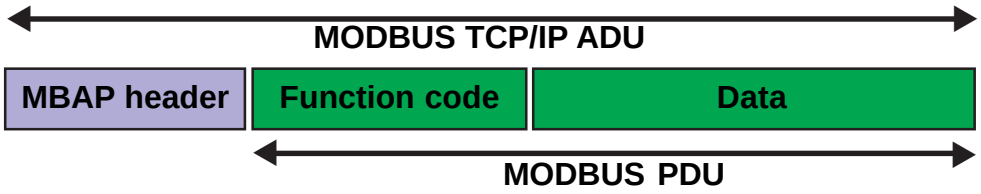


Figure 4-3. *Modbus Application Data Unit (ADU) over serial interface*

MBAP header – MBAP header (Modbus Application Protocol header) is used on TCP/IP to identify Modbus Application Data Unit. It includes Transaction Identifier (2 bytes), Protocol Identifier (2 bytes), Length (2 bytes) and Unit Identifier (1 byte).

Modbus PDU – Same as Modbus PDU in Modbus frame over serial interface (see Section 4.1).

4.3 MODBUS FUNCTION CODES

EN6021/EN6041 implements Modbus function code 4 (0x04), function code 16 (0x10) and function code 43 (0x2B), MEI value 128 (0x80), as shown in Table 4-1.

Table 4-1. *Modbus function codes*

Function Code	Modbus Description	EN6021/EN6041 Function
04 (0x04)	Read input registers	Read Control setting and status from control
16 (0x10)	Write multiple registers	Write Control setting to control
43 (0x2B)	Encapsulated Interface Transport	Read Control setting and status from control, and write Control setting to control

4.3.1 MODBUS FUNCTION CODE 4

Modbus function code 4 is used to read EN6021/EN6041 input port status, weld schedule setting and other control status/data. Request PDU specifies starting address of data and quantity of data. Data in response message are packed as two bytes per word, with binary contents right justified within each byte. For each word (2 bytes) data, first byte contains high order bits and second byte contains low order bits.

When Modbus master requests the EN6021/EN6041, the PDU is 5 bytes in length.

Table 4-2. *Function code 4 data request PDU*

Address Offset (bytes)	Data Name	Value
0	Function code	0x04
1-2	Data starting address	
3-4	Data's quantity in word	

4.3.1 MODBUS FUNCTION CODE 4 (cont.)

When control responds to master, if no error, the PDU is 2 + data quantity x 2 bytes in length.

Table 4-3. *Function code 4 response PDU (without error)*

Address Offset (bytes)	Data Name	Value
0	Function code=0x04	0x04
1	Bytes count	
2-3	First word data: data(0)	
4-5	Second word data: data(1)	
…	…	
…	Last word data: data(quantity-1)	

If control sends back an error message, the PDU will be 2 bytes in length.

Table 4-4. *Function code 4 response PDU (with error)*

Address Offset (bytes)	Data Name	Value
0	Function code	0x84
1	Exception code	1: wrong function code 2: wrong address 3: wrong quantity 4: other errors

The EN6021/EN6041 maps data/control status to addresses shown in Table 4-5.

Table 4-5. *Function code 4 data address and name*

Address	Data Name	Data Description	
700–799	Data for Control status		
…			
900	Input port status of P1	Bit15: PS1 Bit14: NW1 Bit13: TC1 Bit12: Estop Bit11: Start4 Bit10: Start3 Bit9: Start2 Bit8: Start1 Bit7–Bit0: reserved	
901	Input port status of P3	Bit15-0: Input16–Input01	
902	Input port status of P11	Bit15-0: Input32–Input17	
…			
930	Sequencer flag 1	Bit15-0: F16–F01	Ⓓ
931	Sequencer flag 2	Bit15-0: F32–F17	Ⓓ
…			
950	Sequencer analog input1		Ⓓ
951	Sequencer analog input2		Ⓓ
952	Sequencer analog output1		Ⓓ
953	Sequencer analog output2		Ⓓ
…			
1000–5799	Data for weld schedule 0-99		

4.3.1 MODBUS FUNCTION CODE 4 (cont.)

100 words (200 bytes) of control status data are mapped to addresses from 700 through 799 of Modbus function code 4. Structure of control status data is shown in Tables 6-2A and 6-2B.

The Sequencer's flags (F32–F01) are mapped to function code 4 address 930–931. Ⓓ

The Sequencer's analog ports (analog input1, analog input2, analog output1, and analog output2) are mapped to function code 4 address 950–953.

Data's address calculation for weld schedules – Parameters of weld schedules 0 through 99 are mapped to Modbus function code 4 addresses 1000–5799. Each weld schedule includes 48 words (96 bytes) data. Weld schedule 0 starts at address 1000 and ends at address 1047, then each subsequent schedule uses next 48 words address, with last schedule using address 5752–5799. The address offset of each parameters in weld schedules can be found in Table 6-6.

Based on the above description, the following equation (Equation 4-1) is used to calculate a parameter's address for Modus function code 4:

Equation 4-1:

Parameter's address = Start address (1000) + [Weld schedule number x 48] + Parameter address offset in Weld Schedule

For example, to obtain the address of Weld1 time of schedule 5, calculation will be:

Start address = 1000
Weld schedule number = 5
Weld1 time address offset in schedule = 6
Address of Weld1 time of schedule 5 = 1000 + [5x48] + 6 = 1246

4.3.2 MODBUS FUNCTION CODE 16

Modbus function code 16 is used to write EN6021/EN6041 output port state, weld schedule setting and other control states. Request PDU of function code 16 specifies starting address of data and quantity of data. Data in response message are packed as two bytes per word, with binary contents right justified within each byte. For each word (2 bytes) data, first byte contains high order bits and second byte contains low order bits.

When Modbus master requests the EN6021/EN6041, the PDU is (6 + data quantity x 2) bytes in length.

Table 4-6. *Function code 16 data request PDU*

Address Offset (bytes)	Data Name	Value
0	Function code	16 (0x10)
1-2	Starting address	
3-4	Quantity of data	
5	Byte count of data	
6-7	First setting data: data(0)	
8-9	Second setting data: data(1)	
…	…	
…	Last setting data: data(quantity-1)	

4.3.2 MODBUS FUNCTION CODE 16 (cont.)

When control responds to master, if no error, the PDU is 5 bytes in length.

Table 4-7. *Function code 16 response PDU (without error)*

Address Offset (bytes)	Data Name	Value
0	Function code	0x10
1-2	Starting address	
3-4	Quantity of data	

If control sends back an error message, the PDU will be 2 bytes in length.

Table 4-8. *Function code 16 response PDU (with error)*

Address Offset (bytes)	Data Name	Value
0	Function code	0x90
1	Exception code	1: wrong function code 2: wrong address 3: wrong quantity 4: other errors

The EN6021/EN6041 maps data/output states to addresses shown in Table 4-9.

Table 4-9. *Function code 16 data address and name*

Address	Data Name	Data Description	
911	Output port status of P2	Bit15-0: Output16–Output01	(H)
912	Output port status of P10	Bit15-0: Output32–Output17	(H)
…			
930	Sequencer flag 1	Bit15-0: F16–F01	(D)
931	Sequencer flag 2	Bit15-0: F32–F17	(D)
…			
952	Sequencer analog output1		(D)
953	Sequencer analog output2		(D)
…			
1000–5799	Data for weld schedule 0-99		

The Sequencer's flags (F32–F01) are mapped to function code 16 address 930–931. (D)

The Sequencer's analog output port 1 and port 2 are mapped to function code 16 address 952–953.

Data's address calculation for weld schedules is same as function code 4 (Equation 4-1).

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80

This function is used for general data exchange. This frame holds Modbus PDU having function code 43 (0x2B), MEI value 128 (0x80), followed by actual data to be exchanged.

Table 4-10 shows data structure of function code 43 data request PDU.

Table 4-10. *Function code 43 data request PDU*

Address Offset (bytes)	Data Name	Value
0	Function code	0x2B
1	MEI Type	0x80
2	Message ID	
3…	Optional message data	

When EN6021/EN6041 responds to master, if no error, the PDU structure will be (3 bytes + message data) in length, as Table 4-11 shows. If there are errors, the PDU structure will be 3 bytes in length, as Table 4-12 shows.

Table 4-11. *Function code 43 response PDU (without error)*

Address Offset (bytes)	Data Name	Value
0	Function code	0x2B
1	MEI Type	0x80
2	Message ID or ACK	
3…	Optional message data	

Table 4-12. *Function code 43 response PDU (with error)*

Address Offset (bytes)	Data Name	Value
0	Function code	0x90
1	Exception code	1: wrong function code 2: wrong address 3: wrong quantity 4: other errors

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Table 4-13 shows functions which EN6021/EN6041 supports for Modbus function code 43 (0x2B), MEI type 128 (0x80).

Table 4-13. *Functions supported for function code 43 (0x2B), MEI type 128 (0x80)*

Message ID	Function	
0x20	Read control description from control	
0x21	Read control status flags from control	
0x22	Read Weld Log data from control	
0x23	(reserved)	
0x24	Read Error Log data from control	
0x25 – 0x2F	(reserved)	
0x30	Reset Errors	
0x31	Reset Counters	
0x32	Reset Stepper	
0x33	Reset Sequencer	
0x34	Reset Weld Log	
0x35	Reset Error Log	
0x36	Set Weld/No Weld	(A)
0x37	Write time and date to control	(B)
0x38 – 0x3F	(reserved)	
0x40	Read Weld Schedule data from control	
0x41	Write Weld Schedule data to control	
0x42	Read Event data from control	
0x43	Write Event data to control	
0x44	Read Configuration data from control	
0x45	Write Configuration data to control	
0x46	Read Counter data from control	
0x47	Write Counter data to control	
0x48	Read Stepper data from control	
0x49	Write Stepper data to control	
0x4A	Read Sequencer data from control	
0x4B	Write Sequencer data to control	
0x4C	(reserved)	
0x4D	(reserved)	
0x4E	Read I/O Map data from control	
0x4F	Write I/O Map data to control	
0x50	Read Use Schedule data from control	
0x51	Write Use Schedule data to control	
0x52	Read Error Output Map data from control	
0x53	Write Error Output Map data to control	
0x54	Read Calibration data from control	
0x55	Write Calibration data to control	
0x56 – 0x5F	(reserved)	
0x60	Write description to control	

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Detailed descriptions of each Message ID data are explained below.

Message ID 0x20: Read control description from control

Control receives:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	Message ID (0x20)	
Control responds:			
When no error:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	Message ID (0x20)	
	Byte 3–66:	Control description data from Table 6-1	
When error:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	NAK	

Message ID 0x21: Read control status flags from control

Control receives:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	Message ID (0x21)	
Control responds:			
When no error:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	Message ID (0x21)	
	Byte 3–202:	Control status data from Tables 6-2A and 6-2B	
When error:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	NAK	

Message ID 0x22: Read Weld Log data from control

Control receives:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	Message ID (0x22)	
	Byte 3:	High byte of Weld Log record start index (0-57599)	Ⓒ
	Byte 4:	Low byte of Weld Log record start index (0-57599)	Ⓒ
	Byte 5:	Weld Log record count (1–6)	Ⓒ
Control responds:			
When no error:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	Message ID (0x22)	
	Byte 3 up to 194:	Weld Log data from Tables 6-3A and 6-3B	
When error:	Byte 0:	Function Code 0x2B	
	Byte 1:	MEI type 0x80	
	Byte 2:	NAK	

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x24: Read Error Log data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x24)
	Byte 3:	High byte of Error Log record start index (0-999)
	Byte 4:	Low byte of Error Log record start index (0-999)
	Byte 5:	Error Log record count (1–12)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x24)
	Byte 3 up to 194:	Error Log data from Tables 6-4A and 6-4B
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x30: Reset Errors

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x30)
Control responds:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK

Message ID 0x31: Reset Counters

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x31)
	Byte 3:	Reset code (0-3) Reset code=0: Reset none Reset code=1: Reset part counter Reset code=2: Reset weld counter Reset code=3: Reset part counter + weld counter
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x32: Reset Stepper

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x32)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x33: Reset Sequencer

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x33)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x34: Reset Weld Log

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x34)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x35: Reset Error Log

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x35)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x36: Set Weld/No Weld Ⓐ

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x36)
	Byte 3:	State code (0–2)
		State code=0: Set nothing
		State code=1: Set control to No Weld mode
		State code=2: Set control to Weld mode
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x37: Write time and date to control Ⓑ

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x37)
	Byte 3–130:	Time and date data from Table 6-5
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x40: Read Weld Schedule data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x40)
	Byte 3:	Schedule number (0–99)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x40)
	Byte 3:	Schedule number (0–99)
	Byte 4–51:	Weld Schedule data from Table 6-6
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x41: Write Weld Schedule data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x41)
	Byte 3:	Schedule number (0–99)
	Byte 4–51:	Weld Schedule data from Table 6-6
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x42: Read Event data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x42)
	Byte 3:	Schedule number (0–99)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x42)
	Byte 3:	Schedule number (0–99)
	Byte 4–35:	Event data from Table 6-7
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x43: Write Event data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x43)
	Byte 3:	Schedule number (0–99)
	Byte 4–35:	Event data from Table 6-7
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x44: Read Configuration data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x44)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x44)
	Byte 3–66:	Configuration data from Table 6-8
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x45: Write Configuration data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x45)
	Byte 3–66:	Configuration data from Table 6-8
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x46: Read Counter data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x46)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x46)
	Byte 3–35:	Counter data from Table 6-9
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x47: Write Counter data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x47)
	Byte 3–66:	Counter data from Table 6-9
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x48: Read Stepper data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x48)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x48)
	Byte 3–98:	Stepper data from Table 6-10
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x49: Write Stepper data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x49)
	Byte 3–98:	Stepper data from Table 6-10
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x4A: Read Sequencer data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x4A)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x4A)
	Byte 3:	Section number from Table 6-11C
	Byte 4–63:	One section of Sequencer data from Tables 6-11A & 6-11B
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x4B: Write Sequencer data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x4B)
	Byte 3:	Section number from Table 6-11C
	Byte 4–63:	One section of Sequencer data from Tables 6-11A & 6-11B
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x4E: Read I/O Map data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x4E)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x4E)
	Byte 3–202:	I/O Map data from Table 6-12
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x4F: Write I/O Map data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x4F)
	Byte 3–202:	I/O Map data from Table 6-12
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x50: Read Use Schedule data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x50)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x50)
	Byte 3–34:	Use Schedule data from Table 6-13
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x51: Write Use Schedule data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x51)
	Byte 3–34:	Use Schedule data from Table 6-13
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x52: Read Error Output Map data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x52)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x52)
	Byte 3–224:	Error Output Map from Table 6-14
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x53: Write Error Output Map data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x53)
	Byte 3–226:	Error Output Map from Table 6-14
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x54: Read Calibration data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x54)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x54)
	Byte 3–224:	Calibration data from Table 6-15
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

4.3.3 MODBUS FUNCTION CODE 43 (0X2B) MEI TYPE 0X80 (cont.)

Message ID 0x55: Write Calibration data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x55)
	Byte 3–226:	Write Calibration data from Table 6-15
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

Message ID 0x60: Write description to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x60)
	Byte 3–34:	Control description data from Table 6-16
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	ACK
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	NAK

5.0 ETHERNET/IP PROTOCOL ©

5.1 INTRODUCTION

EtherNet/IP (Ethernet/Industrial Protocol) uses CIP (Common Industrial Protocol) which transports data over standard IEEE 802.3 Ethernet networks. Full details may be obtained from www.odva.org. ENTRON Controls, LLC., is a member of ODVA (Vendor ID is 1242).

The EIP/MBTCP Communication Card implements CIP protocol following ODVA specifications – *The CIP Networks Library Volume 1: Common Industrial Protocol* and *The CIP Networks Library Volume 2: EtherNet/IP Adaptation of CIP*.

The EIP/MBTCP Communication Card accepts up to two TCP connections and one UDP connection over port number 44818 (0xAF12). In addition, this card supports one Modbus connection over port number 502. With this feature, control can communicate with a Modbus device when it is communicating with EtherNet/IP devices.

According to CIP specifications and EN6021/EN6041 Control features, the EIP/MBTCP Communication Card transfers data through Explicit Messages. The Explicit Message should be transmitted by Unconnected Messages (UCMM packets) over TCP/IP connections.

5.2 EIP/MBTCP COMMUNICATION CARD OPERATION

The device will send an Explicit Message to Communication Card to get/set the data in the control. The parameters of an Explicit Message should include Service Code, Class Code, Attribute Code, and may include Instance Code.

The Service Code should be 0x0e for getting single Attribute data and 0x10 for setting single Attribute data. Some CIP objects support Get_Attributes_All Service for which Service Code should be 0x01.

The Class Code should be the Class ID of the Objects which Communication Card supports. The current version of the firmware supports three CIP standard Objects: Class Code 0x01 – Identity Object, 0xF5 – TCP/IP Interface Object, 0xF6 – Ethernet Link Object and several ENTRON-defined vendor Objects.

The Instance Code is required when the device tries to get/set Instance Attributes. The Communication Card supports only one Instance, so when the Instance Code is required, value should always be 1.

The Attribute Code is required for all Objects indicating which Attribute data will be gotten/set. Each Object will have a different value range for its Attribute Codes.

5.2.1 SUPPORTED CIP STANDARD OBJECTS

IDENTITY OBJECT

The Identity Object provides identification of and general information about the device. The parameter will be set as:

Service Code:	0x01 – to get all Attribute data 0x0E – to get single Attribute data 0x10 – to set single Attribute data
Class ID:	0x01
Instance ID:	1
Attribute:	1 through 7

Instance Attributes

The Instance Attribute data are described in Table 5-1.

Table 5-1. *Identity Object Instance Attributes*

Attribute Code	Access Rule	Name	Data Type
1	Get	Vendor ID	UINT
2	Get	Device Type	UINT
3	Get	Product Code	UINT
4	Get	Major Revision	USINT
		Minor Revision	USINT
5	Get	Device Status	UINT
6	Get	Serial Number	UDINT
7	Get	Product Name	SHORT STRING

Semantics of Instance Values

1. **Vendor ID** – Vendor IDs are managed by ODVA. ENTRON's Vendor ID is 1242.
2. **Device Type** – ENTRON assigns the value 0x0C to Communication Card as Device Type.
3. **Product Code** – ENTRON assigns the value 10384 to Communication Card as Product Code.
4. **Revision** – Consists of Major and Minor Revisions, identifying Revision of the firmware.
5. **Device Status** – Represents current status of Communication Card. ENTRON only uses the ninth bit (D8) to indicate the Minor Recoverable Fault for this card.

   For EIP/MBTCP Communication Card to work with EN6021/EN6041 Control, set Communication Card parameter to **EIP+MB Ethernet** mode in **Configure Menu** (see Section 1.0). If this card detects that the control has selected the correct communication mode, the D8 bit of Device Status will be 0; if the control selects an incorrect communication mode, the D8 bit of Device Status will be 1.
6. **Serial Number** – Represents the Serial Number of Communication Card.
7. **Product Name** – ENTRON assigns the text string "ENTRON EN6000" to Communication Card as the Product Name.

5.2.1 SUPPORTED CIP STANDARD OBJECTS (cont.)

TCP/IP OBJECT

The TCP/IP Interface Object provides the function to configure a device's TCP/IP network interface. The Interface Attribute data include Communication Card's IP Address, Network Mask, Gateway Address and other settings. The parameter will be set as:

Service Code:	0x0E – to get single Attribute data
	0x10 – to set single Attribute data
Class ID:	0xF5
Instance ID:	1 for getting/setting Instance Attribute only
Attribute:	1 through 7 for getting Class Atrribute
	1 through 6 for getting/setting Instance Attribute

Class Attributes

The Class Attribute data are described in Table 5-2.

Table 5-2. *TCP/IP Object Class Attributes*

Attribute Code	Access Rule	Name	Data Type	Description	Value
1	Get	Revision	UINT	Revision of this object	2
2	Get	Max Instance	UINT	Maximum instance number	1
3	Get	Number of Instances	UINT	Largest instance number	1
4	Get	Optional Attribute List	UINT	List of optional instance attributes utilized in object class implementation	0
5	Get	Optional Service List	STRUCTURE	List of optional services utilized in object class implementation	
		Number Services	UINT		2
		Optional Services	UINT		0x000e
			UINT		0x0010
6	Get	Maximum Number Class Attributes	UINT	Attribute code of last class attribute of class definition implemented in card	7
7	Get	Maximum ID Number Instance Attributes	UINT	Attribute code of last instance attribute of class definition implemented in card	6

Instance Attributes

The Instance Attribute data are described in Table 5-3.

Table 5-3. *TCP/IP Object Instance Attributes*

Attribute Code	Access Rule	Name	Data Type
1	Get	Status	DWORD
2	Get	Configuration Capability	DWORD
3	Get/Set	Configuration Control	DWORD
4	Get	Physical Link Object	STRUCTURE
5	Get/Set	Interface Configuration	STRUCTURE
6	Get	Host Name	STRING

5.2.1 SUPPORTED CIP STANDARD OBJECTS (cont.)

Semantics of Instance Values

1. **Status** – Interface status. Communication Card only uses Bit3-0 to indicate following status:
   - Bit3-0 = 1 Interface Configuration attribute contains valid configuration obtained from DHCP or nonvolatile storage.
   - Bit3-0 = 2 Jumper block is installed on Jumper JP3 on Communication Card; Interface Configuration attribute uses ENTRON default values for configuration.
2. **Configuration Capability** – This attribute is a bitmap that indicates the device's support for optional network configuration capability.

   The value is 0x00000074 which indicates:
   - Bit 2 = 1 Device is capable of obtaining its network configuration via DHCP.
   - Bit 4 = 1 Interface Configuration attribute is settable.
   - Bit 5 = 1 IP Address member of Interface Configuration attribute can be obtained from hardware settings (Jumper JP3).
   - Bit 6 = 1 Device requires a reboot in order for changes to Interface Configuration attribute to take effect.
3. **Interface control flags** – Configuration Control attribute is a bitmap used to control network configuration.

   The value will be 0 or 2 to indicate:

   0 = Device shall use statically-assigned IP configuration values.

   2 = Device shall obtain its interface configuration values via DHCP.
4. **Path to Physical Link Object** – Identifies the object associated with the underlying physical communications interface.

   The values are three words 0x0002, 0xF620 and 0x0124 to indicate:
   - 0x0002 2 parameters
   - 0xF620 Ethernet Link object (Class ID=F6)
   - 0x0124 Instance ID=01
5. **Interface Configuration** – Contains configuration parameters required for Communication Card to operate as a TCP/IP node. The contents of this attribute include parameters shown in Table 5-4.

Table 5-4. *Interface Configuration Attribute*

Name		Meaning	Default Setting
IP address	Get/Set	Communication Card's IP address	192.168.0.100
Network mask		Communication Card's network mask	255.255.255.0
Gateway address		Communication Card's gateway	192.168.0.1
Name server		Communication Card does not use this attribute	
Name server 2		Communication Card does not use this attribute	
Domain name		Communication Card does not use this attribute	

6. **Host Name** – For this Communication Card, the value is "ENTRON EN6000".

5.2.1 SUPPORTED CIP STANDARD OBJECTS (cont.)

ETHERNET LINK OBJECT

The Ethernet Link Object maintains the status information for an IEEE 802.3 communications interface. The Link Attribute data include Interface Speed, Interface Flags and other information. The parameter will be set as:

Service Code:	0x0E – to get single Attribute data
	0x10 – to set single Attribute data
Class ID:	0xF6
Instance ID:	1 for getting/setting Instance Attribute only
Attribute:	1 through 7 for getting Class Atrribute
	1 through 3 for getting/setting Instance Attribute

Class Attributes

The Class Attribute data are described in Table 5-5.

Table 5-5. *Ethernet Link Object Class Attributes*

Attribute Code	Access Rule	Name	Data Type	Description	Value
1	Get	Revision	UINT	Revision of this object	1
2	Get	Max Instance	UINT	Maximum instance number	1
3	Get	Number of Instances	UINT	Largest instance number	1
4	Get	Optional Attribute List	UINT	List of optional instance attributes utilized in object class implementation	0
5	Get	Optional Service List	STRUCTURE	List of optional services utilized in object class implementation	
		Number Services	UINT		2
		Optional Services	UINT		0x000e
			UINT		0x0010
6	Get	Maximum Number Class Attributes	UINT	Attribute code of last class attribute of class definition implemented in card	7
7	Get	Maximum ID Number Instance Attributes	UINT	Attribute code of last instance attribute of class definition implemented in card	3

Instance Attributes

The Instance Attribute data are described in Table 5-6.

Table 5-6. *Ethernet Link Object Instance Attributes*

Attribute Code	Access Rule	Name	Data Type	Description
1	Get	Interface Speed	UDINT	Interface speed currently in use
2	Get	Interface Flags	DWORD	Interface status flags
3	Get	Physical Address	DWORD	MAC layer address

5.2.2 ENTRON-DEFINED OBJECTS

ENTRON defines objects to get/set parameters of EN6021 and EN6041 Controls. The objects are listed in Table 5-7. All these objects only support Instance Attributes; they do not support Class Attributes. When get/set Attribute data, Instance Code must be set to 1. The parameter will be set as:

Service Code: 0x0e – to get data
0x10 – to set data
Class Code: See Table 5-7
Instance Code: 1
Attribute Code: See Table 5-7

Table 5-7. *ENTRON-defined Objects*

Class Code	Attribute Code	Access Rule	Name	Data Structure	
0x96	0x64 – 0xc7	Get/Set	Weld schedule data	Table 6-6	
0x97	0x64 – 0xc7	Get/Set	Event data	Table 6-7	
0x98			General Control data		
	0x64	Get/Set	Configuration data	Table 6-8	
	0x65	Get/Set	Counter data	Table 6-9	
	0x66	Get/Set	Stepper data	Table 6-10	
	0x67	Get/Set	I/O Map data	Table 6-12	
	0x68	Get/Set	Use Schedule data	Table 6-13	
	0x69	Get/Set	Error output map data	Table 6-14	
	0x6A	Get/Set	Calibration data	Table 6-15	
	0x6B	Set	Time and Date data	Table 6-5	
	0x6C	Get	Control description data	Table 6-1	Ⓔ
	0x6D	Get	Control status flags data	Table 6-2A,B	
	0x6E	Set	Reset Error Flags	(No data required)	
	0x6F	Set	Reset Counter		
	0x70	Set	Reset Stepper	(No data required)	
	0x71	Set	Reset Sequencer	(No data required)	
	0x72	Set	Reset Weld Log	(No data required)	
	0x73	Set	Reset Error Log	(No data required)	
	0x74	Set	Set Weld/No Weld data		
	0x75	Set	Control description data	Table 6-16	Ⓔ
0x99	0x64 – 0x6D	Get/Set	Sequencer data	Table 6-11A,B,C	
0x9A			Read Log data		
	0x64	Get	Read Weld Log data	Table 6-3A,B	
	0x65	Get	Read Error Log data	Table 6-4A,B	
0x9B	0x64 – 0x65	Get/Set	Control Interface I/O		
0x9C	0x64 – 0x93	Get/Set	Weld schedule20 data	Table 6-6	
0x9D	0x64 – 0x93	Get/Set	Weld schedule40 data	Table 6-6	

5.2.2 ENTRON-DEFINED OBJECTS (cont.)

Additional detailed descriptions for some Class and Instance Attribute data are explained below.

Class 0x96: Get/Set Weld schedule data

Operation parameters –	Service Code:	0x0e – to get data
		0x10 – to set data
	Class Code:	0x96
	Instance Code:	1
	Attribute Code:	0x64 – 0xc7

When Attribute Code is set from 0x64 through 0xc7, the operation will get/set data of Weld Schedule 0–99. The Weld Schedule data structure is described in Table 6-6.

Class 0x97: Get/Set Event data

Operation parameters –	Service Code:	0x0e – to get data
		0x10 – to set data
	Class Code:	0x97
	Instance Code:	1
	Attribute Code:	0x64 – 0xc7

When Attribute Code is set from 0x64 through 0xc7, the operation will get/set data of Event 0–99. The Event data structure is described in Table 6-7.

Class 0x98, Attribute 0x6F: Reset Counter

Operation parameters –	Service Code:	0x10 – to set data
	Class Code:	0x98
	Instance Code:	1
	Attribute Code:	0x6F

A one-byte Request Parameter data, Reset Code, is required for this operation.

Reset Code=0	Reset none
Reset Code=1	Reset part counter
Reset Code=2	Reset weld counter
Reset Code=3	Reset part counter + weld counter

Class 0x98, Attribute 0x74: Set Weld/No Weld data

Operation parameters –	Service Code:	0x10 – to set data
	Class Code:	0x98
	Instance Code:	1
	Attribute Code:	0x74

A one-byte Request Parameter data, State Code, is required for this operation.

State Code=0	Set nothing
State Code=1	Set control to No Weld mode
State Code=2	Set control to Weld mode

5.2.2 ENTRON-DEFINED OBJECTS (cont.)

Class 0x99: Get/Set Sequencer data

Operation parameters – Service Code: 0x0e – to get data
0x10 – to set data
Class Code: 0x99
Instance Code: 1
Attribute Code: 0x64 – 0x6D

The whole Sequencer data is divided into 10 sections. When Attribute Code is set from 0x64 through 0x6D, the operation will get/set section 0–9 of Sequencer data. The Section number is described in Table 6-11C, the data structure of one section's Sequencer data is described in Table 6-11A and 6-11B.

Class 0x9A, Attribute 64: Read Weld Log data

Operation parameters – Service Code: 0x0e – to get data
Class Code: 0x9A
Instance Code: 1
Attribute Code: 0x64

This operation needs two Request Parameter data to indicate Index number of the start Weld Log data and the amount of Weld Log data. The structure of the Request Parameter data is:

First request parameter (2-byte):	0-57599	Start Index of required Weld Log data
Second request parameter (1-byte):	1-6	Count of Weld Log record required

The Weld Log data structure is described in Table 6-3.

Class 0x9A, Attribute 65: Read Error Log data

Operation parameters – Service Code: 0x0e – to get data
Class Code: 0x9A
Instance Code: 1
Attribute Code: 0x65

This operation needs two Request Parameter data to indicate Index number of the start Error Log data and the amount of Error Log data. The structure of the Request Parameter data is:

First request parameter (2-byte):	0-999	Start Index of required Error Log data
Second request parameter (1-byte):	1-12	Count of Error Log record required

The Error Log data structure is described in Table 6-4.

5.2.2 ENTRON-DEFINED OBJECTS (cont.)

Class 0x9B: Get/Set Control Interface I/O data

Operation parameters – Service Code: 0x0e – to get data
0x10 – to set data
Class Code: 0x99
Instance Code: 1
Attribute Code: 0x64 – 0x65

The Instance Attribute data are described in Table 5-8.
See Appendix A for hardware structure of Control Interface I/O.

Table 5-8. *Control Interface I/O Data Instance Attributes*

Attribute Code	Access Rule	Name/Description	Data Type
0x64	Get	Local I/O Ports	STRUCTURE
		CPU Input Port P1(D15-D8: Main Input Port P1)	UINT
		CPU Input Port P3	UINT
		Expansion Input Port P11	UINT
		CPU Output Port P2	UINT
		Expansion Output Port P10	UINT
		Expansion AC Output Port P14 (D3-0: PO36-33)	UINT
		DC Valve Output on P1 (D3-0: V3-V1)	UINT
0x65	Get/Set	Remote I/O Ports	STRUCTURE
		PLC Input 1 (for CPU Input Port P3)	UINT
		PLC Input 2 (for Expansion Input Port P11)	UINT
		PLC Output 1 (for CPU Output Port P2)	UINT
		PLC Output 2 (for Expansion Output Port P10)	UINT

Class 0x9C: Get/Set Weld schedule 20 data
Class 0x9D: Get/Set Weld schedule 40 data

Operation parameters – Service Code: 0x0e – to get data
0x10 – to set data
Class Code: 0x9C for Weld Schedule 20
0x9D for Weld Schedule 40
Instance Code: 1
Attribute Code: 0x64 – 0x93

These two objects are designed to get/set only one parameter in Weld Schedule 20 or 40. When Attribute Code is set from 0x64 through 0x93, the operation will get/set only one parameter listed in Table 6-6.

6.0 DATA STRUCTURES

6.1 CONTROL DESCRIPTION DATA FOR READ/GET FUNCTIONS Ⓔ

Table 6-1 is used only for read function in MODBUS protocol and get function in EIP protocol. Ⓔ

Table 6-1. *Control description data (64 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Control type	0x80 for **EN6021** 0xA0 for **EN6041**	0	255 Ⓓ
2	2	Firmware version		0	65535
4	10	Control Serial number	Serial number in 10 ASCII characters		
14	10	(reserved)			
24	20	Control description	Description in 20 ASCII characters		
44	20	(reserved)			

6.2 CONTROL STATUS DATA

Table 6-2A. *Control status data (200 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	12	Error Flag[0-5]	96-bit error flag		
12	8	(reserved)			
20	2	Hardware status[1]	Main control + system status		
22	2	Input port1 status	Bit0=In1 to Bit15=In16		
24	2	Input port2 status	Bit0=In17 to Bit15=In32		
26	2	Output port1 status	Bit0=Out1 to Bit15=Out16		
28	2	Output port2 status	Bit0=Out17 to Bit15=Out32		
30	2	Output port3 status	Bit0-3=Out33-36		
32	2	Valve output status	**For EN6021:** Bit0-2=Valve 1-3 **For EN6041:** Bit0-7=Valve 1-8		Ⓓ
34	1	Schedule selected		0	99
35	1	Schedule select mode	0=internal, 1=external	0	1
36	2	(reserved)			
38	2	PV sensing			
40	1	Pressure unit			
41	1	P-sense mode			
42	2	Weld1 pulse width			
44	2	Weld1 current			
46	2	Weld2 pulse width			
48	2	Weld2 current			
50	2	Part counter			
52	2	Weld counter			
54	12	(reserved)			
66	2	Stepper counter			
68	14	(reserved)			

(Table continued on following page)

6.2 CONTROL STATUS DATA (cont.)

Table 6-2A. *Control status data (200 bytes)* (cont.)

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
82	2	(reserved)			
84	14	(reserved)			
98	2	Event output status1	Bit0=Out1 to Bit15=Out16		
100	2	Event output status2	Bit0=Out17 to Bit15=Out32		
102	2	(reserved)			
104	2	AC line voltage	1 unit=1V	0	850
106	2	Analog Unit	Bit=0: unit is mA Bit=1: unit is V D0: Analog input1 D1: Analog input2 D2: Analog output1 D3: Analog output2		
108	2	Analog input1	1 unit=0.1mA or 0.01V	0	1250
110	2	Analog input2	1 unit=0.1mA or 0.01V	0	1250
112	2	(reserved)			
114	2	Analog output1	1 unit=0.1mA or 0.01V	0	1250
116	2	Analog output2	1 unit=0.1mA or 0.01V	0	1250
118	10	(reserved)			
128	2	Sequencer Status	0: Idle 1: Working 2: End 3: Error	0	3
130	2	Sequencer Step			
132	2	Sequencer Input1	Bit0=In1 to Bit15=In16		
134	2	Sequencer Input2	Bit0=In17 to Bit15=In32		
136	2	Sequencer Output1	Bit0=Out1 to Bit15=Out16		
138	2	Sequencer Output2	Bit0=Out17 to Bit15=Out32		
140	2	Sequencer Flag1	Bit0=Flag1 to Bit15=Flag16		
142	2	Sequencer Flag2	Bit0=Flag17 to Bit15=Flag32		
144	2	Sequencer Counter1	Counter1 of Sequencer	0	999
146	2	Sequencer Counter2	Counter2 of Sequencer	0	999
148	2	Sequencer Counter3	Counter3 of Sequencer	0	999
150	2	Sequencer Counter4	Counter4 of Sequencer	0	999
152	2	Sequencer Counter5	Counter5 of Sequencer	0	999
154	2	Sequencer Counter6	Counter6 of Sequencer	0	999
156	2	Sequencer Counter7	Counter7 of Sequencer	0	999
158	2	Sequencer Counter8	Counter8 of Sequencer	0	999
160	2	Weld Log index		0	57600 ©
162	2	Error Log index		0	1000
164	2	Last IPC Current	(reserved for ENLINK)		
166	2	Last IPS Current	(reserved for ENLINK)		

(Table continued on following page)

6.2 CONTROL STATUS DATA (cont.)

Table 6-2A. *Control status data (200 bytes)* (cont.)

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
168	2	Conduction Angle1 **(for EN6021 only)**		0	180	Ⓓ
170	2	Conduction Angle2 **(for EN6021 only)**		0	180	Ⓓ
172	2	Power Factor Delay		0	99	
174	2	Time Year Month	D15-8: Year (0-99=2000-2099) D7-0: Month	0 1	99 12	Ⓑ
176	2	Time Day Hour	D15-8: Day D7-0: Hour	1 0	31 23	Ⓑ
178	2	Time Minute Second	D15-8: Minute; D7-0: Second	0	59	Ⓑ
180	2	**For EN6021:** Stack-up Thickness	Thickness value of Stack-up 1 unit = 1 mil	0	10500	Ⓓ
		For EN6041: Contactor Number	Contactor (SCR) number 0-7=SCR1-SCR8	0	7	Ⓓ
182	2	Stack-up Displacement **(for EN6021 only)**	Displacement value of Stack-up 1 unit = 1 mil	-100	100	Ⓓ
184	2	RT-Thickness **(for EN6021 only)**	Real-time thickness value of Stack-up; 1 unit = 1 mil	0	10500	Ⓓ
186	2	Last THK Current **(for EN6021 only)**	(reserved for ENLINK)			Ⓓ
188	2	Log Memory Status	Bit0: 0=Weld Log is not full 1=Weld Log is full Bit1: 0=Error Log is not full 1=Error Log is full Bit2-15: (reserved)			Ⓕ
190	2	(reserved)				
192	2	(reserved)				
194	2	(reserved)				
196	2	(reserved)				
198	2	Control type	(reserved for ENLINK)			Ⓓ

6.2 CONTROL STATUS DATA (cont.)

The first 6 words of Table 6-2A include 96-bit data and indicates 96 error messages or system status. The detail of word address and bit offset information are shown in Table 6-2B.

Table 6-2B. *Bit definition of Error Flags*

Error Flags		
Word	**Bit address**	**Error description**
0	0	Configuration Error
	1	Calibration Error
	2	Schedule Error
	3	Sequencer Error
	4	Event Error
	5	Counter Error
	6	Stepper Error
	7	I/O Map Error
	8	E-Stop Error
	9	TC1 (Contactor) Error Ⓓ
	10	No Weld (P1-NW)
	11	PS Error
	12	SCR short
	13	Second Stage Error
	14	P Sense Error Ⓓ
	15	Interlock Error
1	0	High Pressure/Force
	1	Low Pressure/Force
	2	High Current1
	3	Low Current1
	4	High Current2
	5	Low Current2
	6	High Voltage
	7	Low Voltage
	8	Counter end
	9	Stepper end
	10	High pulse width1
	11	Low pulse width1
	12	High pulse width2
	13	Low pulse width2
	14	Stepper pre-warn (Tip dress)
	15	AVC error
2	0	Power on with STARTs closed
	1	2-PALM error
	2	PNW
	3	TT1 (Transformer) Error Ⓓ
	4	Safety relay error
	5	No 24V for CPU I/O ports
	6	No 24V for I/O Expansion Board
	7	Prox. lost during weld
	8	2-PALM retract port error Ⓓ
	9	Tip dress on
	10	Prox. not met Ⓓ
	11	
	12	AC120V safety relay error
	13	
	14	No AC120V for I/O Expansion Board
	15	

(Table continued on following page)

6.2 CONTROL STATUS DATA (cont.)

Table 6-2B. *Bit definition of Error Flags* (cont.)

Error Flags		
Word	**Bit address**	**Error description**
3	0	High Pressure pre-limit
	1	Low Pressure pre-limit
	2	High Current1 pre-limit
	3	Low Current1 pre-limit
	4	High Current2 pre-limit
	5	Low Current2 pre-limit
	6	High Stack-up Ⓓ
	7	Low Stack-up Ⓓ
	8	Ⓓ
	9	
	10	
	11	
	12	
	13	
	14	
	15	
4	0	Battery Low
	1	Use Schedule error
	2	
	3	
	4	
	5	
	6	
	7	
	8	Weld Log full
	9	Weld Log warn (80% of full)
	10	Error Log full
	11	Error Log warn (80% of full)
	12	Flash RAM error
	13	
	14	
	15	
5	0	
	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	Retract open error Ⓓ
	9	Retract 2-PALM not ready
	10	Retract input closed
	11	PS not ready
	12	Retract not ready
	13	Second stage not ready
	14	P Sense not ready Ⓓ
	15	Interlock not ready

6.3 WELD LOG DATA

One frame of Weld Log data includes one Start Record Index (2-byte), one Record count (1-byte)Ⓓ and one or multiple Weld records data. When Weld Log data are sent out of EN6021/EN6041, up to 6 weld records can be transferred within one Modbus/EIP package. Table 6-3A shows one frame of Weld Log data in Modbus package. One weld record includes 32 bytes of data – data structure of individual weld record is shown in Table 6-3B.

Table 6-3A. *Weld Log data (35–195 bytes)* Ⓓ

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Start Record Index	Index number of 1st weld record	0	57599 Ⓒ
2	1	Record count	Record count in this message	1	6
3	32	1st Weld record data	32-byte Weld record data		
35	32	2nd Weld record data	32-byte Weld record data		
…					
163	32	(up to) 6th Weld record data	32-byte Weld record data		

6.3 WELD LOG DATA (cont.)

Table 6-3B. *Data structure of individual Weld Log record (32 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
0	2	Index	Index number of weld log record	0	57599	C
2	2	Time Year Month	D15-8: Year; D7-0: Month			
4	2	Time Day Hour	D15-8: Day; D7-0: Hour			
6	2	Time Minute Second	D15-8: Minute; D7-0: Second			
8	2	Counter Value	Count value from counter	0	60000	A
10	2	Schedule number	0-99	0	99	
12	2	Pulse Width	D15-8: PW2; D7-0: PW1			
14	2	Measure I1	x10A			
16	2	Measure I2	x10A			
18	2	Force				
20	2	Data Unit	**For EN6021:** D15-12: Force unit D11-3: Reserved D2-1: Analog input2 unit (24th-27th)* – 0=0.1mA, 1=0.01V, 2=1 mil D0: Analog input1 unit – 0=0.1mA, 1=0.01V			D D
			For EN6041: D15-12: Force unit D11-2: Reserved D1: Analog input2 unit – 0=0.1mA, 1=0.01V D0: Analog input1 unit – 0=0.1mA, 1=0.01V			D
22	2	Analog Input1	0-200: 0-20.0mA or 0-1000: 0-10.00V			
24	2	**For EN6021:** Analog Input2-at-Squeeze or Stack-up thickness*	0-200: 0-20.0mA or 0-1000: 0-10.00V or 0-10000: 0-10000 mil			D
		For EN6041: Analog Input2-at-Squeeze	0-200: 0-20.0mA or 0-1000: 0-10.00V			
26	2	**For EN6021:** Analog Input2-at-Hold or Stack-up displacement*	0-200: 0-20.0mA or 0-1000: 0-10.00V or -100-+100: -100-+100 mil			D
		For EN6041: Contactor (SCR) Number	0-7: SCR1-SCR8			
28	2	Reserved2				
30	2	Reserved3				

* When Weld Log data are created, in each weld record: D

If Analog Input2 is mapped to **Stack-up** function –
24th-25th byte store the Stack-up thickness
26th-27th byte store the Stack-up displacement

If Analog Input2 is mapped to **Sequener** function –
24th-25th byte store Analog Input2 raw value when control is at the end of **Squeeze** step
26th-27th byte store Analog Input2 raw value when control is at the end of **Hold** step

6.4 ERROR LOG DATA

One frame of Error Log data includes one Start Record Index (2-byte), one Record count (1-byte) Ⓓ and one or multiple Error records data. When Error Log data are sent out of EN6021/EN6041, up to 12 error records can be transferred within one Modbus/EIP package. Table 6-4A shows one frame of Error Log data in Modbus/EIP package. One error record includes 16 bytes of data – data structure of individual error record is shown in Table 6-4B.

Table 6-4A. *Error Log data (19–195 bytes)* Ⓓ

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
0	2	Start Record Index	Index number of 1st error record	0	999	Ⓒ
2	1	Record count	Record count in this message	1	12	
3	16	1st Error record data	16-byte Error record data			
19	16	2nd Error record data	16-byte Error record data			
…						
179	16	(up to) 12th Error record data	16-byte Error record data			

Table 6-4B. *Data structure of individual Error Log record (16 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
0	2	Index	Index number of error log record	0	999	Ⓒ
2	2	Time Year Month	D15-8: Year; D7-0: Month			
4	2	Time Day Hour	D15-8: Day; D7-0: Hour			
6	2	Time Minute Second	D15-8: Minute; D7-0: Second			
8	2	Counter Value	Count value from counter	0	60000	Ⓐ
10	2	Error number				
12	2	Error state	D15-1: Reserved D0: 0=Off, 1=On			
14	2	(reserved)				

6.5 TIME AND DATE DATA Ⓑ

Table 6-5. *Time and Date data (128 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Time Year Month	D15-8: Year (0-99=2000-2099) D7-0: Month	0 1	99 12
2	2	Time Day Hour	D15-8: Day D7-0: Hour	1 0	31 23
4	2	Time Minute Second	D15-8: Minute; D7-0: Second	0	59
6	122	(reserved)			

6.6 WELD SCHEDULE DATA

Table 6-6. *Weld schedule data (96 bytes for each schedule)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Squeeze Time	Squeeze time (cycles)	0	99
2	2	Valve Mode	Bit *x* =0: Valve off Bit *x* =1: Valve on **For EN6021:** *x* =0-2: Valve 1-3 **For EN6041:** *x* =0-7: Valve 1-8		
4	2	Squeeze Pressure	Pressure for PV When Pressure Unit is PSI: 0-100, 1 unit=1 PSI When Force Unit is Lb: 0-15700, 1 unit=0.5 Lb When Pressure Unit is mA: 0-200, 1 unit=0.1 mA		
6	2	Pressure Trigger	Trigger Pressure – same as Squeeze Pressure		
8	2	Pressure Limit High	Pressure Monitor High–same as Squeeze Pressure		
10	2	Pressure Limit Low	Pressure Monitor Low–same as Squeeze Pressure		
12	2	Weld1 Time	Weld1 time (cycles)	0	99
14	2	Weld1 Heat	Weld1 pulse width (1%)	0	99
16	2	Weld1 Current	Weld1 current (10A)	0	10000
18	2	Current1 Limit High	Current1 Monitor High (10A)	0	10000
20	2	Current1 Limit Low	Current1 Monitor Low (10A)	0	10000
22	2	Cool1 Time	Cool1 time (cycles)	0	99
24	2	Slope Time	Slope time (cycles)	0	99
26	2	Weld2 Time	Weld2 time (cycles)	0	99
28	2	Weld2 Heat	Weld2 pulse width (1%)	0	99
30	2	Weld2 Current	Weld2 current (10A)	0	10000
32	2	Current2 Limit High	Current2 Monitor High (10A)	0	10000
34	2	Current2 Limit Low	Current2 Monitor Low (10A)	0	10000
36	2	Cool2 Time	Cool2 time (cycles)	0	99
38	2	Hold Time	Hold time (cycles)	0	99
40	2	Off Time	Off time (cycles)	0	99
42	2	Impulses	Pulses Mode	1	99
44	2	Weld Mode	Bit0: Weld1 regulation mode – 0=Phase shift, 1=Constant current Bit1: Weld1 current monitor – 0=Off, 1=On Bit2: Weld2 regulation mode – 0=Phase shift, 1=Constant current Bit3: Weld2 current monitor – 0=Off, 1=On Bit4: Squeeze pressure monitor – 0=Off, 1=On Bit5,6: Pressure Sensing mode (Psense) – 0=Off, 1=Rising, 2=Falling Bit7: Weld1 pulse width monitor – 0=Off, 1=On Bit8: Weld2 pulse width monitor – 0=Off, 1=On Bit9: Pressure/Force pre-limit mon.–0=Off,1=On Bit10: Current1 pre-limit monitor – 0=Off, 1=On Bit11: Current2 pre-limit monitor – 0=Off, 1=On **For EN6021 only:** Bit12: Stack-up thickness monitor – 0=Off, 1=On		

(Table continued on following page)

6.6 WELD SCHEDULE DATA (cont.)

Table 6-6. *Weld schedule data (96 bytes for each schedule)* (cont.)

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
46	2	Cycle Mode	**For EN6021:** 0: Non-repeat 1: Repeat 2: Chained 3: Successive 4: Wait Here	0	4	Ⓓ
			For EN6041: 0: Non-repeat 1: Chained 2: Successive	0	2	Ⓓ
48	2	PW1 High	Pulse width1 monitor-High value	0	99	
50	2	PW1 Low	Pulse width1 monitor-Low value	0	99	
52	2	PW2 High	Pulse width2 monitor-High value	0	99	
54	2	PW2 Low	Pulse width2 monitor-Low value	0	99	
56	2	Current Offset	Current Offset	35	65	
58	2	SQZ Delay Time **(for EN6021 only)**	Squeeze Delay time (Air-over-oil gun=Off) or Advance time (Air-over-oil gun=On)	0 0	99 99	Ⓓ
60	2	**For EN6021:** Block Delay Time	Blocking Delay time for Air-over-oil gun	0	99	Ⓓ
		For EN6041: Contactor (SCR) Number	0-7: SCR1-SCR8	0	7	Ⓓ
62	2	Pressure Pre-limit	Pre-limit for Pressure/Force monitor	0	99	
64	2	Current1 Pre-limit	Pre-limit for Current1 monitor	0	99	
66	2	Current2 Pre-limit	Pre-limit for Current2 monitor	0	99	
68	2	Stack-up High **(for EN6021 only)**	Stack-up monitor-High value 1 unit=1 mil	0	10000	Ⓓ
70	2	Stack-up Low **(for EN6021 only)**	Stack-up monitor-Low value 1 unit=1 mil	0	10000	Ⓓ
72	24	(reserved)				

6.7 EVENT DATA

Table 6-7. *Event data (64 bytes for each schedule)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
0	2	Event Channel[0]	0: Disable 1-32: Output 1-32	0	32	
2	2	Event Channel[1]				
4	2	Event Channel[2]				
6	2	Event Channel[3]				
8	2	Event State[0]	0: Off 1: On	0	1	
10	2	Event State[1]				
12	2	Event State[2]				
14	2	Event State[3]				
16	2	Event Interval[0]	**For EN6021:** 0: Squeeze Delay (Advance) 1: Squeeze (Intensify) 2: 2nd Stage 3: Weld1 4: Cool1 5: Slope 6: Weld2 7: Cool2 8: Hold	0	8	Ⓓ
18	2	Event Interval[1]				
20	2	Event Interval[2]				
			For EN6041: 0: Squeeze 1: 2nd Stage 2: Weld1 3: Cool1 4: Slope 5: Weld2 6: Cool2 7: Hold	0	7	Ⓓ
22	2	Event Interval[3]				
24	2	Event Delay[0]	0-98 cycles	0	98	Ⓔ
26	2	Event Delay[1]				
28	2	Event Delay[2]				
30	2	Event Delay[3]				
32	32	(reserved)				

6.8 CONFIGURATION DATA

Table 6-8. *Configuration data (64 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
0	2	Seam Mode **(for EN6021 only)**	0: Spot 1: Seam1 2: Seam2	0	2	Ⓓ
2	2	Retraction Mode **(for EN6021 only)**	0: None 1: Momentary 2: Maintained 3: 2-PALM	0	3	Ⓓ
4	2	Sch Select Mode	0: Internal 1: External	0	1	
6	2	Force Unit	0: PSI 1: Lb 2: mA 3: Calibrated Lb	0	3	
8	2	Cylinder Diameter	1unit=0.1 inch	10	100	
10	2	Background Pressure		0	9999	
12	2	Regulation Mode	0: Primary 1: Secondary 2: Secondary Feedback using Primary Coil	0	2	Ⓕ
14	2	Beat Mode **(for EN6021 only)**	0: Non-Beat 1: Beat During Squeeze 2: Beat During Squeeze+Weld 3: Allow Wait-Here (CM=4)	0	3	Ⓓ
16	2	AVC Mode	0: Off; 1-10: 1%-10%	0	10	
18	2	Voltage Limit High		160	750	
20	2	Voltage Limit Low		160	750	
22	2	Current Offset	0-15%	0	15	
24	2	Water Saver Delay		0	199	Ⓒ
26	2	Half Cycle Mode	0: Off; 1: +; 2: -; 3: AC	0	3	
28	2	ID Number	ID number of control	1	99	
30	2	Config Mode	Bit0,1: On error – 0=Continue 1=Stop on fault 2=Head lock on fault Bit2: Sequencer – 0=Off, 1=On Bit3,4: Air-over-oil mode **(for EN6021 only)** – 0=Off 1=Air-over-oil without retraction 2=Air-over-oil with retraction Bit5: Vmonitor – 0=Off, 1=On Bit6: 2-PALM – 0=Off, 1=On Bit7: 87° Delay – 0=Off,1=On Bit8: Analog In1 unit – 0=I, 1=V Bit9: Analog In2 unit – 0=I, 1=V Bit10: Analog Out1 unit – 0=I, 1=V Bit11: Analog Out2 unit – 0=I, 1=V Bit12: (reserved)			Ⓓ

(Table continued on following page)

6.8 CONFIGURATION DATA (cont.)

Table 6-8. *Configuration data (64 bytes)* (cont.)

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
32	2	RS Com Mode	0: MB Ethernet 1: MB RS232 RTU 2: MB RS485 RTU 3: Label printing 4: EIP+MB Ethernet	0	4	Ⓒ
34	2	Blanking	Blank cycles	0	99	
36	2	AVC Nominal	AVC nominal voltage	187	633	
38	2	Retract Open Time **(for EN6021 only)**	Retract open time for Air/oil gun	0	99	Ⓓ
40	2	Retract Close Time **(for EN6021 only)**	Retract close time for Air/oil gun	0	99	Ⓓ
42	2	Display Return Delay	Delay time for RPP2 display returning to Status Page 1	0	10	
44	2	Pressure Control Mode	0: No pressure sensing/control 1: Pressure sensing 2: Pressure control 3: Pressure sensing+control	0	3	
46	2	Set Power Factor	0: Auto power factor 1-99: power factor=0.01-0.99	0	99	
48	2	Log Recording Mode	Bit0-3: 0=Recording stops when log is full 1=Rewrite log memory when log is full Bit4-7: –when read data from control, these bits are 0 –when write data to control, these bits must be 10 for control accepting data in Bit0-3 Bit8-15: (reserved)			Ⓕ
50	14	(reserved)				

6.9 COUNTER DATA

Table 6-9. *Counter data (32 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Part Counter	Counter for part	0	9999
2	2	Weld Counter	Counter for weld (Welds-per-part)	0	9999
4	2	Counter Control	0: Disable 1: Enable	0	1
6	2	Max Part Count	Max count for part counter	0	60000
8	2	Max Welds per Part	Max count for weld counter	1	9999
10	22	(reserved)			

6.10 STEPPER DATA

Table 6-10. *Stepper data (96 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Stepper Count[0]		0	9999
2	2	Stepper Count[1]			
4	2	Stepper Count[2]			
6	2	Stepper Count[3]			
8	2	Stepper Count[4]			
10	2	Stepper Count[5]			
12	2	Stepper Count[6]			
14	2	Stepper Count[7]			
16	2	Stepper Count[8]			
18	2	Stepper Count[9]			
20	2	Stepper Heat Inc[0]		0	99
22	2	Stepper Heat Inc[1]			
24	2	Stepper Heat Inc[2]			
26	2	Stepper Heat Inc[3]			
28	2	Stepper Heat Inc[4]			
30	2	Stepper Heat Inc[5]			
32	2	Stepper Heat Inc[6]			
34	2	Stepper Heat Inc[7]			
36	2	Stepper Heat Inc[8]			
38	2	Stepper Heat Inc[9]			

(Table continued on following page)

6.10 STEPPER DATA (cont.)

Table 6-10. *Stepper data (96 bytes)* (cont.)

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
40	2	Stepper Current Inc[0]		0	9999
42	2	Stepper Current Inc[1]			
44	2	Stepper Current Inc[2]			
46	2	Stepper Current Inc[3]			
48	2	Stepper Current Inc[4]			
50	2	Stepper Current Inc[5]			
52	2	Stepper Current Inc[6]			
54	2	Stepper Current Inc[7]			
56	2	Stepper Current Inc[8]			
58	2	Stepper Current Inc[9]			
60	2	Stepper Force[0]	Force stepper % When=0: -100% When=100: +-0% When=200: +100%	0	200
62	2	Stepper Force[1]			
64	2	Stepper Force[2]			
66	2	Stepper Force[3]			
68	2	Stepper Force[4]			
70	2	Stepper Force[5]			
72	2	Stepper Force[6]			
74	2	Stepper Force[7]			
76	2	Stepper Force[8]			
78	2	Stepper Force[9]			
80	2	Stepper Control	0: Disable 1: Current/Heat stepper 2: Force stepper 3: Heat + Force	0	3
82	2	Stepper Count Now		0	9999
84	2	Stepper Pre-warn Count		0	9999
86	10	(reserved)			

6.11 SEQUENCER DATA

The EN6021/EN6041's Sequencer includes up to 200 sequence statements, each statement includes one operation code and one or two parameters. Table 6-11A shows the structure of an individual sequence statement. Table 6-11B shows the value of each operation code and its possible parameters.

Table 6-11A. *Data structure of individual sequence statement*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Op-code[*i*]	Operation Code	0	32 **for EN6021** Ⓔ 30 **for EN6041**
2	2	Seq Variable1[*i*]	Sequencer 1st Variable	0	999
4	2	Seq Variable2[*i*]	Sequencer 2nd Variable	0	999

i=0–199 (maximum 200 steps)

Table 6-11B. *Value of operation codes and parameters (Data in memory / data displayed)*

Operation Code		Parameter 1		Parameter 2		
Value	Function	Min	Max	Min	Max	
0	Blank					
1	Step	0/1	99/100			
2	Sub	0/1	99/100			
3	Await Input ON	0/1	31/32			
4	Await Input OFF	0/1	31/32			
5	Output ON	0/1	31/32			
6	Output OFF	0/1	31/32			
7	Flag ON	0/1	31/32			
8	Flag OFF	0/1	31/32			
9	Delay	1/0.1	999/99.9			
10	Jump	0/1	199/200			
11	GoSub	0/1	99/100			
12	Return					
13	Set Counter	0/1	7/8	1	999	
14	Decrement Counter	0/1	7/8			
15	If Counter>0 Jump	0/1	7/8	0/1	199/200	
16	If Output ON Jump	0/1	31/32	0/1	199/200	
17	If Output OFF Jump	0/1	31/32	0/1	199/200	
18	If Flag ON Jump	0/1	31/32	0/1	199/200	
19	If Flag OFF Jump	0/1	31/32	0/1	199/200	
20	If Input ON Jump	0/1	31/32	0/1	199/200	
21	If Input OFF Jump	0/1	31/32	0/1	199/200	
22	Spot-weld with Sch	0	100			ⒶⒹ
23	Set Analog Output	0/1	1/2	40 or 0	200 or 1000	
24	If Analog Input1 > Jump	40 or 0	200 or 1000	0/1	99/100	
25	If Analog Input1 < Jump	40 or 0	200 or 1000	0/1	99/100	
26	If Analog Input2 > Jump	40 or 0	200 or 1000	0/1	99/100	
27	If Analog Input2 < Jump	40 or 0	200 or 1000	0/1	99/100	

(Table continued on following page)

6.11 SEQUENCER DATA (cont.)

Table 6-11B. *Value of operation codes and parameters (Data in memory/data displayed)* (cont.)

Operation Code		Parameter 1		Parameter 2		
Value	Function	Min	Max	Min	Max	
28	End					
29	If Error ON Jump	0**	96**	0/1	199/200	Ⓓ
30	If Error OFF Jump	0**	96**	0/1	199/200	Ⓓ
31	Seam-weld with Sch **(for EN6021 only)**	0	99			Ⓓ
32	Seam-weld end **(for EN6021 only)**					Ⓓ

** For **Parameter 1** of **Operation Code 29**, value of 0 stands for any errors. Ⓓ
For **Parameter 1** of **Operation Code 30**, value of 0 stands for all errors.
For both codes, value of 1–96 stands for Error 0–95 (displayed as 1–96).

Due to maximum size of Modbus data package, the entire Sequencer's data (200-step sequences' data) are divided into 10 sections, each section include 20 steps of sequence data, so that one Modbus package can carry one section of Sequencer's data. Table 6-11C shows section numbers of Sequencer and step range covered.

Table 6-11C. *Sequencer's section numbers and ranges*

Section Number	Covered Step Range
0	0 through 19
1	20 through 39
2	40 through 59
3	60 through 79
4	80 through 99
5	100 through 119
6	120 through 139
7	140 through 159
8	160 through 179
9	180 through 199

6.12 I/O MAP DATA

Table 6-12. *I/O Map settings data (200 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
0	64	IO Map[I] I=0-31 for input port PI-1 to PI-32 function	0: Primary function 1: Sequencer	0	1	Ⓓ
64	64	**For EN6021:** IO Map[I] I=32-63 for output port PO-1 to PO-32 function **For EN6041:** IO Map[I] I=32-55 for output port PO-1 to PO-24 function	0: Primary function 1: Event function 2: Sequencer 3: PLC output	0	3	Ⓓ
128	2	IO Map[64] for input resource	Bit0-15=Input1-16			
130	2	IO Map[65] for input resource	Bit0-15=Input17-32			
132	8	IO Map[66]: Analog input1 IO Map[67]: Analog input2 IO Map[68]: Analog output1 IO Map[69]: Analog output2	Analog port function – 0: Primary function 1: Sequencer	0	1	
140	2	IO Map[70]: AC Port33 function **(for EN6021 only)**	0: Valve1 output	0	0	Ⓓ
142	2	IO Map[71]: AC Port34 function **(for EN6021 only)**	0: Valve2 output	0	0	Ⓓ
144	2	IO Map[72]: AC Port35 function **(for EN6021 only)**	0: Valve3 output	0	0	Ⓓ
146	2	IO Map[73]: AC Port36 function **(for EN6021 only)**	0: EOS 1: Not ready 2: Tip dress 3: Retraction 4: Count end 5: Error 6: Step end 7: Interlock 8: Water saver	0	8	Ⓓ
148	52	(reserved)				

6.13 USE SCHEDULE DATA

Table 6-13. *Use Schedule data (32 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Use Schedule	Internal Schedule selected	0	99
2	30	(reserved)			

6.14 ERROR CODE OUTPUT MAP DATA

Table 6-14. *Error Code output map data (224 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	192	Error Map[0-95]	Expansion port number for Error Code	0	16
192	32	(reserved)			

6.15 CALIBRATION DATA

Table 6-15. *Calibration data (64 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max	
0	2	Coil Sensitivity	For secondary current feedback: 127-173 = 127-173 mV/kA For primary current feedback **–OR–** secondary feedback using primary coil: 119-161 = 1190-1610 mV/kA			(F)
2	2	Max Current	5-100 kA	5	100	(F)
4	2	Turns Ratio	Transformer's Turns Ratio	10	255	
6	2	(reserved)				
8	2	(reserved)				
10	2	(reserved)				
12	2	(reserved)				
14	2	(reserved)				
16	2	(reserved)				
18	2	(reserved)				
20	2	(reserved)				
22	2	(reserved)				
24	2	(reserved)				
26	2	(reserved)				
28	2	(reserved)				
30	2	(reserved)				
32	2	(reserved)				
34	2	(reserved)				
36	2	Cal IPC I1	Analog input for PT1 of IPC calibration; 1 unit=0.1 mA	30	210	(D)
38	2	Cal IPC F1	Force setting for PT1 of IPC calibration; 1 unit=0.5 Lb	0	16485	(D)
40	2	Cal IPC I2	Analog input for PT2 of IPC calibration; 1 unit=0.1 mA	30	210	(D)
42	2	Cal IPC F2	Force setting for PT2 of IPC calibration; 1 unit=0.5 Lb	0	16485	(D)
44	2	Cal IPS I1	Analog input for PT1 of IPS calibration; 1 unit=0.1 mA	30	210	(D)
46	2	Cal IPS F1	Force setting for PT1 of IPS calibration; 1 unit=0.5 Lb	0	16485	(D)
48	2	Cal IPS I2	Analog input for PT2 of IPS calibration; 1 unit=0.1 mA	30	210	(D)
50	2	Cal IPS F2	Force setting for PT2 of IPS calibration; 1 unit=0.5 Lb	0	16485	(D)
52	2	**For EN6021:** Cal Stack-up I1	Analog input for PT1 of Stack-up calibration; 1 unit=0.1 mA	30	199	(D)
		For EN6041: AC-line Voltage Setting	Setting for AC-line Voltage measurement	140	750	(D)
54	2	Cal Stack-up I2 **(for EN6021 only)**	Analog input for PT2 of Stack-up calibration; 1 unit=0.1 mA	41	210	(D)
56	2	Cal Stack-up Gain **(for EN6021 only)**	Thickness setting for PT2 of Stack-up calibration; 1 unit=1 mil	1	10500	(D)
58	2	(reserved)				
60	2	(reserved)				
62	2	(reserved)				

6.16 CONTROL DESCRIPTION FOR WRITE/SET FUNCTIONS Ⓔ

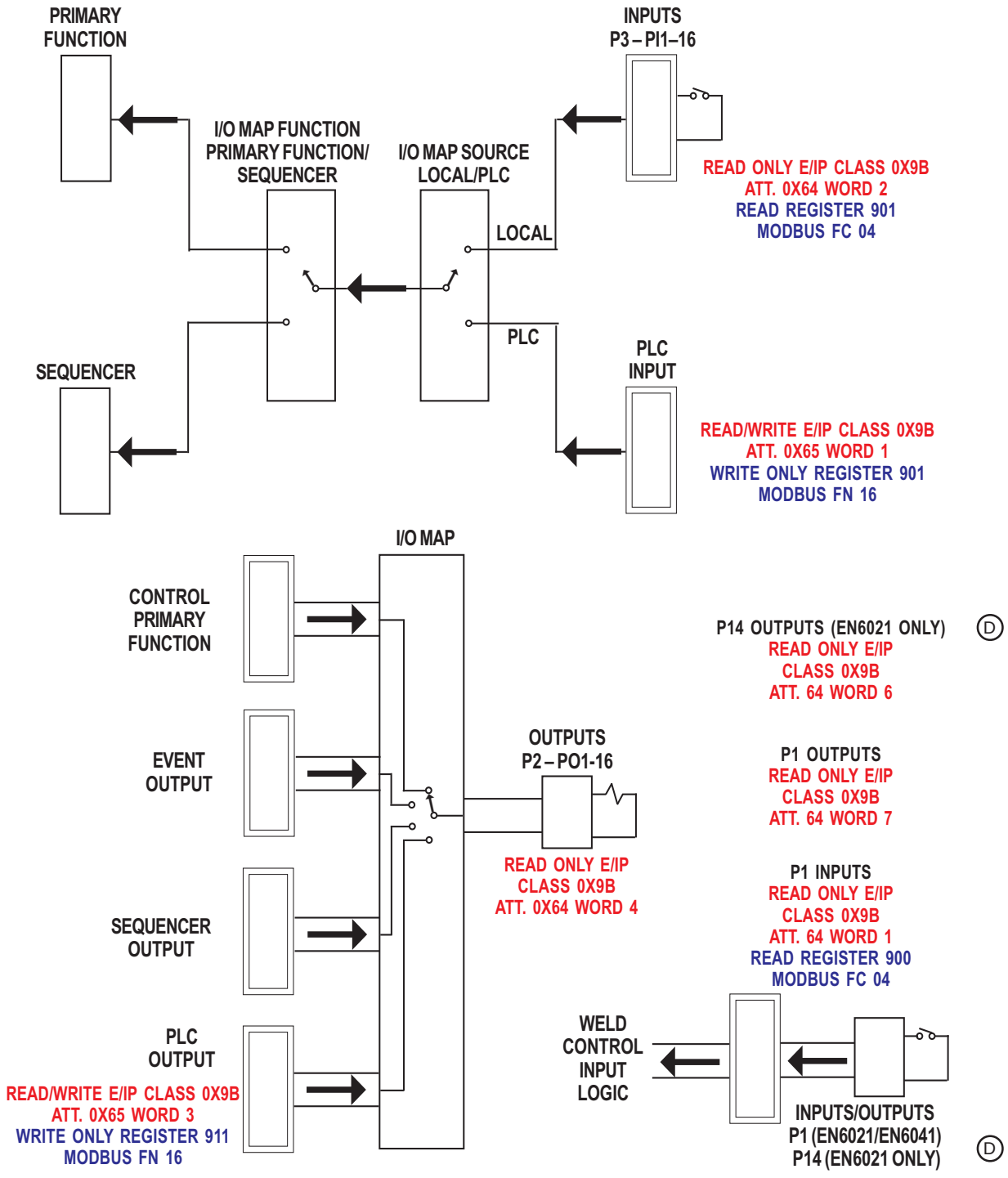
Table 6-16 is used only for write function in MODBUS protocol and set function in EIP protocol. Ⓔ

Table 6-16. *Control description (32 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	20	Description[0-19]	20 ASCII characters for control description		
20	12	(reserved)			

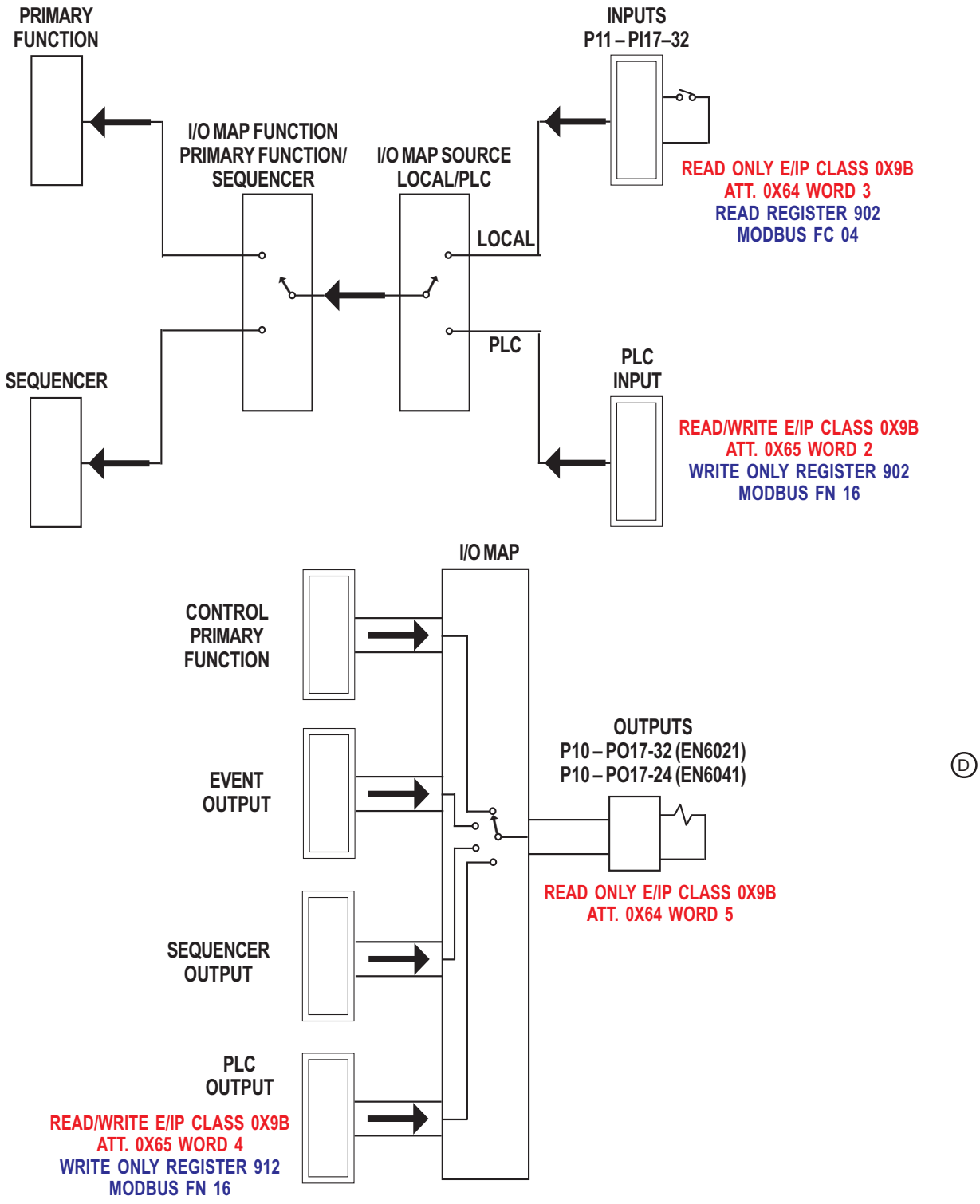
APPENDIX A CONTROL INTERFACE I/O HARDWARE STRUCTURE©

CPU BOARD – P3, P2, P1, P14



APPENDIX A CONTROL INTERFACE I/O HARDWARE STRUCTURE

EXPANSION BOARD – P11 & P10



Ⓓ

PCB 410384 (772056)

06/24/2012 Version V1.1 – Original release

05/01/2013 Version V1.2

Change EIP-child and Modbus-child tasks' function to achieve faster processing speed and more reliable stack structure.

07/012013 Version V1.3

1. Add (Class: 0x98 Attribute: 0x75) – "Set control description" function.
2. Modify (Class: 0x98 Attribute: 0x6C) – "Get Control description" function to be read only function.
3. Fix bug with Class 1 Attribute 4 function call – the bug reverses the addresses of Major-revision and Minor-revision data.

10/21/2014 Version V1.4

Revise firmware to support RJ45 connector (LED pattern: RED/GREEN, RED/GREEN)