INSTRUCTION MANUAL
700231C
COMMUNICATION SPECIFICATIONS FOR EN6001 SERIES CONTROLS

FOR

MICROPROCESSOR BASED
Weld Sequence Controls
With
Solid State Thyristor Contactors

EN6001 Control Manual – Instruction Manual 700230
EN6001 Timer – firmware version 7.02 and higher

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

RWMA MEMBER

TABLE OF CONTENTS

1.0 INTRODUCTION

The EN6001 Control offers an option for communication of data and status information to a remote device. This communication option requires an additional Communication Card, **EIP/MBTCP** Communication Card (P/N 410392), to be installed on the EN6001 Control.

Ethernet communication offers multiple connections between EN6001 Controls and remote device. Typical Ethernet connection is shown in Figure 1-1.

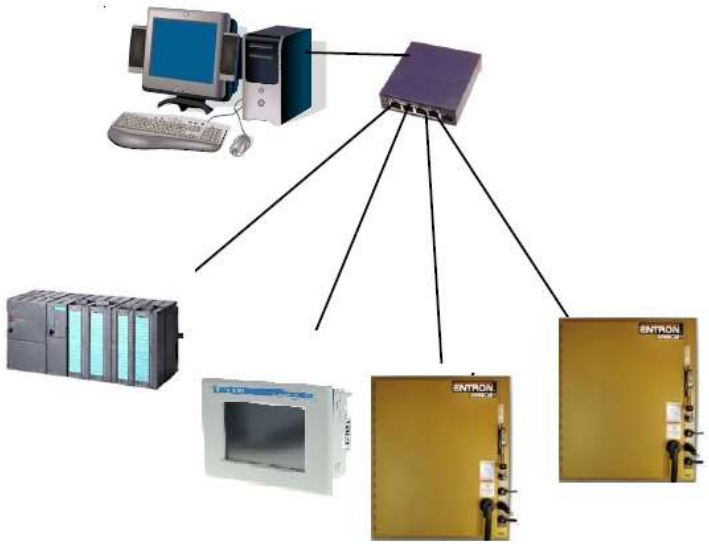

Figure 1-1. *Typical Ethernet connection*

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 EIP/MBTCP COMMUNICATION CARD

2.1 ETHERNET INTERFACE

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

2.2 SPECIFICATIONS

Ethernet:	10/100BASE-T
Speed:	Auto detect 10/100 Mb
Connector:	RJ45
Support cable:	Regular direct Ethernet cable Ethernet cable
Connection supported:	Support up to two TCP/IP connections and one UDP for EIP and 2 TCP/IP connections for Modbus
Ethernet-port used:	Port number 502 for Modbus Port number 44818 for EIP

2.3 CONNECTOR

The connector for Ethernet connection is P9 on the CPU. The Ethernet interface has same pin layout 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	4	Differential Ethernet receive data -
(Not used)		5	
(Not used)		6	
(Not used)		7	
(Not used)		8	
SHIELD			Chassis ground

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

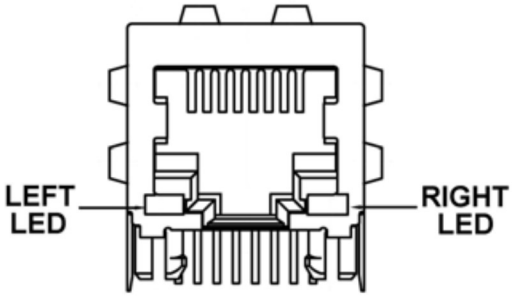

Figure 2-1. *Status Indicator LEDs*

Table 2-2. *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

2.4 ETHERNET INTERFACE SETTING

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

The Ethernet interface configuration can be set through the control panel.

2.5 MAC ADDRESS LABEL

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

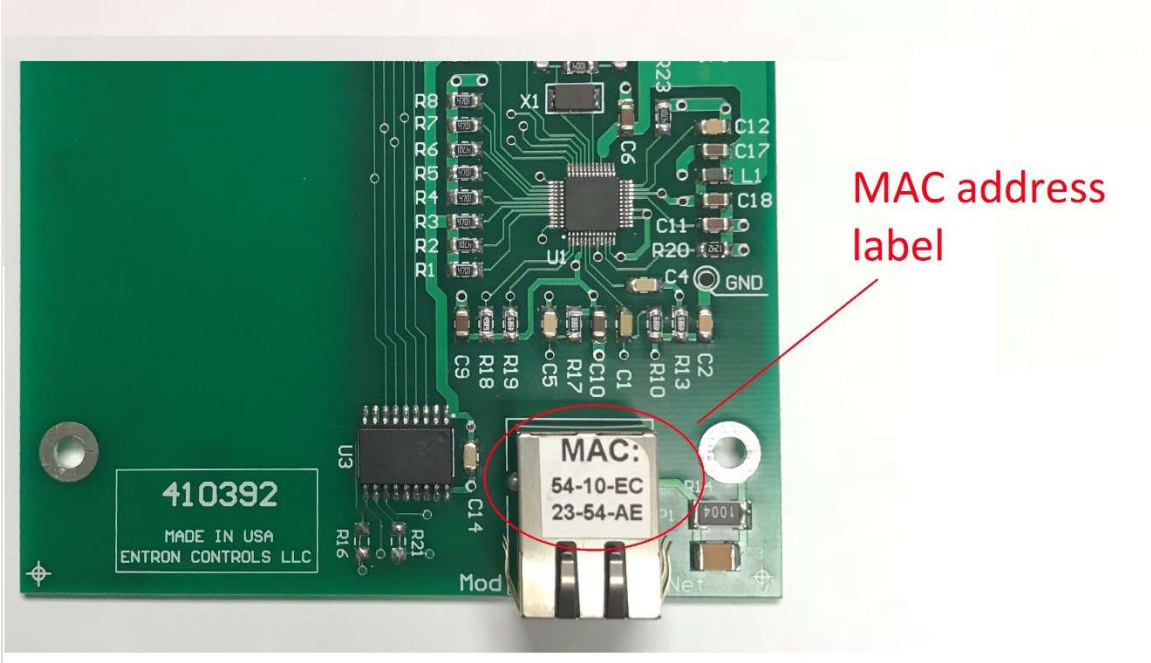

Figure 2-2. *MAC address label*

3.0 MODBUS PROTOCOL

The EN6001 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 3-1 shows.

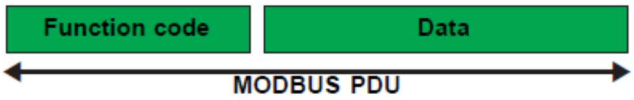

Figure 3-1. *Modbus Protocol Data Unit*

In Modbus PDU, function code indicates to server (EN6001) what kind of action to perform. EN6001 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.

3.1 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 3-2 shows.

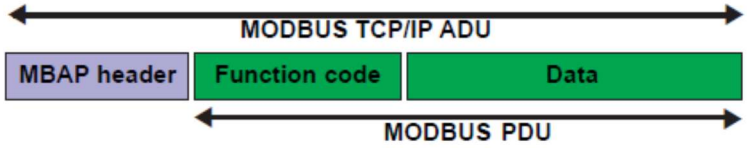

Figure 3-2. *Modbus Application Data Unit (ADU) over serial interface*

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 – Includes function code and optional data as Figure 3-1 shows.

3.2 MODBUS FUNCTION CODES

EN6001 implements Modbus function code 4 (0x04), function code 16 (0x10) and function code 43 (0x2B), MEI value 128 (0x80), as shown in Table 3-1.

Table 3-1. *Modbus function codes*

Function Code	Modbus Description	EN6001 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

3.2.1 MODBUS FUNCTION CODE 4

Modbus function code 4 is used to read EN6001 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 EN6001, the PDU is 5 bytes in length.

Table 3-2. *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	

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

Table 3-3. *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 the control sends back an error message, the PDU will be 2 bytes in length.

Table 3-4. *Function code 4 response PDU (with error)*

Address Offset (bytes)	Data Name	Value
0	Function code=0x04	0x84
1	Exception code	1: wrong function code 2: wrong address 3: wrong quantity 4: other errors

The EN6001 maps data/control status to addresses shown in Table 3-5.

Table 3-5. *Function code 4 data address and name*

Address	Data Name	Value
100	Local input port status	Bit0: FS1 Bit1: FS2 Bit2: TC1 Bit3: ES1 Bit4: Weld/No-weld Bit5: (reserved) Bit11-6: PI6-1 Bit15-12: (reserved)
101	Output port status	Bit3-0: PO4-1
102--103	(Reserved)	
104	Heart-beat counter	
105--119	(Reserved)	
120	Remote input status	Bit5-0: Input6–Input1
121--699	(Reserved)	
700--799	Control status	
800---	(Reserved)	

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 5-2A and 5-2B.

3.2.2 MODBUS FUNCTION CODE 16

Modbus function code 16 is used to write setting data to the EN6001 control. 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 EN6001, the PDU is (6 + data quantity x 2) bytes in length.

Table 3-6. *Function code 16 data request PDU*

Address Offset (bytes)	Data Name	Value
0	Function code=0x10	16(0x10)
1-2	Starting address	
3-4	Quantity of the setting data	

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

Table 3-7. *Function code 16 response PDU (without error)*

Address Offset (bytes)	Data Name	Value
0	Function code=0x04	0x10
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 3-8. *Function code 16 response PDU (with error)*

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

The EN6001 maps data/output states to addresses shown in Table 3-9.

Table 3-9. *Function code 16 data address and name*

Address	Data Name	Value
120	Remote input port status	Bit5-0: Input6-1 Bit15-6: (reserved)

3.2.3 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 3-10 shows data structure of function code 43 data request PDU.

Table 3-10. *Function code 43 data request PDU*

Address Offset (bytes)	Data Name	Value
0	Function code	43(0x2B)
1	MEI Type	128(0x80)
2	Message ID	
3…	Optional message data	

When EN6001 responds to master, if no error, the PDU structure will be (3 bytes + message data) in length, as Table 3-11 shows. If there are errors, the PDU structure will be 3 bytes in length, as Table 3-12 shows.

Table 3-11. *Function code 43 response PDU (without error)*

Address Offset (bytes)	Data Name	Value
0	Function code	43(0x2B)
1	MEI Type	128(0x80)
2	Message ID or ACK	
3…	Optional message data	

Table 3-12. *Function code 43 response PDU (with error)*

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

Table 3-13 shows functions which EN6001 supports for Modbus function code 43 (0x2B), MEI type 128 (0x80).

Table 3-13. *Functions supported for function code 43 (0x2B), MEI type 128 (0x80)*

Message ID	Function
0x20	Read control description from control
0x21	Read control status data from control
0x22 – 0x2F	(reserved)
0x30	Reset Errors
0x31	Reset Counters
0x32	Reset Stepper
0x33-0x35	(reserved)
0x36	Set Weld/No-weld
0x37 – 0x3F	(reserved)
0x40	Read Weld Schedule data from control
0x41	Write Weld Schedule data to control
0x42-0x43	(reserved)
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-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	Read Control Description data from control
0x61	Write Control Description data to control
0x62-0xFF	(reserved)

Detailed descriptions of each Message ID data are explained below.

Message ID 0x21: Read control status data 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 5-2A and 5-2B
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	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)
	Byte 3:	Reset code (0-1)
		Reset code=0: No action
		Reset code=1: Reset errors
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
	Byte 2:	NAK

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

Message ID 0x32: Reset Stepper

Control receives: Byte 0: Function Code 0x2B
Byte 1: MEI type 0x80
Byte 2: Message ID (0x32)
Byte 3: Reset code (0-1)
Reset code=0: No action
Reset code=1: Reset Stepper

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
Byte 2: NAK

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: No action
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
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 (0–63)

Control responds:

When no error: Byte 0: Function Code 0x2B
Byte 1: MEI type 0x80

	Byte 2:	Message ID (0x40)
	Byte 3:	Schedule number (0–63)
	Byte 4–51:	Weld Schedule data from Table 5-3
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	Byte 2:	NAK

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 (0x40)
	Byte 3:	Schedule number (0–63)
	Byte 4–51:	Weld Schedule data from Table 5-3

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
	Byte 2:	NAK

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 5-4
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	Byte 2:	NAK

Message ID 0x45: Read Configuration data from 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 5-4

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
	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 3-34:	Counter data from Table 5-5
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	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 3–34:	Counter data from Table 5-5
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
	Byte 2:	NAK

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 5-6
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	Byte 2:	NAK

Message ID 0x49: Write Steeper 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 5-6
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
	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 3–130:	I/O Map data from Table 5-7
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	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 3–130:	I/O Map data from Table 5-7
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
	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 5-8
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	Byte 2:	NAK

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 5-8
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
	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 3–162:	Error Output Map from Table 5-9
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	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 3–162:	Error Output Map from Table 5-9
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

	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 3–66:	Calibration data from Table 5-10
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	Byte 2:	NAK

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 3–66:	Calibration data from Table 5-10
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
	Byte 2:	NAK

Message ID 0x60: Read Control Description data from control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x60)
Control responds:		
When no error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x54)
	Byte 3–34:	Control description data from Table 5-11
When error:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type
	Byte 2:	NAK

Message ID 0x61: Write Control Description data to control

Control receives:	Byte 0:	Function Code 0x2B
	Byte 1:	MEI type 0x80
	Byte 2:	Message ID (0x61)
	Byte 3–34:	Control description data from Table 5-11
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
	Byte 2:	NAK

4.0 ETHERNET/IP PROTOCOL

4.1 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).

EN6001control 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*.

EN6001 Control transfers data through Explicit Messages. The Explicit Message should be transmitted by Unconnected Messages (UCMM packets) over TCP/IP connections. EN6001control accepts up to two TCP connections and one UDP connection over port number 44818 (0xAF12). In addition, the control supports two Modbus connections over port number 502. With this feature, control can communicate with Modbus devices when it is communicating with EtherNet/IP devices.

To transfer time-critical data with PLC, HMI and other EtherNet/IP devices, EN6001 control offers Explicit Messaging. These time-critical data include remote Input/Output data and the control status data. As ODVA specifications states, Explicit message uses UDP over port number 2222.

4.2 EXPLICIT MESSAGING

A PLC, HMI or any EIP devices will send an Explicit Message to EN6001 control 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. Most of EN6001's Objects only have one Instance; so when the Instance Code is required for those Objects, value should always be 1. For those Objects which support more than one instances, the range of Instance value will be given in this manual.

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.

4.2.1 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 4-1.

Table 4-1. *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 10391 to EN6001 as Product Code.
4. **Revision** – Consists of Major and Minor Revisions, identifying Revision of the firmware.
5. **Device Status** – Represents current status of the control. With the current firmware revision, EN6001 does not use this parameter; the control will return a number "0" for this parameter.
6. **Serial Number** – Represents the Serial Number of Communication Card.
7. **Product Name** – ENTRON assigns the text string "ENTRON EN6001" to EN6001 as the Product Name.

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 Attribute
	1 through 6 for getting/setting Instance Attribute

Class Attributes

The Class Attribute data are described in Table 4-2.

Table 4-2. *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		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 4-3.

Table 4-3. *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

Semantics of Instance Values

1. **Status** – Interface status. The control only uses Bit3-0 to indicate following status:

Bit3-0 = 1 Interface Configuration attribute contains valid configuration obtained from nonvolatile storage.

2. **Configuration Capability** – This attribute is a bitmap that indicates the device's support for optional network configuration capability.

The value is 0x00000010 which indicates that Interface Configuration attribute is settable.

3. **Interface Control Flags** – Configuration Control attribute is a bitmap used to control network configuration.

The value is 0 to indicate that the control shall use statically-assigned IP configuration values.

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 the control to operate as a TCP/IP node. The contents of this attribute include parameters shown in Table 4-4.

Table 4-4. *Interface Configuration Attribute*

Name	Access Rule	Meaning	Default Setting
IP address	Get/Set	EN6001's IP address	192.168.0.100
Network mask	Get/Set	EN6001's network mask	255.255.255.0
Gateway address	Get/Set	EN6001's gateway	192.168.0.1
Name server		EN6001 does not use this attribute	
Name server 2		EN6001 does not use this attribute	
Domain name		EN6001 does not use this attribute	

6. **Host Name** – For this Communication Card, the value is "ENTRON EN6001".

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
Class ID:	0xF6
Instance ID:	1 for getting/setting Instance Attribute only
Attribute:	1 through 7 for getting Class Attribute
	1 through 3 for getting/setting Instance Attribute

Class Attributes

The Class Attribute data are described in Table 4-5.

Table 4-5. *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		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		1
		Optional Services	UINT		0x000e
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 4-6.

Table 4-6. *Ethernet Link Object Instance Attributes*

Attribute Code	Access Rule	Name	Data Type	Description	Value
1	Get	Interface Speed	UDINT	Interface speed currently in use	
2	Get	Interface Flags	DWORD	Interface status flags	
3	Get	Physical Address	ARRAY of 6 USINTs	MAC layer address	

Semantics of Instance Values

1. **Interface Speed** – The Interface Speed attribute shall indicate the speed at which the interface is currently running. The value shall be as follows:

10	the interface is running at 10 Mbps
100	the interface is running at 100 Mbps

2. **Interface Flags –** The Interface Flags attribute contains status and configuration information about the physical interface. EN6001 only use bit0 and bit1, the value shall be as follows:

Table 4-7. *Ethernet Link Object Interface Flags*

Bit	Name	Description
0	Link Status	Indicates whether or not the IEEE 802.3 communications interface is connected to an active network. 0 indicates an inactive link; 1 indicates an active link.
1	Half/Full Duplex	Indicates the duplex mode currently in use. 0 indicates the interface is running half duplex; 1 indicates the interface is running full duplex.

3. **Physical Address** – The Physical Address attribute contains the interface's MAC layer address.

The Physical Address is an array of octets. The recommended display format is "XX-XX-XX-XX-XX-XX", starting with the first octet

4.2.2 ENTRON-DEFINED OBJECTS

ENTRON defines objects to get/set parameters of EN6001 Control. The objects are listed in Table 4-8. 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 4-8
Instance Code:	1
Attribute Code:	See Table 4-8

Table 4-8. *ENTRON-defined Objects*

Class Code	Attribute Code	Access Rule	Name	Data Structure
0x96	0x64 – 0xA3	Get/Set	Weld schedule data	Table 5-3
0x97			(reserved)	
0x98			General Control data	
	0x64	Get/Set	Configuration data	Table 5-4
	0x65	Get/Set	Counter data	Table 5-5
	0x66	Get/Set	Stepper data	Table 5-6
	0x67	Get/Set	I/O Map data	Table 5-7
	0x68	Get/Set	Use Schedule data	Table 5-8
	0x69	Get/Set	Error output map data	Table 5-9
	0x6A	Get/Set	Calibration data	Table 5-10
	0x6B		(reserved)	
	0x6C	Get	Control Information data	Table 5-1
	0x6D	Get	Control status flags data	Table 5-2A,B
	0x6E	Set	Reset Error Flags	
	0x6F	Set	Reset Counter	
	0x70	Set	Reset Stepper	
	0x71	Set	(reserved)	
	0x72	Set	(reserved)	
	0x73	Set	(reserved)	
	0x74	Set	Set Weld/No Weld data	
	0x75	Get/Set	Control description data	Table 5-11
	0x76-0x9A		(reserved)	
	0x9B	Get/Set	Control Interface I/O	

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 – 0xA3

When Attribute Code is set from 0x64 through 0xc7, the operation will get/set data of Weld Schedule 0–63. The Weld Schedule data structure is described in Table 5-6.

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	No action
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: 0x9B
Instance Code: 1
Attribute Code: 0x74

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

State Code=0	No action
State Code=1	Set control to No Weld mode
State Code=2	Set control to Weld mode

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 4-9.

Table 4-9. *Control Interface I/O Data Instance Attributes*

Attribute Code	Access Rule	Name	Data Structure
0x64	Get	Local I/O ports	STRUCTURE
		Local input port Bit0: FS1 Bit1: FS2 Bit2: TC1 Bit3: ES1 Bit4: Weld/No-weld Bit5: (reserved) Bit11-6: PI6-1	UINT
		Local output port Bit3-0: PO4-1	UINT
		(Reserved)	UINT
		(Reserved)	UINT
		Heart-beat counter	UINT
0x65	Get/Set	Remote I/O ports	STRUCTURE
		Remote input port Bit5-0: Input6–Input1	UINT
		(Reserved)	UINT
		(Reserved)	UINT
		(Reserved)	

4.3 IMPLICIT MESSAGING

PLC, HMI or other Ethernet/IP devices can establish a Class 1 connection with EN6001 control, and then exchange remote Input data, Output data and the control status data. The parameters for setting the connection are shown in Table 4-10:

Originator->Target (O->T) message is used to send the control signal data from the PLC (or any Ethernet/IP devices) to EN6001 control. The O->T message include 2-word (4-bye) data; the data structure is shown in Table 4-11:

Target ->Originator (T->O) message is used to send the control status data from EN6001 control to the PLC (or any Ethernet/IP devices). The T->O message include 100-word (200-bye) data; the data structure is shown in Table 5-2A and Table 5-2B.

Table 4-10 *Parameters for Class 1 Connection*

Originator->Target (O->T) Connection	
Connection point	170
Data Size	4 bytes (2 words) with Run/Idle Header
Connection Rate (RPI)	100—10000 mS
Transport Type	Point to Point
Target->Originator (T->O) Connection	
Connection point	120
Data Size	200 bytes (100 words) without Run/Idle Header
Connection Rate (RPI)	100—10000 mS
Transport Type	Point to Point
Transport Trigger	Cyclic
Configuration	
Configuration Instance	1

Table 4-11 Data structure of O->T message

Byte Offset	**Size (bytes)**	**Name**	**Description**
0	2	Control signal [0]	Bit0: FS1 Bit1: FS2 Bit2: PS1 Bit3: Retraction input Bit4: Part counter (PCTR) reset Bit5: Error reset Bit6: TT1 Bit7: Interlock input Bit8: Edit lock Bit9: Escape Bit10: Back step Bit11: 2nd stage Bit12: Weld counter (WCTR) reset Bit13: Stepper reset Bit15-14: (Reserved)
2	2	Control signal [1]	Bit0: Schedule select 1 Bit1: Schedule select 2 Bit2: Schedule select 4 Bit3: Schedule select 8 Bit4: Schedule select 16 Bit5: Schedule select 32 Bit15-6: (Reserved)

5.0 DATA STRUCTURES

5.1 CONTROL INFORMATION DATA FOR READ FUNCTION

Table 5-1 is used only for read function in MODBUS protocol

Table 5-1. *Control information data (64 bytes)*

Byte Offset	Size (bytes)	Name	Units/Description/Notes	Min	Max
0	2	Control type	0x40 for EN6001	0	65535
2	2	Firmware version	Firmware version Bit15-8: Major version number Bit7-0: Miner version number		
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)			

5.2 CONTROL STATUS DATA

The first 5 words of Table 5-2A include 80-bit data and indicate 80 error messages or system status. The detail of word address and bit offset information are shown in Table 5-2B.

Table 5-2A. *Control status data (200 bytes)*

Byte Offset	Size (Bytes)	Name	Units/Description/Notes	Min	Max
0	10	Error Flag[0-4]	80-bit error flags		
10	2	(Reserved)			
12	2	Control output	Bit0: Valve 1 Bit1: Valve 2 Bit2: Valve 3 Bit3: End of sequence (EOS) Bit4: Control not ready Bit5: Major error Bit6: Retraction output Bit7: Count end Bit8: Any error Bit9: Step end Bit10: Interlock output Bit11: Water saver Bit12: Force error (PS1)		

			Bit13: AVC error Bit14: SCR short Bit15: Current error
14	2	(Reserved)	
16	2	Control signal [0]	Bit0: FS1 Bit1: FS2 Bit2: TC1 Bit3: ES1 Bit4: NW1 Bit5: DC Safety Relay status Bit6: PS1 Bit7: Retraction input Bit8: Part counter (PCTR) reset Bit9: Error reset Bit10: TT1 Bit11: Interlock input Bit12: Edit lock Bit13: Escape Bit14: Back step Bit15: 2nd stage
18	2	Control signal [1]	Bit0: Weld counter (WCTR) reset Bit1: Stepper reset Bit2: Schedule select 1 Bit2: Schedule select 2 Bit2: Schedule select 4 Bit2: Schedule select 8 Bit2: Schedule select 16 Bit2: Schedule select 32 Bit8: AC Safety Relay status Bit15-9: (Reserved)
20	2	Input port status	Bit0: FS1 Bit1: FS2 Bit2: TC1 Bit3: ES1 Bit4: NW1 Bit5: DC Safety Relay status Bit6: PI1 Bit7: PI2 Bit8: PI3 Bit9: PI4 Bit10: PI5 Bit11: PI6 Bit12: AC Safety Relay status Bit15-13: (Reserved)
22	2	(Reserved)	
24	2	Output port status	Bit 0: Valve 1

			Bit 1: Valve 2 Bit 2: Valve 3 Bit 3: Safety Relay latch Bit 4: PO1 Bit 5: PO2 Bit 6: PO3 Bit 7: PO4 Bit 15-7: (Reserved)		
26	2	Heartbeat	Heartbeat counter	0	65535
28	2	AC line voltage	1unit= 1V	0	850
30	2	AC line frequency	1unit= 0.1Hz	0	5000
32	2	(Reserved)			
34	2	Schedule selected	Bit7-0: Selected weld schedule 0-63= Weld schedule 0-63 Bit15-8: Schedule select mode 0= internal select mode 1= external select mode		
36	6	(Reserved)			
42	2	Weld1 pulse width		0	99
44	2	Weld1 current		0	12000
46	2	Weld2 pulse width		0	99
48	2	Weld2 current		0	12000
50	2	Part counter		0	60000
52	2	Weld counter		0	9999
54	12	(Reserved)			
66	2	Stepper count		0	9999
68	100	(Reserved)			
168	2	Conduction Angle1	1 unit= 1 degree	0	180
170	2	Conduction Angle2	1 unit= 1 degree	0	180
172	2	Power factor delay	1 unit = 1%	0	99
174	24	(Reserved)			
198	2	Control type	(Reserved for ENLINK)		

Table 5-2B. *Bit definition of Error Flags*

Error Flags		
Word	**Bit address**	**Error description**
0	0	Configuration error
	1	Calibration error
	2	Schedule error
	3	Use schedule error
	4	(Reserved)
	5	Counter error

	6	Stepper error
	7	I/O map error
	8	E-stop error
	9	TC1(Contactor) error
	10	P1 No-weld
	11	PS1 error
	12	SCR short
	13	2nd stage error
	14	TT1(Transformer) error
	15	Interlock error
1	0	(Reserved)
	1	(Reserved)
	2	High current 1
	3	Low current 1
	4	High current 2
	5	Low current 2
	6	High line voltage
	7	Low line voltage
	8	PCTR counter end
	9	Stepper end
	10	High pulse width1
	11	Low pulse width1
	12	High pulse width2
	13	Low pulse width2
	14	Tip dress Pre-warn
	15	AVC error
2	0	Power on with STARTs closed
	1	SYNC error
	2	Panel No-weld
	3	DC Safety relay error
	4	AC Safety relay error
	5	No-coil with Constant Current mode
	6	(Reserved)
	7	(Reserved)
	8	(Reserved)
	9	(Reserved)
	10	(Reserved)
	11	Low current 1 pre-warn
	12	(Reserved)
	13	Low current 2 pre-warn
	14	(Reserved)
	15	(Reserved)
3	0	(Reserved)
	1	(Reserved)
	2	(Reserved)
	3	(Reserved

	4	(Reserved)
	5	(Reserved)
	6	(Reserved)
	7	(Reserved)
	8	(Reserved)
	9	(Reserved)
	10	Retraction input closed
	11	PS1 not ready
	12	Retract not ready
	13	2nd stage not ready
	14	(Reserved)
	15	Interlock not ready
4	0-15	(Reserved)

5.3 WELD SCHEDULE DATA

Table 5-3. *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 x =0-2: Valve 1-3	0	7
4	8	(reserved)			
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 (%)	0	99
20	2	Current1 Limit Low	Current1 Monitor Low (%)	0	99
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 (%)	0	99
34	2	Current2 Limit Low	Current2 Monitor Low (%)	0	99
36	2	Cool2 Time	Cool 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	Bit 0: Weld1 Regulation mode - 0= Pulse width, 1= Constant current Bit 1: Weld1 current monitor – 0= Off, 1= On		

			Bit 2: Weld2 regulation mode - 0= Pulse width, 1= Constant current Bit 3: Weld2 Current Monitor - 0= Off, 1= On Bit 4-6: (reserved) Bit 7: Weld1 pulse width monitor - 0= Off, 1= On Bit 8: Weld2 pulse width monitor - 0= Off, 1= On Bit 9: (Reserved) Bit 10: Current1 pre-limit monitor – 0=Off, 1= On Bit 11: Current2 pre-limit monitor – 0=Off, 1=On Bit 12-15: (reserved)		
46	2	Cycle Mode	0= Non-repeat 1= Repeat 2= Chained 3= Successive 4= Wait Here	0	4
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: (-15% ~ +15%)	35	65
58	2	SQZ Delay Time	Squeeze delay time (Air-over-oil gun =off) or Advance time (Air-over-oil gun =On)	0	99
60	2	Block Delay Time	Blocking delay time for Air-over-oil gun	0	99
62	2	(reserved)			
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	Current1 Pre-limit count	Cycle count to report Current1 Pre-low error	0	99
70	2	Current 2 Pre-limit count	Cycle count to report Current2 Pre-low error	0	99
72	24	(reserved)			

5.4 CONFIGURATION DATA

Table 5-4. *Configuration data (64 bytes)*

Byte Offset	Size (Bytes)	Name	Units/Description/Notes	Min	Max
0	2	Seam Mode	0: Spot 1: Seam1 2: Seam2	0	2
2	2	Retraction Mode	0: None 1: Momentary 2: Maintained	0	2
4	2	Schedule Select Mode	0: Internal 1: External	0	1
6	2	Second Stage Mode	0: After Squeeze 1: Before Squeeze	0	1
8	4	(reserved)			
12	2	Current regulation mode	0: Primary 1: Secondary 2: No-coil	0	2
14	2	Beat Mode	0: Non-Beat 1: Beat During Squeeze 2: Beat During Squeeze + Weld 3: allow Wait-Here mode (CM=4)	0	3
16	2	AVC Mode	0: Off 1-10: 1% - 10%	0	10
18	2	Voltage Limit High	Voltage monitoring high limit (V)	160	690
20	2	Voltage Limit Low	Voltage monitoring low limit (V)	160	690
22	2	Max Current Offset	0-15 %	0	15
24	2	Water Saver Delay	Delay time to turn off Water saver contact (seconds)	0	199
26	2	Half Cycle Mode	0: Off 1: + 2: - 3: Alternative	0	3
28	2	(reserved)			
30	2	Config Mode	Bit0-1: Action on error 0= Continue 1= Stop on Error 2=Head lock on Error Bit2: (reserved) Bit3-4: Air-over-oil work mode 0= Off 1= Air-over-oil without retraction 2= Air-over-oil with retraction		

			Bit5: Voltage Monitoring 0= Off 1= On Bit6: (reserved) Bit7: 87' Delay 0= Off 1= On Bit8-15: (reserved)		
32	2	EOS Mode	0: Pulsed 1: Handshake	0	1
34	2	Blanking	Blanking cycles	0	99
36	2	AVC Nominal	AVC nominal voltage (V)	187	633
38	2	Retract Open Time	Retract open time for Air-over- oil gun (cycles)	0	99
40	2	Retract Close Time	Retract close time for Air-over-oil gun (cycles)	0	99
42	2	Display Return Delay	Delay time for display returning to Status Page 1 0: Will not return to Status Page 1 1~10: Delay time= 1~10 minutes	0	10
44	2	(reserved)			
46	2	Power Factor setting	0: Auto power factor 1-99: power factor=0.01-0.99	0	99
48	16	(reserved)			

5.5 COUNTER DATA

Table 5-5. *Counter data (32 bytes)*

Byte Offset	Size (Bytes)	Name	Units/Description/Notes	Min	Max
0	2	Part Counter	Counter for part	0	60000
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)			

5.6 STEPPER DATA

Table 5-6. *Stepper data (96 bytes)*

Byte Offset	Size (Bytes)	Name	Units/Description /Notes	Min	Max
0	2	Stepper Count[0]	Count numbers for Step 1-10	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]	Heat increments for Step 1-10 (Heat %)	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]			
40	2	Stepper Current Inc[0]	Current increments for Step 1-10 (10A)	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	20	(reserved)			
80	2	Stepper Control	0: Disable 1: Enable	0	1
82	2	Stepper Count Now	Count number	0	9999
84	2	Tip-dressing Pre-warn Count	Count setting for Tip-dressing warning	0	9999
86	10	(reserved)			

5.7 I/O MAP DATA

Table 5-7. *I/O Map settings data (128 bytes)*

Byte Offset	Size (Bytes)	Name	Description/Port functions	Min	Max
0	2	IO Map[0]	Programmable input port PI1- 0: TT1 1: 2nd Stage 2: Back Step 3: PCTR Reset	0	3
2	2	IO Map[1]	Programmable input port PI2- 0: Edit Lock 1: PS1 2: Interlock 3: WCTR Reset	0	3
4	2	IO Map[2]	Programmable input port PI3- 0: Error Reset 1: Sch. Select1 2: Stepper Reset 3: 2nd Stage	0	3
6	2	IO Map[3]	Programmable input port PI4- 0: Interlock 1: Sch. Select2 2: Error Reset 3: PS1	0	3
8	2	IO Map[4]	Programmable input port PI5- 0: Back Step 1: Sch. Select4 2: Retraction 3: TT1	0	3
10	2	IO Map[5]	Programmable input port PI6- 0: Stepper Reset 1: Sch. Select8 2: Edit Lock 3: Escape	0	3
12	2	IO Map[6]	Programmable output port PO1- 0: ANY Error 1: Retraction 2: Force Error 3: Major Error 4: None	0	4
14	2	IO Map[7]	Programmable output port PO2- 0: AVC Error 1: Contactor Error 2: Step End 3: EOS	0	4

			4: None		
16	2	IO Map[8]	Programmable output port PO3- 0: Current Error 1: ANY Error 2: Count End 3: Water Saver 4: None	0	4
18	2	IO Map[9]	Programmable output port PO4- 0: Step End 1: Current Error 2: AVC Error 3: Interlock 4: Major Error	0	4
20	2	IO Map[10]	Input sources for Programmable input port PI1-PI6 – Bit[0-5] for PI1-PI6 Bit[x]=0: local Bit[x]=1: remote1 Bit[6-15]: (reserved)		
22	2	IO Map[11]	Input sources for Control signal Bit[0]: FS1 Bit[1]: FS2 Bit[2]: PS1 Bit[3]: Retraction input Bit[4]: Part counter (PCTR) reset Bit[5]: Error reset Bit[6]: TT1 Bit[7]: Interlock input Bit[8]: Edit lock Bit[9]: Escape Bit[10]: Back step Bit[11]: 2nd stage Bit[12]: Weld counter (WCTR) reset Bit[13]: Stepper reset Bit[14-15]: (Reserved) Bit[0-1] Bit[x]=0: uses signal from local input Bit[x]=1: uses signal from Remote2 (EIP Implicit data) Bit[2-15] Bit[x]=0: uses signal from Programmable input (PI) Bit[x]=1: uses signal from Remote2 (EIP Implicit data)		

24	2	IO Map[12]	Input sources for Control signal Bit[0]: Schedule select 1 Bit[1]: Schedule select 2 Bit[2]: Schedule select 4 Bit[3]: Schedule select 8 Bit[4]: Schedule select 16 Bit[5]: Schedule select 32 Bit[6-15]: (Reserved) Bit[0-3] Bit[x]=0: uses signal from Programmable input (PI) Bit[x]=1: uses signal from Remote2 (EIP Implicit data) Bit[4-5] Bit[x]=0: Off Bit[x]=1: uses signal from Remote2 (EIP Implicit data)
26	102	(reserved)	

5.8 USE SCHEDULE DATA

Table 5-8. *Use Schedule data (32 bytes)*

Byte Offset	Size (Bytes)	Name	Units/Description/Notes	Min	Max
0	2	Use Schedule	Internal Schedule selected	0	63
2	30	(reserved)			

5.9 ERROR CODE OUTPUT MAP DATA

Table 5-9. *Error Code output map data (160 bytes)*

Byte Offset	Size (Bytes)	Name	Units/Description/Notes	Min	Max
0	96	Error Map[0-47]	Error output mode for Error code 0-47 (Display as Error 1 – 48) Error Map[x]=0: Miner error Error Map[x]=1: Major error Error Map[x]=2: None X=0-47	0	2
96	64	(reserved)			

5.10 CALIBRATION DATA

Table 5-10. *Calibration data (64 bytes)*

Byte Offset	Size (Bytes)	Name	Units/Description/Notes	Min	Max
0	2	Coil Sensitivity	For secondary current feedback: 135-165 = 135-165 mV/kA For primary current feedback: 126-154 = 1260-1540 mV/kA		
2	2	Max Current	6-100 KA	6	100
4	2	AC Voltage Scale	Scale for AC line voltage measurement - 80-120: 0.8-1.2	80	120
6	2	Turns ratio	Transformer turns ratio	10	255
8	56	(reserved)			

5.11 CONTROL DESCRIPTION DATA

Table 5-11. *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)			

Document History

Version	Date	Description
Original	5/9/2018	First release
A	6/12/2018	Added the third option " No-coil" for "Current feedback mode" in Config
B	3/8/2019	Revised for firmware version 6.00. Added information for EIP Explicit message Revised I/O Map options
C	5/10/2019	Revised for firmware version 7.02. Added 2nd-stage feature for Config. Added the third options for Error Map. Added the fifth options for I/O Map (POs section)