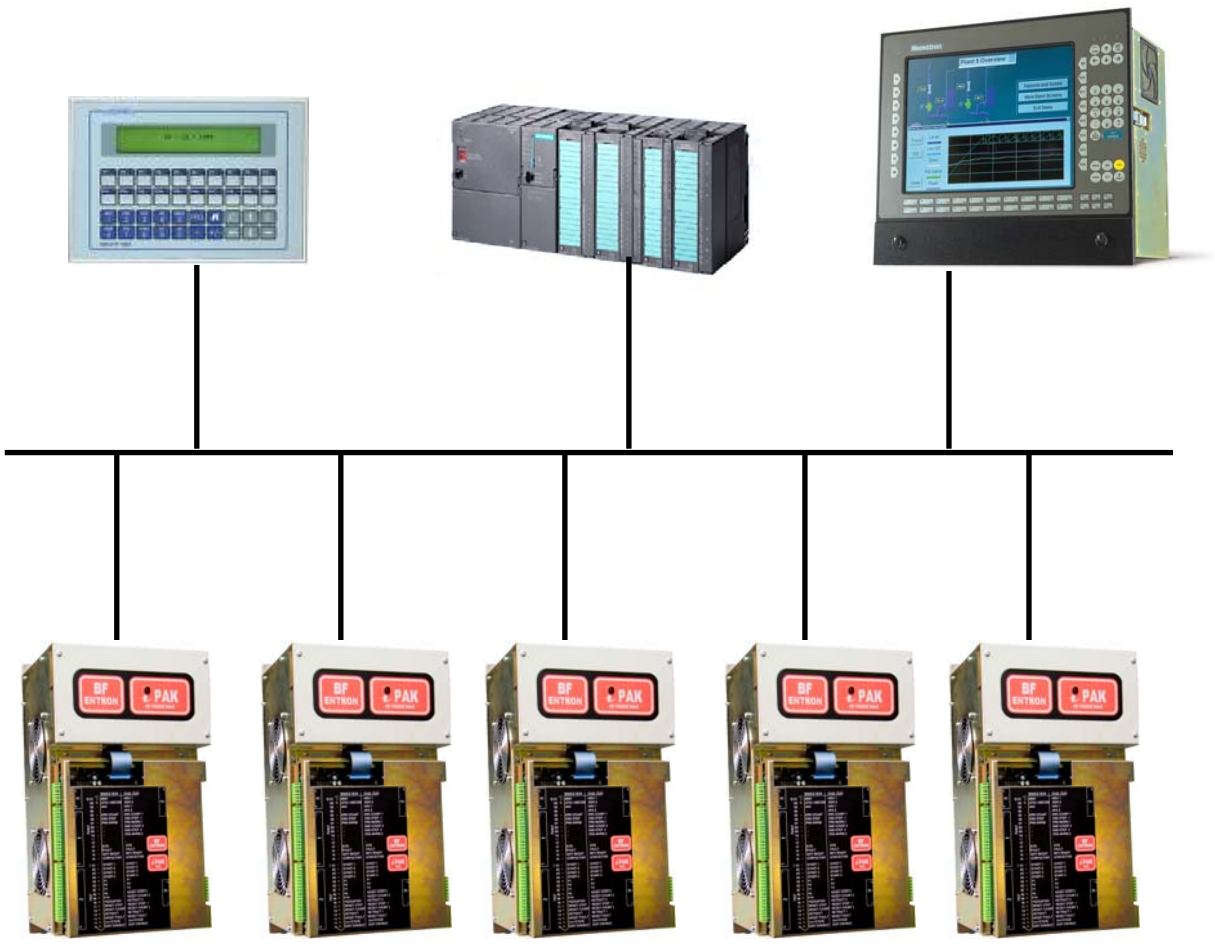
Specification for communications with iPAK



Kevin Cambridge
B.Sc.(Hons), C.Eng, M.I.E.T.
Technical manager
BF Entron ltd.

7/6/18
For firmware version 1.62

Contents

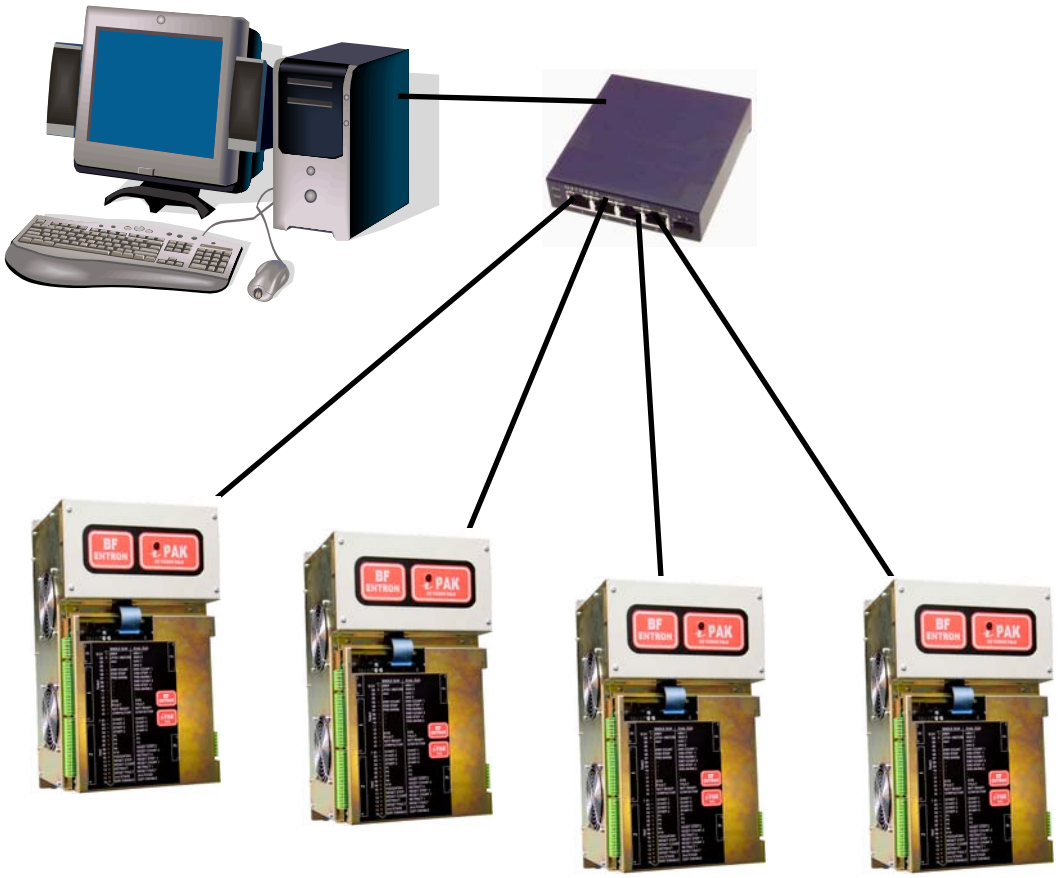
Introduction

The iPAK control offers a number of options for communication of data and status information to a remote device. The choice of which method to use will depend on the availability of software and processing power on the connected device, and on the number of iPAKs to which it must be connected.

Protocol	Connections	Interface		
		RS232	Ethernet	Ethernet/IP
Binary	1	✓	✖	✖
MODBUS-TCP/IP	Multiple	✖	✓	✖
Ethernet/IP	Multiple	✖	✖	✓

The RS232 port is provided as standard. Ethernet and Ethernet/IP require the fitting of an optional adapter board to the iPAK.

Typical Ethernet or Ethernet/IP connection



Typical RS232 connections

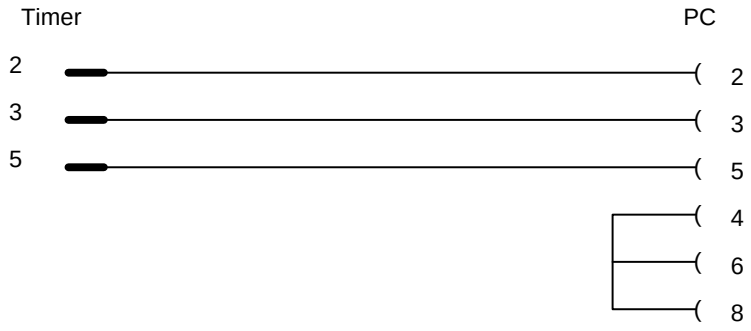
RS232 interface

RS232 3-wire interface, receiver and transmitter only (i.e. no handshaking etc.). Full duplex operation. Connection via standard 9-pin D type connector (Welding control is fitted with female half).

Timer Pin	Timer I/O	Function	Note	Connection to PC
1	Output	+24V	Isolated from RS232	
2	Output	Transmit data		2
3	Input	Receive data		3
4	Input	CTS	Not required by timer	
5		Signal ground		5
6	Output	Pull-up to V+		
7				
8	Output	DTR	Driven high by timer	
9		0V	Isolated from RS232	

You should only require connections on pins 2,3,and 5 at the timer end. At the PC end, you should also link pins 4,6 and 8 together. The cable will require a male connector at the timer end, and a female connector at the PC end. Take care not to connect to pin 1!

Suggested wiring for the cable:



Baud rate.

Baud rate is fixed at 19200.

Character format.

Serial characters are constructed as follows:

1 Start bit
8 Data bits
1 Stop bit
No parity

Ethernet interface

See Ethernet adapter manual.

Ethernet/IP interface

See Ethernet/IP adapter manual.

ASCII protocol

All messages sent to the control must conform to the following format:

[STX]{data}[ETX][HPC_lo][HPC_hi][CR]

where

STX=Start of text character(hex 02).
data=optional data as required.
ETX=End of text character(hex 03).
HPC=Horizontal parity check
(exclusive-or of data).
(sent as 2 ASCII-HEX characters. i.e. 0..9,A..F)
CR= Carriage return (hex 0D).

All messages sent by the control will conform to the same format, with the following exceptions:

1. If no response data is required, the response is a single character ACK (hex 06).

2. If message is not correctly received, the response is a single character NAK(hex 15).

Note that all data is sent least significant digit first, and in ASCII-HEX format.

i.e. data=32 (decimal) (20 hex)
characters sent= [0][2]
ASCII equiv.=[hex 30][hex 32]

i.e. data=254 (decimal) (FE hex)
characters sent= [E][F]
ASCII equiv.=[hex 45][hex 46]

This means that the data field and the HPC field, will only contain the characters 0..9 and A..F.

Example of ASCII protocol
(hexadecimal representation)

ID request:
STX - 78h - ETX - HPC_lo - HPC_hi - CR

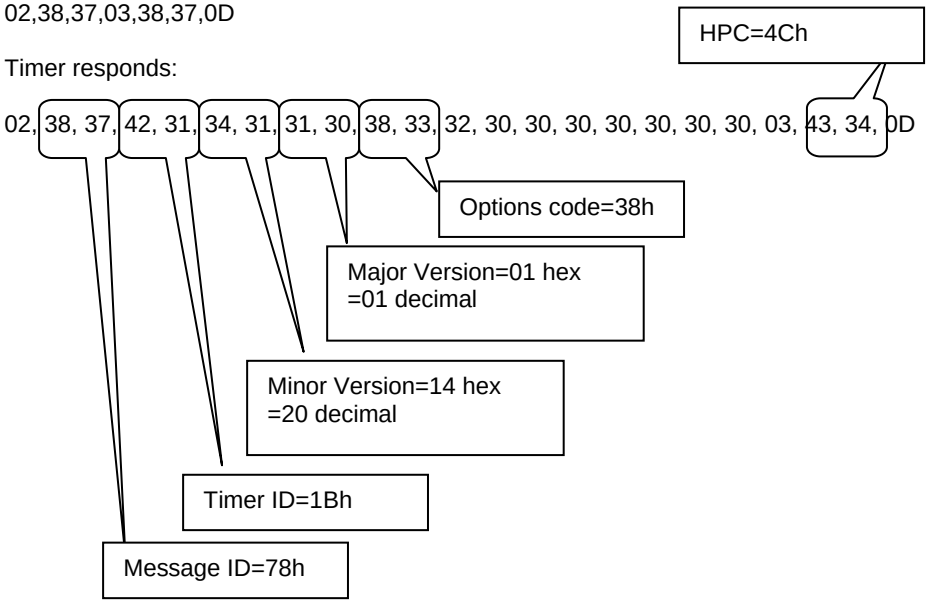
Response:
STX - 78h - < [8 bytes]> - ETX - HPC_lo - HPC_hi - CR

Byte 0 = 1Bh (iPAK ID code)
Byte 1 = 18h (Minor version number)
Byte 2 = 1 (Major version number)
Byte 3 = Options code
Byte 4 = EPLD version no.
Byte 5 = Boot ROM version no.
Byte 6 = 0 (No adapter in slot 1)
Byte 7 = 0 (No adapter in slot 2)

PC sends to timer:

02,38,37,03,38,37,0D

Timer responds:



Binary protocol

All messages sent to the control must conform to the following format:

[STX][13h][00h]{data}[ETX][CRC_lo][CRC_hi]

where

STX=Start of text character(hex 02).
data=optional data as required.
ETX=End of text character(hex 03).
CRC=Cyclic redundancy check (CRC16 algorithm)
Computed on all data from 13h up to end of data field

In order to achieve data transparency, any significant character within the data field must be preceded by a DLE character (10h). Significant characters are: STX, ETX, EOT, ETB, ENQ, DLE, ESC.

All messages sent by the control will conform to the same format, with the following exceptions:

1. If no response data is required, the response is a single character ACK (hex 06).

2. If message is not correctly received, the response is a single character NAK(hex 15).

The binary protocol has the advantage of being up to 50% faster than the ASCII protocol. However, it is more difficult to implement, as computation of the CRC is not trivial.

Example of binary protocol
(hexadecimal representation)

ID request:
STX - 13h - 00h - 78h - ETX - CRC_lo - CRC_hi

Response:
STX - 13h - 00h -78h - < [8 bytes]> - ETX - CRC_lo - CRC_hi

Byte 0 = 1Bh (iPAK ID code)
Byte 1 = 18h (Minor version number)
Byte 2 = 1 (Major version number)
Byte 3 = Options code
Byte 4 = EPLD version no.
Byte 5 = Boot ROM version no.
Byte 6 = 0 (No adapter in slot 1)
Byte 7 = 0 (No adapter in slot 2)

PC sends to timer:

02, 13, 00, 78, 03, CRC_lo, CRC_hi

Timer responds:

02, 13, 00, 78, 10, 1B, 14, 01, 38, 10, 02, 00, 00, 00, 03, CRC_lo, CRC_hi

DLE inserted before 1B(ESC) and 02(STX)

MODBUS TCP/IP protocol(Ethernet)

The iPAK implements Ethernet communications via the MODBUS TCP/IP protocol. This is an open protocol, and full details may be obtained from www.modbus.org.

TCP/IP is used to encapsulate a MODBUS frame. This frame holds a MODBUS PDU consisting of a function code, followed by the actual data to be exchanged.

Thus the general format of the MODBUS frame is:

[MBAP header][Function code]{data}

Details for the MBAP header may be found in the documents from www.modbus.org.

All messages sent by the control will conform to the same format, with the following exceptions:

1. If no response data is required, the response is a single character ACK (hex 06), embedded in the MODBUS frame.

2. General error handling is provided by TCP/IP. Errors within the MODBUS protocol generate MODBUS exceptions, as detailed in the specification.

MODBUS function codes implemented on iPAK

Function code	MODBUS description	MODBUS address range	iPAK function
01(0x01)	Read coils	00001 - 00032	Reads fieldbus input status (see iPAK manual for bit definitions)
02(0x02)	Read inputs	10001 - 10032	Reads fieldbus output status (see iPAK manual for bit definitions)
03(0x03)	Read holding registers Read protocol data	40001 – 40004 42001 - 42255	Reads fieldbus input and output status (see iPAK manual for bit definitions) Data exchange (see data structures)
04(0x04)	Read registers	30001 - 30019	Weld log data (see weld log record data structure –page 20)
05(0x05)	Write coil	00001 - 00032	Set/reset single bit in fieldbus input
15(0x0F)	Write multiple coils	00001 - 00032	Set/reset multiple bits in fieldbus input
16(0x10)	Write holding registers Write protocol data	40001 – 40002 41001 - 41255	Sets fieldbus input status (see iPAK manual for bit definitions) Data exchange (see data structures).
43(0x2B)	Encapsulated interface transport		Data exchange (see data structures).

Any other function code will result in a MODBUS exception.

User data (program parameters, configuration etc) may be accessed either via FC43, or via a combination of FC3 and FC16.

Data exchange via holding registers (FC3 and FC16): The master initiates a data exchange by first writing protocol information to registers 42001 upwards. A response is then obtained by reading registers 42001 upwards.

Example of MODBUS TCP/IP protocol (function codes 3 and 16)
(hexadecimal representation)

ID request: (FC16)
41001 = 0078h (00= null, 78= ID request)

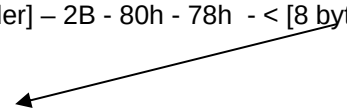
Response: (FC3)
42001 = 0006h (06 = ACK)
42002 = 141Bh (14 = minor version no., 1B = iPAK ID code)
42003 = 3801h (38 = option code, 01 = major version no.)
42004 = 0102h (01 = Boot ROM version no., 02 = EPLD version no.)
42005 = 0000 (00 = No adapter in slot 1, 00 = no adapter in slot 2)

Data exchange via encapsulated interface transport (function code 43): This frame holds a MODBUS PDU having the function code 43(0x2B), the MEI value 128(0x80), followed by the actual data to be exchanged.

Example of MODBUS TCP/IP protocol (function code 43)
(hexadecimal representation)

ID request:
[MBAP header] – 2B - 80h - 78h

Response:
[MBAP header] – 2B - 80h - 78h - < [8 bytes]>



Byte 0 = 1Bh (iPAK ID code)
Byte 1 = 18h (Minor version number)
Byte 2 = 1 (Major version number)
Byte 3 = Options code
Byte 4 = EPLD version no.
Byte 5 = Boot ROM version no.
Byte 6 = 0 (No adapter in slot 1)
Byte 7 = 0 (No adapter in slot 2)

Etehernet/IP protocol

This is an open protocol, and full details may be obtained from www.odva.org. The implementation in the iPAK uses two assembly objects – one to write data and one to read data. The data consists of separate areas for I/O and messaging.

Input assembly

Object	Class	Instance	Attribute
Input (T->O) assembly	4	101	3

data structure

Name	Size (bytes)	Description
Status	2	Module status[1]
Input bits	4	Signal outputs from iPAK –see CPU manual
Comms buffer	240	Message area (response from iPAK)

Status definition

Bit (0=lsb)	Status flag
0	Module error – slave is responding with error codes
1	Module error – communications with slave has timed-out (was previously established)
2	Module error – communication with slave cannot be established at all
3-7	Reserved
8	Module is currently active at factory defaults
9	Module has a new configuration which will take effect upon reset
10-15	Reserved

Output assembly

Object	Class	Instance	Attribute
Output (O->T) assembly	4	102	3

data structure

Name	Size (bytes)	Description
Run/Idle	2	Bit 0 (lsb) = 1(run) or 0(Idle)
Output bits	4	Signal inputs to iPAK –see CPU manual
Comms buffer	240	Message area (to send to iPAK)

Messages

The data field within the transmitted frames consists of a stream of bytes, made up as follows:

Byte #	
0	Message ID
1	Parameter (optional)
2	Data structure (optional)

Binary protocol: When a message requires no parameter, the (optional) data structure will begin at byte#1.

MODBUS protocol (FC43):When a message requires no parameter, the (optional) data structure will begin at byte#1.

MODBUS protocol (FC3/FC16): When using register based transfers (FC3/FC16), the data always begins at register 41002/42002, regardless of there being a parameter value or not at byte #1. Where there is no data to be returned, register 42002 will contain 0006 (ASCII ACK) as an acknowledgement.

Ethernet/IP protocol: Byte#1 always contains either the parameter, or a null. The data always begins at byte#2. Where there is no data to be returned, byte#0 will contain 06 (ASCII ACK) as an acknowledgement, and byte#1 will be 00 (null).

Parameter definitions:

prog: byte value in the range 0..63
casc: byte value in the range 0..15
cntr: byte value in the range 0..7
cali: byte value in the range 0..7
srec byte value in the range 0..250
epm: word value = embedded maximum primary current limit
wlog: byte value in the range 0..63

Data write functions:

Function	Byte		
	#0	**#1**	**#....**
Load cascade program to timer	60h	casc	Cascade data structure - 32 bytes
Response	ACK		
Load welding program to timer	61h	prog	Program data structure - 64 bytes
Response	ACK		
Load limits program to timer	62h	prog	Limits data structure - 32 bytes
Response	ACK		
Load counter to timer	63h	cntr	Counter data structure - 32 bytes
Response	ACK		
Load counter to timer (exclude count)	93h	cntr	Counter data structure - 32 bytes
Response	ACK		
Load calibration to timer	64h	cali	Calibration data structure - 64 bytes
Response	ACK		
Load configuration to timer	65h	Configuration data structure - 32 bytes	
Response	ACK		
Load transformer data to timer	66h	Transformer data structure - 32 bytes	
Response	ACK		
Load 'USE program' setting to timer	67h	'USE program' data structure - 32 bytes	
Response	ACK		
Load stepper data to timer	6Dh	cntr	Stepper data structure - 96 bytes
Response	ACK		
Load stepper data to timer (exclude count)	6Eh	cntr	Stepper data structure - 96 bytes
Response	ACK		
Load adapter setup file to timer	6Fh	Adapter setup data structure - 32 bytes	
Response	ACK		
Load event data to timer	9Ch	prog	Event data structure - 32 bytes
Response	ACK		
Load output map to timer	9Eh	Output map data structure - 32 bytes	
Response	ACK		
Load input map to timer	A0h	Input map data structure - 32 bytes	
Response	ACK		
Load sequencer record to timer	A2h	srec	Sequencer record data structure – 8 bytes
Response	ACK		
Load printer setup to timer	AAh	Printer setup data structure - 32 bytes	
Response	ACK		
Load Servo(SD) setup to timer	4Eh	50h	SD setup data structure – 192 bytes
Response	ACK		

Data read functions:

Function	Byte			
	#0	**#1**	**#....**	
Read cascade program from timer	70h	casc		
Response	70h	casc	Cascade data structure - 32 bytes	
Read welding program from timer	71h	prog		
Response	71h	prog	Program data structure - 64 bytes	
Read limits program from timer	72h	prog		
Response	72h	prog	Limits data structure - 32 bytes	
Read counter from timer	73h	cntr		
Response	73h	cntr	Counter data structure - 32 bytes	
Read calibration from timer	74h	cali		
Response	74h	cali	Calibration data structure - 64 bytes	epm
Read configuration from timer	75h			
Response	75h	Configuration data structure - 32 bytes		
Read transformer data from timer	76h			
Response	76h	Transformer data structure - 32 bytes		
Read 'USE program' setting from timer	77h			
Response	77h	'USE program' data structure - 32 bytes		
Read electrode association from timer	7Bh			
	7Bh	Electrode association data structure - 64 bytes		
Read program link data from timer	7Ch			
Response	7Ch	Program link data structure - 8 bytes		
Read stepper data from timer	7Dh	cntr		
Response	7Dh	cntr	Stepper data structure - 96 bytes	
Read counter dynamic data	7Eh	cntr		
Response	7Eh	cntr	Counter dynamic data structure - 4 bytes	
Read adapter setup file from timer	7Fh			
Response	7Fh	Adapter setup data structure - 32 bytes		
Read event data from timer	9Bh	prog		
Response	9Bh	prog	Event data structure - 32 bytes	
Read output map from timer	9Dh			
Response	9Dh	Output map data structure - 32 bytes		
Read input map from timer	9Fh			
Response	9Fh	Input map data structure - 32 bytes		
Read sequencer record from timer	A1h	srec		
Response	A1h	Sequencer record data structure – 8 bytes		
Read printer setup from timer	A9h			
Response	A9h	Printer setup data structure - 32 bytes		
Read servo(SD) setup from timer	4Eh	47h		
Response	Servo(SD) setup data structure – 192 bytes			
Read telemetry info	E1h			
Response	Telemetry info structure – 6 bytes			
Read telemetry records	E2h	Telemetry record request structure – 4 bytes		
Response	Telemetry records (8 bytes per record)			

Weld log functions:

Function	Byte		
	#0	**#1**	**#....**
Read weld log size from timer	A6h		
Response	A6h	Weld log size data structure - 2 bytes	
Read weld log record from timer	A7h	Record number	
Response	A7h	Weld log record data structure - 46 bytes	
Reset (clear) weld log	A8h		
Response	ACK		
Read most recent weld log record from timer	7Ah		
	7Ah	Weld log record data structure - 46 bytes	

Status functions:

Function	Byte		
	#0	**#1**	**#....**
Read ID from timer	78h		
Response	78h	Timer ID data structure - 8 bytes	
Read status from timer	79h		
Response	79h	Timer status data structure - 93 bytes	
Reset (clear) status	94h		
Response	ACK		
Reset counter	95h	cntr	
Response	ACK		
Reset stepper	96h	cntr	
Response	ACK		
Read sequencer status	A3h		
Response	A3h	Sequencer status data structure - 28 bytes	
Read fieldbus I/O status from timer	A4h		
Response	A4h	Fieldbus I/O data structure - 8 bytes	

Data structures

Program data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Heat 1	0	999	1 unit = 0.1%
2	2	Heat 2	0	999	1 unit = 0.1%
4	2	PV voltage/ current[1]	0	200	1 unit = 0.5V / 80mA
6	2	Presqueeze time	0	1999	ms
8	2	Squeeze time	0	1999	ms
10	2	Weld 1 time	0	999	ms
12	2	Cool 1 time	0	999	ms
14	2	Weld 2 time	0	999	ms
16	2	Cool 2 time	0	999	ms
18	2	Pulsations	1	9	
20	2	Hold time	0	999	ms
22	2	Off time	0	999	ms
24	2	Current 1	0	60000	A
26	2	Current 2	0	60000	A
28	2	Modes[2]			
30	2	Electrode	0	7	Value corresponds to 1..8
32	2	Upslope time	0	999	ms
34	2	Downslope time	0	999	ms
36	2	Power 1	0	65500	1 unit = 2 W
38	2	Power 2	0	65500	1 unit = 2 W
40	2	OHMA gun close time	0	999	ms
42	2	OHMA gun open time	0	999	ms
44	2	OHMA retract close time	0	999	ms
46	2	OHMA retract delay time	0	999	ms
48	2	OHMA retract open time	0	999	ms
50	2	Voltage 1	0	10000	mV
52	2	Voltage 2	0	10000	mV
54	2	Servo(SD) force[3]	0	32767	
56	2	Reserved			
58	2	Servo(SD) position[3]	0	32767	
60	4	Reserved			

[1] Min and max values should also be restricted to lie within the PV calibration points.

[2] Modes is a bit-field variable, encoded as follows:

Bits 3..0:
- 0000= Weld 1 constant pulse width (P/W)
- 0001= Weld 1 constant current, uncalibrated (CCu)
- 0010= Weld 1 constant current, calibrated (CCC)
- 0011= Weld 1 constant voltage (CV)
- 0100= Weld 1 constant power (POW)

Bits 7..4:
- 0000= Weld 2 constant pulse width (P/W)
- 0001= Weld 2 constant current, uncalibrated (CCu)
- 0010= Weld 2 constant current, calibrated (CCC)
- 0011= Weld 2 constant voltage (CV)
- 0100= Weld 2 constant power (POW)

Bits 14..8: Reserved

Bit 15:
- 0= Normal
- 1= Link

[3] See section on SD set-up data structure for scale factors

Cascade data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	No. of steps	0	8	
2	2	Program for step 1	0	63	
4	2	Program for step 2	0	63	
6	2	Program for step 3	0	63	
8	2	Program for step 4	0	63	
10	2	Program for step 5	0	63	
12	2	Program for step 6	0	63	
14	2	Program for step 7	0	63	
16	2	Program for step 8	0	63	
18	14	Reserved			

Program links data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	8	Link flags[3]			Bit 0 = prog 0, bit 63=prog 63

Limits data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Current monitor	0	1	0= Off, 1=On
2	2	Low limit, weld 1	0	99	% of current 1
4	2	High limit, weld 1	0	99	% of current 1
6	2	Pre-limit, weld 1	0	99	% of current 1
8	2	Low limit, weld 2	0	99	% of current 2
10	2	High limit, weld 2	0	99	% of current 2
12	2	Pre-limit, weld 2	0	99	% of current 2
14	2	Pre-limit count	0	99	spots
16	2	Pressure monitor	0	1	0= Off, 1=On
18	2	Low limit, pressure	0	99	% of PV output
20	2	High limit, pressure	0	99	% of PV output
22	2	Wait for pressure	0	1	0= Off, 1=On
24	2	Setdown monitor	0	1	0= Off, 1=On
26	2	Setdown min[4]	0	32767	
28	2	Setdown max[4]	0	32767	
30	2	Reserved			

[3] Link flags is a 64 bit field where each bit = 1 if that corresponding program is set to LINK
[4] See section on SD set-up data structure for scale factors

Event data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Event 1 Output	0	15	Output (Q) number
2	2	Event 1 action	0	1	0 = Off, 1 = On
4	2	Event 1 interval[4]	0	6	
6	2	Event 1 offset	0	999	ms
8	2	Event 2 Output	0	15	Output (Q) number
10	2	Event 2 action	0	1	0 = Off, 1 = On
12	2	Event 2 interval[4]	0	6	
14	2	Event 2 offset	0	999	ms
16	2	Event 3 Output	0	15	Output (Q) number
18	2	Event 3 action	0	1	0 = Off, 1 = On
20	2	Event 3 interval[4]	0	6	
22	2	Event 3 offset	0	999	ms
24	2	Event 4 Output	0	15	Output (Q) number
26	2	Event 4 action	0	1	0 = Off, 1 = On
28	2	Event 4 interval[4]	0	6	
30	2	Event 4 offset	0	999	ms

[4] Intervals are encoded thus:
0 = Presqueeze
1 = Squeeze
2 = Weld 1
3 = Cool 1
4 = Weld 2
5 = Cool 2
6 = Hold

Counter data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Count	0	9999	
2	2	Set count	0	9999	0 = counter off
4	2	Stop	0	1	0 = continue, 1 = Stop at end
6	2	Tip dressing	0	1	0 = Off, 1 = On
8	2	Reserved			
10	2	Max dressings	0	9999	
12	2	Dressings done	0	9999	
14	2	Reset to	0	65535	Point on stepper curve
16	16	Reserved			

Stepper data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Step 0: No of spots	0	9999	
2	2	Step 1: No of spots	0	9999	
4	2	Step 2: No of spots	0	9999	
6	2	Step 3: No of spots	0	9999	
8	2	Step 4: No of spots	0	9999	
10	2	Step 5: No of spots	0	9999	
12	2	Step 6: No of spots	0	9999	
14	2	Step 7: No of spots	0	9999	
16	2	Step 8: No of spots	0	9999	
18	2	Step 9: No of spots	0	9999	
20	2	Step 0: heat increment	0	500	1 unit = 0.1%
22	2	Step 1: heat increment	0	500	1 unit = 0.1%
24	2	Step 2: heat increment	0	500	1 unit = 0.1%
26	2	Step 3: heat increment	0	500	1 unit = 0.1%
28	2	Step 4: heat increment	0	500	1 unit = 0.1%
30	2	Step 5: heat increment	0	500	1 unit = 0.1%
32	2	Step 6: heat increment	0	500	1 unit = 0.1%
34	2	Step 7: heat increment	0	500	1 unit = 0.1%
36	2	Step 8: heat increment	0	500	1 unit = 0.1%
38	2	Step 9: heat increment	0	500	1 unit = 0.1%
40	2	Step 0: current increment	0	30000	A
42	2	Step 1: current increment	0	30000	A
44	2	Step 2: current increment	0	30000	A
46	2	Step 3: current increment	0	30000	A
48	2	Step 4: current increment	0	30000	A
50	2	Step 5: current increment	0	30000	A
52	2	Step 6: current increment	0	30000	A
54	2	Step 7: current increment	0	30000	A
56	2	Step 8: current increment	0	30000	A
58	2	Step 9: current increment	0	30000	A
60	2	Step 0: power increment	0	15000	1 unit = 2W
62	2	Step 1: power increment	0	15000	1 unit = 2W
64	2	Step 2: power increment	0	15000	1 unit = 2W
66	2	Step 3: power increment	0	15000	1 unit = 2W
68	2	Step 4: power increment	0	15000	1 unit = 2W
70	2	Step 5: power increment	0	15000	1 unit = 2W
72	2	Step 6: power increment	0	15000	1 unit = 2W
74	2	Step 7: power increment	0	15000	1 unit = 2W
76	2	Step 8: power increment	0	15000	1 unit = 2W
78	2	Step 9: power increment	0	15000	1 unit = 2W
80	2	Stepper on	0	1	0 = Off, 1 = On
82	2	Stepper stop	0	1	0 = continue, 1 = stop at end
84	4	Spots done	0		Dynamic counter (32 bits)
88	8	Reserved			

Configuration data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Type[5]	0	3	
2	2	Retract 1[6]	0	4	
4	2	Regulation	0	1	0 = primary, 1 =secondary
6	2	Contactor time	0	199	seconds
8	2	Blanking	0	1	0 = Off, 1 = On
10	2	Guns[7]	0	4	
12	2	Toroid test	0	1	0 = Off, 1 = On
14	2	Program select	0	1	0 = binary, 1 = 1 of 4
16	2	Measure	0	1	0=Secondary, 1=Primary
18	2	Stop on fault[8]	0	5	
20	2	2nd stage	0	1	0 = After squeeze, 1 = Before Squeeze
22	2	Sequencer	0	1	0 = Off, 1 = On
24	2	Retract 2[6]	0	3	
26	2	I/O source[9]	0	1	
28	2	Scale factor	1	10	
30	2	Sync with log	0	1	0 = Off, 1 = On
32	2	Pressure units	0	1	0 = kN, 1 = lbf
34	2	Analog output	0	1	0 = PV, 1 = Current
36	28	Reserved			

[5] Type is encoded:
0 = Spot
1 = Roll-spot
2 = Seam (2-heat)
3 = Seam (pre-heat)

[6] Retract is encoded:
0 = Simple / none
1 = Hi-lift+
2 = Hi-lift -
3 = Maintained
3 = OHMA retract (gun 1 only)

[7] Guns are encoded:
0 = single gun
1 = dual gun
2 = Multi-gun
3 = Multi gun cascade
4 = OHMA gun

[8] Stop on fault is encoded:
5 = Head-lock, no EOS
4 = Stop, no EOS
3 = Continue, no EOS
2 = Head-lock, EOS
1 = Stop, EOS
0 = Continue EOS

[9] I/O source is encoded:
0 = discrete
1 = fieldbus (map 1)
2 = fieldbus (map 2)

Calibration data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Toroid sensitivity	100	1999	mV/kA
2	2	S:P ratio	1	199	
4	2	Max Primary current	0	1000	A
6	2	PV point 1 force	0	5000/9999	1 unit = 10 N / 1 lbf
8	2	PV point 1 volts	0	200	1 unit = 0.05V
10	2	PV point 2 force	0	5000/9999	1 unit = 10 N / 1 lbf
12	2	PV point 2 volts	0	200	1 unit = 0.05V
14	2	CT ratio	50	10000	Only if Max I_p is programmable
16	2	Current point 1	0	60000	A
18	2	Power point 1	0	65000	1 unit = 2W
20	2	Heat point 1	0	999	1 unit = 0.1%
22	2	Current point 2	0	60000	A
24	2	Power point 2	0	65000	1 unit = 2W
26	2	Heat point 2	0	999	1 unit = 0.1%
28	2	Analog output scale	0	60000	A for 10V/20mA output
30	2	S:P Trim	-500	+500	0.1%
32	2	S:P Offset	-5000	+5000	A
34	2	Analog input gain (ch3)	900	10000	1 unit = 0.001
36	2	Analog input offset (ch3)	-999	+999	1 unit = 0.01 V
38	10	Unused			

Input map data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Input I01	0	1	0= normal, 1 = sequencer
2	2	Input I02	0	1	0= normal, 1 = sequencer
4	2	Input I03	0	1	0= normal, 1 = sequencer
6	2	Input I04	0	1	0= normal, 1 = sequencer
8	2	Input I05	0	1	0= normal, 1 = sequencer
10	2	Input I06	0	1	0= normal, 1 = sequencer
12	2	Input I07	0	1	0= normal, 1 = sequencer
14	2	Input I08	0	1	0= normal, 1 = sequencer
16	2	Input I09	0	1	0= normal, 1 = sequencer
18	2	Input I10	0	1	0= normal, 1 = sequencer
20	2	Input I11	0	1	0= normal, 1 = sequencer
22	2	Input I12	0	1	0= normal, 1 = sequencer
24	2	Input I13	0	1	0= normal, 1 = sequencer
26	2	Input I14	0	1	0= normal, 1 = sequencer
28	2	Input I15	0	1	0= normal, 1 = sequencer
30	2	Input I16	0	1	0= normal, 1 = sequencer

USE program data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Use program number	-1	63	-1=external select
2	30	Reserved			

Output map data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Output Q01	0	2	0 = normal 1 = event 2 = sequencer
2	2	Output Q02	0	2	
4	2	Output Q03	0	2	
6	2	Output Q04	0	2	
8	2	Output Q05	0	2	
10	2	Output Q06	0	2	
12	2	Output Q07	0	2	
14	2	Output Q08	0	2	
16	2	Output Q09	0	3	0 = normal 1 = event 2 = sequencer 3 = fieldbus
18	2	Output Q10	0	3	
20	2	Output Q11	0	3	
22	2	Output Q12	0	3	
24	2	Output Q13	0	3	
26	2	Output Q14	0	3	
28	2	Output Q15	0	3	
30	2	Output Q16	0	3	

Transformer data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Transformer for electrode 1	0	7	Value corresponds to 1..8
2	2	Transformer for electrode 2	0	7	Value corresponds to 1..8
4	2	Transformer for electrode 3	0	7	Value corresponds to 1..8
6	2	Transformer for electrode 4	0	7	Value corresponds to 1..8
8	2	Transformer for electrode 5	0	7	Value corresponds to 1..8
10	2	Transformer for electrode 6	0	7	Value corresponds to 1..8
12	2	Transformer for electrode 7	0	7	Value corresponds to 1..8
14	2	Transformer for electrode 8	0	7	Value corresponds to 1..8
16	16	Reserved			

Adapter setup data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Fieldbus address[10]			
2	2	Fieldbus baud rate[11]	0	2	0 =128k, 1=256k,2=512k
4	2	IP address[12]	0	255	msb
6	2	IP address	0	255	
8	2	IP address	0	255	
10	2	IP address	0	255	lsb
12	2	Subnet mask[12]	0	255	msb
14	2	Subnet mask	0	255	
16	2	Subnet mask	0	255	
18	2	Subnet mask	0	255	lsb
20	2	Gateway[12]	0	255	msb
22	2	Gateway	0	255	
24	2	Gateway	0	255	
26	2	Gateway	0	255	lsb
28	4	Reserved			

[10] Fieldbus address can be:
0..63 for DeviceNet
1..99 for Profibus

[11] Only for DeviceNet
[12] Only for Ethernet/IP

Timer ID data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	1	Timer type	0x1B	0x1B	0x1B= iPAK
1	1	Minor version no.	0	99	
2	1	Major version no.	0	99	
3	1	Options code[13]	0	255	
4	1	EPLD version no.	0	255	
5	1	Boot ROM version no.	0	255	
6	1	Adapter ID (slot 1) [14]	0	255	
7	1	Adapter ID (slot 2)	0	255	

[13] Options code is a bit-field encoded:
Bit 0: Constant power
Bit 1: Multiplex inverter
Bit 2: Reserved
Bit 3: Embedded parameters
Bit 4: Low-force
Bit 5: Constant voltage
Bit 6: Reserved
Bit 7: Reserved

[14] Adapter ID is encoded:
0xE2 Ethernet TCP/IP
0xE3 Ethernet TCP/IP + MF
0xE4 Ethernet/IP V1
0xE5 Ethernet/IP V2
0xDB Profibus-DP
0xDA DeviceNet
0 No adapter

Electrode association data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	64	Electrode no.[15]	0	7	Value corresponds to 1..8

Weld log size data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	1	Index	0	63	Index to most recent record
1	1	Entries	0	64	No. of records in weld log

Weld log record data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Program no.	0	63	
2	2	Counter	0	9999	
4	2	Heat 1	0	999	1 unit = 0.1%
6	2	Heat 2	0	999	1 unit = 0.1%
8	2	Target current/voltage/power 1	0	65000	A / mV / 2W
10	2	Target current/voltage/power 2	0	65000	A / mV / 2W
12	2	Measured current 1	0	60000	A
14	2	Measured current 2	0	60000	A
16	2	Measured power 1	0	65000	1 unit =2W
18	2	Measured power 2	0	65000	1 unit =2W
20	2	Modes			See program data structure
22	2	PV output	0	200	1 unit = 0.05V
24	2	PV output (force)	0	1000	1 unit = 10 N
26	2	PV input	0	200	1 unit = 0.05V
28	2	PV input (force)	0	1000	1 unit = 10 N
30	1	Weld log status[16]			
31	1	Index	0	255	
32	1	Gun	0	7	Value corresponds to 1..8
33	1	Pulse-width	0	100	%
34	2	Measured voltage 1	0	10000	mV
36	2	Measured voltage 2	0	10000	mV
38	2	Force (SD)	0	32767	See SD data scaling
40	2	Reserved			
42	2	Pre-weld position (SD)	0	32767	See SD data scaling
44	2	Post weld position (SD)	0	32767	See SD data scaling

[15] Each byte holds the electrode number for the corresponding weld program.

[16] Weld log status is a bit-field encoded:
Bit 0: Current monitor on
Bit 1: Weld 1 pass
Bit 2: Weld 2 pass
Bit 3: Pressure monitor on
Bit 4: Pressure pass
Bit 5: Weld ON input status
Bit 6: Weld 1 active
Bit 7: Weld 2 active

Sequencer status data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	1	State	0	3	0=idle, 1=running, 2=end, 3=error
1	1	Reserved			
2	2	Program counter	0	249	
4	2	Error line no.	0	249	
6	2	Input states			Bit 0 = I01…Bit 15=I16
8	2	Sequencer output states			Bit 0 = Q01…Bit 15=Q16
10	2	Memory states			Bit 0 = M01… Bit 7= M08
12	2	Counter C01	0	999	
14	2	Counter C02	0	999	
16	2	Counter C03	0	999	
18	2	Counter C04	0	999	
20	2	Counter C05	0	999	
22	2	Counter C06	0	999	
24	2	Counter C07	0	999	
26	2	Counter C08	0	999	

Sequencer record data structure

Byte offset	Size (bytes)	Name
0	2	Opcode[17]
2	2	Main variable
4	2	Sub-variable 1
6	2	Sub-variable 2

[17] encoding:

Opcode		Main variable		Sub-variable 1		Sub-variable 2	
Value	Function	Min	Min	Min	Max	Min	Max
0	Blank						
1	Step	1	999				
2	Await Input ON	1	16				
3	Await input OFF	1	16				
4	Output ON	1	16				
5	Output OFF	1	16				
6	Memory ON	1	8				
7	Memory OFF	1	8				
8	Delay	1	999				
9	Jump	1	999				
10	Gosub	1	999				
11	Return						
12	Set counter	1	8	1	999		
13	Decrement counter	1	8				
14	If counter>0 Jump	1	8	1	999		
15	If output ON Jump	1	16	1	999		
16	If output OFF Jump	1	16	1	999		
17	If Memory ON Jump	1	8	1	999		
18	If Memory OFF Jump	1	8	1	999		
19	If input ON Jump	1	16	1	999		
20	If input OFF Jump	1	16	1	999		
21	Weld	0	64				
22	Await Analog<	1	3	0	999		
23	Await Analog>	1	3	0	999		
24	If Analog < Jump	1	3	0	999	1	999
25	If Analog > Jump	1	3	0	999	1	999

Fieldbus I/O data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	4	Fieldbus input states			Bit 0 = input bit 0….bit 31=input bit 31
4	4	Fieldbus output states			Bit 0 = output bit 0..bit 31=output bit 31

Timer status data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	8	Status flag structure			64 bit field
8	1	Last error program	0	63	
9	1	Last program used	0	63	
10	2	Input status			Bit 0 = I01, Bit 15 = I16
12	2	Output status			Bit 0 = Q01, Bit 15 = Q16
14	1	Selected program	0	63	
15	1	Program select flag	0	1	0 = external, 1 = internal
16	2	PV output (force)	0	1000	1 unit = 10 N / 1 lbf
18	2	PV input (force)	0	1000	1 unit = 10 N / 1 lbf
20	2	Max. force			1 unit = 10 N / 1 lbf
22	2	Min. force			1 unit = 10 N / 1 lbf
24	2	Counter 0	0	9999	
26	2	Counter 1	0	9999	
28	2	Counter 2	0	9999	
30	2	Counter 3	0	9999	
32	2	Counter 4	0	9999	
34	2	Counter 5	0	9999	
36	2	Counter 6	0	9999	
38	2	Counter 7	0	9999	
40	2	Event Output status			Bit 0 = Q01, Bit 15 = Q16
42	2	Sequencer Output status			Bit 0 = Q01, Bit 15 = Q16
44	1	Reserved			
45	4	Stepper 0, spots done	0		
49	4	Stepper 1, spots done	0		
53	4	Stepper 2, spots done	0		
57	4	Stepper 3, spots done	0		
61	4	Stepper 4, spots done	0		
65	4	Stepper 5, spots done	0		
69	4	Stepper 6, spots done	0		
73	4	Stepper 7, spots done	0		
77	1	Last program links	0	64	No. of links used in last sequence
78	1	Configuration error[18]	0	7	
79	2	Measured current 1	0	60000	A
81	2	Measured current 2	0	60000	A
83	4	Electrode status structure			
87	2	Analog input ch.0	0	10000	mV
89	2	Analog input ch.1	0	10000	mV
91	2	Analog input ch.2	0	10000	mV

[18] Configuration errors are encoded:
0 = Illegal TYPE parameter
1 = Configuration data error
2 = Low-force setup error
3 = System file does not match hardware
4 = Illegal GUN parameter
5 = Power module not configured
6 = Illegal combination of Program Select and gun parameters
7 = OHMA retract selected without OHMA gun.

Status flag structure

Bit offset	Name
0	Configuration error
1	Calibration error
2	Module configuration error
3	Power module error
4	Not ready
5	Retract not ready
6	Data error
7	Weld off
8	No current (weld 1)
9	No current (weld 2)
10	Low current (weld 1)
11	Low current (weld 2)
12	Pre-alarm (weld 1)
13	Pre-alarm (weld 2)
14	High current (weld 1)
15	High current (weld 2)
16	Configuration changed
17	No 2nd stage
18	Toroid overrange
19	Toroid open circuit
20	Toroid short circuit
21	Servo mid<retract
22	Servo mid>ref-pinch
23	Servo setdown error
24	Sequencer error
25	Heatsink hot
26	No 24V supply
27	System suspended
28	Servo Bore/Stroke not set
29	Headlocked
30	No adapter
31	Fieldbus inactive

Bit offset	Name
32	Servo MODBUS error
33	Servo not started
34	Servo obstructed
35	Servo pressure warning
36	Servo pressure error
37	Servo no component
38	Servo not at force
39	Servo contact error
40	Low pressure
41	High pressure
42	Transformer hot
43	Stop
44	Fan fail
45	Waiting for pressure
46	Switch off START
47	No current(weld off)
48	Max. pulse width
49	Max. primary current
50	Short circuit
51	Earth fault
52	Inverter voltage error
53	Inverter disabled
54	
55	
56	No voltage
57	Low power (weld 1)
58	Low power (weld 2)
59	High power (weld 1)
60	High power (weld 2)
61	Servo reference mode
62	Servo headset mode
63	Voltage overrange

Electrode status flag structure

Bit offset	Name
0	Stepper 1 end
1	Stepper 2 end
2	Stepper 3 end
3	Stepper 4 end
4	Stepper 5 end
5	Stepper 6 end
6	Stepper 7 end
7	Stepper 8 end
8	Stepper 1 prewarn
9	Stepper 2 prewarn
10	Stepper 3 prewarn
11	Stepper 4 prewarn
12	Stepper 5 prewarn
13	Stepper 6 prewarn
14	Stepper 7 prewarn
15	Stepper 8 prewarn

Bit offset	Name
16	Counter 1 end
17	Counter 2 end
18	Counter 3 end
19	Counter 4 end
20	Counter 5 end
21	Counter 6 end
22	Counter 7 end
23	Counter 8 end
24	Tip dress 1
25	Tip dress 2
26	Tip dress 3
27	Tip dress 4
28	Tip dress 5
29	Tip dress 6
30	Tip dress 7
31	Tip dress 8

Printer setup data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	printer_mode[19]	0	3	
2	2	lines_per_page	10	99	
4	2	print format	0	1	0=table, 1=ASCII
6-31	26	Reserved			

[19] Printer mode is encoded:
0= Off
1= Print all
2= Print fails
3= Print passes

SD set-up data structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Command			Not used, set internally
2	2	Modes			Not used, set internally
4	2	Inputs			Not used, set internally
6	2	Operation			Not used, set internally
8	2				
10	2				
12	2				
14	2				
16	2				
18	2				
20	2				
22	2				
24	2				
26	2				
28	2	Start up			Not used, set internally
30	2	Baud rate			Not used, set internally
32	2	Retract position	0	32767	
34	2	Mid position	0	32767	
36	2	Weld position	0	32767	
38	2				
40	2				
42	2				
44	2				
46	2				
48	2	Weld force 1	0	32767	
50	2	Weld force 2	0	32767	Not used
52	2	Weld force 3	0	32767	Not used
54	2	Weld force 4	0	32767	Not used
56	2				
58	2				
60	2				
62	2				
64	2	Velocity 1	0	32767	Retract to Mid
66	2	Acceleration 1	0	32767	Retract to Mid
68	2	Velocity 2	0	32767	Mid to Weld
70	2	Acceleration 2	0	32767	Mid to Weld
72	2	Velocity 3	0	32767	Weld to Mid
74	2	Acceleration 3	0	32767	Weld to Mid
76	2	Velocity 4	0	32767	Mid to Retract
78	2	Acceleration 4	0	32767	Mid to Retract
80	2	Velocity 5	0	32767	Reference/Headset
82	2	Acceleration 5	0	32767	Reference/Headset
84	2	Velocity 6	0	32767	Not used
86	2	Acceleration 6	0	32767	Not used
88	2	Velocity 7	0	32767	Not used
90	2	Acceleration 7	0	32767	Not used
92	2	Velocity 8	0	32767	Not used
94	2	Acceleration 8	0	32767	Not used
96	2	Touch force	0	32767	
98	2	Obstruction force	0	32767	
100	2	Position resolution	0	32767	
102	2	Force resolution	0	32767	
104	2	Pinch offset	0	32767	
106	2	Over-travel limit	0	32767	
108	2				
110	2				
112	2	P gain	0	32767	Non-dimensional
114	2	S gain	0	32767	Non-dimensional

116	2	D gain	0	32767	Non-dimensional
118	2	Force gain	0	32767	Non-dimensional
120	2				
122	2				
124	2				
128	2				
130	2	Maximum weight	0	32767	
132	2	Minimum weight	0	32767	
134	2	Offset Force	0	32767	
136	2				
138	2				
140	2				
142	2	Stiction	0	32767	Non-dimensional
144	2	Friction	0	32767	Non-dimensional
146	2	Minimum pressure	0	32767	
148	44				

SD data scaling

Position

ENGLISH
SCALING: Register value = (1024)*Position (inches)
LIMIT: 32767

METRIC
SCALING: Register value = (40.31)*Position (mm)
LIMIT: 32767

Velocity

ENGLISH
SCALING: Register value = (256)*Velocity (inches/s)
LIMIT: 32767

METRIC
SCALING: Register value = (10.08)*Velocity (mm/s)
LIMIT: 32767

Acceleration

ENGLISH
SCALING: Register value = (4096)*Acceleration (g's)
LIMIT: 32767

METRIC
SCALING: Register value = (4096)*Acceleration (g's)
LIMIT: 32767

Force

ENGLISH
SCALING: Register value = (16)*Force (lbf)
LIMIT: +/- 32767

METRIC
SCALING: Register value = (3595.50)*Force (kN)
LIMIT: +/- 32767

Offset Force

ENGLISH
SCALING: Register value = (16)*Force (lbf)
LIMIT: +/- 32767

METRIC
SCALING: Register value = (3595.50)*Force (kN)
LIMIT: +/- 32767

Weight/Mass	ENGLISH SCALING: Register value = (16)*Force (lbf) LIMIT: 32767
	METRIC SCALING: Register value = (35.2)*Mass (kg) LIMIT: 32767
Pressure	ENGLISH SCALING: Register value = (16)*Pressure (psi) LIMIT: 32767
	METRIC SCALING: Register value = (232)*Pressure (bar) LIMIT: 32767

Telemetry info structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Records available	0	4096	8 bytes /record, 1ms /record
2	2	Weld index	0	255	Note 2
4	2	Buffer size (bytes)	240	750	Bytes (Note 1)

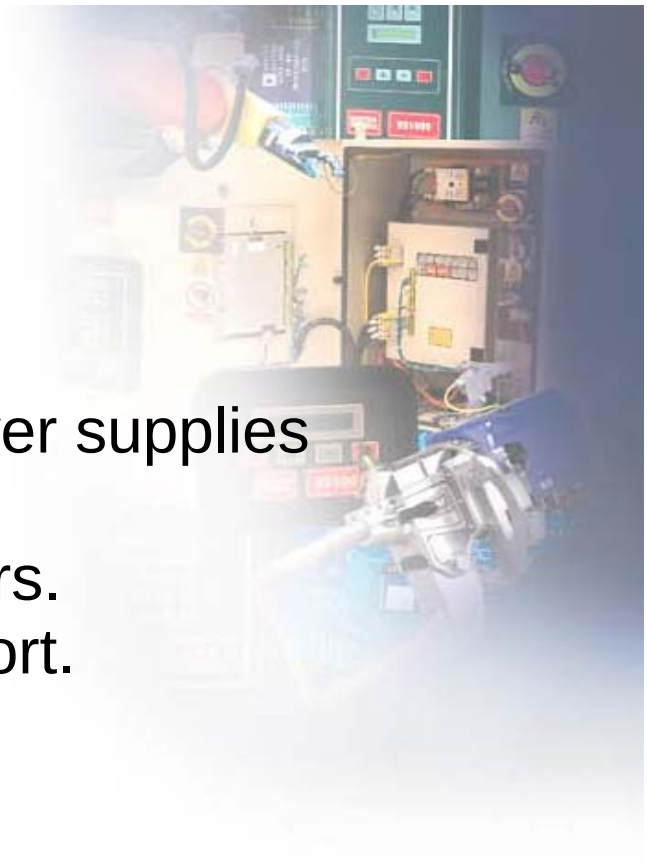
Notes:
1. The buffer size can be used to compute the maximum number of records which may be requested per transaction.
2. The weld index is an arbitrary value which increments each time a weld is completed.
3. Records are made at 1 ms intervals.

Telemetry record request structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Starting record no.	0	4095	
2	2	No. of records to get	1		See note 1 above

Telemetry record structure

Byte offset	Size (bytes)	Name	Min	Max	Units / Notes
0	2	Current	0	60000	A
2	2	Analog ch 1.	0	10000	mV
4	2	Analog ch 2.	0	10000	mV
6	2	Analog ch 3.	0	10000	mV



- Welding controls.
- Thyristors and power supplies
- Power Inverters.
- Meters and Monitors.
- Training and Support.
- Consultancy.

BF Entron Ltd.
Building 80 Bay 1
First Avenue
Pensnett Estate
Kingswinford
DY6 7FQ
England.

Tel: +44 (0)1384 455401
Fax: +44 (0)1384 455551
Email: info@bfentron.co.uk

Web site:
www.bfentron.co.uk