INSTRUCTION MANUAL

700140J

232 OPTIONS

RS232 OPTION for
MICROPROCESSOR BASED
EN1000 Series Weld Controls

WIRING DIAGRAMS
See Appendix D

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

INSTALLATION AND OPERATION MANUAL FOR:
232 Options for EN1000 Series Weld Controls

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

ENTRON Controls, LLC.
Greer, South Carolina 29650

TABLE OF CONTENTS

FIGURES

TABLES

1.0 INTRODUCTION

EN1000 Series Controls with the RS232 option can communicate with a PC, PLC or other devices with the RS232 interface, see Figure 1-1. This manual covers the RS232 Option, RS232 Protocol and Serial Printer usage. See appropriate section for the required information.

The examples in Section 5.4 show what will be displayed on the terminal or computer monitor. The text displayed will differ depending on the software used for Terminal Emulation. Furthermore, the control parameters will differ depending on the model number or type of the control. For example, the EN1001 needs data referring to High and Low Current Limits when in the CONSTANT CURRENT mode. The EN1000 Cascade has 99 schedules and different SCHEDULE parameters as opposed to the EN1000. If an IPSC option exists on the control, corresponding SCHEDULE and EXTENDED FUNCTION data will be displayed.

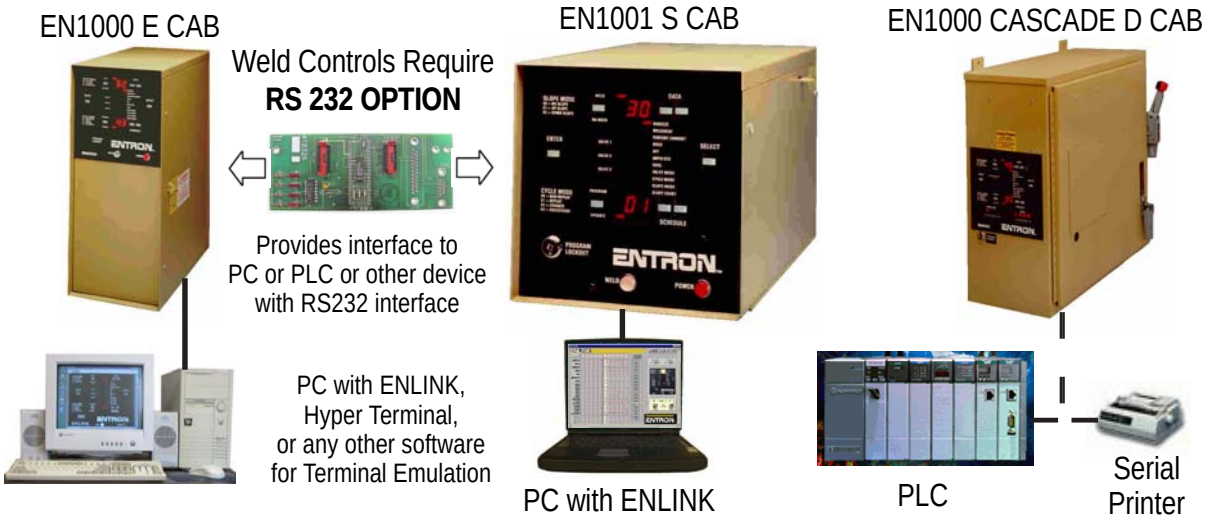


Figure 1-1. *Block diagram*

1.1 EQUIPMENT AND SOFTWARE

1. Any EN1000 Series Control with RS232 option.
2. A 9 pin (straight through) serial cable.
3. A PC with one available COM serial port.
4. Communication software for Terminal Emulation, for example Hyper Terminal (Windows 95/98/2000 or NT) or ENLINK (Windows 95/98/2000), or ProComm or any other communication software for DOS or WINDOWS.
5. RAM Memory Map (see Appendix B) and EEPROM Memory Map (see Appendix C).
6. 485 Options Instruction Manual 700171 (*optional*).

2.0 EN1000 SERIES WELD CONTROLS WITH RS232 OPTION

As mentioned in the first section, EN1000 Series Controls with RS232 option can be used with PC, PLC or other devices with RS232 interface. Control Board connections to the RS232 driver board is shown in Figure 2-1.

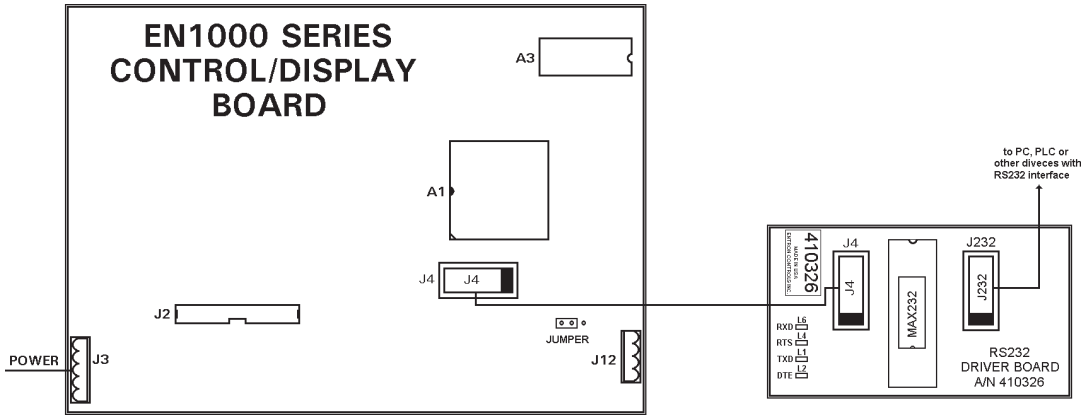


Figure 2-1. *Control Board connection to RS232 driver board*

2.1 PROGRAMMING OF THE CONTROL WITH RS232 OPTION

To program the IDENTIFICATION NUMBER of the control, use the **I.d.** parameter in the EXTENDED FUNCTIONS. This **I.d.** number could be any number from **00** to **99**. IDENTIFICATION NUMBERS **I.d.** = **01** to **64** are reserved for RS485 communication with ENBUS protocol. IDENTIFICATION NUMBERS **I.d.** = **00** and **90** to **99** are reserved for programming baud rate for RS232 serial communication between PC and welding control, as shown in Table 2-1. Default baud rate is 2400 baud, but other standard values are also programmable by programming the appropriate IDENTIFICATION NUMBER. Only EN1001 Cascade and EN1200 (*Obsolete Control*) can be used with 9600 baud.

Table 2-1. *Baud rate*

I.d.	**00**	**01 to 64**	**...**	**93**	**94**	**95**	**96**
Baud rate [baud]	2400	4800	…	1200	2400	4800	9600
Note	default	RS485					

2.1.1 EXTENDED FUNCTION PARAMETERS

The following EXTENDED FUNCTION parameters are relevant to the 232 Options:

I.d. = Control Identification Number for RS485 communications
b.L. = Block Delay, available only if **P.O.**=**07**

C.r. = Constant Current mode
r.A. = Range
t.r. = Turns Ratio

***available only in Constant Current Control* (EN1001 Series)**

C.O. = Current Offset
S.t. = Stepper/Counter Enable

2.1.1 EXTENDED FUNCTION PARAMETERS (cont.)

P.C. = Pressure Control
b.d. = Background Pressure
S.I. = Pressure Sense Input

] ***available only if IPSC Hardware is present***

2.1.2 SCHEDULE PARAMETERS

The associated SCHEDULE functions are accessible by first using SELECT to choose the main function such as SQUEEZE, and then clicking either SCHEDULE push button to access a SCHEDULE function such as **P.r.** or **P.t.**.

SQUEEZE		Associated EXTENDED FUNCTION: **P.C.** = **01** to **06**
P.r. – **P.t.**	Pressure Mode:	**P.C.**=**01** or **02**, where Pressure is in PSI
F.r. – **F.t.**	Force Mode:	**P.C.**=**03** or **04**, where Force is in Lb
A.r. – **A.t.**	mA Mode:	**P.C.**=**05** or **06**, where Current is in mA
WELD		Associated EXTENDED FUNCTION: **S.t.** = **01**
S.C. – **L.S.** – **L.C.**		
CURRENT		Associated EXTENDED FUNCTION: **C.r.** = **10** to **33**
H. i. – **L.o.**		
HOLD through SLOPE COUNT		*No associated EXTENDED FUNCTIONS exist for the SCHEDULE parameters HOLD through SLOPE COUNT.*

NOTICE

For more information and full description of all SCHEDULE and EXTENDED FUNCTIONS parameters, please refer to the appropriate control's Instruction Manual.

2.2 RS232 DRIVER BOARD

An ENTRON RS232 interface (Figure 2-2) is a full duplex device, allowing both transmission and receiving at the same time.

Both LED diodes L1 or TXD and L6 or RXD will be OFF only during communication, otherwise they should be steadily ON. LED L4 should be OFF, and LED L2 ON all the time.

The status of all four LEDs can indicate good connection between this RS232 driver board and Control Board.

A simplified RS232 interface component layout is shown in Figure 2-3.

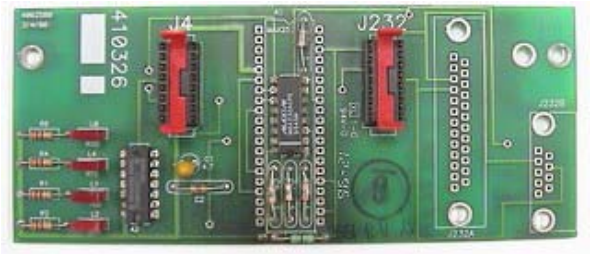


Figure 2-2. *RS232 driver board*

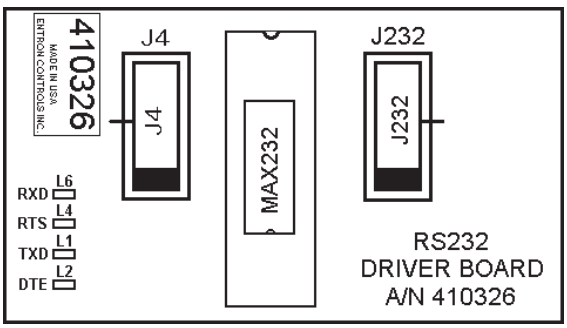


Figure 2-3. *RS232 driver board layout*

3.0 HYPER TERMINAL PROGRAM CONFIGURATION

One of the standard Windows based programs for Terminal Emulation is ***Hyper Terminal***. This program is part of the Windows OS.

1. Start Hyper Terminal Program – With Windows, select: Start / Programs / Accessories / Communication / Hyper Terminal / Hypertrm.exe. Create a new connection and choose a name, for example EN4800. These steps are shown in Figures 3-1 and 3-2.

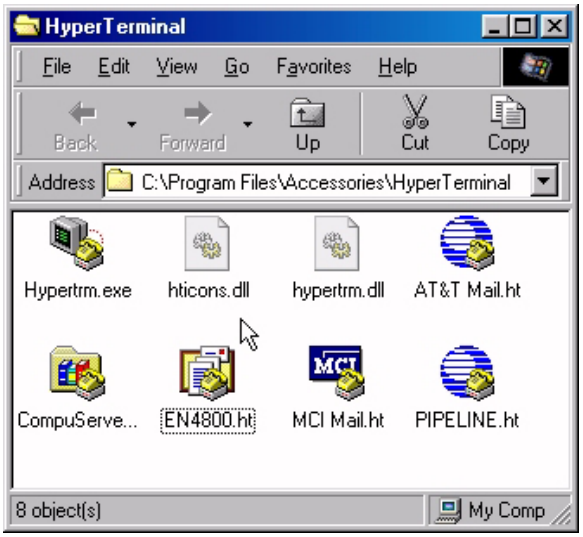


Figure 3-1. *Launching Hyper Terminal program*

Figure 3-2. *Create new connection*

2. Select serial communication port; COM1 is selected in the example, shown in Figure 3-3.
3. Configure COM port: 4800 baud (1200, 2400 or 9600, according to Section 2.1), 8 data bits, No parity, 1 stop bit, and No flow control (Figure 3-4).

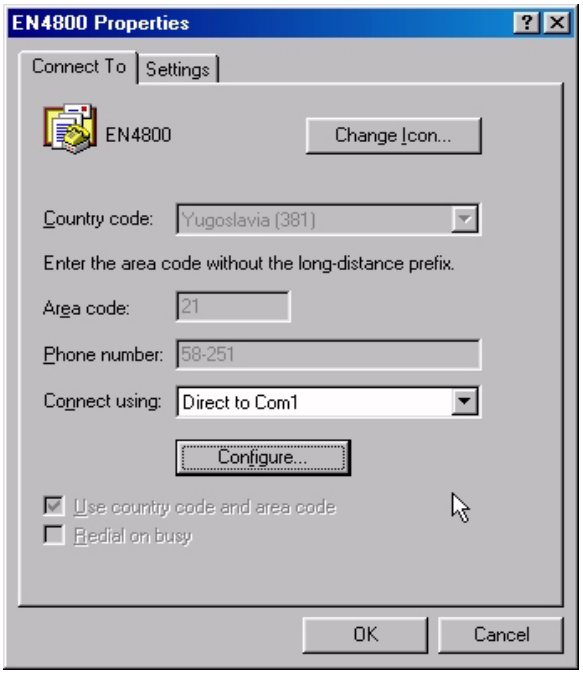


Figure 3-3. *Serial COM port selection*

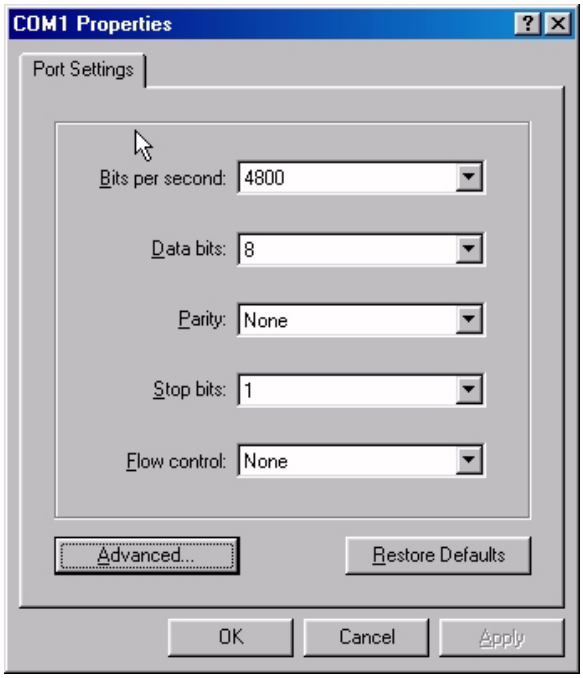


Figure 3-4. *Serial COM port configuration*

3.0 HYPER TERMINAL PROGRAM CONFIGURATION (cont.)

4. Click on the Settings tab (from Figure 3-3 or select Properties and then click on Settings) and adjust ASCII settings as shown in Figures 3-5 and 3-6.

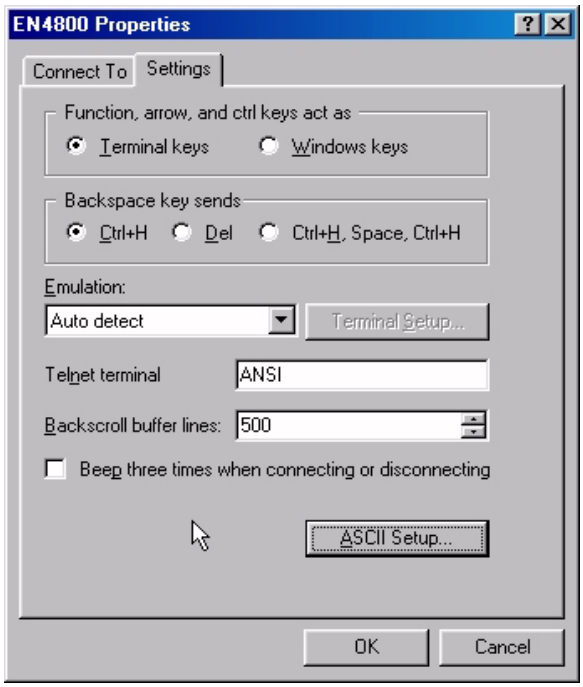


Figure 3-5. *COM port settings*

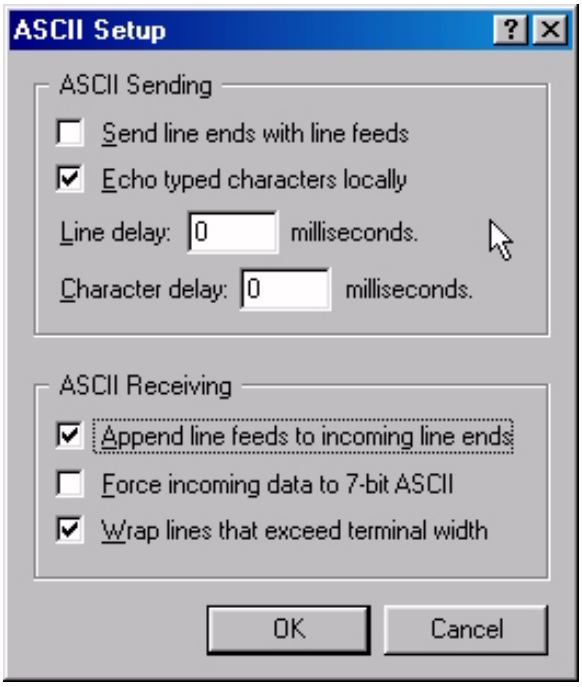


Figure 3-6. *ASCII setup*

4.0 USING ENBUS PROTOCOL OR ENLINK WITH RS232 OPTION

ENLINK for Windows or any software with ENBUS protocol (refer to 485 Options Instruction Manual 700171) can be used with a single weld control with RS232 option. Before using ENLINK, ***I.d.*** number of the control must be between ***01*** to ***64*** (see Section 2.1). ENBUS will be limited to only one weld control (see also Manual 700171). Contact factory for ENLINK software availability.

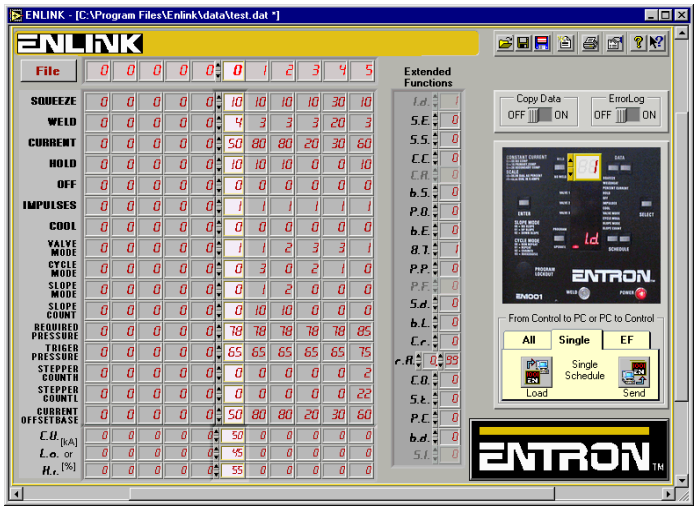


Figure 4-1. *ENLINK for Windows*

5.0 ENTRON RS232 COMMAND/RESPONSE PROTOCOL

ENTRON RS232 Interface is full duplex and operates in ASCII mode with specified RS232 Command/Response protocol. Using ASCII (American Standard Code for Information Interchange) mode, each eight-bit byte in a message is sent as two ASCII characters. The main advantage of this mode is that it allows time intervals to occur between characters without causing an error. In addition, the protocol uses short ASCII commands and an interactive mode.

5.1 COMMUNICATION MODE

EN1000 Series Controls with RS232 Option operate using **ASCII** mode.

5.1.1 COMMUNICATION PARAMETERS

Serial UART parameters are:

- 2400 baud (**default** settings, but 1200, 4800, or 9600 are also available – see Section 2.1)
- 8 data bits
- No parity
- 1 stop bit
- No Flow control

5.1.2 BITS PER BYTE

- 1 **start** bit
- 8 **data** bits
- 1 **stop** bit

5.2 COMMAND/RESPONSE MESSAGE FORMAT

Command Format for serial RS232 protocol is shown in Figure 5-1, and Response Format in Figure 5-2. Table 5-1 illustrates the ENTRON RS232 Command/Response messages format.

START	FUNCTION	DATA	END
–	one or more chars	0 or more chars	–
	—	—	

Figure 5-1. *RS232 Protocol – Command Format*

START	FUNCTION/DATA	END
–	0 or more chars	–
	—	

Figure 5-2. *RS232 Protocol – Response Format*

5.3 ENTRON RS232 PROTOCOL MESSAGES

Table 5-1. *RS232 Commands and Responses*

COMMAND Sent by Host	RESPONSE Sent by Controls (to Host)	DESCRIPTION
1. **Ctrl+E** [**⌐.5.**] or **05** hex	on SCHEDULE display	Connect to the control.
2. **Ctrl+D** or **04** hex	Disconnect...	Disconnect from the control.
3. **HE** or **he** or **Ctrl+H** or **08** hex	ENTRON ENxxxx/RS232-Help:	Print Help.
4. **TW** or **tw**	WELD Enabled or WELD Disabled	Toggle WELD / NO WELD.
5. **xx**	SC**xx:qq,qq,qq,qq,qq,...,qq**	**R**ead Schedule data for schedule **xx.** **xx** = Schedule number, from 00 to 49. **qq**=Schedule sequence data, always two chars. Schedule Sequence parameters are: SQUEEZE, WELD, %, HOLD, OFF,...
6. **Rxx** or **rxx**	Squeeze: **qq** Weld: **qq** ...	**R**ead Schedule **xx** interactively.
7. **RFx1,x2** or **rfx1,x2**	SC**x1:qq,qq,qq,qq,qq,...,qq** **Scxx:qq,qq,qq,qq,qq,...,qq** SC**x2:qq,qq,qq,qq,qq,...,qq**	**Read F**ile schedule data for all schedules, from **x1** to **x2.** **x1**, **x2**=Schedule numbers, from 00 to 49.
8. **RSC** or **rsc**	SC**xx**: ..... and EF:**...**	**R**ead **c**urrent **S**chedule and EF.
9. **Wxx** or **wxx**	Squeeze:yy **qq↵** Weld:yy **qq↵**... ...	**W**rite single Schedule **xx** interactively: **yy** = Schedule sequence data, previous value. **qq** = Schedule sequence data, new value.
10. **SCxx:qq,qq,qq,...,qq↵**		Write all parameters for **Sc**hedule **xx** at the same time.
11. **EX** or **ex**	EF:**qq,qq,qq,qq,qq,...,qq**	Read **Ex**tended Functions parameters**.** EF parameters are: Id,SE,CC,CA,PO,87,PP,PF,....
12. **RE** or **re**	ID Number: 00 SEAM (Spot=00): **qq** ...	**R**ead **E**xtended Functions interactively.
13. **WE** or **we**	ID Number: 00 SEAM (Spot=00):yy **qq↵**...	**W**rite **E**xtended Functions interactively.

5.3 ENTRON RS232 PROTOCOL MESSAGES (cont.)

COMMAND Sent by Host	RESPONSE Sent by Controls (to Host)	DESCRIPTION
14. **EERxx,yy**	EEPROM:**xx,yy,qq**	**R**ead data, one byte, from EEPROM, all parameters in **hex**. **xx**=EEPROM page, **yy**=EEPROM word address, **qq** = EEPROM data.
15. **EEWxx,yy,qq**	EEPROM:**xx,yy,qq**	**W**rite data **qq** to EEPROM, all parameters in **hex**. **xx** = EEPROM page, except A8 and AA page, **yy** = EEPROM word address, **qq** = EEPROM data.
16. **RST** or **rst**	ReSET Control.	**Re**s**e**t **C**ontrol.
17. **RLS** or **rls**	Last Schedule: **xx**	**R**ead **L**ast dialed **S**chedule number. **xx** = Last dialed Schedule No, from 00 to 49.
18. **WLSxx** or **PSxx**	Last Schedule: **xx**	**W**rite **xx** as **L**ast **S**chedule number. **P**ut **S**chedule number, **xx** from 00 to 49.

NOTES:

- **ID** or **ADDRESS** is ***I.d.*** parameter of control, which might be changed through EXTENDED FUNCTIONS.
- ↵ is symbol for Carriage Return or ENTER, ASCII code **0D** hex or 13 decimal.
- **x** and **z** are always only ONE character, **yy** and **qq** are always TWO characters.
- Command characters are not case sensitive, i.e. either upper or lower case might be used.
- For more information about Schedule and EF Sequence, refer to corresponding Manuals.

5.4 EXAMPLES

COMMAND	RESPONSE	DESCRIPTION
1. Ctrl+E or **05** hex	***r.S.*** on SCHEDULE display	Connect to the control.

For EN1001 Control:

```
ENTRON - EN1001.     RS232 Settings:
4800 Baud, 8 Bits, NO Parity, 1 Stop Bit.

Version:En02.03.00            EPROM:619016-002F
Date:03/18/2002

Type HE or <Ctrl+H> for Help.
```

For EN1200 Control (*Obsolete*):

```
ENTRON - EN1001.     RS232 Settings:
4800 Baud, 8 Bits, NO Parity, 1 Stop Bit.

Version:En02.03.00            EPROM:619016-002F
Date:03/18/2002

Type HE or <Ctrl+H> for Help.
```

5.4 EXAMPLES (cont.)

COMMAND	RESPONSE	DESCRIPTION
2. Ctrl+D or **04** hex	Disconnect...	Disconnect from control.

```
Disconnecting...
```

3. **HE**, **he**, Ctrl+H or **08** hex	ENTRON ENxxxx-RS232- Help:	Print Help.

For EN1200 Control (*Obsolete*):

```
ENTRON EN1200/1201-RS232 Help:
Ctrl+E  Connect to Control
Ctrl+D  Disconnect

HE      Help
TW      Toggle Weld/No Weld
Rxx     Read Schedule xx
Wxx     Write Schedule xx
RE      Read EF
WE      Write EF
RFx1,x2 Read File schedules x1-x2
RSC     Read Current Schedule & EF
RLS     Read Last Schedule number
WLS     Write Last Schedule number

Type HE or <Ctrl+H> for Help.
```

4. **TW** or **tw**	WELD Enabled or WELD Disabled	Toggle WELD / NO WELD.

```
TW
 WELD Disabled.


TW
 WELD Enabled.
```

5. **xx**	SC**xx:qq,qq,qq,qq,qq,...,qq↵**	**R**ead Schedule data for schedule **xx.**

Standard schedule parameters: Stepper disabled, and no IPSC option.

```
10
 SC10:10,08,30,00,00,03,02,04,02,00,00,

20
 SC20:05,00,40,00,00,01,00,01,02,01,15,
```

5.4 EXAMPLES (cont.)

COMMAND	RESPONSE	DESCRIPTION
5. **xx** (cont.)	SC**xx:qq,qq,qq,qq,qq,...,qq**↵	**R**ead Schedule data for schedule **xx.**

Standard schedule parameters: **Stepper** enabled, and **IPSC** option exists with **P.r.** = 31 PSI, **P.t.** = 26 PSI.

```
20
 SC20:10,00,0250,0240,0265,00,00,01,00,01,02,01,15,31,26,0258,

20
 SC20:10,00,0250,0240,0265,00,00,01,00,01,02,01,15,002300190258,
```

Standard schedule parameters: **Constant Current** enabled, **Stepper** enabled, and **IPSC** option exists with **F.r.** = 0023 LB, **F.t.** = 0019 LB.

```
20
 SC20:10,00,0250,0240,0265,00,00,01,00,01,02,01,15,0023,0019,0258,
```

For EN1000 Cascade or EN1001 Cascade Controls:

```
10
 SC10:30,10,88,03,00,01,00,AF,FA,00,00,00,05
```
(Constant Current inactive, Valve extension 9-16 installed)

```
10
 SC10:30,10,0062,0000,0000,03,00,01,00,AF,00,00,00,05
```
(Constant Current inactive, Valve ext. 9-16 not installed)

```
10
 SC10:30,10,0000,0000,0000,03,00,01,00,AF,FA,00,00,00,05
```
(Constant Current active, Valve extension 9-16 installed)

```
10
 SC10:30,10,0062,0000,0000,03,00,01,00,AF,FA,00,00,00,05,33,30
```
(Constant Current, Valve extension 9-16 installed, Pressure control active P.r. = 33, P.t. = 30)

```
10
 SC10:30,10,0062,0000,0000,03,00,01,00,AF,00,00,00,05,33,30,0250
```
(Constant Current, Pressure control active P.r. = 33, P.t. = 30, Stepper active)

```
10
 SC10:30,10,0062,0000,0000,03,00,01,00,AF,00,00,00,05,0250
```
(Constant Current, Stepper active)

5.4 EXAMPLES (cont.)

COMMAND	RESPONSE	DESCRIPTION
6. **Rxx** or **rxx**	Squeeze: **qq** ...	**R**ead Schedule **xx** interactively.

```
R10
Squeeze: 10
Weld   : 08
Percent: 30
Hold   : 00
Off    : 00
Impulse: 03
Cool   : 02
Valve  : 04 Valve 3
Cycle  : 02 Chained mode
Slope  : 00 No slope
Count  : 00 cycles of slope
```

For Controls with **Constant Current** mode:

```
R20
Squeeze: 10
Weld   : 00
Current: 0250
CurLow : 0240
CurHigh: 0265
Hold   : 00
Off    : 00
Impulse: 01
Cool   : 00
Valve  : 01 Valve 1
Cycle  : 02 Chained mode
Slope  : 01 Upslope
Count  : 15 cycles of slope
```

For Controls with enabled **Stepper**:

```
R20
Squeeze: 10
      ....
Count  : 15 cycles of slope
Stepper: 0258
```

For Controls with **IPSC** option:

```
R20
Squeeze: 10
      ....
Count  : 15 cycles of slope
Stepper: 0258
PresRef: 31 PSI
PresTrg: 26 PSI
```

5.4 EXAMPLES (cont.)

COMMAND	RESPONSE	DESCRIPTION
7. **RFx1,x2** or **rfx1,x2**	SC**x1:qq,qq,qq,qq,qq,...,qq↵**	**R**ead **F**ile schedule data for all schedules from schedule **x1** to **x2.**

```
RF20,25
SC20:05,00,40,00,00,01,00,01,02,01,15,
SC21:00,04,40,00,00,06,02,01,02,00,00,
SC22:10,30,37,05,00,01,00,01,02,02,10,
SC23:10,00,00,00,00,01,00,02,00,00,00,
SC24:00,00,10,60,00,01,00,04,02,00,00,
SC25:10,48,60,10,00,01,00,01,00,00,00,
```

8. **RSC** or **rsc** — SC**xx**: ..... and EF:**...** — **R**ead **c**urrent **S**chedule and EF.

```
RSC
SC20:
Squeeze: 10
Weld   : 48
Percent: 60
Hold   : 10
Off    : 00
Impulse: 01
Cool   : 00
Valve  : 01 Valve 1
Cycle  : 00 Non repeat mode
Slope  : 00 No slope
Count  : 00 cycles of slope

ID Number      00 Id. Number for RS485
SEAM (SPOT=0)  00 SPOT mode
SCHED SELECT   00 Disabled, internal SS
AVC            00 Disabled
CA Clear DATA  00 Use ONLY to Clear All DATA
BACK STEP      00 Disabled
PROCESS OUTPUT 00 No PO, V3 used as a valve
BEAT           00 Non beat operation
FREQUENCY      00 Default frequency 1000Hz.
CYCLE MULTIPL. 00 Disabled
FIRING OFFSET  00 Disabled
SQUEEZE DELAY  00 No delay before squeeze
BLOCK DELAY    70 cycles of blocking delay
CURRENT OFFSET 00 Disabled
STEPPER        00 Disabled
```

9. **Wxx** or **wxx** — Squeeze:yy **qq↵** — **W**rite single Schedule **xx** interactively:

```
05
 SC05:00,00,00,00,00,01,00,00,00,00,00,
W05
Squeeze: 00 10
Weld   : 00 04
Percent: 00 50
```

5.4 EXAMPLES (cont.)

COMMAND	RESPONSE	DESCRIPTION
9. **Wxx** or **wxx** (cont.)	Squeeze:yy **qq**↵	**W**rite single Schedule **xx** interactively:

```
Hold   : 00 10
Off    : 00 00
Impulse: 01 01
Cool   : 00
Valve  : 00 02
Cycle  : 00
Slope  : 00
Count  : 00

05
 SC05:10,04,50,10,00,01,00,02,00,00,00,
```

10. **SCxx:qq,qq,qq,...,qq**↵ — Write all parameters for **Sc**hedule **xx** at the same time.

```
SC06:20,06,60,20,00,01,00,04,00,00,00

06
 SC06:20,06,60,20,00,01,00,04,00,00,00,
```

11. **EX** or **ex** — EF:**qq,qq,qq,qq,qq,...,qq** — Read **Ex**tended Functions parameters**.**

```
EX
 EF:00,00,00,00,00,00,00,00,00,99,00,00,00,12,02.00,00,00,065,
```

12. **RE** or **re** — ID Number: 00 — **R**ead **E**xtended Functions interactively.

For EN1200 (*Obsolete*):

```
RE
ID Number       00 Id. Number for RS485
SEAM (SPOT=0)   00 SPOT mode
SCHED SELECT    00 Disabled, internal SS
AVC             00 Disabled
CA Clear DATA   00 Use ONLY to Clear All DATA
BACK STEP       00 Disabled
PROCESS OUTPUT  00 No PO, V3 used as a valve
BEAT            00 Non beat operation
FREQUENCY       00 Default frequency 1000Hz.
CYCLE MULTIPL.  00 Disabled
FIRING OFFSET   00 Disabled
SQUEEZE DELAY   00 No delay before squeeze
BLOCK DELAY     70 cycles of blocking delay
CURRENT OFFSET  00 Disabled
STEPPER         00 Disabled
```

5.4 EXAMPLES (cont.)

COMMAND	RESPONSE	DESCRIPTION
12. **RE** or **re** (cont.)	ID Number: 00	**R**ead **E**xtended Functions interactively.

For Controls with **Constant Current** mode:

```
RE
ID Number      00 Id. Number for RS485
SEAM (SPOT=0)  00 SPOT mode
SCHED SELECT   00 Disabled, internal SS
AVC            00 Disabled
CA Clear DATA  00 Use ONLY to Clear All DATA
BACK STEP      00 Disabled
PROCESS OUTPUT 00 No PO, V3 used as a valve
BEAT           00 Non beat operation
87 deg DELAY   00 Disabled
PROGRAMABLE PF 99 Value of programmed power factor
POWER FACTOR   00 Value shown ONLY after weld sequence
SQUEEZE DELAY  00 No delay before squeeze
BLOCK DELAY    00 No blocking delay
TURNS RATIO   065 Turns ratio of transformer
Cr CURRENT REG 12 Primary Coil sensing
rA MAX SCAL 02.00 Max. programmable current
CURRENT OFFSET 00 Disabled
STEPPER        00 Disabled
```

For Controls with **IPSC** option:

```
RE
ID Number      00 Id. Number for RS485
     ....
Cr CURRENT REG 20 Secondary Hall-Effect
rA MAX SCAL 05.00 Max. programmable current
CURRENT OFFSET 00 Disabled
STEPPER        01 Enabled
PRESSURE CNTRL 01 PSI mode
Background Pr. 10 PSI
Sensor Input  027 Pressure Sensor Input (IPSC,IPS).
```

NOTICE

EXTENDED FUNCTIONS parameter ***S.I.*** is scaled to range 00-255 for all three IPSC modes. On display, this parameter is always scaled according to chosen IPSC mode.

13. **WE** or **we**	ID Number: 00	**W**rite **E**xtended Functions interactively.

For EN1200 (*Obsolete*):

```
WE
ID Number      00
SEAM (SPOT=0)  00
SCHED SELECT   00
AVC            00
CA Clear DATA  00
BACK STEP      00
PROCESS OUTPUT 00 22
BEAT           00
FREQUENCY      00
CYCLE MULTIPL. 00 17
FIRING OFFSET  00 90
SQUEEZE DELAY  00
BLOCK DELAY    70 00
CURRENT OFFSET 00
STEPPER        00

EX
 EF:00,00,00,00,00,00,22,00,00,17,90,00,00,00,00,
```

5.4 EXAMPLES (cont.)

COMMAND	RESPONSE	DESCRIPTION
14. **EERxx,yy**	EEPROM:**xx,yy,qq**	**R**ead data, one byte, from EEPROM; all parameters in **hex**.

```
EERA0,00
 EEPROM:A0,00,0A
```

15. **EEWxx,yy,qq**	EEPROM:**xx,yy,qq**	**W**rite data **qq** to EEPROM; all parameters in **hex**.

```
EEWA0,00,05
 EEPROM:A0,00,05
```

NOTICE

Using this command, any of the SCHEDULE and EXTENDED FUNCTIONS parameters can be overwritten. For proper usage, refer to Appendix C – EEPROM Memory Map.

16. **RST** or **rst**	ReSET Control.	**Res**et **C**ontrol. Control will be reset by software.

```
RST
 ReSET Control.
```

17. **RLS** or **rls**	Last Schedule: **xx**	**R**ead **L**ast dialed **S**chedule number.

```
RLS
 Last Schedule: 15
```

18. **WLSxx** or **PSxx**	Last Schedule: **xx**	**W**rite **xx** as **L**ast **S**chedule number. **P**ut **S**chedule number.

```
WLS20
 Last Schedule: 20

PS15
 Last Schedule: 15
```

6.0 ERROR CODES and PROCESS OUTPUTS

6.1 ERROR CODES

Besides standard error codes described in Application Note 700158, some of the following errors may occur only on the Remote Terminal RT4.

E.r. = 40 No active control with the requested **I.d.** is on the bus. This error will occur if the control with the desired **I.d.** is turned off or does not respond. After clearing this error, [—] will be displayed on the RT4 DATA display, which means no data is being received from the control.

E.r. = 41 Message is not received. This error will occur if the control does not respond on request from RT4.

E.r. = 42 Error in **FUNCTION** byte. This error will occur if the Command or Respond FUNCTION byte is not valid; i.e., if the FUNCTION byte does not match the values from the ENBUS protocol.

E.r. = 43 **CHECKSUM** error. This error will occur if the data is corrupt during communication between RT4 and corresponding controls.

Each of the errors may be cleared by pressing any push button on RT4's Front Panel.

6.2 PROCESS OUTPUTS

Two PROCESS OUTPUTS are directly correlated to RS232 protocol and Error and Status Monitoring.

P.O. = 22 Send HI/LO and CURRENT after out of limit window weld or last weld in sequence with RS232 Option (*EN1001 only*).

Will send data for out of limit window weld in sequence or last weld only in completed sequence to master on ENBUS or to serial printer. Display will flash **H. L.** or **L.o.** and measured CURRENT if out of limit window.

P.O. = 23 Send HI/LO and CURRENT if out of limit window with RS232 Option (*EN1001 only*).

Will send data after weld if the CURRENT is out of limit window to master on ENBUS or to serial printer. Display will flash **H. L.** or **L.o.** and measured CURRENT.

If **I.d.** is between **01** and **64**, data will be prepared to be sent to a master PC with ENLINK software or to a master device. With these two PROCESS OUTPUTS, the weld sequence will be stopped and the control will wait for a master to scan or read all necessary prepared data. At the same time, the control will show CURRENT on the display, and, if error exists, it will flash **L.o.** or **H. L.** , until any push button is pressed on the Front Panel. In other words, in order to clear the display of errors, any push button on the Front Panel must be pressed and the master must read the prepared message from the control. Only after an error is cleared will the control allow re-initiation of a new weld sequence.

If the control IDENTIFICATION NUMBER **I.d.** = **00** or **93** to **95** as described in Section 2.1, these PROCESS OUTPUTS can be used to send data to a serial printer at the end of completed weld sequence.

6.3 USING SERIAL PRINTER

Serial printer settings should match control serial UART parameters as described in Section 5.1.

For **P.0.** = **22**, prints a message after out of limit window weld or last weld in sequence. One of the messages might be:

```
CURRENT WITHIN LIMITS  02.500
CURRENT TOO HIGH       03.380
CURRENT TOO LOW        02.120
```

For **P.0.** = **23**, prints a message only when measured current at the end of weld is out of limit window. One of the messages might be:

```
CURRENT TOO HIGH       04.850
CURRENT TOO LOW        02.420
```

7.0 FIELD INSTALLATION OF RETROFIT RS232 OPTION

1. Remove **ALL** power to control and open door.
2. In EN1000 and EN1001 Controls, remove Control PCB from Dial Plate and replace with new Control Board included with kit. In EN1000 Cascade Controls, if revision letter of PROM on PCB 410321 is earlier than new PROM included with kit, remove and replace old PROM with new PROM. **NOTE:** Older Cascade Controls may have PROM with P/N 619011-001 or -002 – these PROMs **must** be replaced with PROM P/N 619044-002 included with kit.
3. In "T/D" Cabinets, mount bracket (P/N 525035) on the two 6-32 x 3/8 studs on the left side of cabinet using the #6 lockwashers and 6-32 hex nuts (P/Ns 557017 and 557018) supplied. In NEMA 12 Enclosures, mount the bracket using 6-32 x 3/8 Phillips Pan Head Screws and #6 lockwashers in the holes provided on the rear panel.
4. In "T/D" and NEMA 12 Enclosures, mount the RS232 PCB Assembly (410326) to bracket using the 6-32 x 1/4 Phillips Pan Head Screws (P/N 557003) supplied. In "B", "S", and "E" Cabinets, mount the PCB to the existing standoffs as indicated on the diagrams below.
5. Mount the 9 Pin D-subminiature Connector end of the J232-J232 Harness to the cabinet (in "B", "S", "E", and "T/D" Cabinets purchased after June 1996 and in "L" Cabinets purchased after October 1996, a mounting has been provided in the cabinet) using the Jack Screws (P/N 331124) supplied. Plug the other end of the harness into the J232 jack on the RS232 PCB Assembly. The red stripe on the harness **MUST** be oriented as shown on the Wiring Diagram.
6. Plug J4-J4 (or J4B-J4B) Harness Assembly into the J4 jack on the RS232 PCB Assembly and the J4 (or J4B) jack on the Indicator PCB Assembly. The red stripe on the harness **MUST** be oriented as shown on the Wiring Diagram enclosed.
7. Close cabinet door, re-apply power.

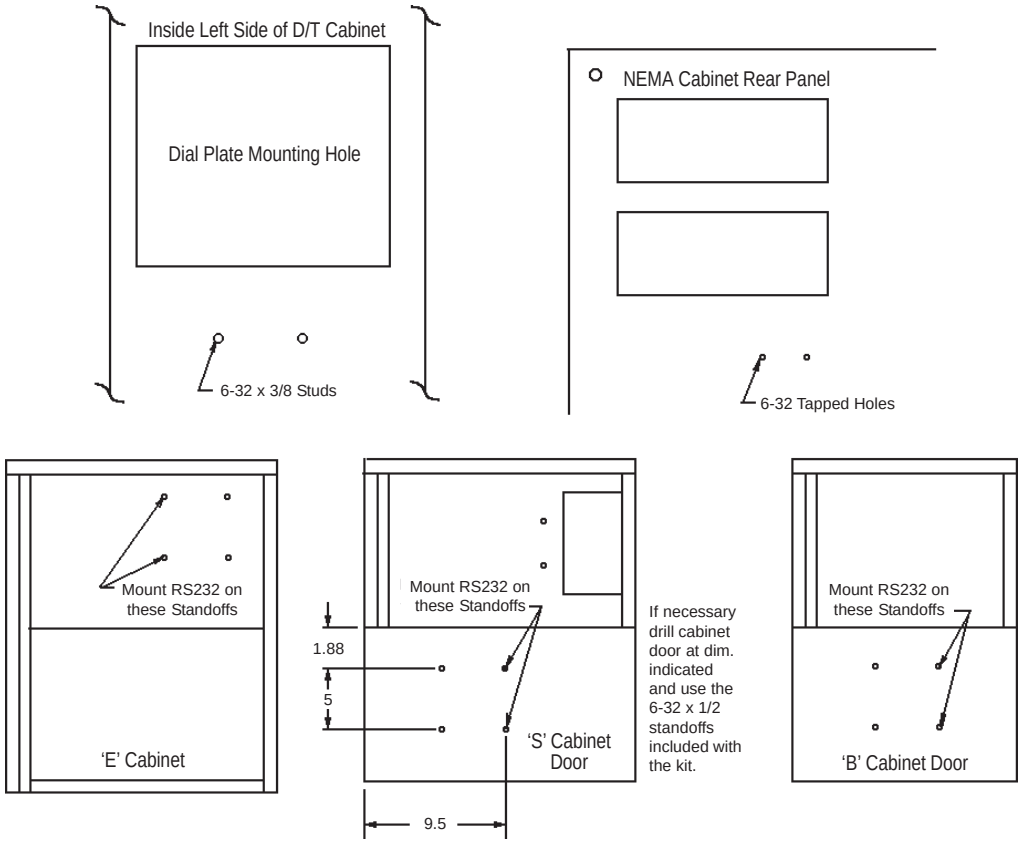


Figure 7-1. *RS232 mounting diagrams*

7.1 RS232 RETROFIT OPTION ASSEMBLIES

RS232 RETROFIT OPTION ASSEMBLIES – WHERE USED CHART

(Determine the correct combination of Control Style and Cabinet Style)

OPTION KITS for	B,S,E	T/D	NEMA 12 ENCL.
EN1000	600619-001	600619-002	600619-003
EN1001	600620-001	600620-002	600620-003
EN1000 Cascade	N/A	600621-002	600621-003

RS232 RETROFIT OPTION KIT BILL OF MATERIALS

600619-001	600619-002	600619-003	600620-001	600620-002	600620-003	600621-002	600621-003	PART NO.	DESCRIPTION
1	1	1	1	1	1	1	1	410326	PCB Assembly, RS232 Option
1			1					322415	Harness, J4-J4, 22-1/2"
	1			1		1		322416	Harness, J4-J4, 26-1/2"
		1			1		1	322417	Harness, J4-J4, 34"
1		1	1		1		1	322369-002	Harness, J232-J232, 23"
	1			1		1		322369-004	Harness, J232-J232, 8-1/4"
1	1	1						600541	EN1000 Control Board
			1	1	1			600572-002	EN1001 Control Board
						1	1	619044-002	EN1000 Cascade PROM version **ORIG** (or later)
	1	1		1	1	1	1	525035	Bracket, PCB Mounting
2	2	2	2	2	2	2	2	331124	Connector, Jack Screw, for .090 Panel
4	2	2	4	2	2	2	2	557003	6-32 x 1/4 PHSMS, Phil, Brite
		2			2		2	557006	6-32 x 3/8 PHSMS, Phil, Brite
	2	2		2	2	2	2	557017	#6 Med. Split Lockwasher, Brite
	2			2		2		557018	6-32 x 1/4 AF Hex Nut, Brite
2			2					555031	Spacer, Hex Threaded Brass, 6-32 x 1/2
2	2	2	2	2	2	2	2	342012	Cable Clamp, 5/32 Flat, Adhesive Back
1	1	1	1	1	1	1	1	342020	Cable Clamp, 2 x .17 Thk, Adhesive Back
2	2	2	2	2	2			460129	Label, "J4"
						2	2	460129	Label, "J4B"
1	1	1	1	1	1	1	1	460156	Label, RS232
1	1	1	1	1	1	1	1	600689	ENLINK 1000/1001 for Windows
1	1	1	1	1	1	1	1	700140	Manual, 232 Options

The Harness Assemblies included with this kit may not match the Harness Assemblies listed on the included Wiring Diagram. Either Assembly will work.

7.1.1 FIELD INSTALLATION OF RETROFIT RS232 OPTION in 24"x16"x8" CABINETS (Use kits for "T/D" Cabinets – see Section 7.1)

1. Remove **ALL** power to control and open door.

2. In EN1000 and EN1001 Controls, remove the Control PCB from the Dial Plate and replace with the new Control Board included with this kit.

3. Modify the cabinet per the diagram below, mount bracket (P/N 525035) on two 6-32 x 3/8 screws installed from the outside of the cabinet using the #6 lockwashers and 6-32 hex nuts (P/Ns 557017 and 557018) supplied.

4. Mount the RS232 PCB Assembly (410326) to bracket using the 6-32 x 1/4 Phillips Pan Head Screws (P/N 557003) supplied.

5. Mount the 9 Pin D-subminiature Connector end of the J232-J232 Harness to the cabinet using the Jack Screws (P/N 331124) supplied. Plug the other end of the harness into the J232 jack on the RS232 PCB Assembly. The red stripe on the harness **MUST** be oriented as shown on the Wiring Diagram.

6. Plug J4-J4 Harness Assembly into the J4 jack on the RS232 PCB Assembly and the J4 jack on the Control PCB Assembly. The red stripe on the harness **MUST** be oriented as shown on the Wiring Diagram enclosed.

7. Close cabinet door, re-apply power.

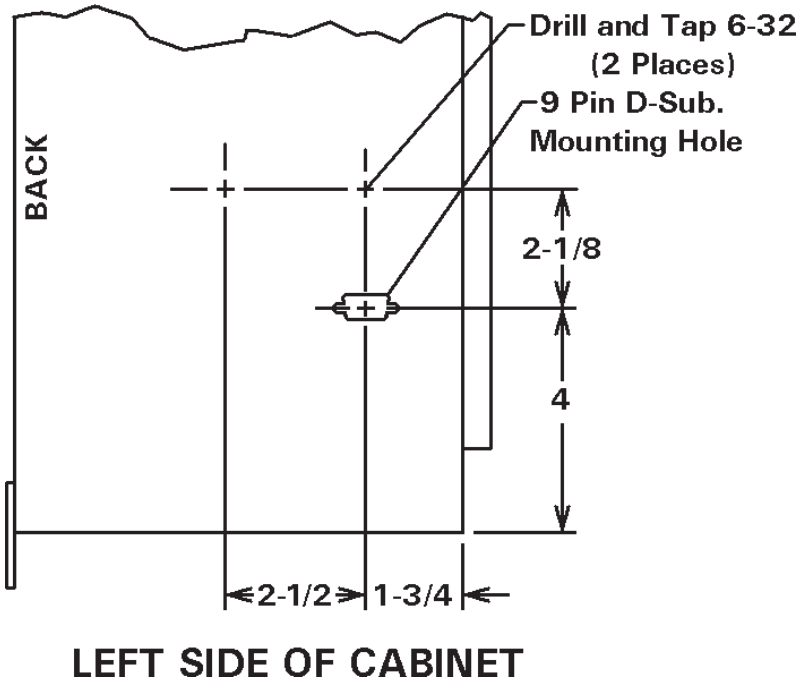


Figure 7-2. *RS232 mounting in 24″ x 16″ x 8″ cabinets*

8.0 ENTRON LIMITED WARRANTY AND FACTORY SERVICE

ENTRON Controls, LLC., warrants that all ENTRON control panels, **EXCEPT** Mid-frequency Inverter controls, silicon controlled rectifiers (SCRs), insulated gate bipolar transistors (IGBTs), SCR and IGBT assemblies, circuit breakers, and electro-mechanical contactors, are free of manufacturing defects for a period of **TWO YEARS** from the date of original purchase and, in the event of a manufacturing defect, ENTRON will repair or replace, at its discretion, the defective part without any cost for parts or labor.

All silicon controlled rectifiers, SCR and IGBT assemblies, circuit breakers, and electro-mechanical contactors in ENTRON control panels are covered by **a limited warranty from the original manufacturer**. If these parts fail because of a manufacturing defect, they will not be repaired or replaced by ENTRON, but will be returned by ENTRON to the original manufacturer in accordance with said manufacturer's warranty.

ENTRON Controls, LLC., warrants that all Mid-frequency Inverter controls are free of manufacturing defects for a period of **ONE YEAR** from the date of original purchase and, in the event of a manufacturing defect, ENTRON will repair or replace, at its discretion, the defective part without any cost for parts or labor.

To obtain repairs or replacement parts under this warranty, the defective part must be returned, prepaid, to ENTRON Controls, LLC., 1402 S. Batesville Road, Greer, SC 29650. Please send your repair to the attention of "Service" with a description of the problem you are experiencing, contact person, and phone number.

EXCLUSIONS: This warranty does not cover damage by accident or misuse, unauthorized repair or modification to any control assembly by the customer.

IMPORTANT NOTE: The warranty period is considered from the date of shipment and is tracked by a serial number code.

USE OF OUT OF WARRANTY REPAIR SERVICE:
To obtain service for any printed circuit board assembly or welding control after the warranty period, send the assembly or control, prepaid, to ENTRON Controls, LLC., and ENTRON will repair the printed circuit board assembly or control and return it to you without further warranty. Additional service charges may be invoiced at time of shipment.

Your ENTRON Controls, LLC., Original Equipment Manufacturers (OEMs), Dealers and Distributors are your first response contact to secure technical assistance on control or welding problems. Should they be unable to assist you, please contact your ENTRON sales representative or the factory directly. Contact the factory at 864-416-0190.

APPENDIX A FIELD MODIFICATION OF "L" CABINET TO MOUNT RS232

1. Remove **ALL** power to control and open door.

2. In "L" Cabinets with Single Function Type Controls (i.e., EN1000, EN1001), drill and tap two 6-32 mounting holes on the left side of the cabinet, below the mounting position for the side mounted Dial Plate per Diagram A below. In "L" Cabinets with Cascade Controls, drill and tap two 6-32 mounting holes on the Rear Panel below the two Terminal Strip boards per Diagram B below. Be sure to remove ALL metal chips from the inside of the cabinet.

3. Continue installation of the RS232 option beginning at Step 2 of the RS232 Field Installation instructions (see Section 7.0).

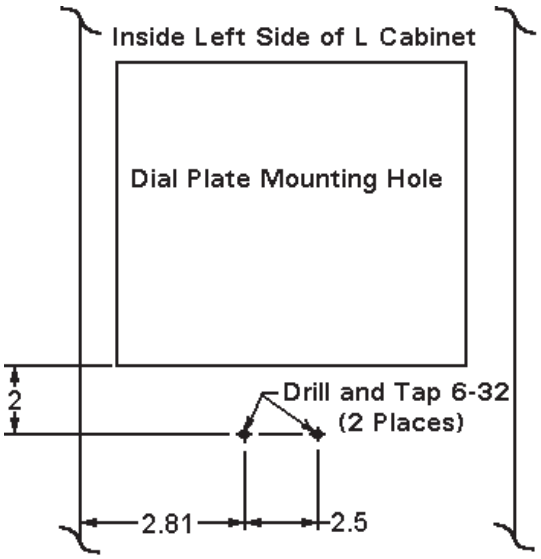


DIAGRAM A

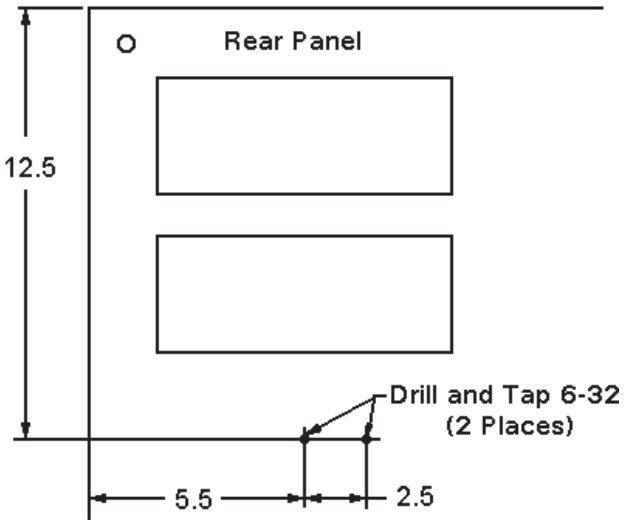


DIAGRAM B

APPENDIX B RAM MEMORY MAP

In EN1000 Series Weld Controls, EEPROM Memory is used for non-volatile storage all SCHEDULE and EXTENDED FUNCTIONS data. Before executing any weld sequence, all SCHEDULE and EXTENDED FUNCTIONS data are loaded into RAM memory. The RAM memory map may be useful when an RS232 or RS485 option is used to read SCHEDULE data, EXTENDED FUNCTIONS data or other useful flags or data from PC or PLC.

To read data from RAM memory, data location (i.e., address of the data) must be known.

FLAGS

The bytes from **0x20** to **0x25** are bit addressable and contain different status flags.

```
=================================================================
at 0x20 bdata byte flags0;          //DATA @ 20h - Status flags
sbit flag_wnw = flags0 ^ 0;         //WELD/NoWELD push button status
sbit flag_program = flags0 ^ 1;     //PROGRAM/OPERATE push button status
sbit flag_PIN_lock = flags0 ^ 2;    //PIN lock enable/disable flag
sbit flag_sw_lock = flags0 ^ 3;     //PLK switch flag
sbit flag_timer = flags0 ^ 4;
sbit flag_valve = flags0 ^ 5;
sbit flag_sync = flags0 ^ 6;
sbit flag_slope = flags0 ^ 7;
_____________________________

at 0x21 bdata byte flags1;          //DATA @ 21h - Status flags
sbit flag_estop = flags1 ^ 0;       //Emergency STOP status
sbit flag_i2c_busy = flags1 ^ 1;    //i2c bus busy flag
sbit first_i2c_err = flags1 ^ 2;    //i2c error flag
sbit first_weld = flags1 ^ 3;       //first weld flag
sbit flag_PO_07 = flags1 ^ 4;
sbit pf_zero = flags1 ^ 5;
sbit wnw_pressed = flags1 ^ 6;
sbit flag_av = flags1 ^ 7;
_____________________________
```

APPENDIX B (cont.)
FLAGS (cont.)

```
at 0x22 bdata byte flags2;          //DATA @ 22h - Status flags
sbit flag_IPSEr = flags2 ^ 0;       //the same functionality as PSW1: flag_IPSEr=1->PCerror
sbit flag_avc=flags2^1;             //
sbit flag_adc = flags2 ^ 2;         //
sbit flag_short = flags2 ^ 3;       //
sbit flag_RS232 = flags2 ^ 4;       //RS232 active flag
sbit flag_tx_active=flags2 ^ 5;     //UART TxD active
sbit flag_msg_ack = flags2 ^ 6;     //master UART has red message if this flag is set
sbit flag_msg_recv=flags2 ^ 7;      //Message (or byte) received on RxD
```

```
at 0x23 bdata byte flags3;          //DATA @ 23h -> STATUS byte: type of the Control
sbit flag_IPC = flags3 ^ 0;         //IPC=Pressure Control option
sbit flag_IPS = flags3 ^ 1;         //IPS=Pressure Sense option
sbit          flags3 ^ 2;           //reserved
sbit          flags3 ^ 3;           //reserved
sbit flag_EN1000C = flags3^4;       //EN1000Cascade
sbit flag_EN1200=flags3 ^ 5;        //EN1200 (Obsolete)
sbit flag_EN1001=flags3 ^ 6;        //if set->EN1001, if cleared->EN1000
sbit flag_ENRT = flags3 ^ 7;
```

```
at 0x24 bdata byte sws;             //DATA @ 24h - Status for all pushbuttons
sbit sw_program = sws ^ 0;          //PROGRAM/OPERATE button pressed if this flag is zero
sbit sw_enter = sws ^ 1;            //ENTER push button pressed if this flag is zero
sbit sw_weld = sws ^ 2;             //WELD/No WELD push button pressed if this flag is zero
sbit sw_select = sws ^ 3;           //SELECT push button pressed if this flag is zero
sbit sw_data1 = sws ^ 4;            //Right DATA push button pressed if this flag is zero
sbit sw_data10 = sws ^ 5;           //Left DATA push button pressed if this flag is zero
sbit sw_schedule1 = sws ^ 6;        //Right SCHEDULE push button pressed if this flag is zero
sbit sw_schedule10 = sws ^ 7;       //Left SCHEDULE push button pressed if this flag is zero
```

```
at 0x25 data byte flags4;           //DATA @ 25h
sbit flag_overcur = flags4 ^ 0;     //reserved
sbit flag_S49 = flags4 ^ 1;         //S49 option status: =1 =>S49 board exists
sbit flag_welddone = flags4^2;      //=1 if Weld mode, WNW swich closed and WELD time>00
sbit flag_PLK_en = flags4 ^ 3;      //enable/disable PLK switch: =1 PLK enabled
sbit flag_sekvencer = flags4^4;     //reserved
sbit flag_reset_range = flags4 ^ 5; //temporary bit
```

==========

APPENDIX B (cont.)
DATA from 0x26 to 0x3F

The bytes from **0x26** to **0x3F** are different status bytes.

```
at 0x26 data byte ;                     //reserved
at 0x27 data byte first_sched;          //temporary byte
at 0x28 data byte z_low;                //temporary counter
at 0x29 data byte z_high;               //temporary counter
at 0x2a data byte SEQptr;               //Schedule Pointer
at 0x2b data byte EFptr;                //Extended Function pointer
at 0x2c data byte led_data;             //Data (two digit DATA display)
at 0x2d data byte led_sched;            //Schedule number (SCHEDULE Display)
at 0x2e data byte *i2c_xptr;            //pointer to IIC MasterTransmitData
at 0x2f data byte *i2c_rptr;            //pointer to IIC MasterReceiveData
at 0x30 data struct ByteLed_a6 Led;     //Structure for DATA and SCHEDULE displays
    //at 0x30 data byte a6_write;       //Led.inst
    //at 0x31 data byte a6_write1;      //Led.ctrl
    //at 0x32 data byte a6_write2;      //Led.DATA10
    //at 0x33 data byte a6_write3;      //Led.DATA1
    //at 0x34 data byte a6_write4;      //Led.SC10
    //at 0x35 data byte a6_write5;      //Led.SC1
at 0x36 data struct ByteLed_a19 Leds;   //Structure for two bytes LED display
    //at 0x36 data byte a19_write;      //Leds.inst
    //at 0x37 data byte a19_write1;     //Leds.ctrl
    //at 0x38 data byte a19_write2;     //Leds.LED1
    //at 0x39 data byte a19_write3;     //Leds.DATA10L
    //at 0x3a data byte a19_write4;     //Leds.LED2
    //at 0x3b data byte a19_write5;     //Leds.DATA1R
at 0x3c data byte Timer;                //0.5 second Timer
at 0x3d data byte a7_read;              //temporary byte for EEPROM reading
at 0x3e data byte a7_write;             //temporary byte for EEPROM writing
at 0x3f data byte a7_write1;            //temporary byte for EEPROM writing
=================================================================
```

APPENDIX B (cont.)
SCHEDULE DATA

The bytes from **0x40** to **0x4F** are SCHEDULE data bytes. Each schedule has 16 bytes (in CONSTANT CURRENT mode, there are three bytes more), with sequence pointer from 00 to 15 (0x0F) as follows:

```
at 0x40 data byte cur_squeeze;        //00. SQ - SQUEEZE
at 0x41 data byte cur_weld;           //01. We - WELD
at 0x42 data byte cur_percent;        //02. CU - PERCENT CURRENT
at 0x43 data byte cur_hold;           //03. HO - HOLD
at 0x44 data byte cur_off;            //04. OF - OFF
at 0x45 data byte cur_impulses;       //05. IM - IMPULSES
at 0x46 data byte cur_cool;           //06. CL - COOL
at 0x47 data byte cur_valve;          //07. VM - VALVE MODE
at 0x48 data byte cur_cycle;          //08. CM - CYCLE MODE
at 0x49 data byte cur_slope;          //09. SM - SLOPE MODE
at 0x4a data byte cur_slcount;        //0A. SC - SLOPE COUNT
at 0x4b data byte pressure_ref;       //0B. Pr - Pressure reference
at 0x4c data byte pressure_trg;       //0C. Pt - Pressure trigger
at 0x4d data union IntData cur_stc;   //0D. StCnt - Stepper counter (one integer)
    //at 0x4d data byte cur_stc_h;    //0D cur_stc.b.hi \  cur_stc.w
    //at 0x4e data byte cur_stc_l;    //0E cur_stc.b.lo /
at 0x4f data byte cur_per_base;       //0F. Cb - Base value for Current Offset
=====================================================================
```

EXTENDED FUNCTIONS DATA

The bytes from **0x50** to **0x64** and **0xea** are EXTENDED FUNCTIONS data as follows:

```
at 0x50 data byte cur_id;             //00. I.d. - ID number of the control
at 0x51 data byte cur_se;             //01. S.E. - Seam
at 0x52 data byte cur_ss;             //02. S.S. - Schedule Select
at 0x53 data byte cur_cc;             //03. C.C. - Automatic Voltage Compensation
at 0x54 data byte cur_ca;             //04. C.A. - Clear All / Setup Constant Current
at 0x55 data byte cur_bs;             //05. b.S. - Back Step
at 0x56 data byte cur_po;             //06. P.O. - Process Output
at 0x57 data byte cur_be;             //07. b.E. - Beat/Non Beat
at 0x58 data byte cur_87;             //08. 8.7. - 87 Degree Delay
at 0x59 data byte cur_pp;             //09. P.P. - Programmable Power Factor
at 0x5a data byte cur_pf;             //0A. P.F. - Power Factor Measuring
at 0x5b data byte cur_sd;             //0B. S.d. - Squeeze Delay
at 0x5c data byte cur_bl;             //0C. b.L. - Blocking Delay
at 0x5d data byte cur_cr;             //0D. C.r. - Current Regulation
at 0x5e data union IntData cur_ra;    //0E. r.A. - Range
    //at 0x5e data byte cur_ra_h;     //cur_ra.b.hi \ cur_ra.w;     0E. r.A. - First Byte
    //at 0x5f data byte cur_ra_l;     //cur_ra.b.lo /               0F. r.A. - Second Byte
at 0x60 data byte cur_co;             //10. C.O. - Current Offset
at 0x61 data byte cur_st;             //11. S.t. - Stepper
at 0x62 data byte cur_pc;             //12. P.C. - Pressure Control
at 0x63 data byte cur_bd;             //13. b.d. - Background(Return)Pressure
at 0x64 data byte cur_si;             //14. S.I. - Sensor Input(reading)
at 0xea idata byte turns_ratio;       //19. t.r. - Turns Ratio of transformer
=====================================================================
```

APPENDIX B (cont.)
DATA from 0x65 to 0xFF

```
at 0x65 data byte ;                 //reserved
at 0x66 data byte temp_percent;     //temporary byte
at 0x67 data union IntData cur_cu;  //cc mode: current in [%] or [kA*100]
    //at 0x67 data byte cur_cu.b.hi; //cur_cu high byte
    //at 0x68 data byte cur_cu.b.lo; //cur_cu low byte
at 0x69 data byte last_percent;     //temporary byte
at 0x6a data byte p1_shadow;        //temporary byte
at 0x6b data byte led_datab;        //temporary byte
at 0x6c data byte led_datab_next;   //temporary byte
at 0x6d data byte rms2dc_ref;       //desired current, scaled from 0x00 to 0xff
at 0x6e data byte rms2dc;           //measured current from A/D converter
at 0x6f data byte rms2dc_old;       //temporary byte
```

```
at 0x70 data byte t_low;            //temporary byte
at 0x71 data byte t_high;           //temporary byte
at 0x72 data byte temp;             //temporary byte
at 0x73 data byte temp1;            //temporary byte
at 0x74 data union IntData TmpRMS;  //Temporary Integer
    //at 0x74 data byte TmpRMS_h;   //TmpRMS.b.hi \  TmpRMS.w
    //at 0x75 data byte TmpRMS_l;   //TmpRMS.b.lo /
at 0x76 data byte load_16byte;      //temporary byte
at 0x77 data byte load_16byte1;     //temporary byte
at 0x78 data byte acc32_3;          //temporary byte
at 0x79 data byte acc32_2;          //temporary byte
at 0x7a data byte acc32_1;          //temporary byte
at 0x7b data byte acc32_0;          //temporary byte
at 0x7c data byte tmp_3;            //temporary byte
at 0x7d data byte tmp_2;            //temporary byte
at 0x7e data byte tmp_1;            //temporary byte
at 0x7f data byte tmp_0;            //temporary byte
```

==========

```
at 0x80 idata byte stack;           //IDATA from 80h to BFh - Stack
- - -
at 0xac idata byte integral_3;      //   \
at 0xad idata byte integral_2;      //    | Temporary bytes for SPOT subroutines
at 0xae idata byte integral_1;      //    |
at 0xaf idata byte integral_0;      //   /
- - -
at 0xb0 idata byte OutBuf[OutBufLen]; //output buffer from B0-C4 = 21 bytes
```

APPENDIX B (cont.)
DATA from 0x65 to 0xFF (cont.)

```
at 0xc5 idata byte ;                    //reserved
at 0xc6 idata byte chars4Tx;            //number of characters for Transmit
at 0xc7 idata byte ScanCode;            //Control status: =0->Control ready; =0xF1 error, ...
at 0xc8 idata byte ErrorNo;             //Error Number: =0->no errors; =01...99->error no.
#define  InpBufLen  23                  //UART input RX buffer length
at 0xc9 idata byte InpBuf[InpBufLen];   //input buffer from C9-DF = 23 bytes
```

```
at 0xe0 idata byte setup_cu;            //temporary byte
at 0xe1 idata byte setup_ctr;           //temporary byte
at 0xe2 idata byte sample_ctr;          //auxiliary counter
at 0xe3 idata byte samp_ptr;            //to store number of samples acquired
at 0xe4 idata byte fa_adjust;           //temporary byte
at 0xe5 idata byte short_count;         //temporary counter
at 0xe6 idata byte last_duty;           //temporary byte
at 0xe7 idata byte adc_18V;             //A/D value of 18 volts power supply
at 0xe8 idata byte adc_18V_cnt;         //temporary byte
at 0xe9 idata byte trend_up;            //temporary byte
at 0xea idata byte turns_ratio;         //Turns Ratio of transformer

at 0xeb idata byte max_dc;              //full range voltage for A/D converter
at 0xec idata byte gain_1;              //gain setting of amplifier
at 0xed idata byte gain_0;              //gain setting of amplifier
at 0xee idata byte gain_ptr;            //temporary byte
at 0xef idata byte rms2dc_rA;           //temporary byte
```

```
at 0xf0 idata byte rms2dc_refLO;        //temporary byte for current monitor
at 0xf1 idata byte rms2dc_refHI;        //temporary byte for current monitor
at 0xf2 idata byte current_mask;        //temporary byte for current control
at 0xf3 idata byte last_ss;             //temporary byte for schedule selection
at 0xf4 idata byte avc_count;           //temporary byte for AVC
at 0xf5 idata byte avc_value;           //temporary byte for AVC
at 0xf6 idata byte avc_value_nom;       //temporary byte for AVC
at 0xf7 idata byte avc_slope;           //temporary byte for AVC
at 0xf8 idata byte OutPointer;          //pointer to RS485 Transmit Data
at 0xf9 idata byte InpPointer;          //pointer to RS485 Receive Data
at 0xfa idata union IntData TmpInt;     //Temporary Integer
    //at 0xfa idata byte TmpInt_h;      //TmpInt.b.hi \  TmpInt.w
    //at 0xfb idata byte TmpInt_l;      //TmpInt.b.lo /
at 0xfc idata union IntData led_data4;  //temporary integer
    //at 0xfc idata led_data4+1;        //led_data4.b.hi  \ led_data4.w
    //at 0xfd idata led_data4+1;        //led_data4.b.lo  /
at 0xfe idata byte a8_read;             //temporary byte
at 0xff idata byte a8_write;            //temporary byte
```

==========

APPENDIX C EEPROM MEMORY MAP

In EN1000 Series Weld Controls, EEPROM Memory is used for non-volatile storage all SCHEDULE and EXTENDED FUNCTIONS data. Memory Map may be useful when an RS232 or RS485 option is used to program SCHEDULE and EXTENDED FUNCTIONS data from PC or PLC.

To read data from EEPROM or write data to EEPROM memory, data location (i.e., EEPROM page and EEPROM word address of the data) must be known.

SCHEDULE DATA

The pages **A0** to **A6** contain SCHEDULE data. EEPROM **page** for any SCHEDULE data can be calculated by using following equation:

$$EEPROM\ page\ [hex] = A0 + 2*\text{int}(\frac{SN}{16})$$

EEPROM word **address** for any SCHEDULE data can be calculated by using following equation:

$$EEPROM\ address\ [hex] = SN * 16 + SP$$

where: *SN* is Schedule Number
SP is Sequence Pointer

Each schedule is contained in a line of 16 bytes, with *SP* from 0 to 15 (0x0F), as follows:

```
00. SQ    - Squeeze
01. We    - Weld
02. CU    - Percent Current
03. HO    - HOLD
04. OF    - OFF
05. IM    - IMPULSES
06. CL    - COOL
07. VM    - Valve Mode
08. CM    - Cycle Mode
09. SM    - Slope Mode
0A. SC    - Slope Count
0B. Pr    - Pressure (Squeeze Pressure)
0C. Pt    - Pressure trigger (Level of Pressure to begin welding)
0D. StCnt - Stepper counter (two bytes, one integer)
0F. Cb    - Base value for Current Offset use
```

APPENDIX C (cont.)
SCHEDULE DATA (cont.)

==========
PAGE A0 hex - SCHEDULE data for Schedule #00 to Schedule #15
==========

SN	*Add*	SQ	WE	CU	HO	OF	IM	CL	VM	CM	SM	SC	Pr	Pt	StCnt	Cb
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
00.	**00**	0A	04	32	0A	00	01	00	01	00	00	00	4E	41	00	00	00
01.	**10**	0A	03	50	0A	00	01	00	01	03	01	0A	4E	41	00	00	00
02.	**20**	0A	03	50	0A	00	01	00	02	00	02	0A	4E	41	00	00	00
03.	**30**	0A	03	14	00	00	01	00	03	02	00	00	4E	41	00	00	00
04.	**40**	1E	14	1E	00	00	01	00	03	01	00	00	4E	41	00	00	00
05.	**50**	0A	03	3C	0A	00	01	00	01	00	00	00	55	4B	00	00	00
06.	**60**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
07.	**70**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
08.	**80**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
09.	**90**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
10.	**A0**	0A	01	46	00	00	03	02	04	02	00	00	4E	41	01	10	00
11.	**B0**	00	3C	14	0A	00	01	00	07	00	00	00	00	00	00	00	00
12.	**C0**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
13.	**D0**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
14.	**E0**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
15.	**F0**	0A	03	3C	0A	00	01	00	01	00	00	00	4E	41	00	00	00

SN	Add	SQ	WE	CU	HO	OF	IM	CL	VM	CM	SM	SC	Pr	Pt	StCnt	Cb

==========
PAGE A2 hex - SCHEDULE data for Schedule #16 to Schedule #31
==========

SN	*Add*	SQ	WE	CU	HO	OF	IM	CL	VM	CM	SM	SC	Pr	Pt	StCnt	Cb
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
16.	**00**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
17.	**10**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
18.	**20**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
19.	**30**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
20.	**40**	0A	05	3C	0A	0A	01	00	01	01	00	00	4E	41	02	20	00
21.	**50**	14	0A	46	14	00	06	05	02	03	00	00	4E	41	00	00	00
22.	**60**	14	0A	14	00	00	01	00	04	02	00	00	4E	41	00	00	00
23.	**70**	00	00	0A	28	00	01	00	04	02	00	00	00	00	00	00	00
24.	**80**	00	00	0A	3C	00	01	00	04	02	00	00	00	00	00	00	00
25.	**90**	0A	03	3C	0A	00	01	00	01	00	00	00	32	28	00	00	00
26.	**A0**	14	01	5A	0A	00	1E	01	07	01	00	00	32	28	00	00	00
27.	**B0**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
28.	**C0**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
29.	**D0**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
30.	**E0**	0A	03	3C	0A	00	01	00	01	00	00	00	00	00	03	30	00
31.	**F0**	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00

SN	Add	SQ	WE	CU	HO	OF	IM	CL	VM	CM	SM	SC	Pr	Pt	StCnt	Cb

APPENDIX C (cont.)
SCHEDULE DATA (cont.)

```
============
```
PAGE A4 hex - SCHEDULE data for Schedule #32 to Schedule #47
```
============
SN  Add  SQ WE CU HO OF IM CL VM  CM SM SC Pr Pt StCnt Cb
          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
32.  00  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
33.  10  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
34.  20  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
35.  30  0A 03 3C 0A 00 01 00 01  00 00 00 00 00 00 00 00
36.  40  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
37.  50  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
38.  60  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
39.  70  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
40.  80  0A 03 3C 0A 00 01 00 01  00 00 00 00 00 04 40 00
41.  90  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
42.  A0  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
43.  B0  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
44.  C0  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
45.  D0  0A 03 3C 0A 00 01 00 01  00 00 00 00 00 00 00 00
46.  E0  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
47.  F0  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
-------------------------------------------------------
SN  Add  SQ WE CU HO OF IM CL VM  CM SM SC Pr Pt StCnt Cb
-------------------------------------------------------

============
```
PAGE A6 hex - SCHEDULE data for Schedule #48 and #49; and EXTENDED FUNCTIONS data
```
============
SN  Add  SQ WE CU HO OF IM CL VM  CM SM SC Pr Pt StCnt Cb
          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
48.  00  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00
49.  10  1E 04 3C 0A 00 01 00 01  01 00 00 00 00 00 00 00
     20  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     30  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     40  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     50  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     60  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     70  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     80  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     90  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     A0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     B0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
     C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
     D0  00 00 00 0A 00 00 14 00  00 1E 00 00 28 00 00 00
EF   E0  01 00 00 00 00 00 00 00  01 00 00 00 00 00 00 63
EF   F0  00 00 00 00 FF FF FF FF  FF 30 FF FF FF FF 4D 00
-------------------------------------------------------
EXTENDED FUNCTIONS (EF) data (starting from address 0xE0):
         Id SE SS CC CA bS PO bE  87 PP PF Sd bL Cr rA rA
         CO St PC bd SI              tr             WN LS
-------------------------------------------------------
```

APPENDIX C (cont.)
SCHEDULE DATA EXAMPLES

Schedule 20, EEPROM page is **0xA2**, has following EEPROM addresses and data:

	SQUEEZE	WELD	PERCENT	HOLD	OFF	IMPULS	COOL	ValveM	CycleM	SlopeM	SlopeC	Pr	Pt	StepperC	% base
addr [*hex*]	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4F
data [*hex*]	0x0A	0x05	0x3C	0x0A	0x0A	0x01	0x00	0x01	0x01	0x00	0x00	0x4E	0x41	0x0220	0x00
data	10	05	60	10	10	01	00	01	01	00	00	78	65	544	00

SCHEDULE DATA RANGES

	SQUEEZE	WELD	PERCENT	HOLD	OFF	IMPULS	COOL	ValveM	CycleM	SlopeM	SlopeC	Pr	Pt	StepperC	% base
MIN [*hex*]	0x00	0x00	0x00	0x00	0x00	0x01	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
MAX [*hex*]	0x63	0x63	0x63	0x63	0x63	0x63	0x63	0x07	0x04	0x02	0x63	0xFF	0xFF	0x27FF	0x63
MAX	99	99	99	99	99	99	99	07	04	02	99	255	255	9999	99

EXTENDED FUNCTIONS DATA

The page **A6** also holds the EXTENDED FUNCTIONS data. EEPROM word **address** for any EXTENDED FUNCTIONS data can be calculated by using following equation:

$$EEPROM\ \ addressEF\ [hex] = 0xE\ 0 + EP$$

where: *0xE0* is hex value of first address for EXTENDED FUNCTIONS data
EP is EXTENDED FUNCTIONS Pointer

These data start at Word address 0xE0, with *EP* from 0 to 20 (0x14), as follows:

```
00. Id - Id number of the control
01. SE - Seam
02. SS - Schedule Select
03. CC - Automatic Voltage Compensation
04. CA - Clear All / Setup Constant Current
05. bS - Back Step
06. PO - Process Output
07. bE - Beat/Non Beat
08. 87 - 87 Degree Delay
09. PP - Programmable Power Factor
0A. PF - Power Factor Measuring
0B. Sd - Squeeze Delay
0C. bL - Blocking Delay
0D. Cr - Current Regulation
0E. rA - Range (Current capacity) First Byte
0F. rA - Range (Current capacity) Second Byte
```

APPENDIX C (cont.)
EXTENDED FUNCTIONS DATA (cont.)

10. CO - Current Offset
11. St - Stepper
12. PC - Pressure Control
13. bd - Background(Return)Pressure
14. SI - Sensor Input(reading)
19. tr – Turns Ratio of transformer

EXTENDED FUNCTIONS DATA RANGES

	I.d.	S.E.	S.S.	C.C.	C.A.	b.S.	P.O.	b.E.	87	P.P.	P.F.
MIN [*hex*]	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
MAX [*hex*]	0x63	0x63	0x03	0x1F	0x0B	0x0A	0x27	0x03	0x01	0x99	–
MAX	99	99	03	31	11	10	39	03	01	99	–

	S.d.	b.L.	C.r.	r.A.	C.O.	S.t.	P.C.	b.d.	S.I.	t.r.
MIN [*hex*]	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x0A
MAX [*hex*]	0x63	0x63	0x21	0x270F	0x27	0x01	0x06	0xFF	–	0xFF
MAX	99	99	33	9999	39	01	06	255	–	255

OTHER DATA

==========

PAGE A8 hex - Calibration data. Should not be accessed.

==========

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
00	40	06	05	0D	40	08	09	20	40	0C	0B	0E	40	0D	10	20
10	40	40	0A	0A	40	40	0E	0C	40	40	12	0E	40	40	26	1D
20	40	40	40	0A	40	40	40	0E	40	40	40	12	40	40	40	17
30	40	40	40	1C	40	40	40	22	40	40	40	29	40	40	40	30
40	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
50	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
60	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
70	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
80	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
90	E4	E5	EB	EB	EA	E9	EA	EB	E3	E4	E8	E7	E9	E9	E9	EB
A0	E4	E5	EB	EB	EA	E9	EA	EB	E3	E4	E8	E7	E9	E9	E9	EB
B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
C0	40	06	05	0D	40	08	09	20	40	0C	0B	0E	40	0D	10	20
D0	40	40	0A	0A	40	40	0E	0C	40	40	12	0E	40	40	26	1D
E0	40	40	40	0A	40	40	40	0E	40	40	40	12	40	40	40	17
F0	40	40	40	1C	40	40	40	22	40	40	40	29	40	40	40	30

APPENDIX C (cont.)
OTHER DATA (cont.)

==========
PAGE AA hex - empty page
==========

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
10	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
30	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
40	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
50	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
60	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
70	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
80	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
90	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

The last two pages **AC** and **AE** contain CONSTANT CURRENT data for EN1001 Weld Controls. These data are stored in groups of 9 bytes per schedule. Schedules from 00 to 24 are stored in EEPROM page 0xAC, and Schedules from 25 to 49 in page 0xAE. For example, for Schedule 00:

- Current is stored in first two locations; i.e., address 0x00 and 0x01; value is 60%;
- ***L.o.*** (Low limit) correspond to two bytes at addresses 0x03 and 0x04; value is 55%;
- ***H. i.*** (High limit) correspond to two bytes at addresses 0x06 and 0x07; value is 65%.

The first byte is the MSB and the second the LSB byte; i.e., they are stored as two bytes integer type. Third byte is calculated internally by the control after programming first two bytes.

==========
PAGE AC hex - CU, Lo, Hi data for CONSTANT CURRENT mode
==========

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	3C	00	00	37	00	00	41	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E0	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

APPENDIX C (cont.)
OTHER DATA (cont.)

==========

PAGE AE hex - CU, Lo, Hi data for CONSTANT CURRENT mode

==========

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E0	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	F5	00	30	FF
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

APPENDIX D WIRING DIAGRAMS

421180-006 W/D, EN1000-Series/232, “S” Cabinet

421210-020 W/D, EN1000-Series/232, “D/T/LS/LF” Cabinet
421210-022 W/D, EN1000-FPX(SCR)/232, 11x11 Flat Plate
421210-035 W/D, EN1000-Series/232, “D/T/LS/LF” Cabinet, wired for 380VAC Operation

421212-005 W/D, EN1000-Series/232, “E” Cabinet

421214-039 W/D, EN1000-(1-8)Series Cascade/232, “D/T” & NEMA 12 Cabinets
421214-044 W/D, EN1000-(1-8)(SCR) Cascade/232, “D/T” & “LS/LF” Cabinets

421262-004 W/D, EN1000-Series/232, “B” Cabinet

421268-002 W/D, EN1001-Series/232, “E” Cabinet

421269-006 W/D, EN1001-Series/232, “D/T/LS/LF” Cabinet
421269-028 W/D, EN1001/S49-Series/232, “D/T/LS/LF” Cabinet

421270-002 W/D, EN1001-Series/232, “S” Cabinet

421324 W/D, EN(2)1001-2200/232, “LF” Cabinet

421330 (OBS) W/D, EN1200-200/232, “A” Cabinet – *Obsolete Control*

421337 W/D, EN(6)1000/SP-1200/232, “XF” Cabinet, SP=1 Valve Transformer & 36x60x12 Cabinet

421343 W/D, EN(4)1000/SP-1200/232, “XF” Cabinet, SP=1 Valve Transformer & 36x42x12 Cabinet

421356 W/D, EN(3)1000/SP-1200/232, “HF” Cabinet, SP=1 Valve Transformer

421396 W/D, EN(3)1000-1200/232, “HF” Cabinet

421453 W/D, EN(4)1000-1200/232, “HF/GF” Cabinet

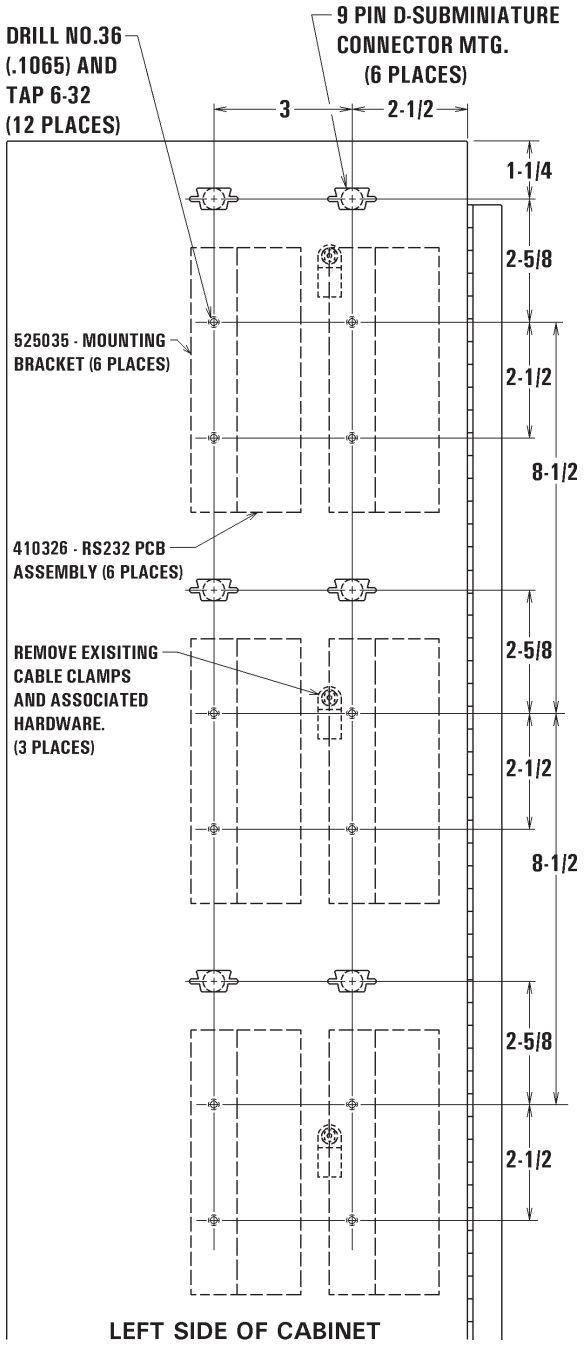
421466 W/D, EN(2)1000-1200/232, “LF” Cabinet

421471 W/D, EN(5)1000/SP-1200/232, “XF” Cabinet, SP=1 Valve Transformer & 36x60x12 Cabinet

421473-001 W/D, EN1001-Series/FPI3/232, “D/T” Cabinet, with Secondary Current Coils

APPENDIX E SUGGESTED CABINET MODIFICATION FOR RS232 OPTION IN EN1000/EN1001 MULTIPLE CONTROLS

Diagram shows suggested cabinet modifications for mounting multiple RS232 Options in EN1000/ EN1001 Multiple Controls. Drill, tap and punch necessary holes for desired number of option assemblies, starting at top of cabinet. To mount assemblies, see Section 7.0 for detailed instructions.



Suggested Cabinet Modification