



Tel : +44 (0)1798 873986
 Fax : +44 (0)1798 872479
 e-mail : hvsales@spellmanhv.co.uk

Broomers Park, Broomers Hill Lane
 Pulborough, W. Sussex, RH20 2RY

Serial Protocol for the MPS Series

Title: Serial Protocol for the MPS Series

Document Number: 47568-21

APPROVALS

Issue	D	1	2
Date	26/02/08	08/12/11	31/10/2013
Issuing Authority	I-1128	6663P	7493F
Engineering Approval			
Sales/Marketing Approval			

Serial Protocol for small modular HV power supplies
 Spellman High Voltage Electronics Limited
 Company Confidential

This document must not be copied or distributed without express permission from Spellman High Voltage Electronics Limited

CHANGE HISTORY

Section	Reason for Change	Issue
All	Created	A
5, 6, Appendix 1	Commands have changed. Removed unused commands.	B
All	Document changed to cover the MPS series	C
All	Change to commands, added RS485	D
Appendix 2	Example of code added	1
3.1 Appendix 1	30kv and 2.5kV units added. 20 Watt information and Command Added	2

Contents

1. Introduction..... 4
2. Hardware 4
3. General message format..... 4
 3.1. Device Types for the MPS Series 5
 3.2. Response from MPS Units 5
 3.3. Checksum calculation 6
Appendix 1 Command set for the MPS Series..... 7
Appendix 2 Code example 9

1. Introduction

The MPS Series protocol is based on the general protocol used with Spellman HV Electronic Ltd power supplies which is described in this document. At the hardware level the protocol runs over an RS232/RS485 two wire interface with a single master and multiple slave units. Up to 9 addresses are available on any one network. Appendix A contains the specific command set for the MPS series power supply.

2. Hardware

The default data rate is 8 bit, no parity at 9600 baud with 1 stop bit. The baud rate can be increased to 19200 using one of the commands listed below.

3. General message format

All data is encoded into ascii format.

The first character of any message is an STX character followed by the one digit address of the unit to which the message is being sent. The next character is used to indicate the type of power module. Following the device type are the characters that make up the data in the message which usually comprises a command and command argument, the exact format of this depends on the type of command but generally the command is three characters followed by a six character argument. The command itself comprises a two character identified followed by a single character operator. The next character is a checksum followed by a line feed to indicate the end of the message.

The general message format is shown below :-

Transmit from host

```
<STX><ADDR><DEVTYPE>< DATA><CSUM><<LF>
```

Response from host

```
<STX><ADDR><DEVTYPE><DATA><CSUM><LF>
```

The host ADDR is always 9

Where:

<STX>	= 1 ASCII 0x02 Start of Text character.
<ADDR>	= 1 ASCII character representing the address of the unit to which the message is being sent.
<DEVTYPE>	= 1 ASCII characters representing the type of power module.
< DATA>	= Command Argument, up to 7 ASCII characters.
<CSUM>	= Checksum (see section 7 for details)
<LF>	= 1 ASCII 0x0A Line Feed character

3.1. Device Types for the MPS Series (10 Watt and 20 Watt)

Model Type	DEVTYPE
MPS0.6	"1" (0x31)
MPS1	"2" (0x32)
MPS2	"3" (0x33)
MPS3	"4" (0x34)
MPS5	"5" (0x35)
MPS10	"6" (0x36)
MPS15	"7" (0x37)
MPS20	"8" (0x38)
MPS30	"9" (0x39)
MPS2.5	"a" (0x61)

20 Watt units

20 Watt units have an added command to tell them apart from 10Watt units. The 10 Watt in response to a device type query 'DT?' remains unchanged

For 10 Watt units:

In response to: <STX><1><1><DT?><XX><LF>

The message would be: <STX><9><0><31><XX><LF>

For 20 Watt units:

In response to: <STX><1><1><DT?><XX><LF>

The message would be: <STX><9><0><31><**32**><XX><LF>

All other commands remain unchanged for both unit types

3.2. Response from MPS Units

The format of the response is very similar to that of the transmitted message except that the address is always 9, the address of the bus master and the unit type is always 0.

i.e. In response to <STX><1><1><V1?><XX><LF> the message would be <STX><9><0><600.0><XX><LF>

An acknowledge message is transmitted in response to any message not requiring an explicit response.

<STX><9><0><XX><LF>

3.3. Checksum calculation

The checksum is calculated as follows:

- Add the <ADDR>, <DEVTYPE>, and <DATA> bytes into a 16 bit (or larger) word. The bytes are added as unsigned integers.
- Take the 2's compliment (negate it).
- Truncate the result down to the eight least significant bits.
- Clear the most significant bit (bit 7) of the resultant byte, (bitwise AND with 0x7F).
- Set the next most significant bit (bit 6) of the resultant byte (bitwise OR with 0x40).

Using this method, the checksum is always a number between 0x40 and 0x7F. The checksum can never be confused with the <STX> or <LF>, since these have non-overlapping ASCII values.

Appendix 1 Command set for the MPS Series

Command	Description	Typical Response	
V1=xxxx.x	Set output voltage	ACK	Where xxxx.x is a voltage from 0.0 to the maximum voltage of the unit.
V1?	Request present value of output voltage	300.0	(See Note 1)
M0?	Read Voltage Monitor	300.0	0V to the maximum voltage of the unit. (See Note 2)
M1?	Read Current Monitor	200.0	0 μ A to the maximum current of the unit (See Note 3)
R0?	Read Voltage Monitor Raw value	XXXX	Where XXXX is ascii hex value in the range 0 to 0x3FFF (See Notes 2)
R1?	Read Current Monitor Raw value	XXXX	Where XXXX is ascii hex value in the range 0 to 0x3FFF (See Notes 3)
EN0	Disable all outputs	No response	(See Note 4)
EN1	Enable all outputs	No response	(See Note 4)
ID?	Read the units address	1	(See Note 4 and 5)
ID=x	Set the unit address	ACK	(See Note 6)
SW?	Request software version number	V1.00R0	
BD?	Read the baud rate	x	Where x is 0 for 9600, 1 for 19200
BD=x	Set the baud rate	ACK	Where x is 0 for 9600, 1 for 19200. Changes will only take place after a power on reset.
DT?	Device type	X X2	10W X is the device type 1 to a 20W X is the device type 1 to a

Notes:

1. The V1 command uses a 16 bit DAC. To calculate the resolution use the maximum voltage / 65536. For example, the MPS10 will give 0.15V resolution.
2. The M0 command uses a 14 bit ADC. To calculate the resolution use:-
 $(5 * \text{Maximum Voltage} / 4.536) / 16384$
 For example, the MPS10 will give 0.7V resolution.
3. The M1 command uses a 14 bit ADC. To calculate the resolution:-
 $(5 * \text{Maximum Current} (\mu\text{A}) / 4.536) / 16384$
 For example, the MPS10 will give 0.06 μ A resolution.
4. Broadcast message – all units will respond to this command.
 An address of 0 is used as a global address and all units on the network will act on

this global command.

5. When reading the address of a unit in a multi slave arrangement, make sure that only this unit is connected to the host as all units will respond to this command.
6. The address of the unit can be set to most ascii character from 0x01 to 0x255. Do not use "9" (0x39) (address of host), STX (0x02), LF (0x0A) or 0x00 (broadcast address). The default address of the unit is set to "1" (0x31).

Example to set the voltage demand of a MPS3 to 3kV

```
<STX>14V1=3000.0v<LF>
```

Where "1" is the default address of the unit, "4" is the MPS3 device type, "V1=3000.0" is the command to set 3kV and "v" (0x76) is the checksum

Response:

```
<STX>90W<LF>
```

Where "9" is the address of the host, "0" is the device type of the host and "W" (0x57) is the checksum.

Example to read the voltage demand of an MPS0.6

```
<STX>11V1?X<LF>
```

Where "1" is the default address of the unit, "1" is the MPS0.6 device type, "V1?" is the command and "X" (0x58) is the checksum

Response:

```
<STX>90600.0c<LF>
```

Where "9" is the address of the host, "0" is the device type of the host, "600.0" is the demand voltage and "c" (0x63) is the checksum.

Appendix 2 Code example

The following "C" source gives an example of message to request the version number. It assumes a unit with an address of "1" and device type "1"

```
CRC = CheckSum(STX + "11SW?")  
Will return:  
CRC = 75Hex
```

```
Function CheckSum(ByVal message As String, ByVal length As Int16) As  
Int16  
    Dim I As Integer  
    CheckSum = 0  
    If Len(message) <= 0 Then Exit Function ' No arguments passed.  
    message = Microsoft.VisualBasic.Mid(message, 2, length)  
  
    For I = 1 To Len(message)  
        CheckSum += Asc(Microsoft.VisualBasic.Mid(message, I, 1))  
    Next I  
    CheckSum = &H200 - CheckSum  
    CheckSum = CheckSum And &H7F  
    CheckSum = CheckSum Or &H40  
End Function
```