Tel : +44 (0)1798 877000
hvsales@spellmanhv.co.uk

Broomers Park, Pulborough,
W. Sussex, UK. RH20 2RY

Product:

MXR Series

Title:

Serial Communications Protocol

Document Number:

81318-21

Issue:

1

Date:

15/November/2021

Change History

ISSUE	DATE	NAME	SECTION	CHANGE
1	Nov 21	JS	All	Created from 47811-21 issue C
			3	Multiple outputs option removed
			3.1	Ramp rate and polarity set command removed (not present in firmware)
			3.1	Internal fault request (FT?) - list of responses corrected

Contents

1. Introduction

The 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 two wire interface. A provision to include an address character has been included such this protocol can be used on a multi device RS485 system. On an RS232 system the address character will always be 0.

2. General message format

The basic RS232 message parameters are as follows:

Data Rate	19200
Parity	None
Stop Bits	1

All data is encoded into ASCII format.

2.1 Message Structure

The first character of any message is an STX character followed by 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 for communication in both directions is shown below :-

<STX><ADDR>< DATA><CSUM><LF>

Where:

<STX> = 1 ASCII 0x02 Start of Text character.

<ADDR> = 1 ASCII address character for RS485 bus usage (always 0 in RS232 operation for both directions of communication).

<DATA> = Command Argument, up to 7 ASCII characters.

<CSUM> = Checksum (see below for details)

<LF> = 1 ASCII 0x0A Line Feed character

Spellman High Voltage Electronics Limited | +44 (0)1798 877000 | hvsales@spellmanhv.co.uk | Broomers Park, Pulborough, W. Sussex, UK. RH20 2RY

TEMPLATE: 40776 ISS B | ECN I-10887 | DATE: 05/11/2021 QA/ENG/SAL

2.2 Checksum Calculation

The checksum is calculated as follows:

- Add the <DATA> and <ADDR> 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.

Below is an example of how to generate the checksum in Visual Basic

```
Dim ADDR As Char = "0"          ' Address is '0' for RS232
Dim comms_message As String = STX + ADDR + "VA=3000.0"         'Set voltage demand to 3kV
Dim CRC As Int16 = CheckSum(comms_message, Len(comms_message))'CRC = 0x5B in this example
comms_message += Chr(CRC) + vbLf
'we need to send as byte array, not a string as VB only sends signed chars due to ascii
encoding. This is required to correctly send the CRC byte.
Dim encoding As New System.Text.UTF8Encoding()
Dim tx_bytes(50) As Byte = encoding.GetBytes(comms_message)
SerialPort1.Write(tx_bytes, 0, tx_bytes.Length + 1)


Public 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 = &H100 - CheckSum
    CheckSum = CheckSum And &H7F
    CheckSum = CheckSum Or &H40
End Function
```

3. Command Protocol

3.1 Command set

Command	Description	Typical Response	Remarks
VA=xxxxx.x	Set Channel A output voltage	VA=xxxxx.x	Where xxxxx.x is a voltage from 0 to the maximum voltage of the unit.
VA?	Request present value of Channel A set voltage	VA=xxxxx.x	
UA?	Request Channel A Voltage Monitor	UA=xxxxx.x	0V to the maximum voltage of the unit.
IA?	Request Channel A Current Monitor	IA=xxxxx.x	0µA to the maximum current of the unit.
SM?	Request Supply rail Monitor	SM=xxxx.xx	Where xxxx.xx is the rail voltage
TM?	Request Temperature Monitor	TM=xxxx.xx	Where xxxx.xx is the temperature in degrees C.
EA0	Disable Channel A output	EA0	
EA1	Enable Channel A output	EA1	
EA?	Request Channel A output state.	EA=x	Where x is 0 for disabled or 1 for enabled.
PA?	Request Channel A polarity	PA=x	Where x is 0 for positive and 1 for negative polarity.
SW?	Request software version number and unit type	V1.00R0 MX30PN24/736	
ID?	Read the units address	ID=x	Address is 0 in RS232 operation
ID=X	Set the unit address	IDx	Not used in RS232 operation
IL?	Request Interlock status	IL=0	IL=0 Interlock open, IL=1 Interlock closed.
FT?	Internal fault status request	FT=0	Internal fault e.g. over temperature or I/P voltage failure. FT=0 No fault, FT=1 Over Temperature Fault FT=2 Voltage Input out of range FT=3 Over Voltage Fault
Invalid Command	Incorrect command or syntax error	ERR	

Spellman High Voltage Electronics Limited | +44 (0)1798 877000 | hvsales@spellmanhv.co.uk | Broomers Park, Pulborough, W. Sussex, UK. RH20 2RY

TEMPLATE: 40776 ISS B | ECN I-10887 | DATE: 05/11/2021 QA/ENG/SAL

3.2 Command Examples

Example to set the voltage demand to 3kV

<STX>0VA=3000.0#<LF> Where “0” is the unit address (always 0 in RS232), “VA=3000.0” is the command to set 3kV and “#” (0x5B) is the checksum

Response:

<STX>0VA=3000.0#<LF> Where “0” is the unit address (always 0 in RS232), “VA=3000.0 is the echo of the command and “#” (0x5B) is the checksum.

Example to read the voltage demand

<STX>0VA?#<LF> Where “0” is the unit address (always 0 in RS232), “VA?” is the command and “#” (0x7A) is the checksum

Response:

<STX>0VA =600.0#<LF> Where “0” is the unit address (always 0 in RS232), “600.0” is the demand voltage and “#” (0x48) is the checksum.

Example to read the polarity

<STX>0PA?#<LF> Where “0” is the unit address (always 0 in RS232), “PA?” is the command and “#” (0x40) is the checksum

Response:

<STX>0PA=0#<LF> Where “0” is the unit address (always 0 in RS232), “PA=0” is the polarity and “#” (0x52) is the checksum.

Example to change the enable HV output

<STX>0EA1#<LF> Where “0” is the unit address (always 0 in RS232), “EA1” is the command and “#” (0x59) is the checksum

Response:

<STX>0EA1#<LF> Where “0” is the unit address (always 0 in RS232), “EA1” is the echo of the command and “#” (0x59) is the checksum.