Remote Control a Single PowerLab 8

Features:

PowerLab 8 is an 8 cell balancing charger, monitor and discharger. It has the ability to safely cycle many different types of battery chemistries. Each battery parameter is defined in 1 of 25 programmable user presets that have over 120 changeable parameters each. Charger Control Software (CCS) tests for valid and safe preset parameters and programs them to the PowerLab 8's non-volatile memory. After the presets are loaded, they can be remotely started, stopped and monitored thru the serial interface.

Serial Data Specification:

PowerLab 8 uses a master/slave form of communication. The master is usually a PC host. The master has to quarry the slave (PowerLab 8) to get a response. The serial interface is single wire full duplex with a 10K pull up resistor tied to 3.6V. A pause of more than 3 bytes should precede any network communication.

Serial data is 19,200 bps, 8 bit, 1 start bit, 1 stop bit and no parity check. Data is non-inverted, which means logical 1 is 3.6V.

Any devices connected to the serial interface should be open collector in order to prevent holding the data line high. If open collector is not available, a diode should be placed on the TX port of any device connected to the network to prevent holding the line high while listening.

All bytes are 8 bit binary (0-255). Multi byte data is MSB first. For example, 16bit is 2 bytes long. 32bit is 4 bytes long.

There are two classes of commands for remote controlling the PowerLab 8.

- Ram – Quarries the PowerLab 8 for a PowerLab 8 status packet
- Sel – Issues a start command or button press

PowerLab 8 Status Packet:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	Ram in	-4 – -2	
Charger ID Num (0-16)	1 in	-1	0 = Master Charger (for multi charger networks) Use Slaves Found to determine which slave addresses are connected to the expansion network (for multi charger networks). Always use 0 to read from the master charger or just a single charger.

Firmware Version	2 out	0 – 1	0.00 thru 655.35
Cell1 Volts	2 out	2 – 3	Volts = 16bit * 5.12V / 65536
Cell2 Volts	2 out	4 – 5	Volts = 16bit * 5.12V / 65536
Cell3 Volts	2 out	6 – 7	Volts = 16bit * 5.12V / 65536
Cell4 Volts	2 out	8 – 9	Volts = 16bit * 5.12V / 65536
Cell5 Volts	2 out	10 – 11	Volts = 16bit * 5.12V / 65536
Cell6 Volts	2 out	12 – 13	Volts = 16bit * 5.12V / 65536
Cell7 Volts	2 out	14 – 15	Volts = 16bit * 5.12V / 65536
Cell8 Volts	2 out	16 – 17	Volts = 16bit * 5.12V / 65536
Synchronous PWM Drive	2 out	18 – 19	0-8191 is Buck, 8192-16383 is Boost
Charge Current Set Point	2 out	20 – 21	Only Valid when Charging Amps = 16bit / 1666
Supply Volts with Current	2 out	22 – 23	Volts = 16bit * 46.96V / 4095 / 16
Supply Volts	2 out	24 – 25	Volts = 16bit * 46.96V / 4095
CPU Temperature	2 out	26 – 27	Tc = (2.5 * 16bit / 4095 - 0.986) / 0.00355
Charge/Discharge Seconds (ChgSec)	2 out	28 – 29	0 to 18*3600 (Use with charge minutes)
Fast Amps Reading	2 out	30 – 31	Amps = 16bit signed / 600
Output Positive Reading	2 out	32 – 33	Volts = 46.96V / 4095
Ahr In to Battery	4 out	34 – 37	mAh = 32bit / 2160
Average Cell Fuel	2 out	38 – 39	Fuel% = 16bit / 10
Start Chg/Dsch Fuel	2 out	40 – 41	Fuel% = 16bit / 10
Average Amps Reading	2 out	42 – 43	Amps = 16bit signed / 600 (Shows on LCD)

Status Flags	2 out	44 – 45	Bit0 = Safety Charge
			Bit8 = Charge/Discharge Complete
			Bit11 = Reduce Amps
			Bit14 = Node Only Charging (3A max current)

RXStatus Flags	2 out	46 – 47	Bit1 = Discharge Running
			Bit4 = Regenerative Discharge
			Bit6 = Charge Running
			Bit7 = Balancers Running

Not Used	2 out	48 – 49	
Status2 Flags	2 out	50 – 51	Bit2 = High Temp (140 deg F)
Internal Resistance Cell 1	2 out	52 – 53	mOhm = (16bit / 6.3984 – VROffset) / VRAmps
Internal Resistance Cell 2	2 out	54 – 55	mOhm = (16bit / 6.3984 – VROffset) / VRAmps
Internal Resistance Cell 3	2 out	56 – 57	mOhm = (16bit / 6.3984 – VROffset) / VRAmps
Internal Resistance Cell 4	2 out	58 – 59	mOhm = (16bit / 6.3984 – VROffset) / VRAmps
Internal Resistance Cell 5	2 out	60 – 61	mOhm = (16bit / 6.3984 – VROffset) / VRAmps
Internal Resistance Cell 6	2 out	62 – 63	mOhm = (16bit / 6.3984 – VROffset) / VRAmps
Internal Resistance Cell 7	2 out	64 – 65	mOhm = (16bit / 6.3984 – VROffset) / VRAmps
Internal Resistance Cell 8	2 out	66 – 67	mOhm = (16bit / 6.3984 – VROffset) / VRAmps
VRAmps	2 out	68 – 69	Amps = 16bit / 600
NiCd Fallback Volts	2 out	70 – 71	Volts = 16bit / 12797 - MaxCell
Not Used	2 out	72 – 73	
MaxCell Volts	2 out	74 – 75	Volts = 16bit / 12797

Status6 Flags	2 out	76 – 77	Bit4 = Constant Voltage
			Bit5 = Preset is Valid and Runable
			Bit8 = Regenerative Discharge Failed

Charge/Dsch Minutes (ChgMin)	2 out	78 – 79	For <18hr use ChgSec For >=18hr Seconds = ChgSec – 64800 + ChgMin * 60
Supply Amps	2 out	80 – 81	Amps = 16bit / 150
Battery Positive	2 out	82 – 83	Volts = 16bit / 12797
Ahr out of Battery	4 out	84 – 87	mAh = 32bit / 2160
Not Used	2 out	88 - 89	
Regen. Volt Set Point	2 out	90 – 91	Volts = 16bit * 46.96V / 4095
Discharge Set Amps	2 out	92 – 93	Amps = 16bit / 600
Internal Discharge PWM	2 out	94 - 95	0 – 8192
Negative Node Volt Drop	2 out	96 – 97	Volts = 16bit * 46.96V / 4095
Positive Node Volt Drop	2 out	98 – 99	Volts = 16bit * 46.96V / 4095
Battery Negative	2 out	100 – 101	Volts = 16bit * 46.96V / 4095
Not Used	2 out	102 – 103	
Starting Supply Volts	2 out	104 – 105	Volts = 16bit * 46.96V / 4095
Not Used	2 out	106 - 107	
Not Used	2 out	108 - 109	
Not Used	2 out	110 - 111	
Not Used	2 out	112 - 113	
VROffset	2 out	114 – 115	mV = 16bit / 6.3984
Slow Average Amps	2 out	116 – 117	Amps = 16bit signed / 600
Preset Set Charge Amps	2 out	118 – 119	Amps = 16bit / 600 (never changes with temp or power)
Slaves Found	2 out	120 - 121	Each bit represents a slave charger that is found
Not Used	2 out	122 - 123	
Balancer 1 PWM	1 out	124	0 – 31
Balancer 2 PWM	1 out	125	0 – 31
Balancer 3 PWM	1 out	126	0 – 31
Balancer 4 PWM	1 out	127	0 – 31
Balancer 5 PWM	1 out	128	0 – 31
Balancer 6 PWM	1 out	129	0 – 31
Balancer 7 PWM	1 out	130	0 – 31
Balancer 8 PWM	1 out	131	0 – 31
Detected Cell Count	1 out	132	1 – 8 (0 = no cells detected)
Mode (Running)	1 out	133	0 = Charger Ready to Start 1 = Detecting Pack 6 = Charging 7 = Trickle Charging 8 = Discharging 9 = Monitoring 10 = Halt for Safety Screen 11 = Pack Cool Down (when cycling) 99 = System Stop Error Occurred
Error Code	1 out	134	Only valid in Mode 99

Chemistry	1 out	135	1 = Lithium Polymer 2 = Lithium Ion 3 = A123 4 = Lithium Manganese 5 = Lithium Cobalt 6 = NiCd 7 = NiMh 8 = Lead Acid 9 = LiFE 10 = Primary 11 = Power Supply
Not Used	1 out	136	
Loaded Preset Number	1 out	137	0 – 24 (Zero based number)
Not Used	1 out	138	
Screen Number	1 out	139	Screen showing on LCD
Not Used	1 out	140	
Not Used	1 out	141	
Cycle Number	1 out	142	0 – 255 (A complete Charge/Discharge is one cycle)
Power Reduced Reason	1 out	143	0 = Full Power Allowed 1 = Input Current Limit 2 = 60A Input Current Limit Reached 3 = Cell Sum Error (Charge) 4 = Supply Noise 5 = High Temp 6 = Low Input Voltage 7 = Constant Voltage Output 8 = Internal Max 100W Discharge 9 = High Temp Discharge 10 = Regen. Max Amps Reached 11 = High Temp Discharge 12 = Cell Sum Error (Discharge) 13 = Regen. Volt Limit Reached 14 = Discharge Reduced (Below Average Charger) 15 = Reduce (Above Average Charger) 16 = Supply Low for High Power
Not Used	1 out	144	
Not Used	1 out	145	
Not Used	1 out	146	
CRC Checksum	2 out	147 – 148	See Text for 16bit calculation

CRC Checksum Calculation:

This sample Visual Basic code will generate the checksum. The result should be compared to the last 2 bytes in the packet (CRC Checksum)

```
RXCRC16 = 2342 'Initialize Checksum for RAM0
For c = 0 To 146 'Do not include RAM0 or CRC Checksum
    CRC16Byte(InBytes(c), RXCRC16)
```

```
Next

Sub CRC16Byte(ByVal Data As Int32, ByRef CRC16 As Int32)
    Dim Temp As Int32
    Dim c As Short
    'Make sure data is not negative
    If Data < 0 Then Data = Data + 256
    'Use Int() to remove rounding
    For c = 1 To 8
        Temp = Data Xor CRC16
        If Int(Temp / 2) = Int((Temp - 1) / 2) Then
            CRC16 = Int(CRC16 / 2)
            CRC16 = 33800 Xor CRC16
        Else
            CRC16 = Int(CRC16 / 2)
        End If
        Data = Int(Data / 2)
    Next
End Sub
```

Select a Preset:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	SelP in	-4 – -1	
Preset Number (0-24)	1 in	0	Zero based number
CRC Checksum	2 out	1 – 2	PowerLab acknowledges receipt of packet

```
RXCRC16 = 4372 'Initialize Checksum for SelP
CRC16Byte(PresetNumber, RXCRC16)
```

Here is an example:

Send to PowerLab 8 ‘S’,’e’,’l’,’P’,0
Read the checksum 56h, B4h

Start Charge Using Bananas:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	SelC in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Here is an example:

Send to PowerLab 8 ‘S’,’e’,’l’,’C’
Read the checksum 05h, DCh

Start Charge Without Bananas:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	Selc in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Here is an example:

Send to PowerLab 8 ‘S’,’e’,’l’,’c’
Read the checksum 05h, DCh

Start Discharge Using Bananas:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	SelD in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Start Discharge Without Bananas:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	Seld in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Start Monitor Using Bananas:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	SelM in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Start Monitor Without Bananas:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	Selm in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Start Cycling Using Bananas:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	SelY in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Start Cycling Without Bananas:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	Sely in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Press Enter Button to Acknowledge a safety screen:

Variable Name	Bytes Direction	Position	Convert to Units
Packet Class	SelE in	-4 – -1	
CRC Checksum	2 out	0 – 1	Always reads 05DCh

Checking Charger Status:

1. Send Ram0 to request a status packet
2. Verify the CRC checksum to confirm the received packet is valid.
3. Gather the following important information from the packet.
    - Cell Voltages
    - Mode
    - Preset Number
    - Charge/Discharge Complete (from Status Flags)

Select a Preset:

1. Send Ram0 (Charger Status)
2. Verify that Mode = 0
3. Send SelP#, where # is the zero based preset number
4. Verify the CRC checksum to confirm the packet was accepted

5. If the checksum is not valid, resend the packet.

Start a Charge/Discharge/Cycle/Monitor:

1. Send Ram0 to check charger status
2. Verify that Mode = 0
3. Send SelC to start the charge using the bananas
4. Verify the acknowledged CRC checksum is 05DCh
5. Send Ram0 (Charger Status) periodically
6. If Mode = 10, send SelE (Enter Press) to acknowledge a safety screen.
   Note: safety screens can be removed for OEM applications. Contact FMA for a custom OEM password.
7. Send Ram0 (Charger Status) periodically
8. Verify that Mode is not 99 (showing an error occurred)
9. When Charge/Dsch Complete (Status Flags) is true, the charge or discharge is finished.

Stop a Charge:

1. Send Ram0 (Charger Status)
2. Verify that Mode = 6, 7, 8, 9 , 11
3. Send SelE (Enter Button)
4. Verify the returned Checksum 05DCh
5. Send Ram0 (Charger Status)
6. Verify that Mode = 0

Clear an Error:

1. Send Ram0 (Charger Status)
2. Verify that Mode = 99
3. Send SelE (Enter Button)
4. Verify the returned Checksum 05DCh
5. Send Ram0 (Charger Status)
6. Verify that Mode = 0

Charger Networking:

PowerLab sends out an expansion network packet to check if other chargers are connected to the network. Only the master charger with address zero sends out the packet. The format of the networked charger communication is not covered in this document. However a few simple rules can allow remote control of all networked chargers.

1) Charger expansion networking can be disabled. This is best if a single PL8 is remote controlled. Run the PC software and go to options / Start Settings / Disable Expansion Network. This makes communication with PL8 simple and the remaining rules in this list can be ignored.
2) The master charger sends out a network packet every second. Custom designed remote control software needs to work around this by only communicating after the expansion network packet. The packet begins with rCS (request charger status). After the packet is detected, wait 50mS for the network to remain quiet before remote control of the PL8. The length of the expansion network packets can change, so it is good to pole the network a couple of times to make sure it is quiet before communicating. Keep the network conflicts to a minimum by letting the master charger set the timing of the network.
3) Multiple charger address can be detected by reading the Slaves Found (address 120) word. Bit zero is the master charger and is always set. Bits 1 thru 15 represent what expansion chargers are detected by the master charger on the network. If the bit is set, an slave/expansion charger has been detected. The slave/expansion chargers can be monitored by changing the address following the Ram command.
4) Keep in mind that more chargers on the network slow down communication. Only one charger can be read by the Ram command every second right after the master charger expansion network packet. That means 4 chargers on the network will require 4 seconds to gather charge data from all the chargers. Sixteen chargers will require sixteen seconds.

PowerLab 8 technical data
Published by
FMA Inc,
5713 Industry Lane, #50
Frederick, MD 21704
USA

Attention please!
The information herein is given to describe certain components and shall not be considered as warranted characteristics.
Terms of delivery and rights to technical change reserved.
We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Information
For further information please contact FMA Inc at 301-668-7614

Warnings
Due to technical requirements, components may contain dangerous substances.
FMA Inc. Components may only be used in life-support devices or systems with the express written approval of FMA Inc., if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health
of the user or other persons may be endangered.