



## CAN bus protocol for BMS16v2

Ian Hooper, Revision 1.0, May 2019

### Introduction

---

This document is intended to assist people integrating the ZEVA BMS16 V2 with their own devices over CAN bus. It describes the required specifications for the CAN bus, packet format and communications protocols.

### CAN bus settings

---

The standard bus speed is 250kbps and uses CAN 2.0B, with 29-bit IDs. Units intended for CAN integration with inverters are normally 500kbps and 11-bit IDs instead. CAN protocol for supported inverters is not covered in this document - please contact us for more information if required. Similarly please contact us if you require modules programmed to suit a different bus speed or packet format.

All packets are the Data frame type, though some require no actual data. Remote frames were not used for request packets since they are not supported by some CAN devices and protocols which may be on the same bus.

### Physical Layer

---

The BMS16 has four screw terminals for CAN bus connections, being Ground, CAN L, CAN H and a 12V supply output for powering CAN devices. The 12V terminal is limited to 200mA max, and will be powered whenever the BMS is turned on and not in sleep mode.

We recommend using Shielded Twisted Pair (STP) cable, with two pairs of conductor – one pair for CAN H and CAN L, the other for Ground and +12V supply. Conductors should be around AWG20-24 for sufficient mechanical strength and current rating. Belden 8723 is a good option. CAN buses work best as a single chain of devices, without any branching, and with 120Ω termination resistors at both ends to prevent signal reflection. There is no separate shield terminal at the BMS but the shield wire can be connected to the Ground terminal.

### Packet Summary

---

For consistency across product lines, the BMS16 uses the same CAN protocol format as the EVMS3 and BMS12 devices, but some data is not applicable / available. These will be indicated as “Reserved” bytes in message format descriptions below, and should be read / written as zero.

Note that all IDs given are in *decimal* (not hex). CAN bus was invented as a realtime, low latency bus so the standard only supports small data packets up to 8 bytes in length. This sometimes results in the need for strange bit-packing and numerical scaling schemes in order to fit the required data within the 8-byte limit.

### Packet IDs and Structure

---

#### **ID 30: Broadcast Status (Tx)**

The BMS broadcasts this packet at 4Hz, containing its current status and operating information.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Error code					Status code		
Byte 1	Battery amp-hours remaining, 0.1Ah resolution, high byte							
Byte 2	Battery amp-hours remaining, 0.1Ah resolution, low byte							
Byte 3	Battery voltage, 0.1V resolution, high byte							
Byte 4	Battery voltage, 0.1V resolution, low byte							
Byte 5	<i>Reserved</i>							
Byte 6	<i>Reserved</i>							
Byte 7	Temperature (°C+40)							

### ID 32: Receive Configuration Data 1 (Rx)

These 8 bytes contain the first part of the new configuration parameters sent to the BMS. Note that settings will not be saved to memory until all four config packets are received, and values will automatically be limited to their allowable range. See also the BMS16 Manual for more detailed explanation about the function of each setting. Minimums and maximums give the range of raw values to be sent (after scaling as listed in Units field).

	Description	Units	Minimum	Maximum
Byte 0	Pack Capacity	Ah x5	1	250
Byte 1	SoC Warning	%	0	99
Byte 2	Full Voltage	V x2	5	251
Byte 3	Warn Current	A x10	1	121
Byte 4	Trip Current	A x10	1	121
Byte 5	<i>Reserved</i>			
Byte 6	Number of Cells*		8	16
Byte 7	Current shunt type*	Enum	0	3

\* These two are different from the EVMS parameters in the same location, only applicable to the BMS16 and replacing parameters only applicable to the EVMS3. Current shunt enumeration values are no shunt (0), 100A (1), 200A (2), 500A (3) – all 75mV shunts.

### ID 33: Receive Configuration Data 2 (Rx)

These 8 bytes contain the second part of the new configuration parameters sent to the EVMS.

	Description	Units	Minimum	Maximum
Byte 0	<i>Reserved</i>			
Byte 1	<i>Reserved</i>			
Byte 2	<i>Reserved</i>			
Byte 3	<i>Reserved</i>			
Byte 4	<i>Reserved</i>			
Byte 5	BMS Min Voltage	1.50 + 0.01n V	0	250
Byte 6	BMS Max Voltage	2.00 + 0.01n V	0	250

Byte 7	Balance Voltage*	2.00 + 0.01n V	0	252
--------	------------------	----------------	---	-----

\* For Balance Voltage, static thresholds of 2.00–4.50 are achieved with values of 0-250. For dynamic threshold, use value 251. To disable balancing, set value to 252.

**ID 34: Receive Configuration Data 3 (Rx)**

These 8 bytes contain the third part of the new configuration parameters sent to the EVMS.

	Description	Units	Minimum	Maximum
Byte 0	BMS Hysteresis	x0.01 V	0	50
Byte 1	BMS Min Temp	n-40 °C	0	141
Byte 2	BMS Max Temp	n-40 °C	0	141
Byte 3	Max Charge Voltage*	V	0	255
Byte 4	Max Charge Current*	A	0	127
Byte 5	Reserved			
Byte 6	Reserved			
Byte 7	Reserved			

\* Bottom 7 bits of Current limit byte are used for current limit, 0-127A range. Top bit in Current limit byte is used as a 9th bit for Target voltage, to extend range to 0-511V. Generally this extended range is not applicable to the BMS16, so the only thing to remember is to limit current to 0-127, to keep the top bit of this byte zero.

**ID 35: Receive Configuration Data 4 (Rx)**

These 6 bytes contain the final part of the new configuration parameters sent to the EVMS. The final two bytes are reserved for possible future settings and should be ignored.

	Description	Units	Minimum	Maximum
Byte 0	Reserved			
Byte 1	Reserved			
Byte 2	Reserved			
Byte 3	Reserved			
Byte 4	Reserved			
Byte 5	Stationary Mode		0	1
Byte 6	Reserved			
Byte 7	Reserved			

**ID 37: Acknowledge Error (Rx)**

This packet should contain one byte, being the ID of the error being acknowledged.

The following errors will not reoccur during a single running session after being acknowledged: BMS Overtemp, BMS Undertemp, Overcurrent Warning, Low SoC. Other errors can be cleared by sending an acknowledge packet, but may reoccur if the error condition reappears (or may not clear if the error condition is still present when the Acknowledge message is sent).

**ID 38: Reset State of Charge (Rx)**

If the BMS's state of charge appears to have lost synchronisation, you can send a packet ID 38 to reset the SoC to 100%. Any data in this packet will be ignored.

**ID 40: Battery Current (Rx)**

If the BMS16 is measuring current using its built-in shunt interface, it will send out the instantaneous current value at 4Hz on ID 40, as a single 24-bit value (big endian format), in milliamps. The value is unsigned but with a 8388608 offset, i.e subtract this number from the received value to get a signed value for current in milliamps. If the shunt type is set to "none", the BMS will not send out this message, but it may instead be sent by hall effect current sensors on the CAN bus, in the same format.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Battery current, high byte							
Byte 1	Battery current, middle byte							
Byte 2	Battery current, low byte							

**ID 301: Cell Voltages 1-4 (Tx)**

Cell voltages are sent in the same format that the BMS12 uses, emulating two BMS12s of module IDs 0 and 1 and having 8 cells each. Each CAN message contains four voltages as 16-bit values, in millivolts (big endian format).

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Voltage 1, high byte							
Byte 2	Voltage 1, low byte							
Byte 3	Voltage 2, high byte							
Byte 4	Voltage 2, low byte							
Byte 5	Voltage 3, high byte							
Byte 6	Voltage 3, low byte							
Byte 7	Voltage 4, high byte							
Byte 8	Voltage 4, low byte							

**ID 302: Cell Voltages 5-8 (Tx)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Voltage 5, high byte							
Byte 2	Voltage 5, low byte							
Byte 3	Voltage 6, high byte							
Byte 4	Voltage 6, low byte							
Byte 5	Voltage 7, high byte							
Byte 6	Voltage 7, low byte							
Byte 7	Voltage 8, high byte							
Byte 8	Voltage 8, low byte							

**ID 311: Cell Voltages 9-12 (Tx)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Voltage 9, high byte							
Byte 2	Voltage 9, low byte							
Byte 3	Voltage 10, high byte							
Byte 4	Voltage 10, low byte							
Byte 5	Voltage 11, high byte							
Byte 6	Voltage 11, low byte							
Byte 7	Voltage 12, high byte							
Byte 8	Voltage 12, low byte							

**ID 312: Cell Voltages 13-16 (Tx)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Voltage 13, high byte							
Byte 2	Voltage 13, low byte							
Byte 3	Voltage 14, high byte							
Byte 4	Voltage 14, low byte							
Byte 5	Voltage 15, high byte							
Byte 6	Voltage 15, low byte							
Byte 7	Voltage 16, high byte							
Byte 8	Voltage 16, low byte							

**Status and Error Codes**

The broadcast status packet contains a 3-bit value for BMS status and a 5-bit value for any pending errors. Status codes use the following enumeration:

Value	Name
0	Idle
1	<i>Reserved</i>
2	Running
3	<i>Reserved</i>
4	Stopped
5	<i>Reserved</i>

The error code enumeration is as follows:

Value	Name	Information
0	No error	
1	Corrupt settings	Invalid settings value detected in memory
2	Overcurrent warning	Current has exceeded the warning threshold
3	Overcurrent shutdown	Current has exceeded fault threshold (drive shut down)
4	Low cell warning	One or more cells below minimum voltage threshold

5	BMS shutdown	Vehicle shutdown due to undervoltage cell for 10+ seconds
6	High cell warning	One or more cells above maximum voltage threshold
7	BMS ended charge	Charger has been stopped due to overvoltage cell for >1sec
8	BMS over-temp	The temperature has exceeded upper limit
9	BMS under-temp	The temperature is below lower limit
10	Low SoC warning	Battery state of charge has passed the low warning level
11	<i>Reserved</i>	<i>Reserved</i>
12	<i>Reserved</i>	
13	<i>Reserved</i>	
14	<i>Reserved</i>	
15	<i>Reserved</i>	
16	CAN error	A CAN communications error was detected

### Revision History

Version	Date	Description
1.0	May 2019	First cut, based on information from the EVMS3 and BMS12 CAN protocol documents, with required changes made.  Please get in touch if you come across any possible errors.