CAN bus protocol for EVMS3

Ian Hooper, December 2017

Introduction

This document is intended to assist people integrating the ZEVA EVMS3 with their own devices over CAN bus. It describes the required specifications for the CAN bus, packet format and communications protocols.

Standard bus settings

The standard bus speed is 250kbps. Packet format is CAN 2.0B, which uses 29-bit IDs. 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

Most ZEVA devices use 5-pin Molex Eurostyle connectors for the CAN bus. Two pins are for the CAN H and CAN L signals, two pins for the 12V supply and Ground, and the fifth pin for transferring shield grounding along the chain of devices (and should be attached to the vehicle chassis at one end of the CAN bus chain). 12V power is supplied to the CAN bus by the EVMS Core whenever active.

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 choice. 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.

Packet Summary

There are 10 different packet types used by the EVMS3, plus those associated with ZEVA motor controllers, BMS12 modules and TC chargers. Please refer to the documentation for those devices for information about their CAN protocol.

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.

ID 30: Broadcast Status *(Tx)*

The EVMS3 broadcasts this packet at 4Hz, containing it's 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	Auxiliary voltage, 0.1V resolution							
Byte 6	Headlights	Isolation integrity (0-100%)						
Byte 7	Temperature (˚C+40)							

The *Headlights* bit will be 1 if the MPI function is set as Headlight Sense and the MPI terminal has 12V applied.

ID 31: Set State *(Rx)*

Writing a one-byte packet to CAN ID 31 will allow you to put the EVMS into Setup mode remotely. Send a 1 value to enter Setup mode, or a 0 value to return to Idle.

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

These 8 bytes contain the first part of the new configuration parameters sent to the EVMS. 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 EVMS3 Manual for more detailed explanation about the function of each setting. Minimums and maximums give the range of binary 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	EVMS Temp Warning	˚C	0	151
Byte 6	Min Aux Voltage	V	8	15
Byte 7	Min Isolation	%	0	99

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	Tacho PPR		1	6
Byte 1	Fuel Gauge Full	%	0	100
Byte 2	Fuel Gauge Empty	%	0	100
Byte 3	Temp Gauge Hot	%	0	100
Byte 4	Temp Gauge Cold	%	0	100
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

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	255
Byte 5	Alt Charge Voltage*	V	0	255
Byte 6	Alt Charge Current*	A	0	255
Byte 7	Sleep Delay	minutes	1	6

** 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.*

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	MPI Function		0	3
Byte 1	MPO1 Function		0	6
Byte 2	MPO2 Function		0	6
Byte 3	Parallel Strings		1	20
Byte 4	Enable Precharge		0	1
Byte 5	Stationary Mode		0	1
Byte 6	*Reserved*			
Byte 7	*Reserved*			

ID 36: Receive BMS cell numbers *(Rx)*

Writing this packet to the bus allows the battery configuration to be updated in the EVMS. The packet should contain 8 bytes, arranged as 16x 4-bit values starting from BMS ID 0 in the low bits of the first byte.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Module 1 cells				Module 0 cells			
Byte 1	Module 3 cells				Module 2 cells			
Byte 2	Module 5 cells				Module 4 cells			
Byte 3	Module 7 cells				Module 6 cells			
Byte 4	Module 9 cells				Module 8 cells			
Byte 5	Module 11 cells				Module 10 cells			
Byte 6	Module 13 cells				Module 12 cells			
Byte 7	Module 15 cells				Module 14 cells			

ID 37: Acknowledge Error *(Rx)*

This packet should contain one byte, being the ID of the error being acknowledged. Note that some EVMS errors will automatically re-spawn if still present after an acknowledge. Any data in this packet will be ignored.

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

If the EVMS'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)*

The EVMS should receive battery current over CAN bus on ID 40. The packet contains a single 24-bit value (big endian format) for instantaneous battery current, 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.

	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							

Status and Error Codes

The broadcast status packet contains a 4-bit value for EVMS status and a 4-bit value for any pending errors. Status codes use the following enumeration:

Value	Name
0	Idle
1	Precharging
2	Running
3	Charging
4	Setup
5	Stopped

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	A BMS module has reported a temperature above limit
9	BMS under-temp	A BMS module has reported a temperature below lower limit
10	Low SoC warning	Battery state of charge has passed the low warning level
11	Overtemperature	Temp input on Core has exceeded warning level
12	Isolation error	Insulation fault detected above warning level
13	Low 12V	12V / aux battery voltage below warning level for >5 sec
14	Precharge failed	An error was detected during precharge (failed to start or failed to complete, possibly due to faulty wiring)
15	Contator switch error	Mismatch between contactor state and its auxiliary switch (faulty or seized contactor likely)
16	CAN error	A CAN communications error was detected