BMS BMU – Vehicle Communications Protocol

25 February 2013

©2013 Tritium Pty Ltd

Brisbane, Australia

http://www.tritium.com.au

TABLE OF CONTENTS

1 INTRODUCTION

This document describes the communications protocol used between the Battery Management Unit (BMU) and the vehicle in the Tritium BMS.

Internal communications between the BMS BMU and BMS Cell Management Units (CMUs) are described in another document (TRI67.009), but should not be needed for a typical end-user.

2 OVERVIEW

The BMS CMUs communicate with each other and with the BMU using a CAN bus that is separate from the vehicle CAN bus.

The BMU collates and summarises data from the CMUs, including minimum and maximum cell voltages and temperatures, and interfaces to the vehicle on a second CAN bus. This is the communications link detailed in this document.

As a general overview, after power is applied to the system, the CMUs report their (factory programmed) serial numbers to the BMU. The BMU allocates a CAN ID and number of cells to monitor for each CMU and builds a table of CMUs that can be used for fault detection (failed CMUs) and other diagnostics.

Once running, each CMU reports its serial number, status, temperature and up to eight cell voltages at approximately 1Hz. Support for high-speed minimum/maximum voltage updates (~100Hz) is currently being added, and is not detailed in this document.

The CMUs use two redundant measurement channels to take cell voltage readings, and also cross-check their supply voltage and A/D references for faults. One channel uses a high-accuracy (24-bit) slow speed (Hz) converter, and the other channel uses a high-speed (kHz) mid-accuracy (12-bit) converter. Each CMU also measures one external 10k NTC temperature sensor and it's own microcontroller die temperature.

The BMU measures additional data itself: total pack voltage on either side of the precharge contactor, total pack current from a shunt, pack isolation from the vehicle, the status of the precharge control, and the presence of the 12V supply voltage for the contactors. This data, along with the minimum and maximum cell voltage and temperature, is reported by the BMU onto the vehicle CAN bus. The BMU can also echo the individual cell voltage and temperature data to the vehicle CAN bus, if configured to do so.

3   CAN ID ALLOCATIONS

By default, the BMS BMU ships with a CAN base ID set to 0x600. This can be adjusted by the end user. Assuming this base ID, the following CAN IDs are used on the vehicle bus by the BMS BMU:

CAN ID (hex)	Description
0x600	BMU heartbeat & serial number
0x601 - 0x6EF	CMU status, temperature and voltage telemetry, if set to relay to vehicle
0x6F0	Reserved for factory configuration & calibration commands
0x6F1 - 0x6F3	Reserved for future configuration system update
0x6F4	Pack SOC
0x6F5	Balance SOC
0x6F6	Charger Control information
0x6F7	Precharge status, 12V status
0x6F8	Minimum / Maximum cell voltage
0x6F9	Minimum / Maximum cell temperature
0x6FA	Battery pack voltage & current
0x6FB	Battery pack status
0x6FC	Fan & 12V supply status
0x6FD	Extended battery pack status
0x6FE - 0x6FF	Reserved, currently unused
0x7F0 - 0x7F4	Reserved for bootloader triggering and data transfer (these IDs do <u>not</u> move with a change in BMU CAN base ID)

4   DATA FORMAT

The 8-byte data on the CAN bus is formatted using a 'union' overlay structure in the BMS CMU microcontroller firmware. NOTE: this depends on the byte ordering and word size of the CPU, and is not portable across architectures – however the byte order is correct for both the MSP430 microcontroller in the CMUs and BMU and the x86 CPU in a PC.

The structure is defined on the (16-bit) MSP430 as follows:

```
typedef union _group_64 {
        float data_fp[2];
        unsigned char data_u8[8];
        char data_8[8];
        unsigned int data_u16[4];
        int data_16[4];
        unsigned long data_u32[2];
        long data_32[2];
} group_64;
```

Data values data_u8[0] through to data_u8[7] are the bytes transmitted on the CAN bus.

5 CAN PACKET FORMATS

5.1 BMU HEARTBEAT AND SERIAL NUMBER

This packet is transmitted by the BMU and allows it to be located and identified on the CAN bus. It contains an identifying string, and the serial number of the BMU device.

CAN ID	0x600
Interval	*1 Hz*
data_u32[0]	v5 and later: Device ID: 0x00001000
	v4 and earlier: ASCII ID string: 'T', '0', '6', '7'
data_u32[1]	Device serial number as programmed at the factory

5.2 CMU STATUS, TEMPERATURE AND VOLTAGE TELEMETRY

These packets are transmitted by the CMUs and contain the telemetry values from the cells. The BMU will relay these packets through to the vehicle CAN bus if configured to do so – this is enabled by default.

Each CMU transmits a group of three packets, with the ID of the first packet being granted by the BMU during the ID request/grant phase at startup.

CMU 1 will be assigned an ID of 0x601, and will therefore transmit telemetry on 0x601, 0x602, and 0x603. CMU 2 will be assigned an ID of 0x604, CMU 3 will be assigned to 0x607, and so on.

Reported cell voltages also indicate measurement status. Measurements with a high confidence (both redundant measurement channels agree closely) will be reported as a reading in mV. Readings where the redundant channels disagree ('trust' error) will be reported as **negative mV**, with the reported number coming from the high-accuracy / slow-speed converter channel.

Note that with the current CMU firmware, during very rapid voltage changes (high charge/discharge current pulses), the channels may report as a trust error for a few update cycles due to the sampling timing for the two measurement channels not being simultaneous.

Readings with *all* cell voltages reported as negative indicate that either the 3.3V power supply voltage or one of the A/D reference voltages is out of specification.

A CMU reporting any of these conditions (-ve mV) will potentially be reporting erroneous readings. A cell reporting a -ve mV reading will not balance that cell, to avoid balancing at a potentially incorrect voltage. Any cell reporting as a -ve mV reading should be flagged in the higher-level system and dealt with appropriately. It should not necessarily trigger an immediate system shutdown, but should probably be latched and indicated to the user to seek servicing promptly.

Readings from cells above the number defined by the BMU to each CMU during startup are sent as **-32768mV** (maximum negative) to indicate a 'cell not present / not monitored'. A cell that is read by the CMU as having voltage present, when that cell has been configured as 'not present', will report as **-32767mV** to indicate a possible 'extra cell' fault. These cells should not be included in any min / max / average cell voltage calculations.

CAN ID	0x601, 0x604, 0x607, etc
Interval	*Approx 1 Hz (CMU timebases are not synchronised and will drift relative to each other, and to actual time)*
data_u32[0]	CMU serial number (allocated at factory)
data_16[2]	PCB temperature (1/10th °C)
data_16[3]	Cell temperature (1/10th °C)

CAN ID	0x602, 0x605, 0x608, etc
Interval	*Approx 1 Hz*
data_16[0]	Cell 0 voltage (mV). +ve = OK, -ve = channel mismatch
data_16[1]	Cell 1 voltage (mV). +ve = OK, -ve = channel mismatch
data_16[2]	Cell 2 voltage (mV). +ve = OK, -ve = channel mismatch
data_16[3]	Cell 3 voltage (mV). +ve = OK, -ve = channel mismatch

CAN ID	0x603, 0x606, 0x609, etc
Interval	*Approx 1 Hz*
data_16[0]	Cell 4 voltage (mV). +ve = OK, -ve = channel mismatch
data_16[1]	Cell 5 voltage (mV). +ve = OK, -ve = channel mismatch
data_16[2]	Cell 6 voltage (mV). +ve = OK, -ve = channel mismatch
data_16[3]	Cell 7 voltage (mV). +ve = OK, -ve = channel mismatch

5.3 PACK STATE OF CHARGE (SOC)

This packet is transmitted by the BMU to show the current state of charge during normal pack operation.

CAN ID	0x6F4
Interval	*1 Hz*
data_fp[0]	SOC (Ah). Shows Ah consumed from the pack. 0 = Full, and counts upwards towards the user-set pack capacity number as Ah are used. Resets to 0 when max cell reaches balance threshold.
data_fp[1]	Shows the SOC as a percentage. 100% = full, 0% = empty

5.4 PACK BALANCE STATE OF CHARGE (SOC)

This packet is transmitted by the BMU to show the current state of cell mismatch during balancing at top of charge.

CAN ID	0x6F5
Interval	*1 Hz*
data_fp[0]	Balance SOC (Ah). Shows Ah supplied to the pack since the first cell began balancing. This number will continue to count up until all cells in the pack are balancing, therefore showing the Ah mismatch that has been corrected during this balancing session.
data_fp[1]	Shows the balancing SOC as a percentage, in other words, the percentage mismatch between cells this session.

5.5   CHARGER CONTROL INFORMATION

This packet is transmitted by the BMU to allow an external charger to control itself based in detailed cell information from the BMS, without having to know about the various user-configured cell setup parameters.

CAN ID	0x6F6
Interval	10 Hz
data_16[0]	Charging cell voltage error (mV). This value is the user-configured "Balance Threshold" voltage minus the maximum cell voltage. The charger should run a charge current control loop to try and bring this value to 0mV. An Integral type control loop is suggested.
data_16[1]	Cell temperature margin (1 /10th °C) This value is the maximum cell temperature minus the user-configured "Maximum Cell Temperature" limit. The charger should reduce charge current such that this value will never reach 0, as the BMS will disconnect the pack if the maximum cell temperature is exceeded. A Proportional type control loop is suggested.
data_16[2]	Discharging cell voltage error (mV) This value is the user-configured "Zero SOC Threshold" voltage minus the minimum cell voltage. This value can be used by devices that are discharging the battery (eg motor controllers in vehicles) to gradually limit their consumption as the minimum cell approaches being fully discharged.
data_u16[3]	Total pack capacity (Ah) This value can be used by the charger / discharger to calculate control loop gain constants for the installation. It is simply the user-set configuration value rounded to the nearest Ah.

5.6   PRECHARGE STATUS

This packet is transmitted by the BMU to indicate the current state of the precharge system.

CAN ID	0x6F7
Interval	1 Hz, and on each precharge state change
data_u8[0]	Precharge contactor driver status 0x01 = Error status of contactor 1 driver (0 = OK, 1 = error) 0x02 = Error status of contactor 2 driver 0x04 = Output status of contactor 1 driver (0 = Off, 1 = On) 0x08 = Output status of contactor 2 driver 0x10 = 12V contactor supply voltage OK (0 = Fault, 1 = OK) 0x20 = Error status of contactor 3 driver 0x40 = Output status of contactor 3 driver 0x80 = Unused

data_u8[1]	Precharge state (in order of normal appearance when starting)
	0 = Error
	1 = Idle
	5 = Enable Pack
	2 = Measure
	3 = Precharge
	4 = Run
data_u16[1]	12V contactor supply voltage, mV (only on v4 or earlier BMU)
	for v5 or later BMU, refer to binary bit 0x10 in data_u8[0]
data_u16[2]	Unused, reports as 0x0000
data_u8[6]	0x01 = Precharge timer elapsed (Don't care if timeout disabled)
	0x00 = Precharge timer not elapsed
data_u8[7]	Precharge timer counter (10ms per count)

5.7   MINIMUM / MAXIMUM CELL VOLTAGE

This packet is transmitted by the BMU to show the highest and lowest voltage cells in the pack.

CAN ID	0x6F8
Interval	***10Hz, will likely move to higher frequency in future version***
data_u16[0]	Minimum cell voltage (mV)
data_u16[1]	Maximum cell voltage (mV)
data_u8[4]	CMU number that has the minimum cell voltage
data_u8[5]	Cell number in CMU that is the minimum cell voltage
data_u8[6]	CMU number that has the maximum cell voltage
data_u8[7]	Cell number in CMU that is the maximum cell voltage

5.8   MINIMUM / MAXIMUM CELL TEMPERATURE

This packet is transmitted by the BMU to show the highest and lowest temperature cells in the pack.

CAN ID	0x6F9
Interval	*1Hz*
data_u16[0]	Minimum cell temperature ($1/10^{th}$ °C)
data_u16[1]	Maximum cell temperature ($1/10^{th}$ °C)
data_u8[4]	CMU number that has the minimum cell temperature
data_u8[5]	Unused, reads as 0x00
data_u8[6]	CMU number that has the maximum cell temperature
data_u8[7]	Unused, reads as 0x00

5.9 BATTERY PACK VOLTAGE / CURRENT

This packet is transmitted by the BMU to show the total pack voltage and current.

CAN ID	0x6FA
Interval	10Hz
data_u32[0]	Battery Voltage (mV)
data_32[1]	Battery Current (mA)

5.10 BATTERY PACK STATUS

This packet is transmitted by the BMU to show the status of the overall pack.Battery Pack Status

This packet is transmitted by the BMU to show the status of the overall pack.

CAN ID	0x6FB
Interval	1Hz
data_u16[0]	Balance voltage threshold – rising (balance resistor turns on)
data_u16[1]	Balance voltage threshold – falling (balance resistor turns off)
data_u8[4]	Status Flags 0x01 = Cell Over Voltage 0x02 = Cell Under Voltage 0x04 = Cell Over Temperature 0x08 = Measurement Untrusted (redundant channel mismatch) 0x10 = CMU Communications Timeout (lost CMU) 0x20 = Vehicle Communications Timeout (lost EVDC) 0x40 = BMU is in Setup mode 0x80 = CMU CAN bus power status Present for backwards compatability with older software For newer software, please view the flags in the extended battery pack status ID packet
data_u8[5]	BMS CMU count
data_u16[3]	BMS BMU Firmware Build Number

5.11 BATTERY PACK FAN STATUS

This packet is transmitted by the BMU to show fan speeds and 12V power consumption.

CAN ID	0x6FC
Interval	1Hz
data_u16[0]	Fan speed 0 (rpm)
data_u16[1]	Fan speed 1 (rpm)
data_u16[2]	12V current consumption of fans + contactors (mA)
data_u16[3]	12V current consumption of CMUs (mA)

5.12   EXTENDED BATTERY PACK STATUS

This packet is transmitted by the BMU to show extended pack status data.

CAN ID	0x6FD
Interval	*1Hz*
data_u32[0]	Status Flags 0x00000001 = Cell Over Voltage 0x00000002 = Cell Under Voltage 0x00000004 = Cell Over Temperature 0x00000008 = Measurement Untrusted (channel mismatch) 0x00000010 = CMU Communications Timeout (lost CMU) 0x00000020 = Vehicle Communications Timeout (lost EVDC) 0x00000040 = BMU is in Setup mode 0x00000080 = CMU CAN bus power status 0x00000100 = Pack Isolation test failure 0x00000200 = SOC measurement is not valid 0x00000400 = CAN 12V supply is low - about to shut down 0x00000800 = A contactor is stuck / not engaged 0x00001000 = A CMU has detected an extra cell present
data_u8[4]	BMU Hardware version
data_u8[5]	BMU Model ID
data_u16[3]	Unused

5.13   RESERVED IDS

Do not transmit data on reserved IDs, as this may trigger configuration or bootload events and mode changes.

6   INCOMING CAN PACKETS

The BMU is expecting regular (>5Hz) CAN packets from Tritium's EV Driver Controls, with the current state of the switches. Specifically, it is looking for the current state of the IGNITION RUN and IGNITION START switches, to use in controlling the precharge state machine.

The base ID of the EVDC can be set in the BMS configuration software, with the switch position packet being sent at +5 above the base ID. By default the EVDC base ID is 0x500, making the switch position ID at 0x505.

6.1   EV DRIVER CONTROLS SWITCH POSITION

This packet is transmitted by the EVDC to show switch status.

CAN ID	0x505
Interval	*10Hz*
data_u16[0]	0x0020 = Ignition Run
	0x0040 = Ignition Start
data_u16[1]	Unused
data_u16[2]	Unused
data_u16[3]	Unused

7   REVISION RECORD

REV	DATE	CHANGE
1	7 July 2011	Document creation (JMK)
2	25 February 2013	Added SOC, ground fault, charger control, fan speeds, extended flags, additional contactors, switch positions (JMK)