CANID\_CMD\_BMS\_MISCsmR

Battery module BMS nodesBattery module BMS node “**MISC**ellaneous **s**tring & **m**odule **R**esponse”

payload layout

11/06/2021

**Initiator** of command:

EMC (or pc) sends and module(s) respond with requested reading

CANID\_CMD\_BMS\_MISCQ'

payload [0] U8: Module identification

[7:6]

11 = All modules respond

10 = All modules on identified string respond

01 = Only identified string and module responds

00 = spare; no response expected

[5:4] Battery string number (0 – 3) (string #1 - #4)

[3:0] Module number (0 – 7) (module #1 - #16)

payload [1] U8: Command code

0 = reserved for heartbeat

1 = status

2 = cell voltage: calibrated

3 = cell voltage: adc counts

4 = temperature sensor: calibrated

5 = temperature sensor: adc counts

6 = isolated dc-dc converter output voltage

7 = charger hv voltage

8 = Hall sensor: calibrated

9 = Hall sensor: adc counts

10 = Highest cell voltage

11 = Lowest cell voltage

12 = FET on/off discharge bits

13 = Turn on Dump FET for no more than ‘payload [3]’ secs

14 = Turn off Dump FET

15 = Enable Heater mode to ‘payload [3] temperature

16 = Turn Heater mode off.

payload [2] U8: item #

for command codes 1 & 2: cell number (1-18)

for command codes 3 & 4: thermistor number (1-3)

for command codes 13 & 14: bits applicable to modules on string

payload [3-6] four bytes (uint32\_t, int32\_t, or float)

**Response** to command:

29b CAN ID maps to string and module.

CANID\_CMD\_BMS\_MISCsmR

where:

s = string number

m = module within string

[7:6]

11 = All modules respond

10 = All modules on identified string respond

01 = Only identified string and module responds

00 = spare; no response expected

[5:4] Battery string number (0 – 3) (string #1 - #4)

[3:0] Module number (0 – 15) (module #1 - #16)

payload [1] U8: Command code

0 = Heartbeat

… as defined above

payload [2] U8:

cell or item number as defined above

payload [3] U8:

spare for additional data

payload [4:7] X4: four bytes (uint32\_t, int32\_t, float)

**Narrative:**

Unique identification of a CAN msg requires specifying the winch, string, battery module on the string. There will be more than one CAN id for each module which makes the potential number of CAN ids rather large. To minimize the number of CAN ids per battery module the msg payload carries command and identification bytes which specify the contents of the payload.

Since the CAN msgs are a low priority with respect to the active launch CAN msgs, and to simplify the management of CAN ids, 29b CAN addresses are used with the low order bits used to specify winch, string, and module. This is only needed for CAN msgs that originate from the module.

The module will respond to a command CAN msg sent by the EMC (or PC). The command specifies which strings and modules are to respond, with the options: all, specified string, specified string and module.

There is one case where the module will generate a heartbeat msg if there has not been a command msg received within a time limit.

Winch number could be added to logging files and eliminate the identification in the CAN msgs, however unlike fairleads and level winds assemblies, battery modules might be swapped between winches, and cell performance tracking data would be confounded. Therefore, provision for winch identification is included.