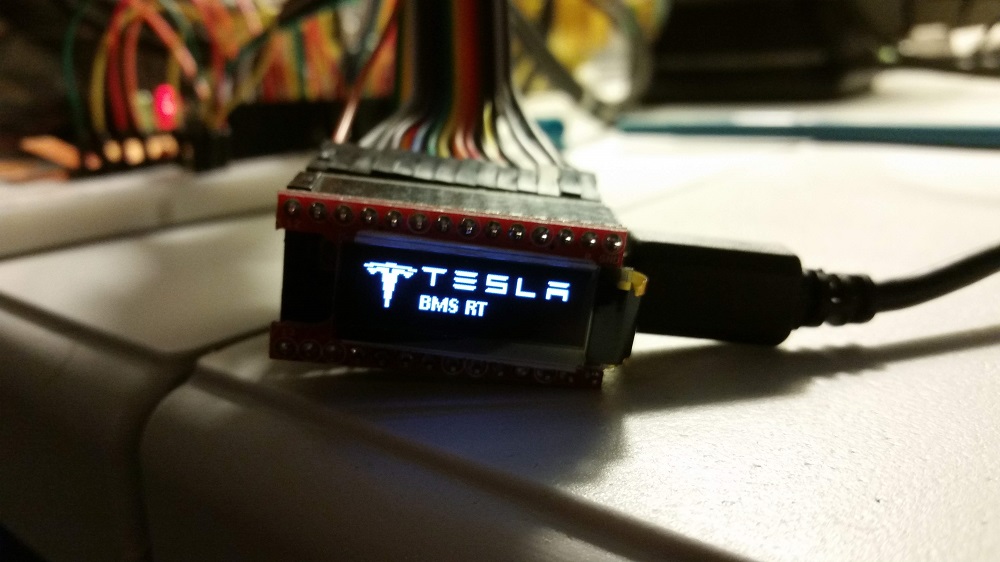
Tesla BMS BL V1.0

Tesla Battery Management System Big Loop Version 1.0



Contents

[Overview 3](#_Toc13773448)

[Installation 3](#_Toc13773449)

[Operation 4](#_Toc13773450)

[OLED display interface 4](#_Toc13773451)

[Tesla BMS BL and eSidewinder splash screens 4](#_Toc13773452)

[Faults 4](#_Toc13773453)

[sFaults (Sticky Faults) 5](#_Toc13773454)

[VBat and TBat 5](#_Toc13773455)

[VClo VChi 5](#_Toc13773456)

[VCmin VCmax 5](#_Toc13773457)

[VCdiff Tmax 5](#_Toc13773458)

[State 5](#_Toc13773459)

[Firmware Update 5](#_Toc13773460)

[Command Line Interface (CLI) 6](#_Toc13773461)

# Overview

The Tesla BMS BL was conceived in the scope of a 1981 Sidewinder conversion to electricity. Although the main function of the controller is to monitor and balance the cells in the Tesla model S modules, it also performs the following functions:

* Controls the Electric Vehicle Charger Controller (EVCC) to initiate a charge.
* Monitors up to two battery compartments for flooding (coolant or water) using water sensors.
* Logs various faults obtained from different sources.
* Controls the coolant pump for the cooling of the tesla modules.
* Monitors and manages the 12V battery by controlling a DC2DC converter.

The controller is optimized to draw very little power from the power pack when in standby.

# Installation

The installation of the Tesla BMS BL must be done according to the following pinout with consideration that the controller functions at 3.3V and is 5V tolerant. Its serial port can talk directly with the Tesla BMS Modules without a voltage converter.

The firmware sources are at <https://github.com/Vigeant/teslaBMSBL>

The pinout configuration is set in the CONFIG.h file in the project.

Typical IO circuitry to convert and protect the controller:

[insert IO diagram here]

# Operation

The Tesla BMS BL implements a finite state machine that controls all outputs in a predictable way.

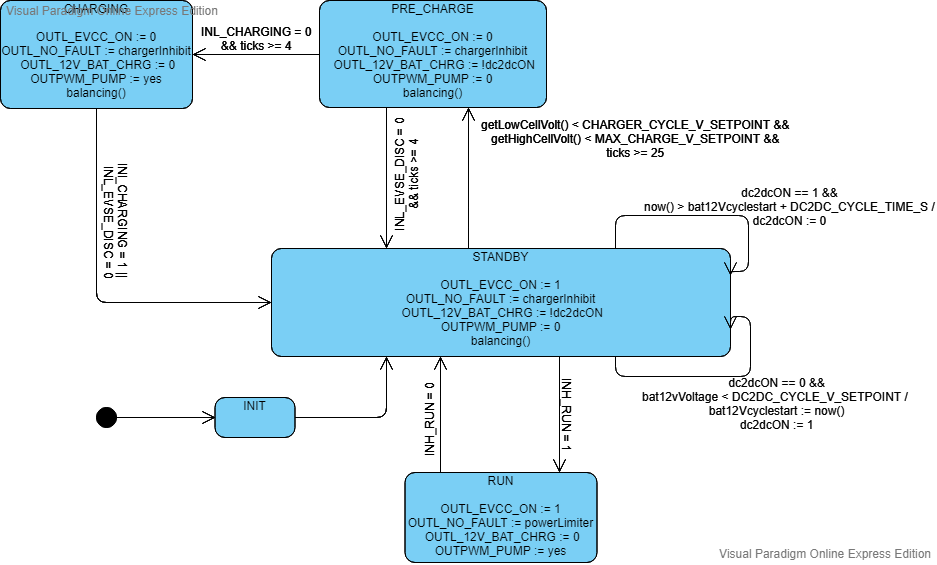


Figure Tesla BMS BL state machine

The state machine is implemented in the controller.cpp file. For simplicity, it is implemented as a mix of a meeley and a moore state machine. The content of the states are the actions and outputs and the transitions typically show the trigger for this transition. When a “/” is present it is used to separate the trigger from the actions on a transition.

The state machine is run at every tick and the default tick rate is 5Hz or every 200ms. The exception is when the controller is in low power mode in the standby state where the controller will sleep for 5 seconds before taking another action.

## OLED display interface

The OLEd display cycles through different formats every 3 seconds to display a quick health check of the modules.

### Tesla BMS BL and eSidewinder splash screens

This is the first format and along with the eSidewinder splash indicates the beginning of the formats cycle.

### Faults

If the controller detects a fault, it will display a format indicating 1 letter fault codes:

|  |  |
| --- | --- |
| Fault Code | Definition |
| A | Modules Fault Loop |
| B | Battery Monitor Fault |
| C | BMS Serial communication Fault |
| D | BMS Cell Over Voltage Fault |
| E | BMS Cell Under Voltage Fault |
| F | BMS Over Temperature Fault |
| G | BMS Under Temperature Fault |
| H | BMS 12V Battery Over Voltage Fault |
| I | BMS 12V Battery Under Voltage Fault |
| J | BMS Water Sensor 1 Fault |
| K | BMS Water Sensor 2 Fault |

Figure faults table

### sFaults (Sticky Faults)

When a fault is resolved, the fault will disappear from the faults format but all faults that occurred since the last reset will be displayed in the sticky faults format.

### VBat and TBat

VBat Displays the total voltage of the Battery.

TBat display the average Temperature of the battery.

### VClo VChi

VClo displays the voltage of the cell with the lowest voltage.

VChi displays the voltage of the cell with the highest voltage.

### VCmin VCmax

VCmin displays the lowest voltage a cell reached since the last reset.

VCmax displays the highest voltage a cell reached since the last reset.

### VCdiff Tmax

VCdiff displays the delta between the highest cell voltage and the lowest cell voltage. This is a good indicatore of how balanced the battery pack is.

Tmax displays the highest temperature reading of all the sensors.

### State

Displays the name of the state the controller is currently in accordance with Figure 1.

## Firmware Update

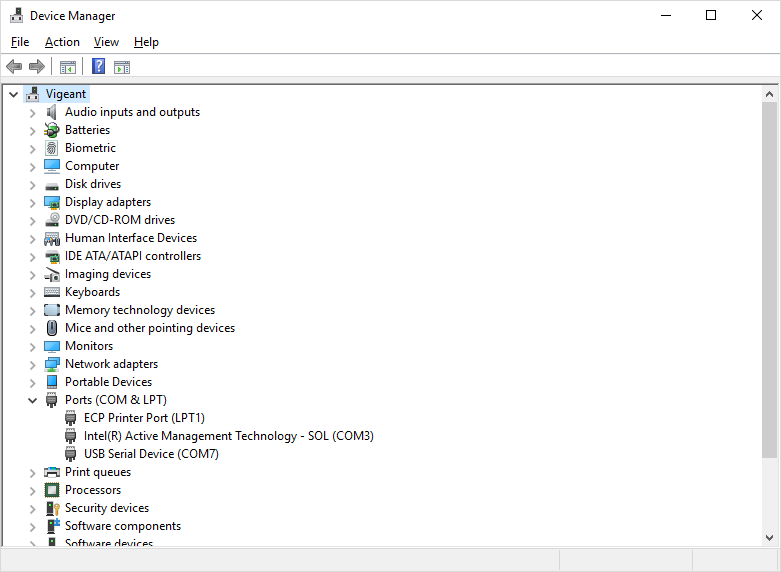
The firmware must be updated using the sources and the Arduino IDE tool chain for the Teensy 3.2.

## Command Line Interface (CLI)

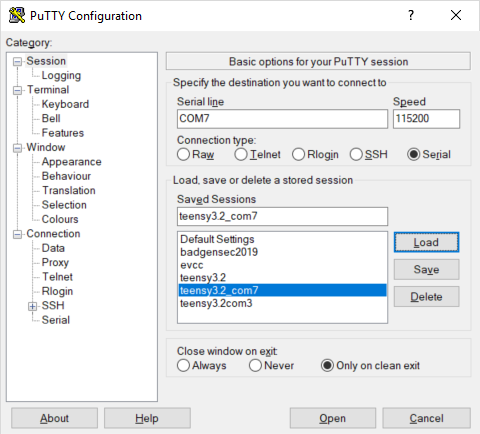
The command line interface offers a more detailed view of the battery packs to help investigate issues. It offers various verbose levels and three main functions.

To connect to the CLI you must:

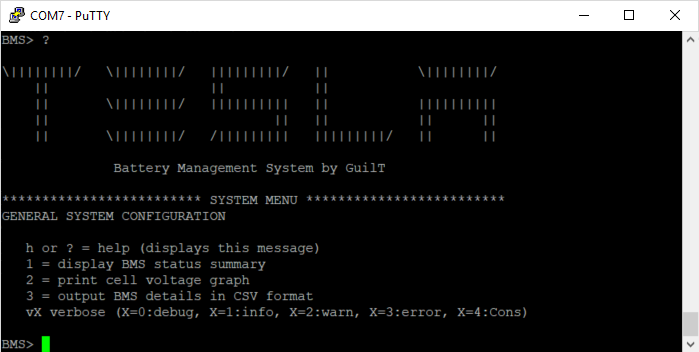
1. Connect a PC to the Teensy 3.2 via a usb cable.
2. Identify the COMX that was assigned to the USB to Serial adapter. In the screenshot below, it is COM7.



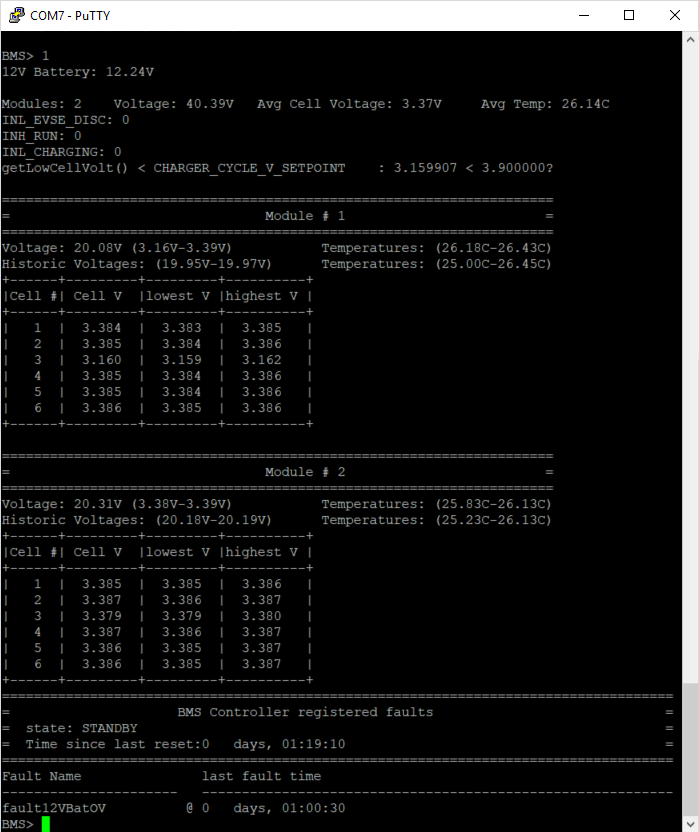
1. Using a terminal emulator of your choice, connect to the serial port using a baud rate of 115200. For example, if using putty, it would look as follows. The connection will fail when the board in sleeping. To ease the connection, the board is prevented from sleeping for 1 minute following a reset and also does not sleep when in any other state than standby. The controller will not sleep as long as a serial connection is established.



1. Once connected, you will be presented with a black screen and can press enter to show the prompt. Typing help or “?” will display the help for the CLI.

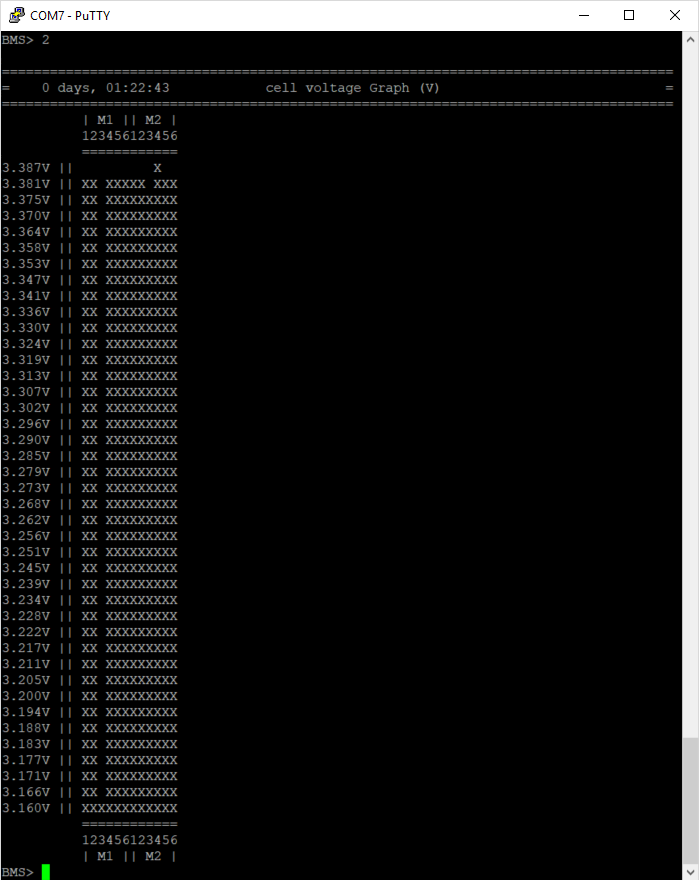


1. Here is an example of the “1” option when connected to two tesla modules:

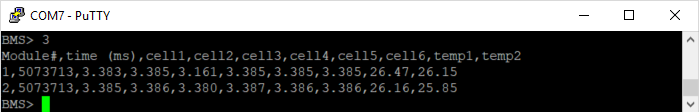


Notice how it provides all the detailed information about each cell in each module as well as the state of the other inputs and the 12V battery. It also provides the last time a fault occurred.

1. Option 2 provides an ascii art bar graph of the voltage of each cell. This is useful to get a quick glance at the balancing of the cells. Notice that the voltage is automatically scaled to cover the delta between the highest and lowest cell.



1. Option 3 simply displays the states of each cells and the two temperature sensors in CSV format for easy export into excel.



1. When done, simply close the putty window and the board will go back into normal operation sleeping in the standby state.