BMSSPV - Battery Management System for Solar Powered Vehicles applied to Técnico Solar Boat prototype

Sebastião Maria Antelo de Holbeche Beirão
sebastiao.beirao@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2021

Abstract

This paper focuses on the design, implementation and testing of a Battery Management System (BMS) for Técnico Solar Boat's (TSB's) solar-powered boats. Since 2015 TSB, a student's project from Instituto Superior Técnico develops manned competition boats powered exclusively by green energies designed to participate in international competitions that take place in Portugal, The Netherlands and Monaco.

The boat on which the developed BMS will be used has a 44.4 V Lithium-Ion Polymer (LiPo) battery, 6 m^2 of monocrystalline solar cells and a drive train composed by two BLDC motors.

The system being developed is a centralized BMS that monitors the battery pack and sends the acquired information to the boat's Controller Area Network (CAN) Bus as well as to a Graphical User Interface (GUI) if a computer is connected. Besides monitoring the battery, the BMS is also responsible for controlling the different relays within the battery box as well as deciding whether the battery is safe and consequently if the boat can be used or not. The final BMS has to respect a set of requirements imposed by the competition's regulations and others imposed by the team.
Keywords: BMS, Battery Management System, Solar Powered Boat, Electrical Vehicle, Lithium-Ion Polymer Battery

1. Introduction

In the past few years, not only due to environmental concerns, but also due to the instability in fuel prices, we have witnessed an increasing demand for Electric powered Vehicles (EVs). This demand is not only seen in terrestrial applications, such as cars, buses and even trucks, but also in the maritime and, more recently, aviation industries [1].

Regardless of the type of vehicle used, all of them share the same typical architecture composed by an energy storing unit, at least one electric motor and a power converter. This work will be focused mainly on the first one, the energy-storing unit, and more specifically on its BMS.

BMSs are real-time systems that control many functions vital to the safety and reliability operation of energy storing units in EVs. This includes tasks such as: monitoring temperature, individual cell voltage, current, battery health and State of Charge (SoC) and handling the communication with other system components. This system should be capable of identifying any potentially dangerous situation, such as over-voltage, and work in order to stop this faulty condition.

Given that the competition's regulations limit both the battery capacity and the area of the solar panels, the battery and its BMS have to be designed in such a way that all the energy available is used but, at the same time, the boat is able to finish the races. This is especially important during the endurance race, where we need to use all the available energy while making sure that the battery remains within its Safe Operating Area (SOA)[1]. For such a race, it is crucial to have an accurate estimate of how much energy there is left on the battery, i.e. the battery SoC.

1.1. Motivation and objectives

When TSB started, in 2015, we tried to develop our own BMS. However, problems with its development, some lack of knowledge and the limited time until our first competition forced us to search for a commercial alternative that would meet, at least partially, our needs.

Although this solution was used for competing for two years, it was far from perfect and required additional hardware to make it comply with the competition's regulations. To account for the missing

[1]The SOA is defined as the temperature, voltage and current conditions over which the cell or battery can be expected to operate without self-damage or degradation.

functionalities, we ended up developing a custom board. This board was named Battery Surveillance System and its main functionalities were: reading temperatures, controlling the solar panels' relays, telling the BMS how to control the charger and load and communicating with the BMS to retrieve its data and broadcasting it to the rest of the boat's systems, using a Universal Serial Bus (USB) serial protocol.

Although this solution proved to be functional and didn't cause any major problems to the boat's performance, except for the low reliability of the implemented serial protocol, but it was far from optimized. There was much room for improvement. Learning the problems we faced during the 2018 competitions, we wanted to increase the reliability of the boat's electrical system for the 2019 season. We completely redesigned and simplified this system. At the time, the complete BMS was composed of two separate boards, and the battery assembly was very disorganized due to all the cables. We wanted to simplify this. Besides these improvements, we also wanted to have more control over the battery and all the systems that were connected to the BMS. These were the main reasons that gave us the motivation to restart the development of our own BMS.

With that in mind, the objective of this work is to develop an all-new BMS to replace the previously used system. This new BMS should set a new standard in the project and be developed in such a way that allows future team members to easily make modifications to it, whether these are related to software or hardware. Hereto, special care must be taken to make sure it is well documented. Adopting a self-developed solution will create, within the team, the know-how needed to improve this critical boat system in future project iterations.

2. State of the Art

In the early ages of lithium battery technology development, BMSs were analog. This meant that hardware modification had to be made to modify parameters such as the cell's maximum or minimum voltage. With the technology improvements in Lithium-ion (Li-ion) cells throughout the years, the demand for more complex and feature-rich BMSs increased, and with it, the shift towards digital BMSs started. Figure 1 summarizes the main functions that most generic digital BMSs have nowadays. The heart of such BMSs is the Battery Stack Monitor (BSM) Integrated Circuit (IC), which is responsible for at least measuring the voltages of each cell. There is also some part responsible for temperature and current measuring, as well as electronically controlled switches to control the load and charger. Finally, there is also a Microcontroller Unit (MCU) which communicates with the BSM, acquires its data and broadcasts all the BMS data to other systems. The MCU is where the algorithm which is responsible for making sure that the battery pack is safe runs.

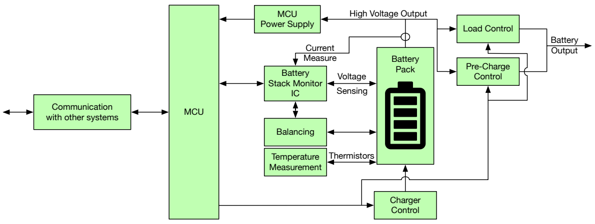


Figure 1: Generic BMS functional block diagram.

Nowadays, the majority of the BMSs available on the market are digital since they allow an easier (re)configuration of the critical parameters (voltage, temperature and current limits) while allowing the acquisition and transmission of valuable data, such as battery SoC, current, voltage and temperatures to other interfaces and systems. Additionally, in case there is any problem with the battery, most digital BMSs are capable of not only reporting this faulty condition but also identifying exactly where the fault happened [2]. Another important task of this type of BMS is to report how much energy is left in the battery, which is very important in TSB's application because it allows the team to use all the available energy during the competition.

2.1. BMS topologies

According to [2],there are four different BMS topologies: centralized, modular, master-slave and distributed. There is no clear choice regarding which topology should be used for a given application. It depends a lot on its specific needs: safety, cost (parts, assembly and maintenance), and reliability are determinating factors.

Except for the centralized topology, which is primarily used in low voltage systems, with few cells in series (typically not more than 15), all the others are more frequently used for higher voltage battery packs, such as the ones used in EVs.

3. Architecture

This chapter will cover the overall system architecture, mainly concerning the BMS itself, but to do so, we first need to specify the battery pack configuration. Although this configuration is essential for the BMS design, it is out of the scope of this work, and thus it will only be briefly summarized in Section 3.1. Other parts of the boat's electrical system that are considered to be relevant for designing the BMS will also be addressed.

Although the team can define most of the BMS requirements, there are a few constraints imposed by the competition regulations, like the obligation to provide a way to monitor individual cell voltages during the battery tests [3]. This requirement implies that the BMS must be digital since such requirement is impossible to satisfy with an analog BMS.

3.1. Battery design

Regarding the battery design, the only constraints imposed by the regulations besides the need to use commercially available cells are the maximum system voltage and the maximum energy stored onboard. The former should not be greater than 52 V, which for Li-ion and LiPo cells leads to a maximum of 12 cells in series, while the latter is limited to 1.5 kWh.

The battery is composed by 24 LiPo pouches with a 3.7 V nominal voltage and a capacity of 16.8 Ah, arranged in a 12s2p[2] configuration. The cells' tabs are laser welded to nickel-plated copper bus bars. This not only improves the connection reliability but also decreases the contact resistance and the assembly complexity.

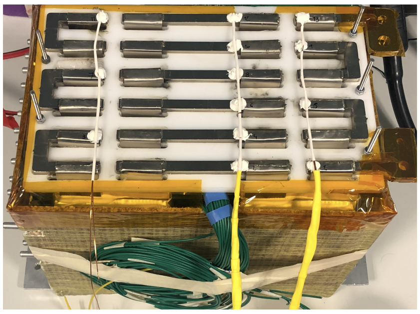


Figure 2: TSB's battery pack.

Compared to the past battery packs, this new battery implements a new cooling system for the cells using ultra-fine water-cooling plates between each cell. Doing so helps to maintain the battery temperature below 45 °C, which is the maximum allowed temperature for charging, ultimately improving the boat's performance.

3.2. BMS requirements

As already mentioned, the BMS must comply with the requirements imposed by the competition regulations, which for our class and according to [3] are: Monitor cell aggregates and battery voltage, temperature, charge and discharge currents; be able to shut the high power system down when necessary; control too high currents.

All the measurements of the above parameters should have a minimum refresh rate of 2 Hz. The BMS must also be able to balance the battery since, according to [3], the battery must be balanced when starting the capacity test. Moreover, this is a necessary task in order to use all of the battery's energy. Besides these requirements imposed by the regulations, the team also wants the following: high accuracy cell voltage measurements; measure at least 16 temperatures (with a refresh rate of at least 0.5 Hz); sense the battery voltage at three different points; capability to control, at least three relays; control the precharge of the motor controllers' capacitors; calculate battery information (such as SoC and number of cycles); control the battery fans; low-power consumption, and high standby life (at least years without draining the battery); be capable of storing user-configurable parameters in non-volatile memory; Watchdog Timer functionality.

Given that the Maximum Power Point Trackers we currently use don't have a communication interface and that the BMS will be nearby them, we also want the BMS to be able to: measure the voltage of the five solar arrays (with a refresh rate of at least 0.5 Hz); measure the total current coming from the solar panels.

The proposed BMS should communicate with the rest of the boat's systems with a CAN bus and broadcast all the acquired data to the boat's CAN Bus so that it can be displayed to the pilot and logged on the boat's CAN logger. The bus should run at 1 Mbit s^{-1} given that this is the boat's CAN Bus baud rate.

Apart from the communication with other boat systems, the BMS should also be able to communicate with a computer that can be used to configure it (change parameters such as cut-off voltage and temperature limits, for example) and also to graphically display the BMS's data with the help of a GUI. For this task, an USB serial connection should be used.

Finally, the BMS should log relevant events to an onboard micro-SD card so that errors can be checked later, have a buzzer to signal any fault(s) and have a Real Time Clock (RTC) to keep track of time.

3.3. BMS architecture

Although all the topologies presented in Section 2.1 can be used for the proposed system, one of them must be chosen. Bearing in mind that our battery design is composed of a single stack, where cells are first connected in parallel and, only after that, in series in order to produce the most compact design and reduce the amount of measuring points and the overall system complexity, the topology that makes the most sense is the centralized.

With both the requirements and the topology already defined, it is time to define the system's general architecture. The main task of the BMS is to measure the cells' voltage, the pack voltage, the pack current and the cells' temperatures. The volt-

[2] XsYp stands for X cells in series and Y in parallel. Hereinafter this parallel will be referred to as cell aggregate.

age measurement task can be done in multiple different ways. However, the most common way of doing it is by using a BSM IC.

In order to monitor the battery temperature, thermistors must be added to the battery assembly. These thermistors may be multiplexed to reduce the total number of analog pins needed to measure all the temperatures. A current transducer or shunt must be installed to measure the current flow to and from the battery. To control too high currents, the BMS must be capable of opening the load relay. The requirements related to the measurement of both battery voltages and cells' voltages are satisfied by the BSM itself. Regarding the control of the precharge relay, a comparator should be used to determine when the precharge is finished.

An optocoupler should be used to provide isolation between the battery and the low voltage General Purpose Input Output (GPIO) pins of the MCU to sense the battery voltage. The measurement of the voltage of the five solar arrays may also be multiplexed so that only one analog pin needs to be used. Another current transducer or shunt must be used to measure the solar panels' current.

Regarding MCU, apart from the already mentioned 1 Mbit s^{-1} Serial Peripheral Interface (SPI) bus, we need at least 15 GPIO pins (4 relays, 1 buzzer, 1 for fans, 2 for Chip Select pins for SPI, 3 battery voltage sense pins, 3 multiplexer channel select pins and one precharge end detector). Finally, we need up to 9 analog inputs for temperature and solar panels' voltage measurements, depending on whether the ones from the BSM are used or not.

4. Implementation

In this chapter, the components and modules necessary to achieve the previously mentioned BMS architecture are presented both in what concerns the hardware and software sides of the system.

The first step required to advance with the design of the proposed BMS is to choose which BSM will be used. In [4] there is a list that compiles the most common BSMs. This list divides the chips by manufacture and classifies them into three different categories: not recommended, recommended and highly recommended. The ICs that are listed as not recommended and recommended were discarded as well as chips that are not available in the main distributors. Considering the requirements and the number of cells to be monitored, the list can be reduced to the following two candidates: LTC6811 from Analog Devices and bq76PL455 from Texas Instruments.

Analog Devices' solution was chosen. Nonetheless, there are two versions of this IC: the LTC6811-1 and the LTC6811-2. By reading the datasheet [5] of the IC, Analog recommends the use of the LTC6811-2 when a single device is to be used since it requires fewer external components and consumes less power, especially when the communication interface is configured as a 4-wire SPI. Given that the chosen topology is centralized and each IC can monitor a maximum of 12 cells in series, and our battery has a 12s2p configuration, the LTC6811-2, hereinafter referred to as BSM is the one that will be used.

Taking into consideration the requirements previously defined, as well as the choice made in what concerns the BSM IC the next step that should be taken in order to proceed with the development of the BMS is to choose which MCU will be used. By analyzing the software complexity, a 16-bit MCU would be able to handle the task. Considering that the team may want to add new functionalities to the proposed BMS in future project iterations, a 32-bit MCU provides a more future-proof solution that prevents the need to adapt the software from 16-bit to 32-bit in the future. Besides, nowadays, 32-bit MCUs are considered the industry standard since most embedded system R&D effort is focused on 32-bit cores, and thus both architectures can achieve similar power consumptions [6]. There are two main architectures available for 32-bit MCUs: AVR and ARM. The latter is the more common and most advanced. Given that, the ARM architecture is the choice.

Different options of the ARM architecture were considered, but ultimately, the choice was made based on what the team uses on the rest of the systems. Within the team, all systems are based on Teensy boards. Their microcontrollers are all based on the ARM architecture, manufactured by NXP, ranging from a Cortex-M0+ to a Cortex-M7. [7] outlines the most relevant differences between the Teensy boards. After considering the various options and the needs of the proposed BMS, the choice went towards the Teensy 3.2.

Inspired by our colleagues from FST Lisboa [8], we decided to build a BMS that would be on the top of the battery. The computer aided design for this approach can be seen in Figure 3.

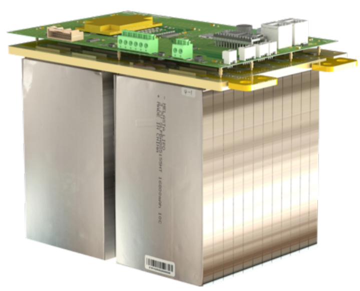


Figure 3: SR-02 battery assembly with spring loaded pins connecting the BMS to the cells.

Figure 4 displays the most relevant blocks of the proposed BMS. Starting from left to right, we have both communication interfaces of the system to the exterior followed by the MCU. The MCU communicates over SPI with the BSM and also controls the contactors with GPIO pins. The BSM is then connected to each cell aggregate of the battery pack to measure their voltage. The same connections are also used for balancing. The connection of the cells to these resistors is controlled by the balancing Metal Oxide Semiconductor Field Effect Transistors (MOSFETs), which are controlled by the BSM. The thermistors that measure the cells' temperature are connected to multiplexers which are connected to the BSM GPIO pins which convert the analog voltage measured to a temperature value. Finally, there are two contactors for the charger and load and one relay for the precharge circuit.

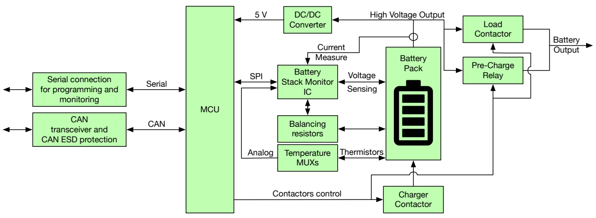


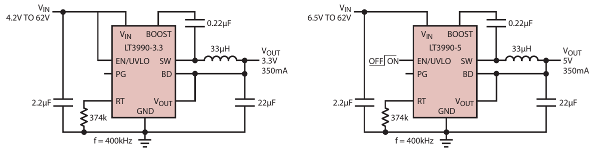
Figure 4: BMSSPV functional block diagram.

4.1. Power supply

For providing the different voltages needed, two Linear Technology's LT3990 step-down regulators were used. One was configured to supply 5 V while the other supplies 3.3 V. These regulators have a maximum quiescent current from $V_{IN} = 1.2\,\mu A$ when in sleep mode, which makes them perfect for low power applications. The sleep mode is controlled by the $EN/UVLO$ pin, which, in the case of the 5 V regulator should be connected to the LTC6811's $DRIVE$ pin, automatically putting LT3990 into its lower power mode when the LTC6811 is in the $SLEEP$ state.

The 3.3 V regulator should always be powered; otherwise, the MCU would be turned off. Nonetheless, given that there are still other components onboard supplied by the 3.3 V bus that don't need to be powered while the system is sleeping, like the micro-SD card, a P-channel MOSFET was added in between the supply voltage and these components.

The schematic for this power supply is shown in Figure 5, and it followed the datasheet recommendation. In LT3990-3.3's case, the $EN/UVLO$ pin was connected to V_{IN} since this Direct Current to Direct Current (DC-DC) should always be powered. In contrast, in the LT3990-5's case, this pin was connected to the LTC6811 $DRIVE$ pin, as suggested on the datasheet.



(a) (b)

Figure 5: Power supply schematic for both LT3990-3.3 (a) and LT3990-5 (b), extracted from [9].

4.2. Cell monitoring

The BSM is, without doubt, the most important Hardware part of the BMS. It is the part responsible for measuring the voltage of each cell aggregate, identifying unsafe conditions (like cell Over-Voltage (OV) or Under-Voltage (UV)), measuring both battery and solar currents, as well as the temperature sensors connected to the cells. At the same time, the BSM also controls the balancing of the cell aggregates, thus allowing us to extract the maximum energy from it.

Regarding the cells' voltage measurement task, each cell aggregate is connected, as shown on Figure 6, to a fuse and then connected to the BSM's C_n inputs. Between each fuse and the BSM, there is an Resistor-Capacitor low pass decoupling filter that helps to reduce the measurement noise, especially in the fast conversion modes, and to reject potentially damaging high energy transients.

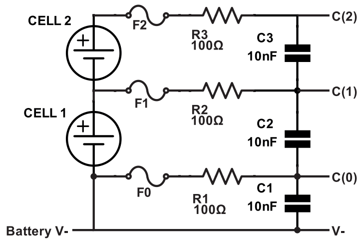


Figure 6: Differential capacitor filter connection scheme for the first two cells.

Concerning the communication part, there are two options to choose from, SPI or Analog Devices' proprietary isoSPI protocol. Using SPI reduces the number of components. Besides, Analog recommends that when only one BSM is needed, the LTC6811-2 should be used and configured to communicate over SPI in order to reduce power consumption. To achieve this, $ISOMOD$ and $A0$ to $A3$ pins should be connected to V^-.

4.3. Cell balancing

The last task of the BSM is to manage the balancing of the battery. As seen before in Section 3.2, this is not only desired, it is a mandatory feature. There are two options for balancing: passive and active. While the former is achieved by wasting the excess energy in heat, the latter redistributes the battery energy.

For this application, where there is plenty of time to charge the battery and solar panels are used, the extra weight and volume required to accommodate an active solution don't outweigh the disadvantages of passive balancing. Moreover, the battery needs to be balanced at the start of the competition, so an unbalancing situation is not expected to happen during the competition given the very low self-discharge current of LiPo batteries. With that said, the chosen solution is passive balancing.

For this, two options are possible: internal or external MOSFETs. The BSM's internal MOSFETs only allow a balancing current of 60 mA or less which, for the cells being used, the balancing time would be around 28 h for a 5 % SoC error. This is just the time needed to balance one cell and assuming that the BSM would not overheat during this process. For this reason, the option with external MOSFETs is the choice. This way, a higher balancing current can be used without overheating the BSM.

In order to balance a cell aggregate, one PMOS transistor, Q_1, and a load, R_1 to R_3, are needed. The transistor's gate is connected to S_n while the source and drain are connected to the positive, C_n, and negative, C_{n-1}, terminals of the cell aggregate, respectively. The balancing load, in this case, a resistor, should be placed between the drain and the negative side of the aggregate. The circuit for this solution is shown in Figure 7. This circuit is replicated 12 times, one for each series of the battery pack.

A balancing current of around 520 mA. This results in a balancing time of 3.2 h for a 5 % SoC error which is 89 % lower than what can be obtained while using the internal MOSFETs. Besides, multiple cell aggregates can be balanced together, thus reducing the overall balancing time even further. To achieve such a balance current, a series of three resistors with 2.7 Ω will be used.

Care must be taken while choosing these resistors to make sure their power rating is respected. The worst-case scenario is when balancing a fully charged cell. The power dissipated in each resistor is given by

$$P = R \times I^2 = 2.7\,\Omega \times 0.52\,\mathrm{A}^2 = 730\,\mathrm{mW} \quad (1)$$

so resistors with a power rating of at least 1 W must be used. It is also important to ensure a proper way for the heat to be dissipated away from the resistors to ensure that neither the Printed Circuit Board (PCB) nor the resistors overheat. For this, a heatsink placed on top of the resistors is to be used.

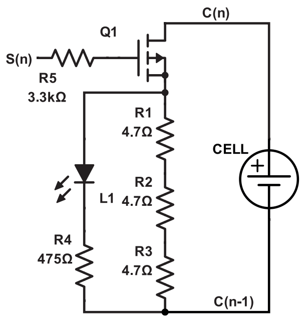


Figure 7: Balancing circuit generic schematic.

4.4. Current measurement

To measure the battery and solar current, two options are possible. Either Hall effect current transducers or current shunt can be used. The choice was the IVT-S from Isabellenhütte, which is capable of not only measuring the current by using a shunt resistor, but it can also measure up to three voltages referenced to the battery ground. This device includes an onboard energy counter which will be very useful for calculating the battery SoC. Finally, the device is capable of sending all the acquired data to the boat's CAN Bus. The IVT-S is powered by the boat's 24 V bus, which means that when the boat is off, the sensor will automatically be off, and thus it will not consume any power.

By having access to the current measurement, the BMS can detect an over-current situation. To satisfy the requirements, the BMS must not only detect this situation but also needs to stop it. To do so, high-current contactors are used.

4.5. Temperature measurement

Considering the new battery water cooling system presented in Section 3.1 and to make it possible to evaluate the performance of the aforementioned system, a total of 40 external thermistors inputs were included. The idea behind this is to allow us to have one thermistor per cell (24) and another one per bus bar (13), totalizing 37 thermistors. By placing the cell's thermistors near the bottom of its body and another near its tab on the bus bars, we will be able to evaluate the cooling performance and decide whether more cooling is needed in future iterations and where. In order to read the 40 thermistors, five 8-channel multiplexers were added to the circuit. The multiplexers' output is measured by the BSM's GPIO pins.

The other temperature measurements that are required are the ambient temperature, the balancing resistors temperature (2) and the BSM temperature. For the first two, Negative Temperature Coefficient (NTC) thermistors were added to the PCB. Regarding the BSM die temperature, this can be obtained by sending the `ADSTAT` command to the BSM so no temperature sensor is needed. The

Teensy analog pins measure the three NTC thermistors directly, given that there are no more inputs available on the BSM. The only drawback of this solution is that the thermistors should be powered with 3.3 V instead of 5 V, given that this is the maximum supported voltage for analog signals applied to the Teensy.

4.6. Sleep mode

There are essentially three situations that should cause the system to wake up: when the boat is turned on, when the USB cable is plugged into the Teensy, and periodically in order to check the battery status while it is not being used. To wake up the system when the boat is turned on, a new voltage sensing port was added compared to V1. This port is connected to the emergency switch. As soon as the switch is closed, the battery voltage enters the optocoupler and subsequently in one of the MCU pins.

For the USB wake up, the Teensy has a pin called V_{USB} which is connected to the USB cable power conductor. By connecting this pin to another digital pin of the MCU, we can also sense that the USB cable has been connected and thus wake up the system.

These pins must be chosen carefully, and it is important to pay attention to the Teensy's Snooze library documentation [10]. Here, all the different ways to wake up the Teensy are explained as well as the pins that can be used for pin-based wake-ups.

To wake the Teensy up periodically, the timer or alarm drivers of the Snooze library can be used, and no additional hardware components are needed.

4.7. Remaining sub-systems

In order to sense the battery voltage in specific points of the system, an optocoupler circuit was designed. This provides the necessary isolation between the battery voltage and the Teensy GPIO pins. To control the different relays and contactors, the BMS cuts the GND supply to the coils. The positive side of the coils is connected to the 24 V emergency DC-DC[3].

The battery fans are also controlled using an N-channel MOSFET acting as a low side switch like the relays using a Pulse Width Modulation (PWM) capable pin to drive the MOSFET's gate so that the speed of the fans can be controlled.

To measure the voltage of the solar arrays, an approach similar to the one used for measuring temperatures was used. The measuring points are connected to a multiplexer (with proper signal conditioning), and then its output is connected to one of Teensy's analog pins.

Regarding the communications aspect of the BMS, both CAN Bus and USB serial are needed. For CAN the only thing needed is to add a suitable CAN transceiver to the board. On the other hand, using the USB serial doesn't require any additional hardware, given that Teensy's USB programming port can also be used as a serial port.

The only three aspects of the requirements still not covered are the RTC, the buzzer and the micro-SD card. The Teensy 3.2 already as a RTC incorporated onboard, which only needs an external 32.768 kHz crystal to be added to the dedicated through-hole pins. The micro-SD is connected to the Teensy using its SPI port, which is shared with the BSM, except for the chip select line. Finally, the buzzer is driven by one of the Teensy's PWM pins through an NPN transistor.

4.8. Software

The main tasks the BMS's software has to accomplish are: acquire battery data (both from the BSM and other sensors), process this data (i.e. check if all parameters are within the SOA) and finally broadcast the battery's data to other boat systems through CAN and to the GUI when a computer is connected. These tasks should be run at a specific refresh rate and should thus be time-triggered. These main tasks should run in the order presented above. Otherwise, there is no new information to process or send, and so they must be run cyclically, one after the other. These tasks should respect their own refresh rate, and thus a higher priority will be attributed to their interruptions.

Other tasks of the BMS such as waking up from sleep, reacting to the voltage sensors (charger, motors, and boat on) and processing received CAN messages are less restrictive. Thus their interruptions can have a lower priority. Such tasks are event-triggered because they are triggered by a rising or falling edge on one of the MCU's digital pins.

The Software (SW) can be divided into the following main parts: Initialization; Time-based interruptions; Pin-change interruptions; Communication interruptions; BMS routine; Main loop; Sleep.

The general flowchart of the BMS's SW is presented in Figure 8. The SW starts by initializing all the subsystems and performing a few checks. If they fail, the BMS enters in a critical error state. Otherwise, the SW proceeds to the main loop where it stays unless an interruption happens or the conditions to enter the sleep mode are satisfied. After performing the Interruption Service Routines, it returns to the main loop. If the conditions to enter the sleep mode are assured, the SW places the Teensy in a low power state from which it can only exit with a wake-up interruption.

[3] This DC-DC is a 44.4 to 24 V DC-DC which turns on as soon as the boat is turned on (both emergency and electronic switches are closed) and it powers the positive side of the coils of all relays within the battery box. This part is only necessary given that the team couldn't source relays with 48 V nominal coils.

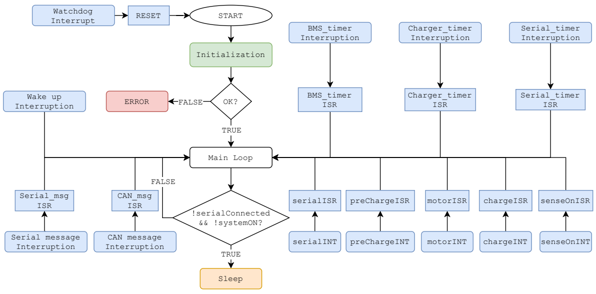


Figure 8: General BMS flowchart.

An improved version of the library provided by Linear Technology was used as an interface to communicate with the BSM but the main code was completely re-written from a command-line interface to a full BMS SW that requires no user input to monitor the battery and is capable of sending the acquired data to the boat's CAN bus and PC based GUI.

Apart from the BMS's software, a GUI was developed in parallel with the BMS development by another member of the team using the Qt framework[4]. It can run both in macOS and Windows. It is responsible for showing the BMS data to the end-user and the competition's inspection team, simply and intuitively. The GUI is also used to configure the BMS's user-configurable parameters.

5. System Validation

This section serves the purpose of detailing the tests performed on the developed BMS in order to validate that it serves its purpose as well as it respects the predefined requirements.

5.1. Balancing

For this test, a balancing threshold of 20 mV was set, meaning that every cell aggregate that is more than 20 mV above the minimum cell aggregate voltage needs to be balanced. Figure 9 shows the voltage of each cell aggregate as well as the standard deviation between the 12 voltages. In this case, cell aggregate 4 is the one with the lowest voltage, and thus, every aggregate at least 20 mV above its voltage should be balanced.

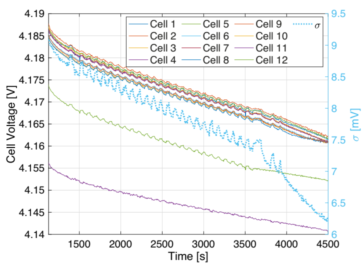


Figure 9: Cell voltages and their standard deviation while balancing.

As Figure 9 shows, the standard deviation between the voltage of the 12 aggregates decreased from around 9 mV to 6.3 mV which shows that, overall, the battery is more balanced at the end of the 75 min period. Due to the time it takes, the balancing procedure was manually stopped before all cells were within the specified threshold. Nevertheless, the acquired data already proves that balancing is working as expected. Another note should be made to explain why the voltage of the lowest cell aggregate is also decreasing during this process. This is due to the battery being in use and powering other peripherals, like the battery box fans, while balancing is taking place.

5.2. Battery capacity

Figure 10 shows the results of this test which took 58 min. The blue line represents the energy discharged from the battery measured with Isabellenhütte's IVT-S smart battery shunt, while the red line shows the lowest cell aggregate voltage. The stop criteria for the test is the moment when the lowest cell aggregate reaches 3 V. The test measured a battery capacity of 1493 Wh which is just 2 Wh above its design capacity.

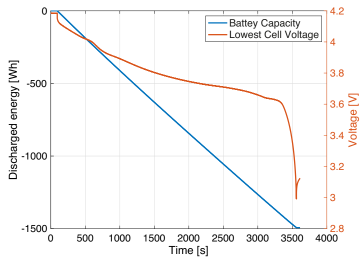


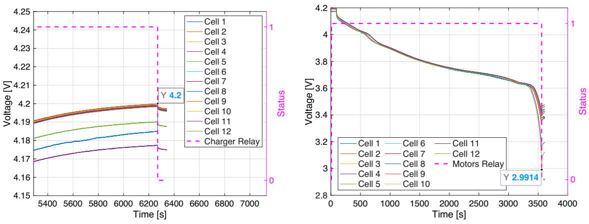
Figure 10: SR-03 battery capacity test.

5.3. Safety features

In this section, the performance of the BMS's various safety mechanisms were tested in order to validate that the BMS is able to keep the battery within its SOA.

Figure 11 shows the BMS's action in both OV and UV situations. For this test the OV threshold was set to 4.2 V while the UV threshold was set to 3.0 V. As it can be seen in Figure 11a, the BMS opens the charger relay (dashed line) as soon as it detects that one of the cells has reached the defined threshold. Similarly, in Figure 11b, the BMS opens the motors' relay as soon as it detects that one of the cells reaches the defined threshold.

Figure 12 shows the protections of the developed BMS against two different types of over-current. Over-current while charging is shown in Figure 12a while over-current while discharging is shown in Figure 12b. In this test, the default over-current thresholds were intentionally reduced to 10 and −20 A for charging and discharging respectively.

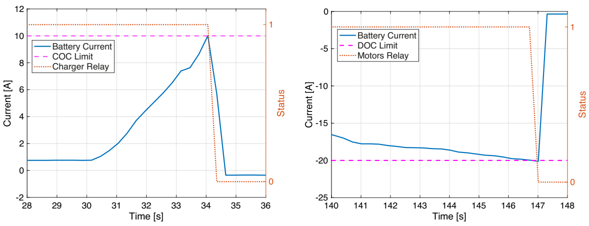


(a) (b)

Figure 11: BMSSPV V2 over-voltage (a) and under-voltage (b) protections.

These thresholds can be identified by the dashed line.

As shown in Figure 12a, as soon as the battery current reaches 10 A, the BMS immediately opens the charger relay, as the dotted line shows. On the other hand, Figure 12b, shows that once the battery current is lower than −20 A the BMS opens the motors' relay. In both situations the over-current condition is stopped.



(a) (b)

Figure 12: BMSSPV V2 charge (a) and discharge (b) over-current protections.

Figure 13 shows the current being pulled from the battery as well as the maximum temperature registered by the battery thermistors. Both bus bars and cells' body thermistors were considered. The right axis represents the status of the motors' relay as well as the BMS's over-temperature warning. For this test the battery over-temperature threshold was set to 30 °C using the BMS's GUI. As it can be seen after drawing 100 A from the battery, it reached the over-temperature threshold, and the respective warning flag was activated. Given that the current continued to increase and with it the battery's temperature, when it reached 35 °C the motors' relay was opened to prevent the battery from heating even more.

Figure 14 shows the temperature measured by both thermistors placed below the balancing resistors' heatsink and also whether the balancing is active or not (dashed line). The maximum acceptable temperature for the resistors is set to 70 °C. As shown, if one of the heatsink's thermistors reaches 70 °C, the balancing process is paused until the temperature is below 60 °C. Then, the balancing is turned on again. This 10 °C hysteresis allows the resistors to cool down properly before turning the balancing on again. Otherwise, the balancing would be turning off and on constantly.

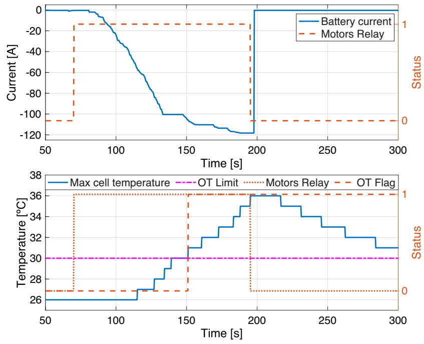


Figure 13: Battery over-temperature while drawing high currents without cooling.

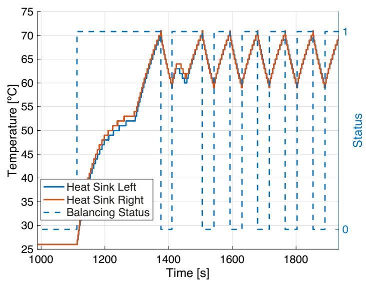


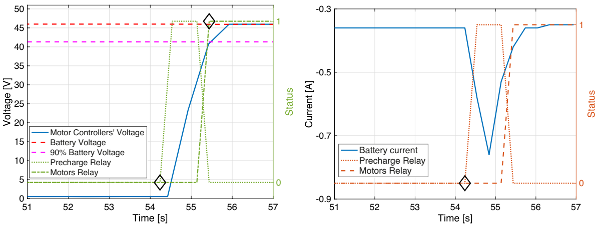
Figure 14: Over-temperature while balancing.

5.4. Precharge

Figure 15 shows the proper functioning of the implemented precharge circuit. Figure 15a shows the voltage at the motor controllers' input, while Figure 15b shows the battery current while the precharge is happening. In both figures, the precharge and motors' relays status is shown. The first marker on both figures shows the moment when the precharge relay is closed, leading to an initial current spike of 0.78 A. After this, the current slowly decreases as the load's voltage gets closer to the source voltage. When the voltage is 90 % below the source voltage, the motors' relay is closed (second marker), and the precharge relay is opened, thus allowing all the current to flow to the load.

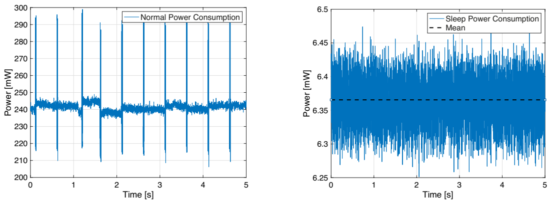
5.5. Power consumption and sleep mode

Figure 16a shows the power consumption in normal mode, while Figure 16b shows the power consumption while sleeping (i.e. the battery is not being used). We can see that while in normal mode the BMS consumes, on average, 241 mW. By analyzing Figure 16b we can conclude that while sleeping it consumes, on average, 6.37 mW which translates to a lifetime of around 27 years in sleep.



(a) (b)

Figure 15: Load voltage (a) and battery current (b) during BMSSPV V2 pre-charge sequence.



(a) Normal. (b) Sleep.

Figure 16: BMSSPV V2 power consumption under the different power modes.

6. Conclusions

This paper addresses the problem of the battery management that TSB was having, and more broadly, an electric vehicle that may have multiple different sources of power may face when using commercially available solutions. Given that the proposed system is to be applied in a specific context and has to respect some regulations, the solution presented here consists of a centralized solution that implements all battery management in a single PCB. If, for example, the number of cells in series wasn't so constrained, the solution presented here would most probably have been different.

Besides the laboratory tests carried out in this thesis, the implemented system has already been extensively tested on multiple occasions in different scenarios. The system has been actively used for more than six months in TSB's most recent solar-powered prototype São Rafael-03, with which the team participated in two different international competitions, which took place in Monaco and Portugal. In Portugal, we conquered second place amongst six international competitor teams.

The system was also used for multiple consecutive hours during TSB's own event, Odisseia TSB, which took place in the Madeira Archipelago last September. During this event the boat sailed more than 40 NM for more than 10 h divided in three days. Both during the competitions and Odisseia TSB the BMS never had any problems and was always able to protect the battery regardless of the situation. These real-world tests prove that the developed system is reliable both in laboratory conditions and in the boat, where it faces a lot more stress due to vibrations, marine environment, ambient temperature and so on.

Overall the implemented BMS can be considered a success. The objective defined by the team was to have a more straightforward BMS that could alone implement all the features needed to guarantee the battery's safety. This objective was successfully achieved. Moreover, the implemented BMS allows the team to have way more control over the way the battery behaves, which ultimately means an improvement in the overall boat's performance.

6.1. Future work

Digital signal processing should be implemented so that decisions are not taken based on a single measurement. Up to now, not using it has not brought any problems. Nonetheless, it should be added to further improve the developed system.

The library used to communicate with the BSM can be updated given that Analog Devices released a new version with a few improvements.

Regarding the BMS's power consumption, theoretically, there is still room for improvements. To achieve this, a good approach would be to add 0 Ω resistors in series with the power supply of each device so that the circuit can be interrupted and the power consumption measured.

Finally, entering a low power mode in between measurements is something that can be thought of, but care should be taken to guarantee that the BMS can wake up with all the interrupts used.

References

[1] E. Gelber. (2018) Solar is here to stay, but where are the boats?

[2] D. Andrea, *Battery Management Systems for Large Lithium Ion Battery Packs*, 1st ed. Artech House, 2010.

[3] Solar Sport One. Technical Regulations 2021.

[4] D. Andrea. (2018) Comparison of BMS ICs for Large Li-ion Battery Packs.

[5] Analog Devices. LTC6811's Datasheet.

[6] A. Samiappan and J. Kathuria. (2017) Upgrading 8- and 16-bit MCU designs: 32-bit MCU architectures.

[7] P. Stoffregen and R. Coon. Teensy Technical Specifications.

[8] FST Lisboa. FST Lisboa Cell Stack Configuration.

[9] Analog Devices. LTC3990's Datasheet.

[10] C. Duffy. Teensy Snooze Library.