Battery Management System applied to Projecto FST's EV prototype

Bruno Santos
Projecto FST — Instituto Superior Técnico (IST)
1049-001 Lisbon, Portugal
Email: brunomanuelsantos@tecnico.ulisboa.pt

***Abstract*—This thesis has the purpose of designing, building and testing a new Battery Management System (BMS) for the recent electric prototypes designed by Projecto Formula Student Técnico (Projecto FST), in particular the current *FST-05e* and the future *FST-06e*. The new design should accommodate core changes in the technology used by the Projecto FST team in accordance with the applicable rules from the Formula Student (FS) competition.**

In both mentioned prototypes, the team developed / is developing a battery powered Electric Vehicle (EV) with a 600V battery of pouch lithium cells, but potentially different configurations — dependent on cell capacity and other car specifications like battery geometry. With such a sensitive system, the BMS needs to be very reliable and safe, even in a noisy environment and subject to vibrations as is the case of a vehicle, even more due to the competition nature of the project. At the same time, the system should still fit the budget of the team as well as comply with space (and weight) limits and provide all the appropriate interfaces.

The solution presented here is a hybrid BMS with modular and distributed features composed by one master module and several slave modules communicating through a Controller Area Network (CAN) bus. This system also allows for more than one battery container, therefore more than one master module may be present in the complete system.

I. Introduction

Battery Management Systems (BMSs) have been a major factor in the successful integration of batteries in all sorts of applications. One such application is Electric Vehicles (EVs), which represent a trend in environment friendly mobility.

These EVs have used several battery technologies, usually requiring high voltage and capacity installations. Consequently, these are among the most challenging applications where performance is greatly influenced by the battery technology in use.

With the greatest shortcoming of batteries being the energy density, these impacts are seen in a significant weight increase for the powertrain system and reduced autonomy. Moreover, high voltages present in these vehicles as well as the general instability of current technologies introduce a new paradigm in product reliability and longevity, but also passenger safety.

Indeed, the highest performing battery chemistries are the ones raising the greatest concerns. These are lithium-ion batteries, which excel in energy density and discharge capabilities. However, these require constant monitoring to avoid early failures and thermal runaways. Failure to provide these monitoring devices on a per cell basis may result in fires or even explosions.

The BMS proposed here targets a system where high performance is a major concern. Naturally, it's the lithium-ion family of batteries that are targeted. The application is electric Formula Student (FS) car prototypes developed by the FS team from Instituto Superior Técnico (IST) — Projecto Formula Student Técnico (Projecto FST).

Figure 1: Prototype *FST-05e* and Projecto FST team in Germany 2013. [Courtesy of Projecto FST]

A. Motivation

FS is an international competition for university students governed by institutions like Society of Automotive Engineers (SAE) and Institution of Mechanical Engineers (IMechE).

The goal is the design of a formula type car and competing, both in pitching design concepts and business plans to a jury, building and racing for best times and efficiency. For these competitions, Formula SAE (FSAE) issues the base set of regulations [1].

Throughout its history, FS has seen an increasing interest in alternative powertrains, and currently combustion and electrical prototypes compete directly with each other.

In search for greater competitiveness, Projecto FST saw a great departure from the *FST-04e* 'safe' and 'simple' approach. Indeed, the successful BMS developed for *FST-04e* [2] was deemed neither appropriate, nor scalable for following prototypes.

In 2013, the team developed its second electric prototype, the *FST-05e*. This prototype faced several mechanical and electrical issues in 2013 season, not being able to run. One of the main issues was the inability of the in-house developed BMS to properly monitor its battery.

This work was then coordinated to develop a new generation of BMS to avoid the same problems in *FST-06e*, which should reach the testing phase early 2015. Additionally, the system would be prototyped in *FST-05e*, allowing its on-track debut in Silverstone, United Kingdom, in 2014.

Several commercial solutions exist, but, given the critical nature of this system and the expected level of integration within the prototype, an in-house solution was proposed. More so, extending the know-how and usage of self-developed electronics and tools has long been a goal for Projecto FST.

B. Scope and objective

This work comprises the identification of the available technologies and an analysis regarding the best architecture to comply with the team's goals. The proposed solution is a digital distributed BMS with one or more *master* modules, each controlling a set of *slaves*.

The integration of the system took place within the 2014 season time frame, in *FST-05e*. However, this car posed certain design constraints that needed to be addressed and avoid compromising in *FST-06e* and future solutions at the same time.

More so, in favor of early integration and faster development, the original master module was used. This module also provided some interfaces that were intended to be deprecated in the new design. However, all the electronics and software should be made targeting an unification of these systems.

With this approach, and the tools developed throughout this work, the new BMS should provide a future proof solution for a broad range of battery configurations. Additionally, a unified code base and hardware development should yield a greater maintainability and longevity of the prototypes with great benefits to design iterations.

All the developed hardware and software referenced throughout this document are hosted at https://bitbucket.org/projectofst with the prefixes *FST BMS* and *FST CAN tools*.

C. The prototypes

One of the objectives of this BMS design is to provide a scalable and generic solution in hardware and software. However, it also targets a close interaction and integration within two specific prototypes: *FST-05e* and *FST-06e*.

Tables I and II resume some specifications for each car. A noteworthy difference is the cell arrangement where these two solutions, while different, only represent a small subset of possible configurations in future designs[1].

Table I: *FST-05e* battery specifications.

Parameter	***Value***
Maximum voltage	600 V
Nominal voltage	532.8 V
Minimum voltage	432 V
Maximum peak current	630 A
Main fuse current	100 A
Maximum charging current	18 A
Capacity	9.5 A h
Total energy	5.7 kW h
Total number of cells	288
Cell configuration	144s2p
Number of stacks	6

Table II: *FST-06e* battery specifications.

Parameter	***Value***
Maximum voltage	600 V
Nominal voltage	532.8 V
Minimum voltage	432 V
Maximum peak current	250 A
Main fuse current	160 A
Maximum charging current	10 A
Capacity	10 A h
Total energy	6 kW h
Total number of cells	144
Cell configuration	144s1p
Number of stacks	12 (2 containers)

II. Battery management system

The proposed system is a distributed approach where, optionally, more than one master should work cooperatively. Each slave module monitor a group of 12 cells and each master should support a minimum of 12 slaves. This allows for great scalability in a FS prototype, whose batteries have a maximum of 600 V.

The slave modules are responsible for monitoring each individual cell. This comprises voltage measurements, temperature measurements and balancing. The master coordinates the slave modules, controls the availability of power and implements protective measures to prevent damage to the Tractive System (TS).

The current of the battery is also measured by the master module, but the balancing current must be accounted for by the slave modules. To communicate with each other, the modules use an internal Controller Area Network (CAN) bus through a galvanic isolation that ensures complete isolation

[1] Battery configuration will be represented as XsYp, referring to X cells in series and Y in parallel.

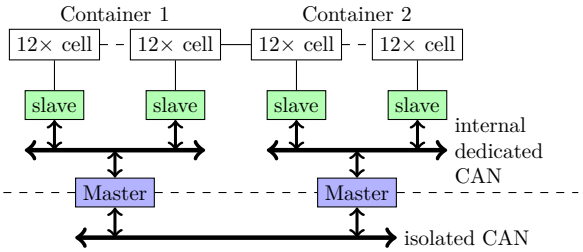


Figure 2: Proposed topology for new BMS in a 2 container configuration.

between the TS and the low voltage system of the car.

Table III summarizes the targeted specifications for each measurement subsystem. These specifications were competitively set against *FST-04e* [2], commercial solutions [3; 4] and established practices and recommendations as per [5] among others.

Table III: Measurement limits.

	Parameter	Value
Voltage	maximum	4.7 V
	minimum	1.6 V
	error	10 mV
	channels per cell	1
	frequency	1 Hz
Temperature	maximum	120 °C
	minimum	−30 °C
	error	5 °C
	channels per cell	0.5
	frequency	0.2 Hz
Current	maximum	200 A
	minimum	−50 A
	error	5 A
	frequency	20 Hz

For historical reasons, the proposed system has the versioning scheme *v3.x* and *v4.x*, for slave and master modules respectively. This is to denote a new iteration in the teams' history rather than any dependency or implied similarity with previous versions.

The tool used to develop both the master and slave modules was Altium Designer. Module programming was achieved through a PICkit™ 3 and the IPE interface [6; 7].

The code is written mostly in C with a few assembly calls, compiled through the XC16 C compiler and MPASM™ assembler (free versions) [8; 9]. Most embedded software is C90 compliant, but some Microchip extensions and assembly macros are used.

A. Slave module

The slave is composed of a Microprocessor Unit (MPU) and a BMS Integrated Circuit (IC). Together, through an internal Serial Peripheral Interface (SPI) bus, they measure voltages and temperatures.

The choice for an integrated solution for the voltage measurements provides great compactness and precision without calibration processes — one of the most relevant drawbacks of the previous system.

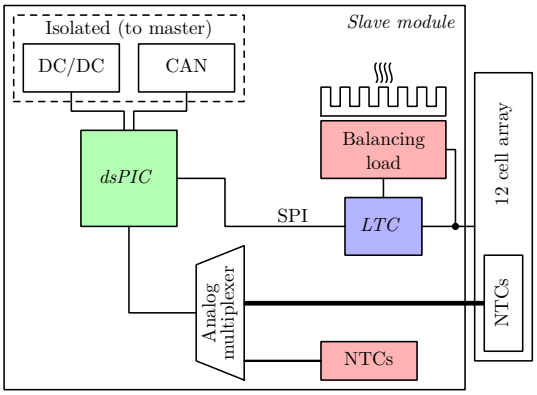


Figure 3: Slave module architecture (*FST-05e*).

The chosen IC was the Linear Technology *LTC6804*, however, given its unavailability for the production time frame, the similar but lower performing *LTC6803* was chosen [10]. Nevertheless, hereby referred to as *LTC*, this IC provides fast and accurate voltage measurements and integrate the balancing of the cells preventing balancing related errors in measurements.

As virtually all solutions on the market, these ICs offer very low support for temperature measurements. The industry accepts 2 to 3 sensors to monitor a large pack of cells, but the same doesn't happen in FS, where more strict rules apply.

For this reason cell temperatures are measured through the MPU directly connected to an array of Negative Temperature Coefficient thermistors (NTCs). For monitoring its own state, the slave has 3 Surface Mounted Device (SMD) NTCs spread through its Printed Circuit Board (PCB). All these NTCs would require significant power requirements, thus being multiplexed and only one — if any — is active at any time.

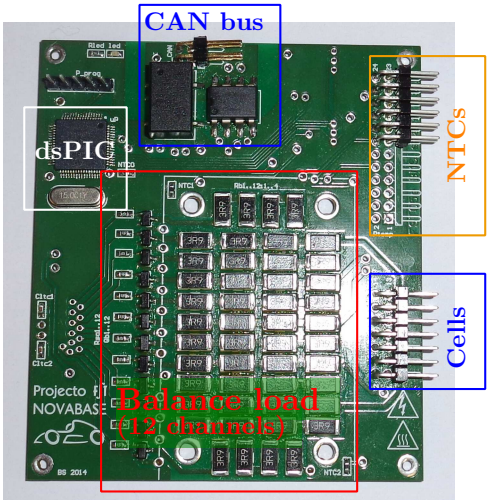
These temperatures are directly monitored by the MPU, a Microchip *dsPIC30F6012A* [11]. This model, hereby referred to as *dsPIC*, was chosen for its extensive use within other modules of the car, and thus familiarity. Together with a high performance level and plenty of memory, this allowed a faster development, but there are cheaper alternatives that should require minimal adaptations.

Another important task of the slave modules is the determination of State of Charge (SoC), i.e. the amount of 'fuel' left. This is a very complex task as this parameter has a strong dependency with a vast number of variables. Not being an essential feature, and given limitations within *FST-05e* regarding current intensity measurements, the proposed solu-

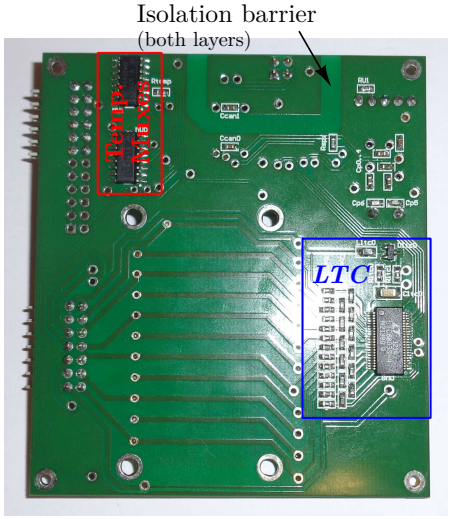
tion is very simple and only takes voltage into account.

a) Layout:

FST-05e had very strict requirements regarding the layout of the slave module. For this reason, the new module borrows its layout from the original one, despite offering a different solution.



(a) Top layer.

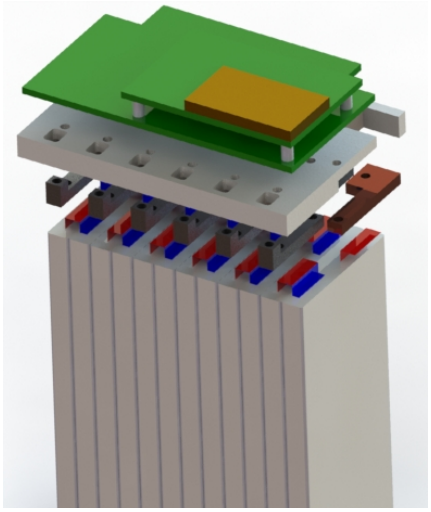


(b) Bottom layer.

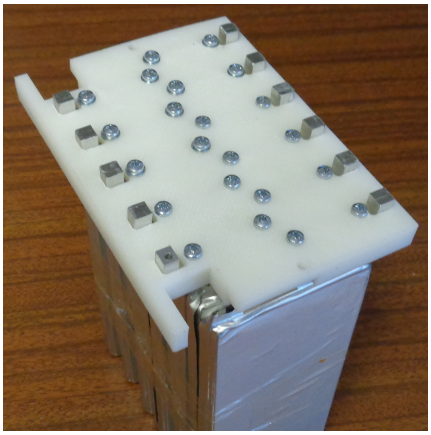
Figure 4: Slave module v3.1 (*FST-05e*).

This solution still relies heavily on wire connections to the cells and temperature sensors, result of *FST-05e* design. For *FST-06e* however, several solutions were proposed to include the BMS design in the *stack* design (a subdivision of the battery). Figure 5 illustrates one such solution currently being prototyped by Projecto FST.

This solution should provide easier assembly of the stack, more accurate readings and, above all, reliability and easier assemblage.



(a) Partial render of proposed stack solution in exploded view.



(b) Early proof of concept. The slave module should connect to the exposed part of the bus bars.

Figure 5: *FST-06e* stack concept. [Courtesy of Projecto FST]

b) Self checks:

With the emphasis on safety, reliability and advanced diagnosis, several safety features are implemented within the slave module, beyond the over/undervoltage or low/high temperature detection.

These features detect software faults through a watchdog timer, cell connection faults and temperature sensor faults. More so, even embedded systems are part of intrinsic and regular sanity checks that detect faults within the components soldered in the slave PCB.

The detection is made on a per connection, per module way, which enables the slave module to precisely report the nature and origin of the identified problem. Indeed, the only fault not covered by the self diagnosis routines is a failure of the MPU itself.

This situation needs to be covered through proper error handling within the master module.

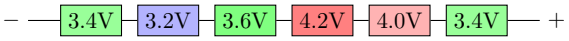


Figure 6: Example of an unbalanced array of cells.

c) Balancing:

Balancing is a universal problem for arrays of batteries. An unmaintained battery may become unbalanced to the point where it no longer allows safe charging or discharging.

To maximize the available capacity of a battery, one of several balancing techniques may be employed. There are two major approaches: passive and active balancing.

For the system in question, an active balancing solution has no advantage in terms of cost or performance. Due to the fast discharge projected even for the longest event in FS — endurance event of 22 km —, an active system wouldn't provide any significant increase of range and would likely result in a heavier and bulkier battery.

Therefore, for this application, a passive balancing system yielded better performance overall and it is also simpler and cheaper to implement. In the proposed system, the balancing is done through one bank of resistors per cell that dissipate excessive energy of the most charged cells. This is done solely during the charging process.

To improve the charging performance, balancing should occur sooner rather than later. Otherwise the charging current may need to be reduced while most (or any) cell is still significantly discharged. This logic is implemented by the slave module, which decides whether each cell should be balanced, and the master, which enables balancing only in specific situations.

The balancing load may be used (or disabled) by the slaves in some emergency situations regardless of the master's commands. These exceptions include safety precautions against any harmful situation to the cells and the slave modules themselves.

Finally, to monitor all 12 channels with only 2 temperature sensors, a heat sink needs to be installed for thermal coupling, which also serves the purpose of improving heat dissipation and, therefore, balancing performance.

For *FST-05e*, the best option was a ¼ brick low profile heat sink, illustrated within a fully assembled slave module in figure 7.

B. Master module

The master module was developed only as a new software for the original master with proper protocols and error handling, the missing components of the previous design. Indeed, from an hardware point of view, the original master was not very technological, but was mostly functional except for some safety issues in the TS activation.

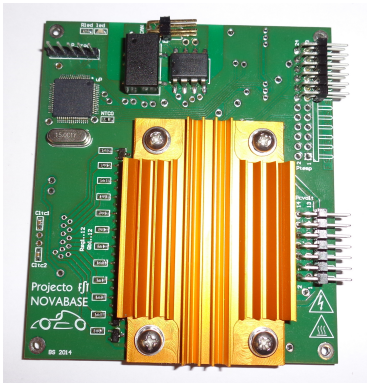


Figure 7: Slave module v3.1 (*FST-05e*) fitted with a low profile heat sink.

In a latter phase, a new module was designed. However, the software was developed with all the features in mind and the MPU in the original master was the same *dsPIC* used in the slave modules and in the new master. In fact, the new module was designed to be a drop in replacement, but account for a richer feature set and better safety.

Because of time constraints, the new solution has not been prototyped completely — i.e. some hardware functions were not yet validated — nor manufactured or deployed in any prototype. However, other than the software routines and protocols, all following references to the master module refer to the new proposed solution.

The overall master architecture is represented in figure 8.

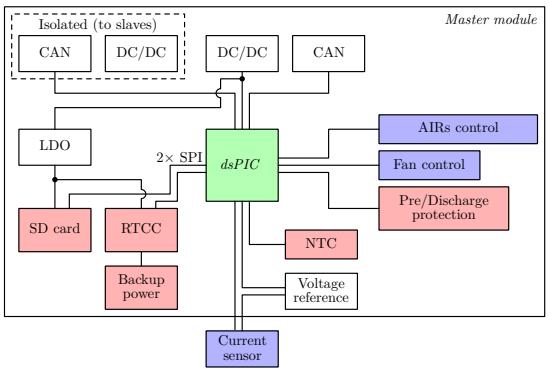


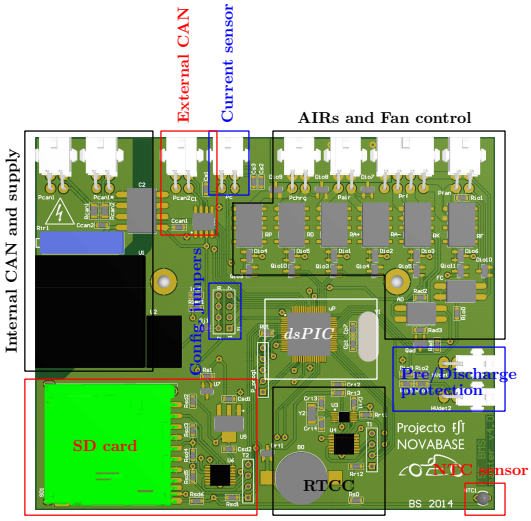
Figure 8: Master module architecture.

d) Layout:

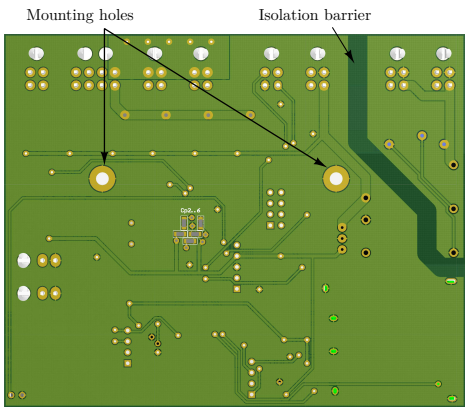
The new module required a higher component density than the previous one, yet it includes more features. These are a Secure Digital card (SD card), a Real Time Clock & Calendar, extra headers and support for an extra safety feature for the precharge and discharge circuits.

This new master version is represented in figure 9.

e) Network responsibilities:



(a) Top layer.



(b) Bottom layer.

Figure 9: Render of master module v4.0.

The master module connects to two CAN buses. One is only shared with the slave modules forming a closed network.

In this network, the master is the sole responsible for querying the slave modules to prevent fault cases in which the slaves would be unable to report any emergency — e.g. a wire fault in the CAN bus. But it also needs to coordinate the slave modules while charging to ensure all slave modules are balancing towards a common goal rather than just balancing the cells under their supervision.

And finally, a SD card and a RTCC allow higher complexity features to be implemented and event logging with time and date. RTCC functions may also be provided to other modules residing in the outer network.

Regarding multi-master topologies, the proposed solution is that one or more masters are *slaves* to another single master, configured through a set of jumpers. This avoids group decision trees with race conditions and synchronization problems.

To guarantee correct operation of the master module, a watchdog enforces a complete reset of the module triggering a fail-safe deactivation of the TS in case of software lockup. However, a *ping* system is available and may be configured with any other module in the outer CAN bus with access to the TS safety circuit as, for instance, a second master.

f) Tractive system control:

The master manages the availability of high power from the battery through a pair of Accumulator Insulator Relays (AIRs). With these relays, the master can interrupt any discharge or charge beyond the limits of the battery.

However, the control of the tractive system is a safety critical task. Namely its activation requires a soft charge feature and rules mandate a fast discharge as well.

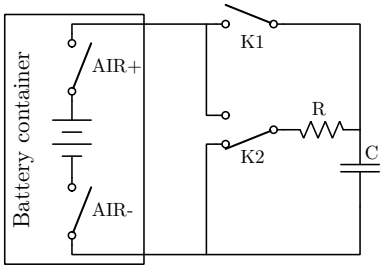


Figure 10: Pre-charge and discharge circuit. All switches represent relays controlled by the master module.

If any of these circuits experience a failure, the TS may become damaged with varying levels of danger. This was an issue with the original master module. The revised master should detect all failure modes, but that requires some exterior hardware.

A complete solution was proposed taking advantage of the already existing Tractive System Active Light (TSAL) module — another rule imposed module, responsible for the detection of high voltage in the TS. Two of these modules should be installed in the car, one inside the battery to monitor availability of high voltage at the battery poles — as already required by the rules —, and the second one across the resistor R (figure 10).

The outputs of both TSAL modules should then be available to the master.

III. CAN tools

To interface with the BMS through a common computer, a set of tools were developed. The first one is a translator module capable of converting CAN messages into Universal Serial Bus (USB) and vice versa.

The remaining tools are embedded in a software package that provide an interface to the translator and high level interfaces to several modules. Among

these high level tools, only one was developed. This interface targets the BMS, providing several controls and displaying several parameters from within the BMS in a user-friendly manner.

This software package targets x86_64 Linux, making use of GNU is Not Unix (GNU) tool chain (gcc v4.8.3). Moreover, to provide a Graphical User Interface (GUI), it's used the Qt framework 5.3.2.

All code is C11 compliant C++.

A. CAN-USB translator

This module is a *Módulo CAN PIC FST* [12], extensively used by Projecto FST, with an embedded *dsPIC*. The focus was then on the software and integration, rather than the hardware.

This translator introduces a major improvement relative to previous translator units used by Projecto FST: a binary frame.

The previous solution used American Standard Code for Information Interchange (ASCII) encoding for their messages. While significantly simpler to develop, this imposes a significant overhead and bandwidth penalty, leading to less than 50 messages per second against the maximum of ≅6200 (CAN bus 80 % full).

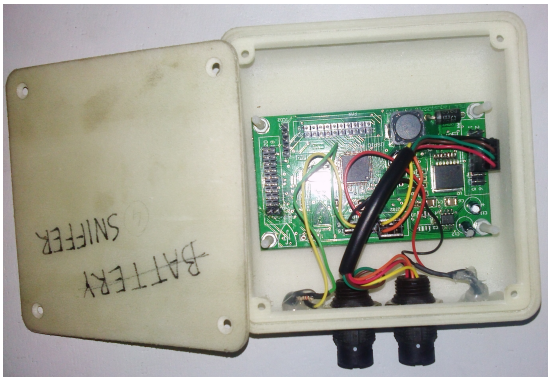


Figure 11: Translator module. Left connector connects to CAN and the right one to USB; Light Emitting Devices (LEDs) indicate bus activity.

Together with the new hardware and a buffer implementation, the new translator unit is capable of handling upwards of 3000 messages per second and even higher rates in small bursts of messages.

A binary frame requires the usage of a Cyclic Redundancy Check (CRC) for message consistency and identification. This CRC is in fact one of the greatest performance limiting factors and paid versions of the Microchip compilers could introduce significant improvements through optimizations. Nevertheless, this *dsPIC* is still able to double its current frequency, effectively halving the time required for CRC calculation and duplicating the baudrate of the serial connection.

B. FST CAN interface

The underlying functionality of the interface and the translator are not tied to the BMS and should then be made as generic and extensible as possible. Figure 12 shows the architecture for this interface, where grayed out modules were not relevant for the objectives of this thesis, and therefore not developed, but still accounted for in the design.

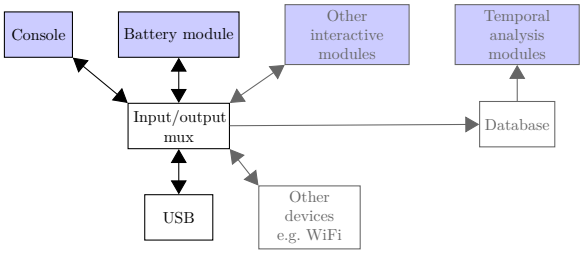


Figure 12: *FST CAN interface* architecture. Blue blocks correspond to GUI interfaces. Grayed out blocks represent features not implemented.

g) Console module:

The console GUI module is represented in figure 13. This module allows to input and receive any CAN message without distinguishing between senders or purposes. For this reason, the console is particularly appropriate for early development, debug unpredicted situations and test newly added features.

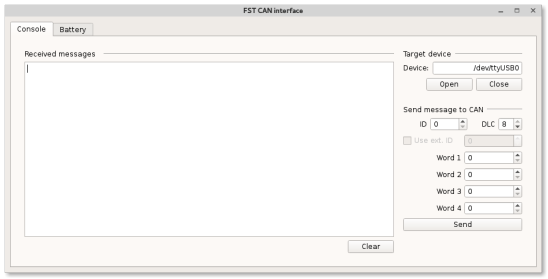


Figure 13: Console tab of *FST CAN interface*.

h) Battery module:

Unlike the console, the battery module has knowledge of the message format, codes and protocols. This allows it to represent BMS related information in an easy to read interface.

This module, illustrated in figure 14, has a main interface that functions as a control panel. Here, only broad information is given to the user, but an optional interface may be activated to show details of every cell being monitored, including any wire faults or lack thereof.

To prevent the display of outdated information and detect the battery has gone offline, this module implements a series of watchdog timers that clear the output.

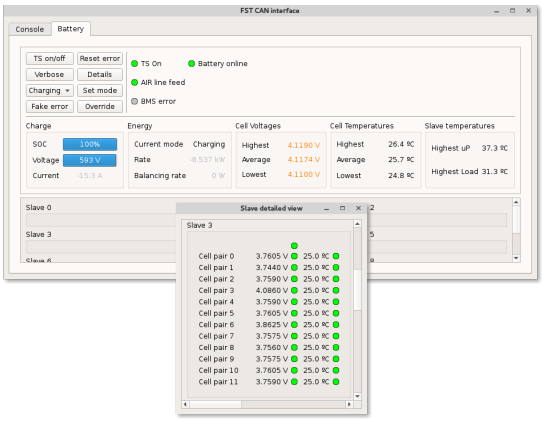


Figure 14: Battery module of *FST CAN interface*. Screenshots taken at different times. Small window has a scrolling area showing all slave modules and not just one.

In the image, the background window shows the battery completely charged, just before the automatic cutoff by the master module. The window in the foreground shows the detailed information for one slave in a previous, uncharged and unbalanced, stage.

IV. Tests and deplyment

The first milestone for the manufacture phase of this BMS was the completion of the slave module. This module was first prototyped in-house in IST Taguspark facilities. The resulting module is illustrated in figure 15.

This phase consisted in a verification of design specifications and the rigorous test of each functionality. With this prototype, all the hardware components were verified to be working as expected

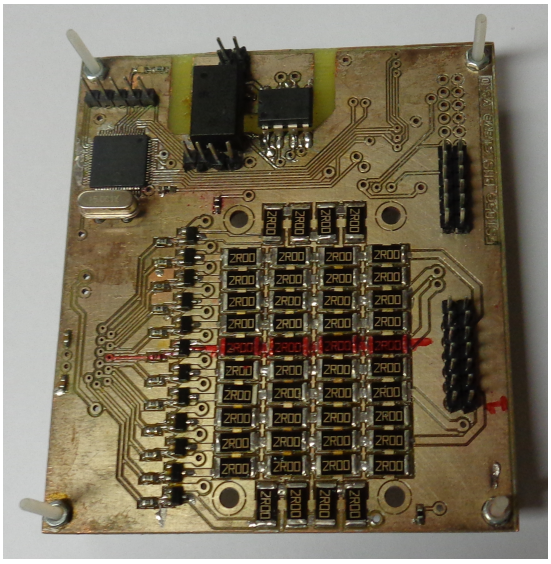


Figure 15: Prototype slave v3.0 top layer.

and a few revisions were made prior to send for production.

The protocols between slave and master were also set and enforced still before the installation within *FST-05e*. The majority of features and all the safety check were implemented as well.

After scaling the system with one master and two slaves, the system was installed within the battery of the fifth prototype for the next phase of tests and development.

A. Deployment

The fully assembled system is shown in figure 16. From this point onwards, feature implementation became a second priority with stability, safety and reliability being the main issues. This was in part due to tests that required a fully installed battery, including tests on track. This goal was well balanced with the integration of the higher level interface.

The system demonstrated great stability after initial sensitivity issues. In fact, most problems within the battery container were soon result of installation problems related to the dependency on wires and stack assemblage difficulty rather than BMS related.

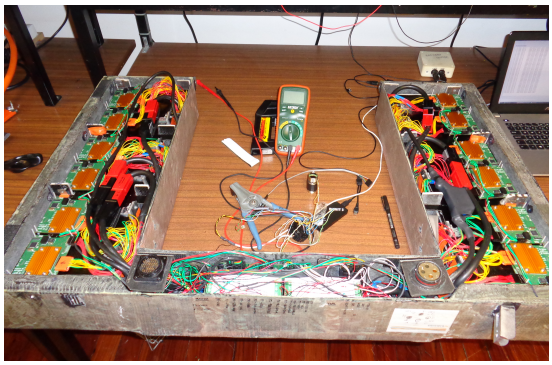


Figure 16: *FST-05e* fully assembled battery without lid. Connector on the left of the middle section is low voltage only. Connector to the right is for high voltage and current.

The BMS error handling proved to be essential in the early integration process. The 300 wires connecting the slaves to the cells and respective sensors proved to be a major source of early BMS faults. However, the BMS was consistently able to identify which wire(s) needed to be replaced.

Using the v3.4 hardware rather than the revised v4.0, the master was also able to handle the 12 slaves comfortably and assure the safe operation of the battery.

Finally, the British competition was an important goal. Despite several successful tests prior to the competition, mechanical and electric problems played a huge role in a rather unsuccessful competition in its dynamic events. Indeed, the car only took

Figure 17: Manuel Ferreira driving *FST-05e* in endurance event, Silverstone, *FSUK2014*. [Courtesy of Projecto FST]

part in the endurance event — the most difficult to finish — and only completed a handful of laps.

One of the problems was found within the battery itself. Improper handling and assemblage of the cells are suspected to be the cause of the early failure of the battery. The cell capability to retain energy was severely damaged as a result.

More so, these cells were installed for the 2013 season and never fully used given BMS problems, but they were also largely unmaintained as well. Therefore, it's difficult to pinpoint the cause for the reduced lifespan of the battery.

Because of financial constraints, rebuilding a battery with all new cells was not feasible, but since the competition, the battery was rebuilt with the few spare cells which only allowed a configuration of 144s1p. Therefore, the battery has currently half the design specified capacity, but the extra space within the container also allowed to solve some of the assemblage problems of the previous solution.

Indeed, the battery is now proving to be more consistent and is expected to allow extensive tests to *FST-05e*, invaluable for the design of new prototypes.

V. Conclusion

This thesis addresses the problem of battery management within a high performance vehicle with a partially distributed solution. Most of the objectives were indeed fully addressed, providing a stable, safe and accurate system.

On one hand, these installation issues are highly dependent in a new battery design and not only in a new BMS architecture. On the other hand, both the charger and better performance parameters like the SoC had to be sacrificed given the large scope of the project. However, in a fast paced environment like FS, priority was rightfully given to the actual testing, safety and stability of the car.

The final solution is not at feature parity with commercial solutions just yet, but clearly has all the capabilities to do so. Weighting in the very competitive cost of the proposed system, and the versatility it offers when compared with the same commercial solutions, the proposed system becomes more attractive.

Opportunities for improvements to achieve this feature parity (and even surpass it) were found all around, which is not surprising given the broad scope of the project. However, the most relevant directions seem to be in SoC estimation algorithms. For better performance, greater integration of mechanical design seems to be the greatest challenge and *FST-06e* should be a step closer to a definite solution.

On a competition perspective, the recovery of *FST-05e* has been a difficult process with a myriad of mechanical and electrical problems, mainly Electromagnetic Interference (EMI) related. This prevented a better performance in *FSUK2014*, but each failure proved to be an invaluable lesson, also reflected in the proposed system.

Finally, tests proved quality assurance and good mechanical design to be the main priorities in a battery design as no BMS system is able to replace these. It can help however.

References

[1] SAE, "2014 Formula SAE Rules," 2014.

[2] M. Guedes, "Battery Management System for Formula Student," Master's thesis, Instituto Superior Técnico, 2011.

[3] elithion, "Lithiumate™ pro Product technical information," 2014.

[4] LION E-Mobility AG, "LION Smart Li-BMS v3 — Allgemeine Beschreibung des modularen Batterie-Management-Systems," tech. rep., LION E-Mobility AG, 2014.

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

[6] Microchip, *PICkit™ 3 Progammer/Debugger Users' Guide*, 2010.

[7] Microchip, *Integrated Programming Environment (IPE) Users' Guide*, 2013.

[8] Microchip, *MPLAB XC16 C Compiler Users' Guide*, 2013.

[9] Microchip, *MPASM™ Assembler, MPLINK™ Object Linker, MPLIB™ Object Librarian User's Guide*, 2013.

[10] Linear Technology, *LTC6803-2/LTC6803-4 Multicell Battery Stack Monitor*, 2011.

[11] Microchip, *dsPIC30F Family Reference Manual*, 2006.

[12] V. Almeida, "Volante Electrónico para o FST," Master's thesis, Instituto Superior Técnico, 2009.