# Distributed Battery Management System Prototype for UTA-Racing Electric Vehicle 2027

*George Boone, Bryan Gonzalez, Khai Nguyen, Roman Rattanakone, Zakir Muqueet, Jayde Brown*
UTA Senior Design Fall 2025 - Spring 2026, *BeeMS Team*
*yongboone1@gmail.com* *glezbryan12@gmail.com*

***Abstract*— Design theory of the 2025 senior design team *BeeMS*. System, Architecture, and our implementation for a distributed BMS intended for use in UTA Racing's E27 battery pack. *This is not a pure research paper*, topics are intended to support anyone pursuing this project in the future.**

## I. Introduction

Multi-cell battery packs are cornerstone technology in mobile and stationary energy storage. With the sensitive operating conditions and potential for violent failures otherwise, active management is a necessity; a Battery Management System (BMS) serves this function. For the purposes of this paper, battery packs involved are for FSAE Electric Vehicles (EV) operating approximately at 80kW 600V, lithium based - because of relatively performant capacity and operational characteristics for use in a Formula Society of Automotive Engineers (FSAE) EV competition - however contents are still relevant outside of this category. For FSAE, teams with custom BMS's have a competitive advantage by means of shrinking packaging, enabling unrestricted pack configuration, and enabling team specific customizations. It also provides the experience and basis for non FSAE related battery pack implementations. We will cover the implementation of a distributed BMS; everything from architecture to UI; following what we believe best for the purposes of a UTA Racing BMS; but by no means limited to such.

## II. Background

This document is put together by the "*BeeMS*" team at the end of Senior Design 2. The team is composed of 6 members - 1 Computer Science, 5 Computer Engineering. Strictly by the SD proposal we have a success - by creating a system where all necessary DAQ and compute for column counting fuel gauging is available, and continued development does not require mixed field expertise. However we have failed in delivering a BMS capable of safe and autonomous operation. We believe our findings, subsystems, complications, parts of our work, etc are worth documenting and of value to anyone involved in future BMS development; continuing this work or otherwise. We emphasise future groups focus on developing the hardware to support a known implementation of fuel gauging (e.g: column counting or kalman filters), rather than developing both hardware and the algorithm in parallel as the BeeMS team attempted.

## III. BMS Theory & Concepts

A BMS's role is to monitor and protect a battery pack from exceeding the cell's thresholds throughout its entire lifetime and operation. Battery Packs are described in terms of series and parallel cells; main intentions being series to increase voltage, and parallel to increase ampacity; *individual cells* in parallel are considered a single cell. UTA Racing's 2025 EV has a pack arrangement of *individual cells* 5 in parallel, repeated 144 times, notation being 144s5p , for scale the Rivian R1T pack is 108s72p. Each parallel set is considered a single cell by the BMS, as they all functionally behave as a single cell with increased capacity.

By FSAE competition rules the battery pack is physically subdivided into smaller bundles of cells (Module); each Module being <120V and 6kW during operation.

Voltage measurements are performed for all cells, and temperature measurements ideally per cell as well, however only 20% coverage is required by FSAE regulations. Temperature measurements of each cell should be from the negative terminal as in cylindrical cells it is more thermally coupled to the internal cell than the positive terminal because of the internal 'tab' (present even in some 'tabless' cells). Current sensing is achieved through a hall effect sensor.

### A. Failure conditions and Metrics

A cell fails in conditions of over/under temperature, voltage, current, or charge. The measurement of charge is a percentage called State of Charge (SoC), or alternatively measured by depletion - referred to as Depth of Discharge (DoD). SoC is not a measure of State of Energy (SoE). Cumulative factors such charge discharge cycles, age, abuse, etc are represented by the State of Health (SoH) - relative to the ideal/original pack characteristics. SoC, DoD, SoH, and SoE are all percentage/relative metrics; determining and measurement of them in an accurate manner usually involves learning algorithms as these vary significantly with pack history..

### B. Cell Balancing

Because of manufacturing & assembly deviations *individual cells* have non-identical characteristics even between the same model and batch, therefore all cells within a pack must be individually monitored. Should any cell fail, the entire pack fails. Because of these small deviations, cells charge & discharge at different rates, and as such it is possible

for some cells to over discharge/charge during operation. The solution is to evenly distribute charge across all cells; this is referred to as cell balancing. There are 2 common methods of doing this, active and passive balancing.

- Passive : discharge all cells to match the lowest cell. Commonly though shorting the cell. Energy inefficient, low complexity & cost.
- Active : transfer charge from over charged to under charged cells. Commonly though muxed capacitors or buck/boost systems. Energy efficient, increased complexity and cost.

Cell balancing during "heavy" discharge of the pack is not usually performed due to measurement imprecisions. By FSAE regulation balancing of any kind is now allowed during operation.

### C. External Interfaces

Safe operation of the pack is maintained by digital communications with loads, chargers, and the operator; additionally an analog fail safe is included in the form of a vehicle interlock. During operation the BMS must at a minimum communicate pack limits - max discharge, charge, SoC, etc.

In modern vehicle systems the network is a Controller Area Network (CAN) or Ethernet; for our purposes it is CAN/CAN-FD (CAN Flexible Data Rate).

Due to the risk potential of HV, the battery pack is physically disconnected from the rest of the vehicle by Accumulator (battery pack) Isolation Relays (AIRs). The BMS is responsible for the operation of precharge and discharge of the prior to connecting the pack, a precharge is performed to reduce voltage across the AIRs, a discharge is performed after AIR's disconnect to drain residual charge.

### D. Cells

Physically the common lithium cell categories are prismatic, pouch, and cylindrical; these differ in physical packaging and behavior. There are factors to consider depending on cell type such as expansion for pouch and lower cell density for prismatic. For our purposes we design for cylindrical cells, the main effect being assumed high cell density.

### E. SoC

The simplest and the minimum required pack metric to track, SoC has a non linear relation with SoE and represents the accumulated charge of the pack. The primary methods of calculation being coulomb counting and correlation to Open Cell Voltage (OCV) of the pack. Unless active cell balancing or otherwise charging the pack, the pack's effective SoC is limited by the weakest cell in it.

It is the responsibility of the BMS to ensure no cell crosses 100% SoC during charge and 0% SoC during discharge.

During storage, there is an ideal SoC cells should be kept at to minimize dendrite formation and other wear. Varies by cell model generally 30% to %70.

### F. Battery Management System Architecture (BMSA)

There is no universally adapted standard for classification for BMSAs, this paper will be with reference to the classifications defined in [1]. The leading two operational theories are distributed and centralized; referring to the distribution of compute within the system. For our purposes the system is of the 'static' category.

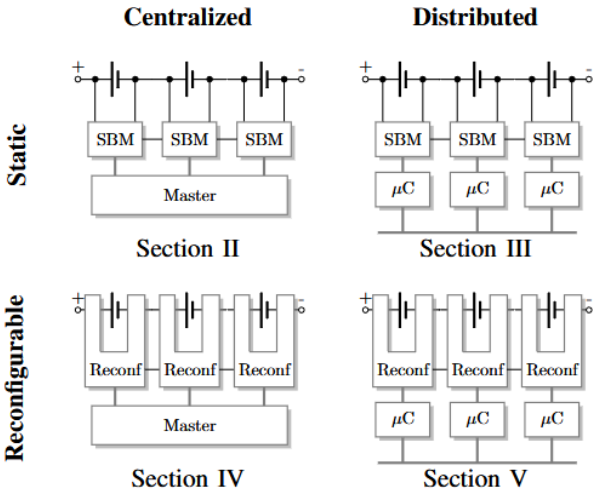


*Fig. 1. BMSA categorization. Sensing and Balancing Module (SBM). Adapted from [1]*

The automotive industry trends centralized static systems[1]. The OrionBMS currently used by UTA Racing is a centralized static system; a unique feature being that it is a monolithic system with internal SBMs. This paper details the design of a distributed static system.

### IV. ICs

Battery packs are established in demand technology and as such there exist dedicated hardware for their purposes. As each cell needs individual voltage and balancing capabilities at a minimum, scaling discrete solutions becomes a packaging and economic challenge; as such there exist dedicated SBM IC's, common terms for such are *Fuel Gauges*, *Battery Monitors*, and *Battery Management*. For the purposes of this paper these will be referred to as SBM IC's. Small series of cells - common options range from 1 to >28 cells - are connected to the SBM IC from which cell voltage and passive balancing can be performed; external circuitry integrated in the cell sense line can be added to offload passive balancing load to external sources; passive solutions discussed in [2].

A common configuration in centralized static systems are daisychainable device level interfaces (e.g: SPI, i2C, isoSPI, etc), so that SBM IC's may be distributed across the pack and controlled from a central Pack Management Unit (PMU).

## V. Centralized BMSA

See [1] for in depth detailing. An implementation where the SBM or MMUs are entirely controlled by the PMU. The approach is the simplest though implementation is highly hardware dependent on the SBM choice. General trend is

software inflexibility and pack dependent, but ease of implementation.

## VI. Distributed BMSA

See [1] for in depth detailing. Each module has an onboard MCU that handles real time operations. In a distributed system a central PMU is still present to perform higher level control. In a fully distributed system no PMU is required, and MMUs autonomously coordinate, however there are new challenges in determining pack configuration in a fully autonomous system.

***Note*— Sections VII and after, are only about the approach and conclusions of the *BeeMS* team.**

## VII. Deciding factors of BMSA

Considerations are drawn from SBM hardware choice between daisy chainable IC's such as the LTC6813 and BQ79616, against pure standalone solutions such as the BQ76952 [2].

The primary difference is complexity in Data Acquisition (DAQ). In a distributed system an communication abstraction layer must be made to coordinate PMU and MMUs over a system level network such as CAN, whereas this layer is supplied by the specific hardware in a centralized system as it uses a device level interface such as isoSPI or UART.

Communication abstraction has the advantage of removing SBM hardware dependency.

Hardware availability and comparable performance was the primary reason for choosing the BQ79616 over the LTC6813. Potential for SBM hardware abstraction to avoid vendor lock-in was the primary reason for choosing the BQ76952 over the BQ79616.

## VIII. System Design

Each 'module' (by FSAE definition) of the battery pack is split into 2 virtual modules of the BMS, each virtual module being a 1 SBM IC and 1 MCU; together these form the Module Management Unit (MMU). This was done because of the 16 cell limitation of the BQ76952 (SBM IC used), and the complexity of legally incorporating LV board-to-board signals between the top and bottom sides of the physical 'module'(FSAE). Each module (virtual) is instead digitally joined to a vehicle level network and independent of the physical packaging.

Digitally the MMU offers 2 interfacing methods, a register based one - meaning operation is controlled and reported by 'read' and 'writes' to virtual addresses - and a BMS custom protocol - series of custom CAN packets. The register based operation is intended for simple & complete control for 3rd party interfacing, while the custom interface is tailored for operations as a FSAE BMS - primarily by removing packet overhead and enabling non register based operations. A significant motivator for the register based interface is that each MMU can act as a Modbus device, thereby allowing the use of preexisting Human Machine Interface (HMI) software; the challenge of creating a custom UI is exchanged for creating a Modbus-CANbus translator.

The compute available with each MMU allows practical edge computing for real time monitoring and response; preprocessed data is then forwarded to the PMU for non real time control and compute intensive modeling. This provides a definitive resource, and logical divide between PMU and MMU responsibilities - a major benefit being the mitigation of single point of failure vulnerabilities. A catastrophic failure now requires simultaneous hardware or software failure across multiple distinct physical locations and distinct software stacks.

## IX. System Implementation

All software and CAD is available in this git repository : *https://github.com/Gregification/BeeMS/tree/main*

The SBM IC used is the BQ7697204PFBR (BQ769xxx family) - a non-daisy chainable, 3-16 cell, 0-7 onboard thermistors, and onboard current sensing & column counting across an external user defined shunt (50uΩ minimum). This was chosen because it provides all DAQ necessary for module level fuel gauging when in independent operation. The MCU use is the MSPM0G3507-Q1 of the MSPM0 series, this was chosen because of its competitive cost/power/compute performance (~$3 single, ~50mA, 80Mhz w/ NPU and math accel), integrated CAN-FD controller, and 1-to-1 code portability across all MSPM0 series MCUs of varying peripherals and performance.

### *A. PMU*

Referred to in documentation as the 'Master board', the PMU acts as the vehicle interface to the BMS. The onboard MCU interfaces vehicle interlock, CAN bus, and the inter-module CAN bus. The module side CAN bus is electrically separate and logically distinct from the vehicle CAN bus for bandwidth and safety.

The PMU is the means of runtime pack control and monitoring; it provides the control logic necessary for SoC modeling, software fault detection, and pack operation. The Master board enforces safety interlocks and reports the state of the vehicle over a separate CAN bus and the Modbus/Ethernet interface. The MSPM0G3507-Q1 utilizes FreeRTOS for resource efficiency and development simplicity.

The primary safety output of the PMU is the grounded low voltage (GLV) interlock relay, which gates power delivery to the vehicle's drivetrain. This interlock is driven by a high drive, active-low GPIO pin read back by a separate sense pin. This enables the firmware to verify a write has been completed. The GLV interlock relay can be overridden by the user for debug purposes. Otherwise, the interlock will respond

to the software state, opening the relay if any fault condition is detected anywhere in the system.

In order to communicate with each MMU, the Master board uses a 29 bit can ID and CAN-FD at a arbitration rate compatible with normal vehicle operations (500kbs) and a data rate of 2Mbs; datarate chosen as a compromise between necessary band-width - experimentally determined - and low enough frequency that no special considerations were needed in network layout. The inter-module CAN packets abide by the J1939 standard. Each ID space is partitioned by a custom field split into four channels. There is a MMU to PMU, PMU to MMU, Modbus, and broadcast channel. Each MMU has its own unique address that the Master board maintains in a stored array. If the Masterboard somehow receives a message from an unknown address, the data is discarded. Under the MMU to PMU channel, the MMU can transmit three different packet types. CELLV packets contain per-cell voltages (mV). CELLT packets contain per-cell temperatures in decimal degrees Celsius. Finally, STATUS1 packets contain data reflecting MMU operating state, pack stack voltage (cV), IC safety status, and an error flag. CAN messages are processed every 5ms (200 Hz).

The current of the pack is measured by a LEM DHAB S161 dual-range Hall-effect transducer, read by the Master board's two 14b >100ksps ADCs (MCP33151). This is the same hall sensor used by the OrionBMS, intention being minimal swap in cost. The hall effect outputs are a mixed current device and as such has notable signal variations on the mixed quality hand assembled PCB's; in our experimentation we have also found a significant variation in the idling voltage after a power cycle of the PMU - cause undetermined but suspected assembly issue. This is resolved by 'zeroing' the reference; though this is a temporary solution for the purposes of prototyping. The signal is fed through a passive RC filter with a 4.7kHz -3db cutoff - for general system noise - and a single stage digital IIR filter (~4% cutoff) was manually tuned as a follow up. For testing purposes the effective filtering preformed is a ~191Hz low pass; tuned to best match PMU samples with oscilloscope measurements across an inline shunt. In a proper implementation the cutoff frequency should have a close relation to the sampling required by the pack modeling algorithm. Though note that the primary noise source is the vehicle inverter, in the case of E25, its nominal 250kHz, and lithium cell chemistry has a limited frequency response in terms of current delivery, ~<100kHz; therefore any signals >100kHz were not considered useful for our purposes in coulomb counting.

A hard overcurrent occurs when an instantaneous current exceeds the max current value. In response, a fault is triggered and the interlock opens. A surge overcurrent occurs when a current within permitted amperage is measured past *maxTime* milliseconds. If a current is considered to be oversurging, a fault is triggered and the interlock fails.

The only analog fail safes used are an isolated unidirectional analog coupling of the inter-module interlock to the vehicle interlock, and an external watch dog IC monitoring the MCU. Both of these signals are capable of overriding software control of the vehicle interlock; and both analog signals, and the one software controlled signal must indicate safe system state for the vehicle side interlock to engage.

The software interlock control is representative of all other fault conditions. When a fault occurs, the PMU will detect if the system transitioned from healthy to faulted. If so, a snapshot of when the system faulted and other data is recorded to onboard non volatile storage for user analysis. There is intentionally no runtime fault clear path. Once the interlock is open, the only recovery is a system restart. This is a deliberate system design choice made to comply with safety constraints.

### *G. Module*

Responsibilities are module level DAQ - cell voltage, cell temperature, module level cooling, etc - breaking the inter-module interlock during critical failure of the module, and providing a CAN interface for BMS operations and abstract register based interface.

All hardware except the SBM IC is powered by doubly isolated vehicle LV - such isolation is not required. An isolated 2MHz spi bus is used to join the SBM with the MCU. An onboard dc-dc converter provides the necessary power and protections required to operate on LV between 8V to 35V. The analog interface being the shared intermodule interlock, and inter-module CAN bus.

The MMU MUC is identical to the one used on the PMU, but with a smaller footprint; it similarly runs FreeRTOS and similar software stack. Its primary process is to perform DAQ,monitor cell voltage, cell temperature, engage module cooling, perform passive cell balancing when requested, and throw the inter-module interlock. Because signaling a failure on the inter-module interlock means shorting to ground; because it's a milli-amp current source supplied by the PMU, this can be achieved with very small form factor solid state relays (SSR). Our implementation used a normally open (NO) SSR though a through NC SSR would have a more appropriate failure mode.

## ACKNOWLEDGMENT

Special thanks to professor Robert L. Woods and Gregory K. Turner for supporting our efforts in this project; and to the UTA Racing TSBP (Tractive System Battery Pack) team for data and cooperation.

## REFERENCES

[1] S. Steinhorst *et al.*, "Distributed reconfigurable battery system management architectures," in *Proc. 21st Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Macao, 2016, pp. 429–434, doi: 10.1109/ASPDAC.2016.7428050. Available: https://s-steinhorst.github.io/PDF/2016-ASPDAC-Distributed%20Reconfigurable%20Battery%20System%20Management%20Architectures.pdf

[2] M. Sunna, "Cell Balancing With BQ769x2 Battery Monitors," Texas Instruments, Dallas, TX, USA, Appl. Note SLUAA81A, 2021. [Online]. Available: https://www.ti.com/lit/an/sluaa81a/sluaa81a.pdf

[3] F. Rizzoto, "Battery Management System for a Formula SAE Car Concept," M.S thesus, Dept. Inf. Eng., University Padova, Padua, Italy, 2022. [Online]. Available: