Whitepaper

Version: V1.0
Date: 2026-06-18

# Software-free battery pack BMS application




## Abstract

The core architecture of a software-free battery pack battery management system (BMS) can be summarized as “domain or zonal controller + software-free (SWF) gateway + analog front-end (AFE)/battery junction box (BJB)/pack sensor.” The domain controller communicates with the SWF gateway through CAN or Ethernet, while the dedicated SWF gateway connects to and forwards data among the AFE, high-speed data (HSD)/ low-speed data (LSD) connectors, EEPROM, security devices, pressure sensors and other in-pack resources.

In high-voltage applications, the software-free approach migrates most decision-making functions of the traditional local battery management unit (BMU) microcontroller (MCU) to the domain controller, while retaining only the necessary measurement, communication-bridging and actuator-interface capabilities inside the pack. In this way, the BMS evolves from a “local independent controller” into a “high-reliability sampling and actuation node managed by the domain controller.”

In low-voltage applications, the same concept can also be applied to low-voltage BMS (LVBMS) by replacing the traditional MCU + system basis chip (SBC) structure, covering 12 V, 24 V and potentially 48 V battery-management scenarios.

Therefore, the key conclusion of this white paper is that software-free battery pack BMS architectures are well suited to systems with clearly defined functional boundaries, high-cost sensitivity and an overall domain-controller-based architecture. For traction batteries, high-voltage energy storage and systems requiring high-level functional safety as well as network/data feedback loops, the recommended architecture is "software-free battery pack BMS + domain-controller supervision." This application remains highly consistent with the functional distribution of current mainstream devices and complete high-voltage BMS reference designs.

## Concept definition and applicable boundaries?

### What is a "software-free battery back BMS"?

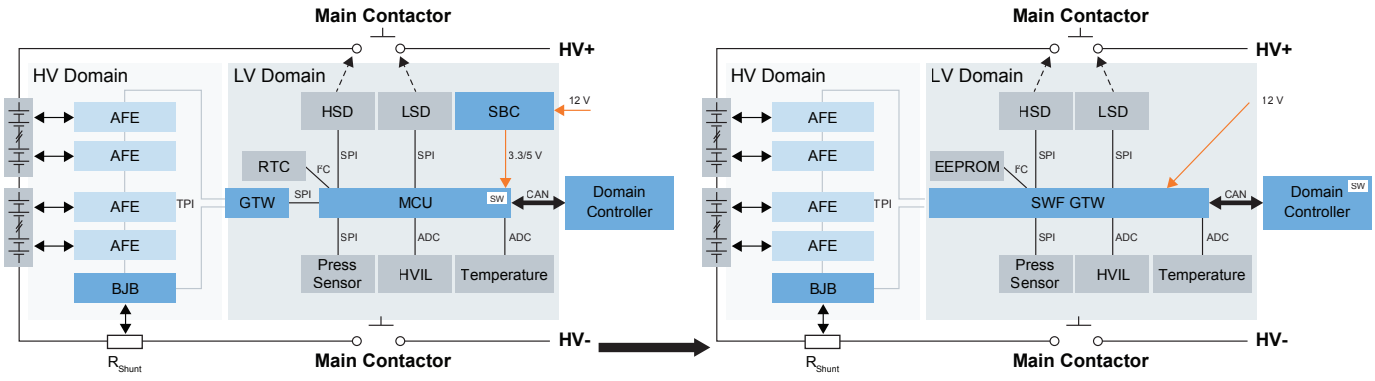
A software-free battery pack BMS refers to an architecture in which no user-developed MCU application-layer or control-layer software is deployed inside the battery pack. Instead, the pack mainly relies on the hardware protection chain, state machines, comparators, OTP/EEPROM parameter tables and fixed logic inside BMS chips to perform core monitoring and protection actions. It is usually not "zero digitalization," because some chips still contain registers, digital filters, analog to digital converters (ADCs) and configurable logic; however, it does not rely on continuously running user firmware for closed-loop control. NXP clearly states that no MCU or related power-supply bill of materials (BOM) is required inside the battery pack, while an external MCU remains optional. This not only significantly reduces the cost of the in-pack BMS per set, but also simplifies the cumbersome process of software debugging, programming and flashing inside the battery pack.

### Differences from a traditional BMS

A traditional BMS, especially in electric vehicle (EV) and energy storage systems (ESS) applications, generally not only monitors voltage, current and temperature, but also performs state of charge (SOC), state of health (SOH), state of power (SOP), state of life (SOL) and other estimations, balancing-strategy optimization, thermal-management coordination, fault classification, logging, network communication, diagnostic services, over the air (OTA) and functional-safety diagnostics. NXP defines a complete typical BMS structure as BMU (MCU + SBC) + BJB + cell management unit (CMU) with an AFE, accompanied by software drivers and safety documentation.
By contrast, the main difference of a software-free battery pack BMS is that, while maintaining the complete functionality of a conventional BMS at the system level, the MCU and software portion are moved upward. There is no computing unit inside the battery pack; only execution and sampling functions are retained.

## Industrial value of a software-free battery pack BMS

NXP summarizes the value of software-free BMS as lower cost, reduced complexity, improved reliability and reduced software-maintenance burden. This indicates that the goal is not "functional reduction," but "functional repositioning." It aligns with centralized computing architectures in vehicles and equipment, with the domain controller taking on more battery-management decisions.




Current BMS Arch SWF BMS Arch

Please provide title

### Value 1: Lower development complexity and shorter time to mass production

In scenarios where complex algorithms and in-vehicle networking are not required, a software-free architecture can significantly reduce the workload associated with MCU firmware development, bootloaders, communication protocol stacks, software testing and software functional-safety processes. Protection thresholds, delays and recovery mechanisms can be stored in EEPROM, while basic pack control is automatically executed by internal state machines. Autonomous balancing and protection actions are also supported, enabling mass production with less software involvement. The number of devices is reduced, MCUs and SBCs are minimized, the burden of in-pack software version management is lowered and the manufacturing process becomes more suitable for automated production lines, reducing flashing, version-proofing and manual intervention.

### Value 2: Low standby power, simple structure and more controllable BOM

A software-free solution typically does not require a continuously running MCU or external high-frequency peripherals, so standby power consumption can be very low. In general, the sleep/periodic-check mode current of a software-free battery pack BMS can be as low as 60 µA. NXP provides multiple low-power modes, with typical current below 5 µA in deep-sleep mode. These characteristics are highly attractive for voltage battery pack applications such as 12 V, 24 V and 48 V automotive battery packs, as well as small ESS systems.

### Value 3: More direct hardware protection path

NXP points out that placing more fault-control execution on the AFE side rather than in the MCU side can shorten the response path and reduce MCU resource consumption. Advanced AFEs directly use ADC readings and configuration values to detect faults and then execute protection through hardware, such as MOSFET shutdown during overcurrent, HSD and LSD control, insulation detection and high-voltage interlock detection.

## Technical capability boundaries of software-free BMS

### What it can do

A software-free battery pack BMS can usually perform the following tasks reliably:

- Cell voltage, pack voltage, current and temperature acquisition; threshold-comparison protection; and cell balancing
- Abnormal-condition detection and hardware shutdown for overcharge, overdischarge, overcurrent, short circuit, overtemperature, under-temperature and open-wire faults
- Sensor acquisition for insulation detection, high voltage interlock loop (HVIL) detection, pressure, crash signals, coolant temperature, and related signals
- High-side or low-side MOSFET control and EEPROM data storage
- Chemical fuse/secondary protection-chain driving, fault self-recovery and low-power sleep
- Fast/slow charging control and battery-heating control

### What it cannot replace

The limitations of software-free BMS mainly lie in advanced state estimation and system coordination:

- Accurate SOC/SOH/SOP estimation usually requires cell models, synchronized voltage/current sampling, temperature compensation, SOC coulomb counting, open-circuit voltage (OCV)/model/filtering algorithms and related mechanisms. High-accuracy SOC/SOH implementation relies on MCU and algorithm cooperation rather than simple threshold protection.
- Complex communication functions, such as unified diagnostic services (UDS), remote upgrades, log traceability and cloud-platform data, are generally not strengths of a software-free battery pack BMS. NXP high-voltage solutions treat these system-level capabilities as standard functions of the upper-layer controller above the software-free battery pack BMS.

## Target application scenarios

### Highly suitable scenarios

Software-free battery pack BMS is generally recommended for the following scenarios:

- High-voltage traction battery packs (400 V/ 800 V EV), compatible with BMU/CMU/BJB layering, ASIL architecture, software drivers and system-level communication, with SOC/SOH estimation models and algorithms moved upward
- Energy storage systems (ESS), compatible with complex thermal management, parallel-system coordination, insulation monitoring, alarm classification, station-level EMS communication, and operations and management (O&M) data feedback loops
- Distributed BMS for commercial vehicles: even if the overall battery system ultimately has a master controller, hardware-autonomous monitoring can first be used on slave boards, removing the slave-board MCU and moving basic software and algorithms upward. NXP supports this layered implementation of "autonomous monitoring + master-controller supervision"

### Other suitable scenarios

Software-free battery pack BMS is also suitable for the following scenarios:

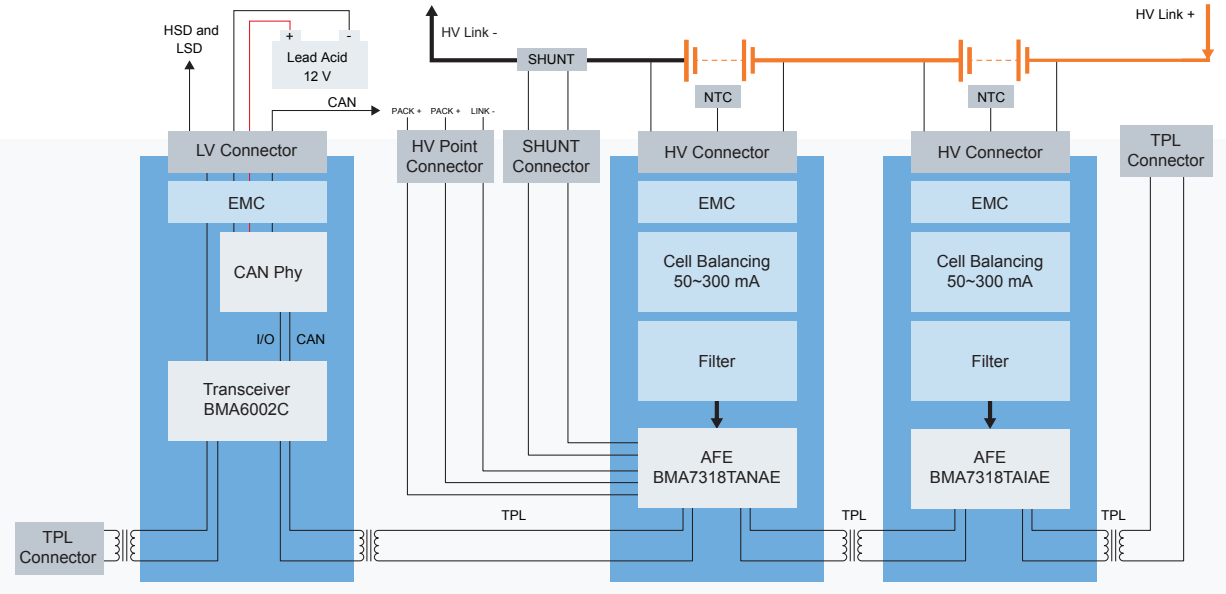
- Power tools, garden tools, vacuum cleaners, wearable/portable devices: medium-to-low cell counts, strong cost sensitivity and protection requirements exceeding complex-algorithm requirements
- Light electric vehicles (E-bike/E-scooter/LEV): requiring voltage/current/temperature protection, basic balancing and low-power standby, but not necessarily traction-battery-level complex state estimation
- Backup power, uninterrupted power supplies (UPS), medical portable devices and small energy-storage modules: emphasizing reliable protection and ultra-low-power static strategies

## Typical technical architecture

For customers that have already planned a centralized computing platform, want to reduce in-pack software versions and are willing to let the domain controller assume more BMS decision-making responsibilities, NXP's software-free battery pack BMS solution is particularly attractive.

From the perspective of customer-scenario fit, it is recommended to prioritize the following projects: first, high-voltage traction battery-pack projects that need to simplify BMU software and hardware complexity; second, low-voltage or auxiliary battery-management projects that aim to implement basic BMS functions with a lower BOM; and third, projects with a high degree of automated manufacturing that wish to reduce production-line software flashing and version-management costs.

A suitable BMS system architecture or software-free battery pack BMS architecture should be selected based on specific requirements such as system functional-safety needs, communication-failure risk assessment and whether the contactor requires strong MCU closed-loop control.



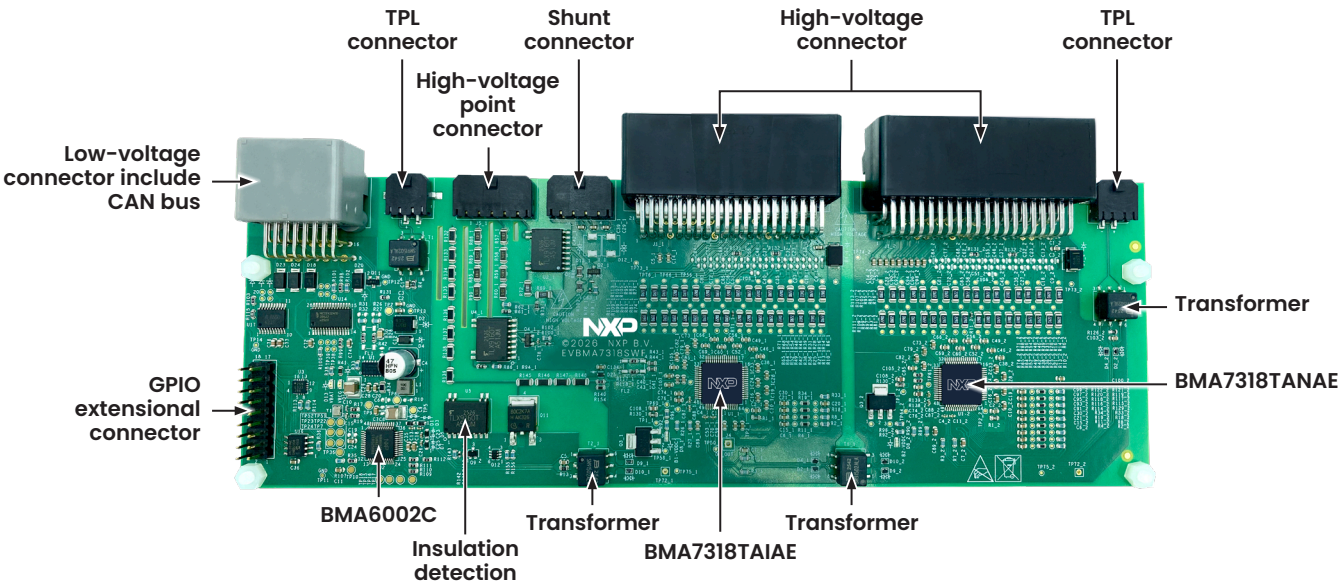



Figure 3. NXP software-free evaluation board based on KIT7318SWFEVB

## Simplified software-free solution

The first category is the "simplified software-free solution," whose goal is extreme cost reduction: the first AFE also undertakes part of the BJB functions, isolated power supplies are minimized as much as possible and EEPROM or a secure integrated circuit (IC) is used to store shunt calibration data, manufacturing information, factory security keys, battery-status data and event logs.

The following figure shows a typical simplified software-free BMS solution based on the NXP BMS kit, including BMA6002C gateway (GW), BMA7318TAIAE battery cell controller (BCC) and BMA7318TANAE BCC.

- EEPROM/security chip is used to store shunt resistor calibration data, manufacturing information, factory security keys, battery-status data and event logs.
- Insulation detection: the IO of the BMA6002 GW controls the opt-MOS, while the AFE analog channel (AIN) is used for measurement.
- BJB functions are implemented; the first AFE has slightly higher power consumption, but provides better voltage/current synchronization performance.
- The BMA6002 GW can be powered by external KL30 (clamp 30) and supports KL15 (clamp 15) wake-up.
- Internal signals or other LSD and HSD peripherals are handled through an IO expander.
- The entire system has no MCU and external communication is performed through the CAN interface of the BMA6002C GW.

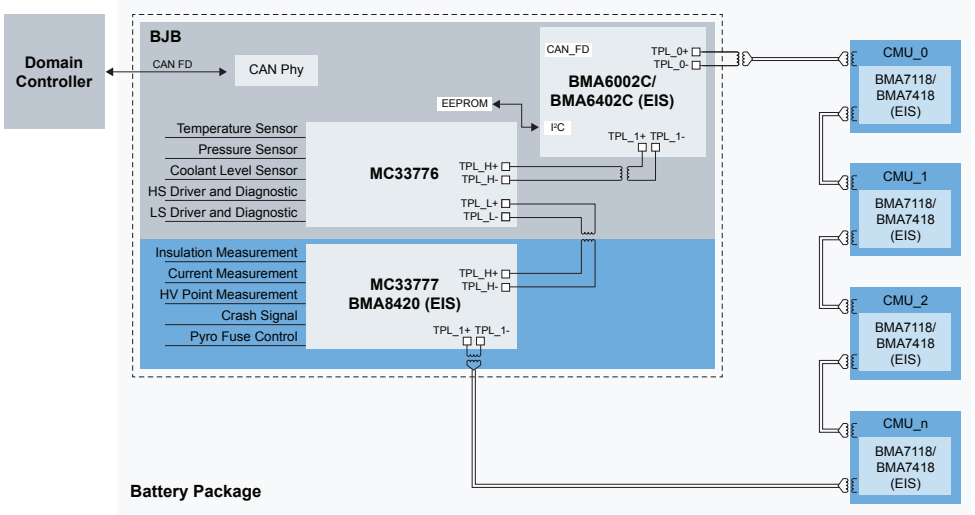



Figure 4. System block diagram of the fully-featured software-free solution

### The fully-featured software-free solution

The second category is the "fully-featured software-free solution," whose goal is to retain more high-voltage battery-pack-level BMS capabilities, such as Pyro Fuse support, insulation detection, pressure-sensor interfaces, secure IC, more ADC/GPIO resources and even different combinations with or without Electrochemical Impedance Spectroscopy (EIS) features. For more details about EIS visit nxp.com/BMS-EIS.

If a software-free + EIS solution is selected, the recommended NXP kit combination is BMA6002C GW, BMA8420 EIS BJB, BMA7418 EIS BCC and MC33776 BJB. If EIS is not required, the kit combination BMA6002C GW, MC33777 BJB, BMA7118 BCC and MC33776 BJB can be used.

#### Advantages

- Supports Pyro Fuse driving.
- Functional-safety target is ASIL D; the gateway supports end-to-end (E2E) communication (QM) and MC33776 BJB supports E2E checking and ASIL D.
- Supports EIS and is pin-to-pin compatible with non-EIS chips.
- Supports crash-triggered Pyro Fuse.
- Supports insulation detection.
- Supports pressure sensors through SPI interface.
- Supports data security through a security chip via SPI interface.
- The BJB provides 16 ADC/GPIO pins plus 5 additional GPIO resources.

#### Limitations

- PWM signal acquisition is not yet supported for HVIL and sensors.
- The gateway has only one SPI master port, creating common-cause-failure risk for connected devices.
- It still relies on intelligent HSD/LSD for contactor control and hold functions.
- IO resources are limited, requiring an additional MC33776 BJB for IO expansion.
- The isolated DCDC output needs to be controlled within the 21–25 V range.
- The SPI master supports parity checking, but its functional-safety capability is limited.
- CAN bus open circuit will cause the battery pack to lose control.

## Key hardware design points

### The contactor control chain must be defined first

The most critical point in a software-free BMS architecture is not sampling accuracy, but "who makes the final decision when communication is lost."

### In-pack storage responsibilities must be clear

Even if local software is greatly reduced, requirements for data integrity, traceability and secure storage still remain.

### Communication design must follow functional-safety thinking

Communication must be handled with functional-safety thinking and not only by considering bandwidth. Redundant CAN FD, E2E checking, parity checking, fault-interrupt inputs and special reset commands should be incorporated into system design as early as possible.

### I/O and peripheral resources must be evaluated at system level

Because software-free battery pack BMS is a new architecture and must also cover almost all BMU functions in a traditional BMS architecture, peripheral resources such as I/O must be re-planned. Software-free BMS solutions can easily reach the limits of GPIO, SPI master, PWM and sensor interfaces, so temperature, HVIL, pressure, insulation, contactor, precharge and fault-input resources must all be included in the resource list early in the project.

### Keep "safety actions" on the pack hardware side

Overcharge, overdischarge, short circuit, overcurrent, overtemperature, open-wire and FET failure protections should be executed by hardware driven by BMS monitoring chips as much as possible. The most critical safety actions should not rely on polling and communication from an external controller.

### Leave "intelligence and data" to the upper-layer domain controller

If the system needs truly usable SOC/SOH, logs, connectivity or customer visualization, these functions should not be forced into the software-free battery pack. They should instead be assigned to the upper-layer host or system controller, providing better maintainability and algorithm-upgrade space. Advanced functions are generally handled by upper-layer processors.

### Select device parameters properly

Basic BMS functional parameters such as voltage-sampling accuracy, current-sampling accuracy, temperature-sampling accuracy and device functional-safety level must be selected reasonably according to system requirements. In addition, balancing current, balancing resistance, measurement timing and the number of simultaneously balanced cells affect device temperature rise and measurement accuracy, so balancing design must consider thermal behavior, accuracy and time together.

### Production-line configuration and misconfiguration prevention must be institutionalized

Software-free battery pack BMS solutions rely heavily on local OTP/EEPROM/register configuration, production-line programming, version management, threshold tables, sampling verification and mismatch-proofing. These must be guaranteed through institutionalized processes.

## Functional safety and regulatory compliance

### Functional-safety design

The IEC/ISO partnership defines functional safety as reducing risk to a tolerable level by means of safety mechanisms in electronics and related software. IEC 61508/ISO 26262 provide a full-lifecycle framework for E/E/PE systems and define SIL 1–4/ASIL A–D. This means that as long as a system carries safety functions, it should be designed from the perspectives of hazard analysis, risk assessment, system decomposition, diagnostic coverage and verification/validation, rather than merely considering "whether there is an MCU."

The functional-safety design focus of a software-free battery pack BMS is how to ensure reliable communication and functional safety of contactor control. Other aspects, such as voltage, current, and temperature sampling, can rely on the functional-safety mechanisms inherent in AFE/BJB and related chips.

When there is no local MCU, functional-safety assurance for HSD and LSD mainly relies on the following measures: 1) for HSD controlled through SPI communication, SPI parity checking is mainly used to ensure reliable and correct command delivery; 2) for I/O-controlled LSD, the upper controller reads back the configuration result and performs secondary verification.

For daisy-chain communication (TPL) inside the battery pack and CAN communication out of the battery pack, E2E (end-to-end) checking is adopted, enabling an ISO 26262 ASIL D level.

However, in a battery pack without a local MCU, once CAN or Ethernet communication between the pack and the outside is lost, causing loss of communication with the upper-layer domain controller or controller, functional safety inside the battery pack will be affected.

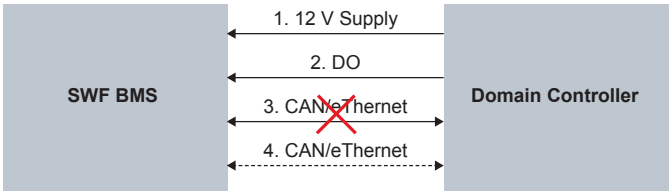



**Figure 5. Communication loss between software-free BMS and domain controller**

**For the above issue, NXP proposes the following solutions:**

| Safety goal | Safe state | Safety level | Workaround |
|---|---|---|---|
| Avoid thermal runaway caused by overcharge | Open the contactor | ASIL C | When CAN/Ethernet communication is interrupted but the system still needs to open the contactor:<br>1. ② (In Fig 5) DO directly drives the HSD to open the contactor<br>2. Turn off the ① 12 V power supply<br>3. Use redundant CAN/Ethernet to maintain communication |
| Avoid thermal runaway caused by overdischarge | Open the contactor | ASIL C | |
| Avoid thermal runaway caused by overtemperature | Open the contactor | ASIL C | |
| Avoid thermal runaway caused by overcurrent | Open the contactor | ASIL C | |
| Avoid unintended contactor opening | Contactor hold (remain closed) | ASIL A | When communication is lost but the system needs to keep the contactor closed:<br>1. ② DO directly drives the HSD to control contactor state<br>2. Keep the BMS chip state unchanged for 10 seconds<br>3. Use redundant CAN/Ethernet to maintain communication |

### Automotive regulatory constraints

UN regulation No. 100 sets clear requirements for vehicle electric-drive systems and rechargeable energy storage systems (REESS) safety. Its annexes cover test items such as vibration, thermal shock and cycling, mechanical shock, mechanical integrity, fire exposure, external short circuit, overcharge, overdischarge, overtemperature and overcurrent, and define key terms such as REESS, SOC, thermal runaway and thermal propagation.

When adopting a software-free battery pack BMS architecture, the battery pack must still meet regulatory requirements in terms of electrical safety, thermal safety, mechanical safety and abnormal-condition testing. Software-free does not mean lower testing requirements; on the contrary, if software is absent, hardware design and fault-tree closure must be even more robust.

## Recommended devices

### Software-free GW

#### MC33665 TPL transceiver

- On the MCU side, it supports CAN/CAN-FD (5 Mbps max), SPI (10 Mbps max) and UART (6 Mbps max);
- Four TPL interfaces
- Eight digital I/Os compatible with I²C master interface
- Built-in FIFO
- Compatible with ASIL D system design
- BMA6002C/BMA6402C gateway
- On the MCU side, it supports CAN/CAN-FD (5 Mbps max)
- Two TPL interfaces
- Eight digital I/Os multiplexable as I²C master and SPI master interfaces
- Built-in FIFO
- Compatible with ASIL D system design
- BMA6402C GW is the version compatible with EIS solution

### AFE

#### BMx7318 battery cell controller

- 4–18 channels
- 10/12 GPIOs, multiplexable as I²C master and over-current protection path
- High-accuracy cell-voltage and current sampling
- Passive balancing compatible with external balancing
- 2 Mbps TPL communication and SPI communication compatibility
- ASIL C functional safety level

#### BMA7118/7418 battery cell controller

- 8–18 channels
- 9 GPIOs, multiplexable as I²C master and SPI master interfaces
- High-accuracy cell voltage sampling
- Passive balancing compatible with external balancing
- 2 Mbps TPL communication and SPI compatibility
- ASIL D functional-safety level
- BMA7418 BCC is the version compatible with EIS solution

### Battery junction box

**MC33777/33776 battery junction box**

- 16 GPIOs, multiplexable as I²C master and SPI master interfaces
- Four current-sampling channels
- 2 Mbps TPL communication and SPI compatibility
- ASIL D functional-safety level
- MC33777 BJB has a Pyrofuse driver, while MC33776 BJB does not

**BMA8420 battery junction box**

- 16 GPIOs, multiplexable as I²C master and SPI master interfaces
- Four current-sampling channels
- 2 Mbps TPL communication and SPI compatibility
- ASIL-D functional-safety level
- Includes a Pyrofuse driver
- BMA8420 BJB is the version compatible with EIS solution

## Concept of a dedicated software-free GW

In software-free battery pack BMS applications, the software-free gateway plays a crucial role in the system architecture. Without an MCU, it must assume almost all MCU functions except computation, support data routing to the outside and support direct CAN or Ethernet output. Because the communication data volume is generally large, sufficient bandwidth is required while maintaining adequate real-time performance.

To meet the demand for next-generation BMS evolution toward centralized E/E architecture, NXP proposes a new SWF gateway concept. By introducing dedicated functional modules, it comprehensively improves system integration and reliability:

- Integrated dedicated BMS processing core: provides an independently running BMS functional unit inside the domain controller, supporting key measurement and control tasks and reducing dependency on central vehicle computing resources.
- Redundant CAN FD communication architecture: supports dual CAN FD design to realize:
  - Open-load detection
  - Communication redundancy
  - High-bandwidth data transmission, improving functional-safety level and communication reliability.
- Multi-channel high-accuracy data acquisition (ADC): integrates ADC modules to support unified acquisition to improve measurement consistency and system integration.
  - Pack voltage
  - Temperature
  - HVIL
  - Various analog sensor signals
- Peripheral interfaces and actuation capability: the SWF gateway provides rich standard interfaces for direct control of key pack actuators and peripheral devices.
  - The SPI master interface connects peripherals such as HSD, LSD and pressure sensors.
  - The I²C master interface supports EEPROM and security chips.
  - The PWM control module supports HVIL signal-integrity monitoring, analog/digital sensor driving and flexible modulation.
- General-purpose IO and wake-up management
  - Supports multi-source wake-up detection
  - Configurable IO levels and IO retention during reset, effectively supporting low-power scenarios and system-state management.
- Functional safety and diagnostics: for automotive-grade system design, the SWF gateway should provide multi-level safety and diagnostic mechanisms.
  - Fault-interrupt input and fast-response mechanism: supports external failure-interrupt input and realizes fast direct response to the domain controller through a hardware path, reducing latency.
  - CAN-based remote diagnostics and control: supports device reset and state control through dedicated CAN commands, improving maintainability and remote diagnostic capability.
- Supports TPL communication for AFE loopback communication and BJB communication.

## System reference verification

### Electrical verification

- Cell over-voltage (OV)/under-voltage (UV) thresholds, tolerances, hysteresis and debounce time.
- Over-current protection (OCP)/short-circuit detection (SCD) trip thresholds and repeatability under different temperatures.
- FET turn-on/turn-off, pre-charge and abnormal-recovery behavior.
- Balancing current, balancing thermal behavior, and measurement deviation during balancing.

### Environmental and abuse verification

- Thermal shock, vibration, mechanical shock, external short circuit, overcharge, overdischarge, overtemperature, fire exposure, etc., should be performed according to the standards of the target market. The annexes of UN R100 clearly provide the REESS safety-test framework.

#### Functional Safety and failure analysis

- Failure mode and effects analysis (FMEA/FMEDA).
- Real-time performance and reliability of data communication.
- Diagnostic strategies for open wire, sampling drift, FET sticking, temperature-probe failure, reference-source abnormality and power loss.
- Redundancy of critical protection chains and fault-injection testing. The IEC functional-safety framework emphasizes full-lifecycle risk analysis and verification/validation.

## References


- NXP BMx7318 BCC Datasheet
- NXP BMA7118 BCC Datasheet
- NXP BMA6002/BMA6402 GW Datasheet
- NXP MC33665 TPL transceiver Datasheet
- NXP MC33777/MC33776 BJB Datasheet
- NXP BMA8420 BJB Datasheet
- NXP High Voltage Battery Management System (HVBMS)
- NXP EIS HVBMS reference design
- Monolithic Power Systems, How to Design a Battery Management System (BMS)
- Analog Devices, Automotive Battery Management Systems
- IEC, Safety and functional safety/ IEC 61508 overview
- UN/ECE, UN Regulation No. 100
- Analog Devices, A Closer Look at SOC and SOH Estimation Techniques for Batteries
- MDPI Batteries, State of Charge and State of Health Estimation in Electric Vehicles





Document Number: EN: SFBPBMSWP REV 0