Application Brief

Increasing Flexibility in Your Battery Management Designs With a Low-Cost MSPM0 MCU

In today's highly connected world, more products than ever rely on battery power. Batteries are everywhere, from cordless power tools to robot vacuums, and even the e-bikes and electric vehicles you see on your commute. These products and many others like them utilize battery management systems (BMS) to ensure power is safely supplied throughout the product.

A BMS is a subsystem that monitors and regulates the charging and discharging of batteries. While each BMS design will vary in components depending on the specific power requirements of the product, most designs will include at least one microcontroller (MCU). The MCU is capable of filling a variety of roles within the battery management system. In a small system the MCU may act as the main battery monitor and controller, whereas in a large system the MCU may simply interface the main processor to the other analog components in the BMS.

Low-power, cost-optimized MSPM0 MCUs can fill many roles in a BMS - delivering the high-performance processing features needed to increase system efficiency.

Figure 1. Applications With BMS

What is the role of a battery management system (BMS)?

A BMS is responsible for ensuring the safe charge and discharge of a battery within a product. A BMS is typically designed to perform the following functions:

- Protect: Incorporate a simple hardware device that responds to unsafe conditions by disconnecting the battery from the load(s).
- Monitor: Measure individual cell voltages, current and temperature, and control protections in order to ensure efficient usage of the battery.
- Gauge: Report the remaining battery capacity, runtime and state-of-charge (SoC) so users know when it is time to recharge the battery.

You can perform all of these functions using TI devices such as the MSPM0 MCUs and the BQ series of battery management products.

Why is MSPM0 suitable for BMS applications?

The main requirement for an MCU in a battery management system is that it has low power consumption. This feature allows the MCU to efficiently carry out its role in the BMS without drawing significant amounts of power from the same battery it manages. Beyond this, the requirements for an MCU change depending on the function being performed.

Small battery management systems typically reduce cost by absorbing as many functions into the microcontroller as possible. In these systems, the MCU requires a higher level of analog and software integration in order to carry out these various functions on its own. On the other hand, a more complex BMS might use discrete devices to monitor or gauge the battery and rely on the MCU to log and process data before passing it along to the main processor using UART, I2C, SPI or CAN-FD.

TI's **scalable** MSPM0 MCU portfolio features an Arm® Cortex-M0+ core, with a maximum CPU speed of 80 MHz. The pin-to-pin compatible portfolio covers from 4KB to 512KB of flash memory with optional analog integration, motor control peripherals, and CAN-FD. With extensive analog and digital integration, these MCUs can provide a **low-power**, **low-cost**, **high-performance**, **reliable** solution for battery pack designs.

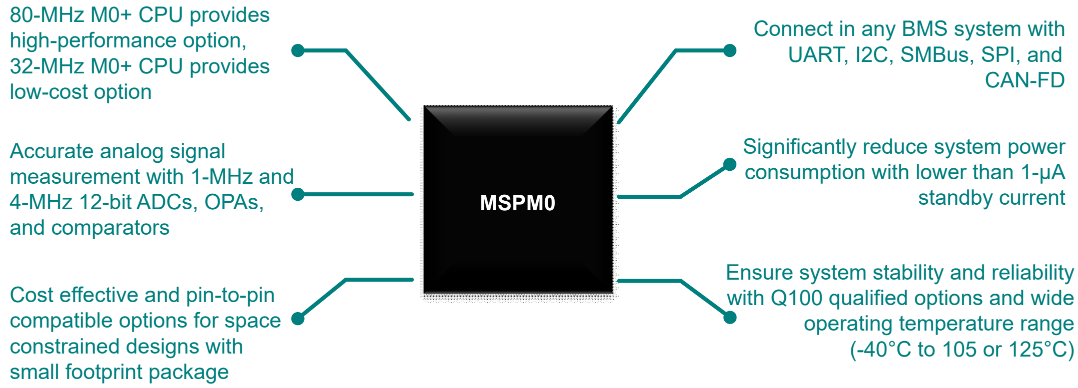

Figure 2. Advantages of MSPM0 Platform

What can MSPM0 enable in BMS applications?

For systems needing low-accuracy gauging algorithms

In some low-cost products with one to two battery cells, such as MP3 players, cameras or electronic cigarettes, the BMS is designed to protect and gauge the battery with low accuracy. This type of system typically includes a hardware protector and one, main MCU that calculates the remaining battery capacity by measuring the voltage directly. The main requirements for a microcontroller in this role are a 12-bit ADC and enough memory for the MCU to act as a battery gauge. With its low-power, low-price, broad portfolio, and high-performance features, MSPM0 MCUs are a great choice for these types of products.

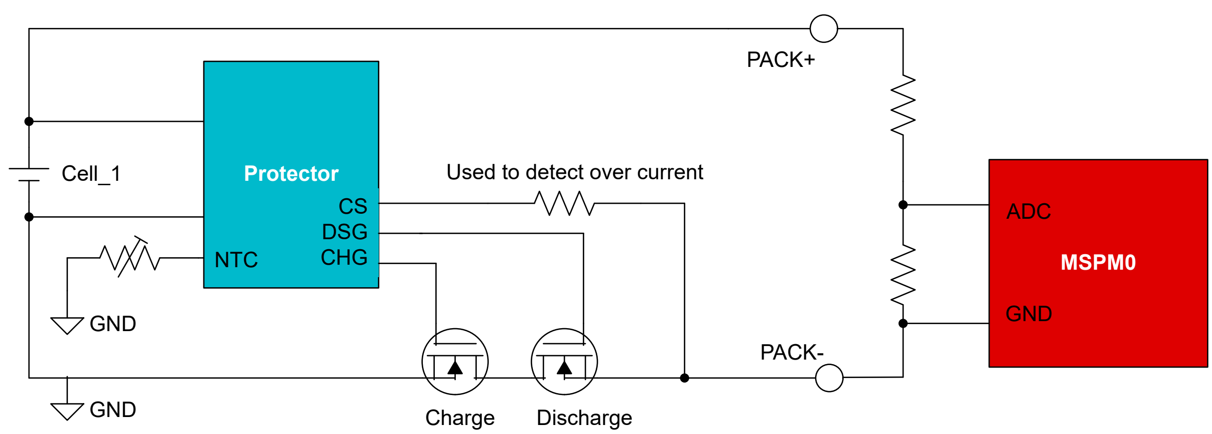

Figure 3. Block Diagram of Systems Needing Low-Accuracy Gauging Algorithms

Key feature requirements for MCUs in these applications:

- Low-power modes
- 12-bit ADC
- GPIOs

Function examples on MSPM0:

- Continuously monitor cell undervoltage, overvoltage
- Detect battery insertion/removal
- Status indication of battery (for example, LEDs)

For systems with 1-s to 6-s battery packs needing high-accuracy gauging algorithms

Some high-end products, like personal computers, require high-accuracy gauging capabilities. Typically, a discrete gauge implements this functionality. However, some systems add an MCU as an adapter between the gauge and host processor to reduce the complexity of the software design. In this case, the most important MCU requirement is to support for comprehensive communication interfaces. With support for I2C, SPI, CAN-FD, and UART interface options, MSPM0 MCUs are a great fit for this role.

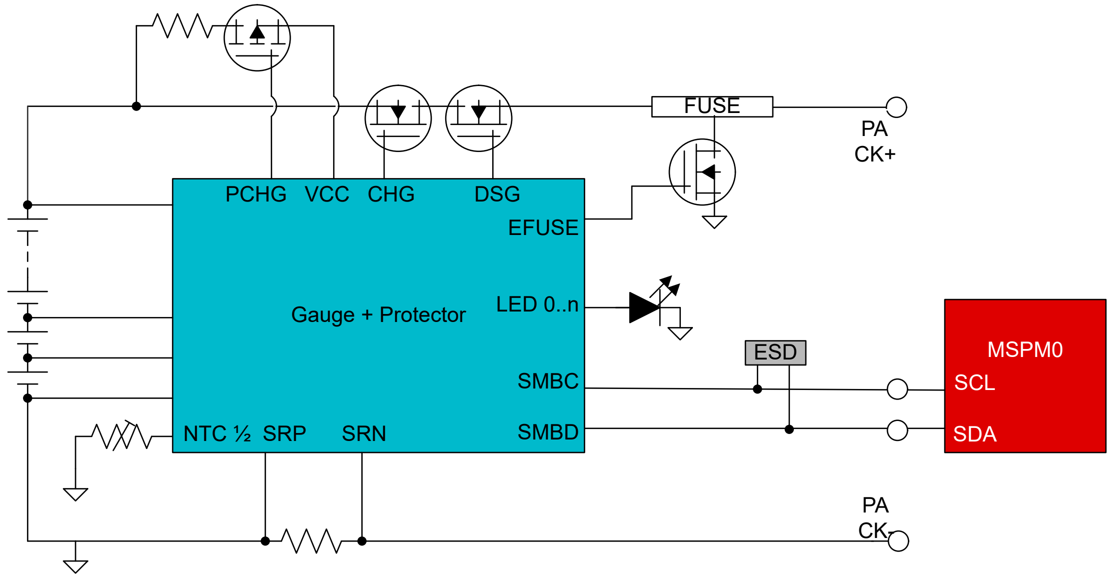

Figure 4. Block Diagram of Systems With 1-s to 6-s Battery Packs Needing High-Accuracy Gauging Algorithms

Key feature requirements for MCUs in these applications:

- Low-power modes
- Communication protocols (UART, SPI, I2C, CAN-FD)
- GPIOs

Function examples on MSPM0:

- Initialize fuel gauge IC based on battery chemistry and other characteristics
- Periodic communication of battery or fuel gauge state to the host processor using I2C, CAN-FD, UART, SMBus, or PMBus
- Support firmware updates for battery/fuel gauge
- Authenticate batteries used in the system

For systems with 6-s to 23-s battery packs needing high levels of protection

Products like power tools, e-mobilities (e-bikes, e-scooters, and e-motorcycles), uninterruptible power supply (UPS) and vacuum cleaners typically have a more complex BMS system and therefore require strong battery protection to off balance the increased risk coefficient. In these applications, battery protection is performed by a monitor, together with an MCU and a protector. The main MCU requirement for this type of system is low-power and high-performance. With the ability to handle complex gauging algorithms, the inclusion of enhanced peripherals, and a good balance of performance and power, MSPM0 MCUS are a great fit for these applications.

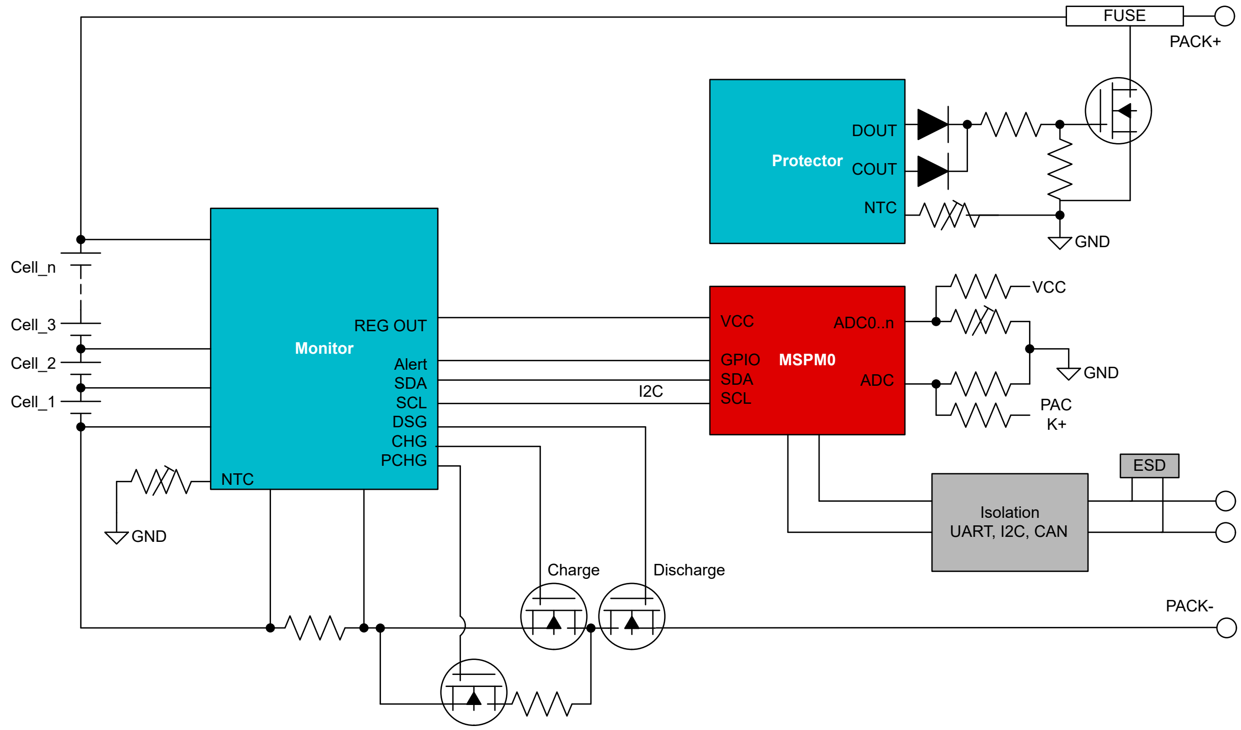

Figure 5. Block Diagram of Systems With 6-s to 23-s Battery Packs Needing High Levels of Protection

Key feature requirements for MCUs in these applications:

- Low-power modes
- Communication protocols (UART, I2C, CAN-FD)
- 12-bit ADC
- GPIOs

Function examples on MSPM0:

- Monitor safety of the battery and detect failure
- Continuously monitor cell undervoltage, overvoltage and temperature
- Continuously monitor status such as temperature, voltage, remaining capacity and average current
- Periodic communication of battery state to the host processor over I2C, CAN-FD, USB, SMBUS or PMBus

Resources

To start evaluating MSPM0 for your battery management system, order an MSPM0 LaunchPad development kit! Jump start your BMS design with MSPM0 code examples and interactive online trainings. You can also find other resources under these links:

- MSPM0 Academy
- MSPM0 landing page
- LP-MSPM0L1306 LaunchPad development kit