AN236

How MPS Fuel Gauges Work

By M. Sanchez, A. Corbella, Carter Wu, Steven Shi

May 2026

INTRODUCTION

This application note provides an overview of MPS's fuel gauges, including their operation and proper configuration. It covers MPS's standard fuel gauges (SFG) and MPS's controller fuel gauges (CFG). While both fuel gauge types share the same fuel gauge algorithm, they support different system architectures and implement different state machines. The SFG is controlled via an intermediate microcontroller unit (MCU); this MCU extracts the battery measurement data from the battery monitor and protector (an analog front-end (AFE)) and loads this information into the SFG (see Figure 1). The CFG directly controls an MPS AFE, so no intermediate MCU is required (see Figure 2). When this application note refers to commands to the CFG, this also describes the commands between the CFG and the AFE device.

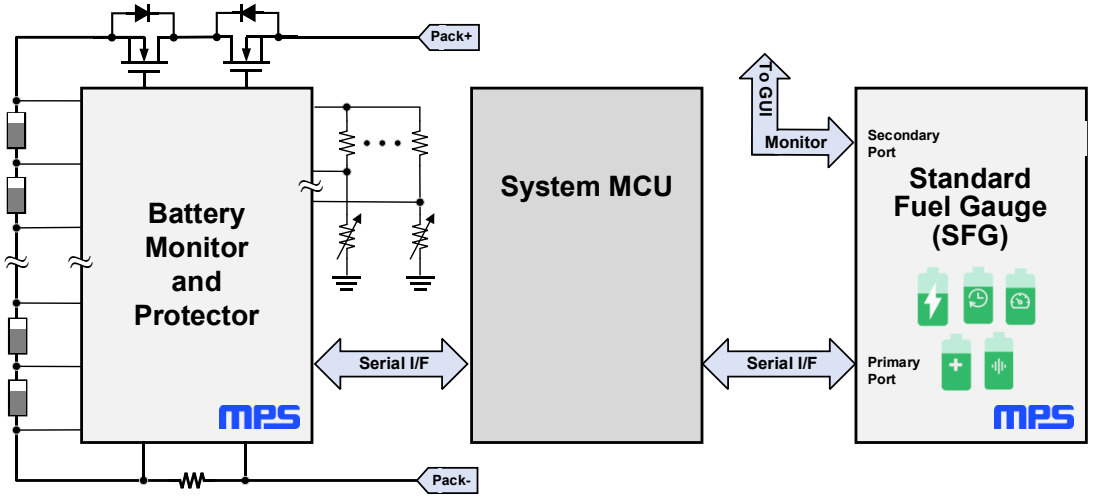

Figure 1: BMS Architecture Based on an MPS Standard Fuel Gauge (SFG)

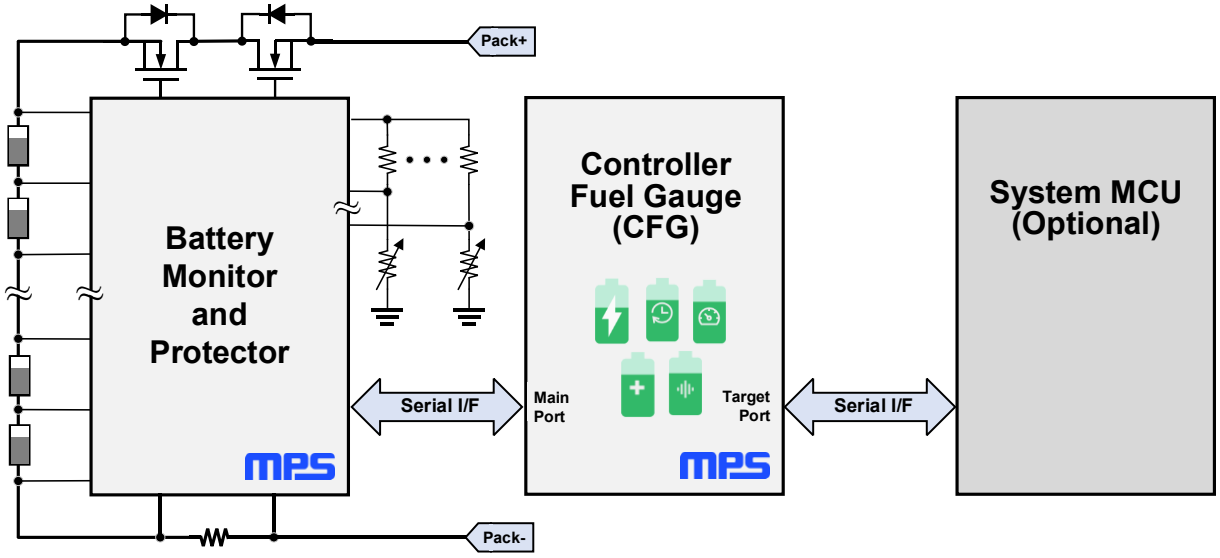

Figure 2: BMS Architecture Based on an MPS Controller Fuel Gauge (CFG)

This application note uses the nomenclature of "target" and "initiator" when referring to the two types of I²C interfaces. The SFG includes a target I²C interface that is controlled via the initiator I²C on an MCU. The CFG includes an initiator I²C interface that directly controls the target I²C on the AFE.

FUEL GAUGE AND CONTROLLER FUEL GAUGE OPERATION

Standard Fuel Gauge (SFG) Operating Modes

Figure 3 shows an SFG power state diagram. See the Communication section on page 7 for details on the process of changing power states.

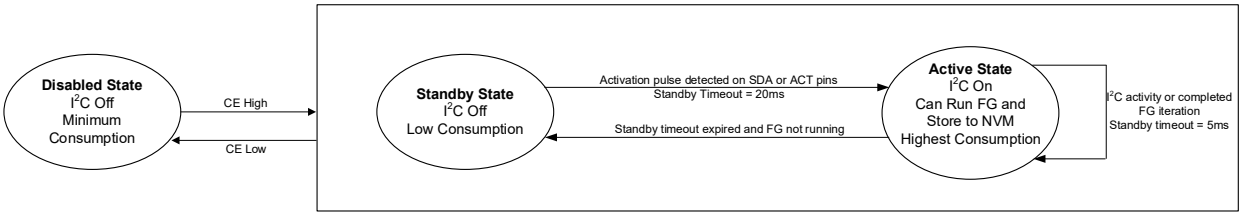

Figure 3: Standard Fuel Gauge Power States Diagram

Controller Fuel Gauge (CFG) Operating Modes

Figure 4 shows a CFG state diagram for the MPF42780. The CFG supports 10 operating modes, which can be read using the BMS_STATUS register.

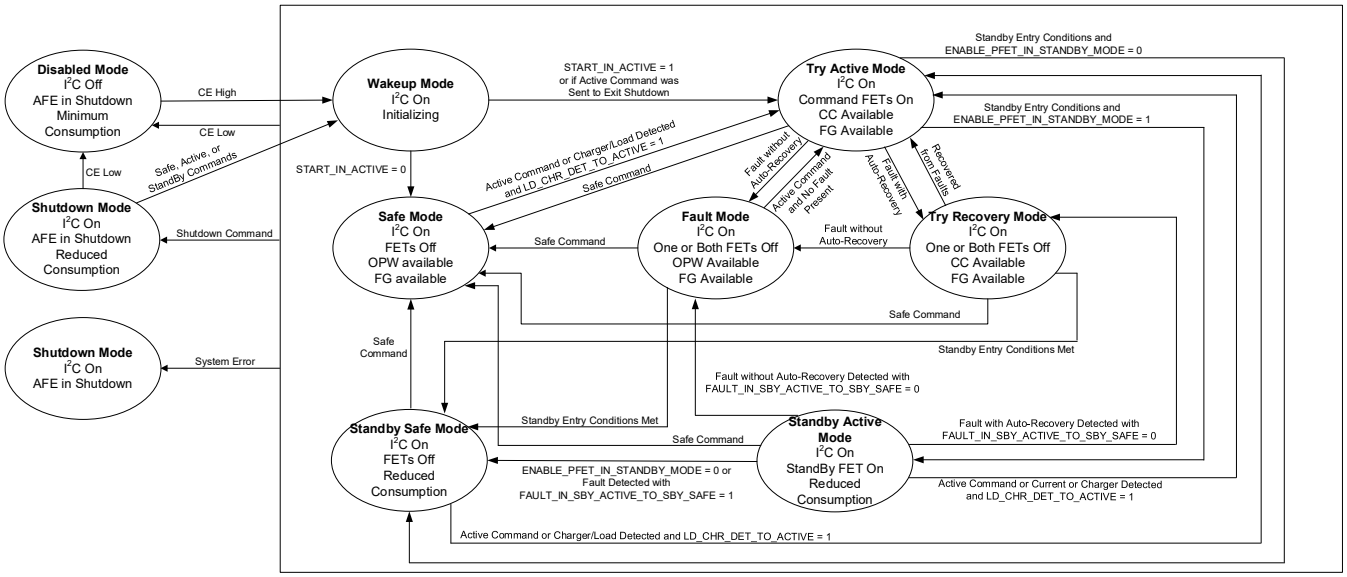

Figure 4: MPF42780 State Diagram

Figure 5 on page 4 shows a CFG state diagram for the MPF42781.

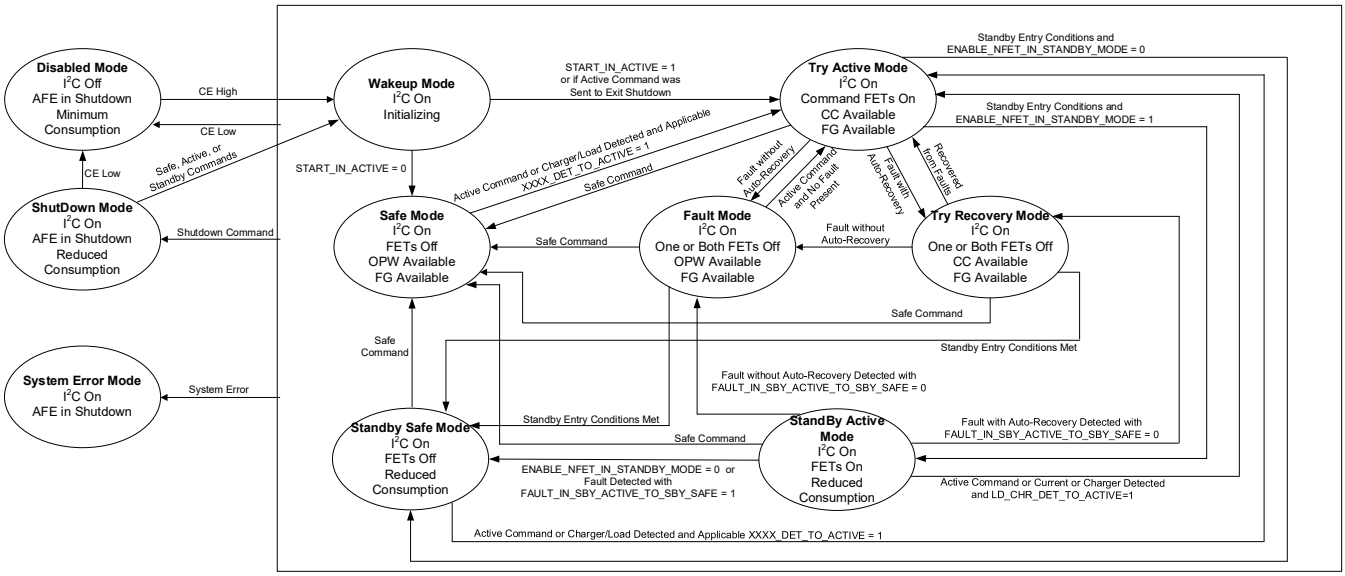

Figure 5: MPF42781 State Diagram

The following sections explain the operating modes. See the device's datasheet for more details regarding the related registers.

Wake Up

The wake-up state is a temporary state before device initialization concludes. While in this state, the protection MOSFETs stay open. The START_IN_ACTIVE register defines the target state when initialization is complete. If START_IN_ACTIVE = 0 at the end of initialization, the operating mode is safe mode; if START_IN_ACTIVE = 1 at the end of initialization, the CFG attempts to enter try active mode.

The related registers are BMS_STATUS and START_IN_ACTIVE.

Safe

In safe mode, the CFG commands the battery monitor to keep the protection MOSFETs open, which keeps the battery in a safe state. The SAFE_MODE_CMD register can be sent to the CFG to enter safe mode. If the previous state was either the try recovery state or fault state, the CFG attempts to clear the faults while transitioning to safe mode. In safe mode, short-circuit (SC) and over-current (OC) monitoring can be enabled via ENABLE_SCOC _SAFE register. In this mode, the device can read the SAFE bit to check whether the MP279x if is in safe state.

The related registers are BMS_STATUS, SAFE_MODE_CMD, SAFE, SAFE_FLAG, and ENABLE_SCOC_SAFE.

Try Active

In try active mode, the CFG attempts to close the AFE's protection MOSFETs when there is no fault. The ACTIVE_MODE_CMD command can be sent to the CFG to attempt reaching try active mode, even if the previous state was the fault state; the CFG attempts to clear the faults to enter try active mode. In this mode, the device can read the ACTIVE bit to confirm whether the AFE is in the active/normal B state.

The related registers are BMS_STATUS, ACTIVE_MODE_CMD, ACTIVE, and ACTIVE_FLAG.

Try Recovery

Try recovery mode occurs when the CFG or AFE has noticed one or multiple fault conditions, but all the present fault conditions have auto-recovery enabled. The applicable protection MOSFET(s) open until the faults can be cleared, and then the IC attempts to reach try active mode.

The related registers are BMS_STATUS, BMS_ALARMS, BMS_RT_ALARMS, FAULTS_EN, FAULTS_REC_EN, FAULTS_1FET_EN (for the MPF42780), and FAULTS_2FET_EN (for the MPF42781).

Fault

Fault mode occurs if the CFG or AFE have detected one or multiple fault conditions that do not have auto-recovery enabled. The applicable protection MOSFET(s) are open. The faults must be manually cleared to exit this mode. In this mode, the device can read the FAULT bit to confirm whether the AFE is in a fault/normal C state.

The related registers are BMS_STATUS, FAULT, BMS_ALARMS, BMS_RT_ALARMS, FAULTS_EN, FAULTS_REC_EN, FAULTS_1FET_EN (for the MPF42780), and FAULTS_2FET_EN (for the MPF42781).

Standby Safe

Standby safe mode is similar to safe mode, but it has reduced power consumption. The fuel gauge algorithm is disabled, the high-resolution (HR) scans from the AFE are requested only every 3s, and the CFG reverts to the sleep power-saving state when there is no action to perform. The CFG can enter standby safe mode either via the STANDBY_MODE_CMD command or automatically, if the automatic standby safe mode entry conditions are met.

The related registers are BMS_STATUS, STANDBY_MODE_CMD, ENABLE_AUTOSBY, TIME_TO_STANDBY, STANDBY_CURRENT_THRESHOLD, AUTOSBY_FROM_ACTIVE, ENABLE_PFET_IN_STANDBY_MODE (for the MPF42780), ENABLE_NFET_IN_STANDBY_MODE (for the MPF42781), FAULT_IN_SBY_ACTIVE_TO_SBY_SAFE, AUTOSBY_FROM_FAULT, and AUTOSBY_FROM_RECOVERY.

Standby Active

For the MPF42780, standby active mode differs from standby safe mode, as the AFE protection MOSFETs are open but the standby discharge (SBYDSG) MOSFET is closed. For the MPF42781, this mode differs from the standby safe mode because it keeps the protection MOSFETs closed. If the AFE measures the current outside of the standby range, or if a charger is detected for the MPF42780, the device automatically exits this mode and returns to the try active mode.

Entering this mode can only be done from try active mode if the ENABLE_NFET_IN_STANDBY_MODE bit = 1 and either the STANDBY_MODE_CMD command is received, or if the automatic standby entry conditions are met.

The related registers are BMS_STATUS, STANDBY_MODE_CMD, ENABLE_AUTOSBY, TIME_TO_STANDBY, STANDBY_CURRENT_THRESHOLD, AUTOSBY_FROM_ACTIVE, ENABLE_PFET_IN_STANDBY_MODE (for the MPF42780), ENABLE_NFET_IN_STANDBY_MODE (for the MPF42781), and FAULT_IN_SBY_ACTIVE_TO_SBY_SAFE.

Shutdown

In shutdown mode, the AFE is commanded into a shutdown state and the CFG reverts to the sleep state. In this mode, the protection MOSFETs are kept open, the FG is disabled, and the battery parameters are not monitored. The CFG target I^2C remains available while in shutdown mode, and can be used to transition to a different mode.

The related registers are BMS_STATUS, SHUTDOWN_MODE_CMD, and SHUTDOWN_FLAG.

Disabled

Similar to shutdown mode, in disabled mode, the AFE is commanded into shutdown, the protection MOSFETs are kept open, the fuel gauge is disabled, and there is no monitoring of the battery parameters. However, the target I^2C is not available in disabled mode, and the CFG is disabled. Disabled mode is achieved by setting the CE pin low. There is no related I^2C register.

System Error

System error mode is only achieved when there is a loss of communication with the AFE or some other unexpected system behavior. The only way to exit this mode is by a reset, either via toggling the nRST pin, cycling the power supply, or the IC_RST_CMD command; in addition, in system error mode, SAFE_MODE_CMD register and ACTIVE_MODE_CMD register also generate a reset.

The related registers are BMS_STATUS, SYSTEM_ERROR_RT, SYSTEM_ERROR, IC_RST_CMD, SAFE_MODE_CMD, and ACTIVE_MODE_CMD.

Controller Fuel Gauge (CFG) Power States

During operation, MPS's CFGs switch between different power states depending on what the device must do (see Figure 6). Whenever possible, the CFG reverts to the power state with lowest current consumption to minimize current drawn from the battery.

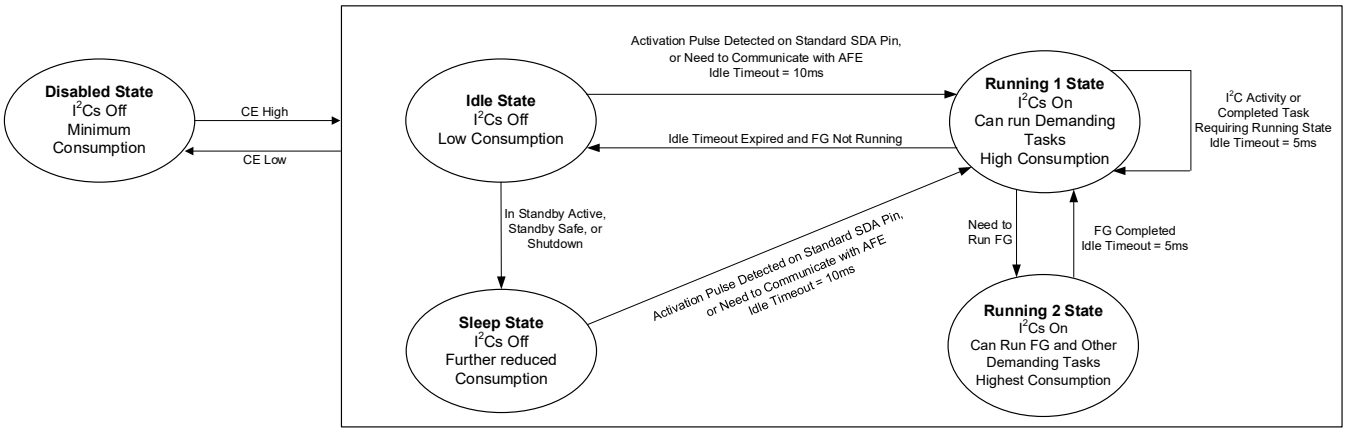

Figure 6: MPF42780/81 Power States Diagram

The power states are described below.

Running States

When the CFG must perform a demanding task, it transitions to one of the running states to successfully complete these tasks.

The running 2 state achieves the maximum internal computing speed and is used during initialization, storing fuel gauge configurations to the non-volatile memory (NVM), and while running the fuel gauge algorithm.

The running 1 state achieves enough processing capability to successfully communicate through the I²C ports. This state is used when the running 2 state is not needed but communication is required with either the AFE or through the target I²C.

Idle State

To reduce current consumption when no demanding task is required, the MPF42781 reverts to an idle state with much lower current consumption compared to the running states.

Sleep State

When in shutdown or in any of the standby modes, the sleep state takes priority over the idle state for further reduced current consumption.

Disabled State

When CE is low, the CFG is in the disabled state to achieve the lowest possible current consumption.

COMMUNICATION

Target Address Setting

The target address is 0x08 by default. Writing to this register takes immediate effect, so the user must adapt the target address in the following I²C transaction. The I2C_ADDRESS register contains the device's 7-bit I²C target address; during the address frame, it is left-shifted by 1 bit to accommodate the read/write (R/W) bit. With the default 0x08 value, the address frame should be 0x10 (write) or 0x11 (read).

The related register is I2C_ADDRESS.

Entering Active/Running Power State

To force a transition from a standby/idle/sleep state to an active/running state, a low pulse of at least 1ms is required on the SDA or ACTx lines. Therefore, transitioning to the active/running state requires a start condition followed by a stop condition 1ms later. Note that it takes about 6ms after the start of the low pulse for the fuel gauge to fully reach the active/running state and for the I²C to be ready.

After the CFG reaches the running state, there is a 10ms window to start communication. For the SFGs entering the active state, this time window is 20ms. Otherwise, the fuel gauge will timeout, and the device transitions back to the standby/idle/sleep state. After the operations are completed (e.g. I²C communication with the user or AFE, or fuel gauge iteration) there is a 5ms timeout before the IC goes to a standby/idle/sleep state.

Clock Stretching

A target I²C cannot receive or transmit a complete byte of data while performing other tasks, but it can hold the SCL line low to force the initiator I²C into a wait state (clock stretching) after the 9th pulse. Then, when the target I²C is ready, data transfer continues, and the clock line (SCL) is released (see Figure 7).

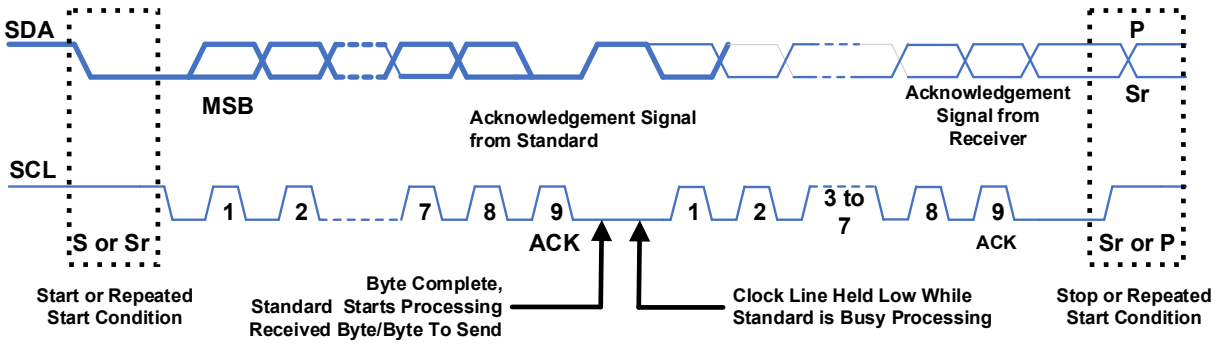

Figure 7: Clock Stretching in the I²C

Protocol Layer

The fuel gauge protocol uses 2 bytes for the register and command addresses. A length field is also provided to declare the number of data bytes in each read or write transaction. The maximum allowed transaction length is 82 bytes of data (not including the register address, length byte, or cyclic redundancy check (CRC) bytes) unless otherwise specified in the device's datasheet.

Cyclic Redundancy Check (CRC)

A CRC ensures the transaction's integrity. When enabled, the last 4 bytes of the transaction correspond to the CRC. The CRC spans the register address, length, and data payload. It is generated in blocks of 4 bytes. If the number of bytes is not a multiple of 4, the block is padded with 0x00.

The I2C_CRC_EN bit determines whether CRC is required in the I²C transactions. If the CRC is disabled, a write transaction does not need to include the CRC bytes to be accepted; it will be accepted regardless of the presence and/or validity of the CRC bytes. During a read, the FG does not include the CRC, so the not acknowledge (NACK) signal and stop command from the initiator I²C come after the last data byte.

The CRC algorithm used is the CRC-32/MPEG-2 with CRC-32 polynomial ($1 + x^1 + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32}$), initial value of 0xFFFFFFFF, no bit reversal and no final XOR.

The algorithm to generate the CRC based on C language is shown below.

```
unsigned long crc32 (unsigned short Reg_Address, unsigned char length, unsigned char *data){
  short i;
  unsigned long crc = 0xffffffff;
  unsigned char dataTemp[4];
  for (i=-1; i<len; i++){
    if(i==-1) {
      dataTemp[0]=len;
      dataTemp[1]=Reg_Address&0x00FF;
      dataTemp[2]=(Reg_Address&0xFF00)>>8;
      dataTemp[3]=0;
    }
    else dataTemp[i%4]=data[i];
    if((i%4)==3 || i == len-1 || i == -1) {
      for (char j=0; j< 4; j++) {
      crc ^= dataTemp[3-j] << 24;
        for (char k = 0; k < 8; ++k) {
          if ((crc & 0x80000000) != 0)
            crc = (crc << 1) ^ 0x04C11DB7;
          else
            crc <<= 1;
        }
      }
      dataTemp[0]=0;
      dataTemp[1]=0;
      dataTemp[2]=0;
      dataTemp[3]=0;
    }
  }
  return crc;
}
```

The related register is I2C_CRC_EN.

I²C Write Transaction

An I²C write transaction with an MPS fuel gauge includes the target I²C address byte with the R/W bit set to 0 (write), 2 bytes containing the 16-bit register address, a byte indicating the length in bytes of data to be written, the data bytes themselves and, if enabled, 4 CRC bytes (see Figure 8).

The process to write 0x09C4 (2 bytes: 0xC4, 0x09) to register 0x10C3 is described below.

1. The initiator I²C pulls SDA low for 1ms to 6ms.
2. For the start of the low pulse, wait for >5ms and <13ms (for the CFG) or <23ms (for the SFG).
3. The initiator I²C sends a start command. The target I²C address byte + write (0x10 if I2C_ADDRESS = 0x08) and check whether the fuel gauge acknowledges the command (FG ACK) (this means the fuel gauge is in active/running mode).
4. Write to the register address byte (0x10C3), write 0xC3 and check the FG ACK, then write 0x10 and check FG ACK.

   For this step and all steps below, ensure that new bytes are sent sooner than 5ms from the previous byte.
5. Write the length of the data (0x02) and check FG ACK.
6. Write Data 0 -byte/bits[7:0] (0xC4) and check FG ACK.
7. Write Data 1 -byte/bits[15:8] (0x09) and check FG ACK.
8. If CRC is disabled, proceed to step 13. If CRC is enabled, the correct 4 CRC bytes must be sent for the write to be accepted. This sequence results in CRC = 0x5385D9AD.
9. Write CRC Byte 0, bits[7:0] (0xAD) and check FG ACK.
10. Write CRC Byte 1, bits[15:8] (0xD9) and check FG ACK.
11. Write CRC Byte 2, bits[23:16] (0x85) and check FG ACK.
12. Write CRC Byte 3, bits[31:24] (0x53) and check FG ACK.
13. The initiator I²C can now send a stop command to end the transmission or repeated start command to start a new transmission.

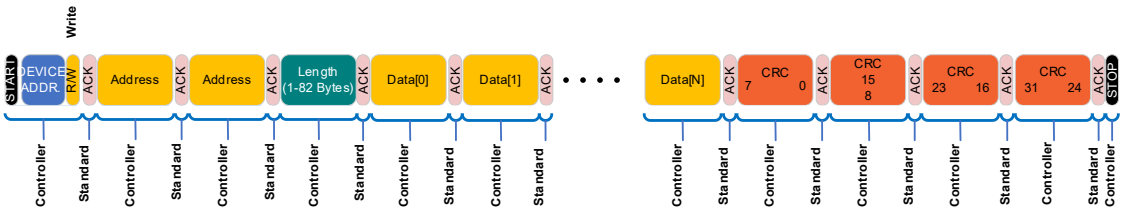

Figure 8: I²C Write Transaction with CRC

I²C Read Transaction

An I²C read transaction with an MPS fuel gauge includes the target I²C address byte with the R/W bit set to 0 (write), 2 bytes containing the 16-bit register address, a byte indicating the length in bytes of data to be read, then a stop + start or repeated start, a second target I²C Address Byte (but with the R/W bit set to 1 (read) to start reading), the data bytes being sent from the fuel gauge and, if enabled, 4 CRC bytes (see Figure 9 on page 10).

The process to read 4 bytes from register 0x12C0 is described below.

1. The initiator I²C pulls SDA low for 1ms to 6ms.
2. For the start of the low pulse, wait for >5ms and <13ms (for the CFG) or <23ms (for the SFG).

3. The initiator I²C sends a start command: target I²C address byte + write (0x10 if I2C_ADDRESS = 0x08) and check FG ACK (this means the fuel gauge is in active/running mode).
4. Write register address byte (0x12C0), write 0xC0, and check FG ACK, and then write 0x12 and check FG ACK.

   For this step and all steps below, ensure that new bytes are sent sooner than 5ms from the previous byte.
5. Write the length of the data (0x04) and check FG ACK.
6. Send either a stop command followed by a start command or a repeated start command. Follow it with the target I²C address byte + read (0x11 if I2C_ADDRESS = 0x08) and check FG ACK.
7. Read Byte 0 and return ACK.
8. Read Byte 1 and return ACK.
9. Read Byte 2 and return ACK.
10. Read Byte 3.
11. If CRC is disabled, proceed to step 16. If CRC is enabled, continue receiving the 4 CRC bytes.
12. Read CRC Byte 0 and return ACK.
13. Read CRC Byte 1 and return ACK.
14. Read CRC Byte 2 and return ACK.
15. Read CRC Byte 3.
16. The initiator I²C returns a NACK and can now send a stop command to end the transmission, or a repeated start command to start a new transmission.

In this example transaction, if the data bytes were {0xF4, 0x01, 0x64, 0x00}, the CRC would be 0x469B7F60. This means that the CRC bytes would have to be {0x60, 0x7F, 0x9B, 0x46} to ensure integrity on the transmission.

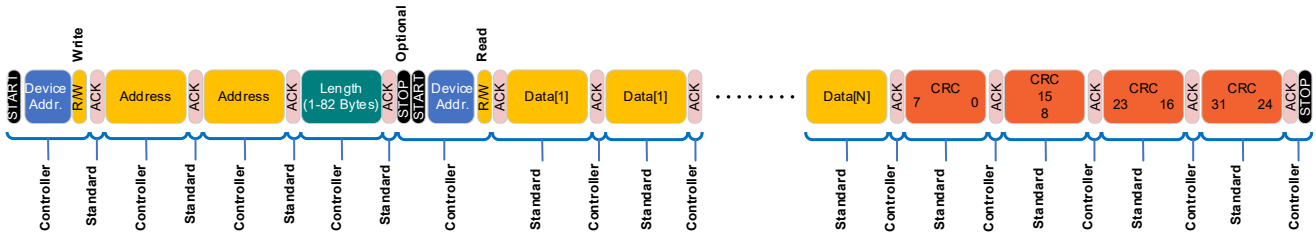

Figure 9: I²C Read Transaction with CRC

Fuel Gauge Algorithm (Both SFG/CFG)

MPS's fuel gauges use a mix-model state-of-charge (SOC) algorithm to calculate the cell/pack SOC. Compared to the common open-circuit voltage (OCV) method or the Coulomb counter method, the MPS fuel gauge can provide accurate SOC results under transient circumstances and does not need full charge and discharge cycles to update the fully charged capacity. This fuel gauge model reduces Coulomb counting accumulation errors. For some specific applications that do not have a chance to reach full charge or full discharge, MPS's fuel gauges achieve excellent SOC results without calibration.

This section of the application note focuses on the registers related to the FG algorithm and how to use them. Readers will understand the definition of these registers and how these registers work if they change the default value.

Some registers show under expert mode in the GUI. These sections can be related to the CFG or SFG.

A fuel gauge iteration in which the SOC is calculated follows the procedure below:

1. Obtain the readings. The readings below should be obtained from the AFE.
    a. Cell and pack voltage. If the pack voltage cannot be provided, disable the pack voltage reading (VRDG_PACK_EN bit).
    b. Cell and pack current. If the cell current cannot be provided, disable cell current reading (IRDG_CELLS_EN bit).
    c. Temperature points.
    d. Coulomb counting. If Coulomb counting cannot be provided, disable Coulomb counting (CCRDG_EN bit).

For an SFG, the user must send this measurement data to the fuel gauge on each fuel gauge iteration to ensure correct operation. For a CFG, the CFG triggers a high-resolution scan every 500ms to obtain all measurements needed for fuel gauging and complete battery monitoring.

1. Update the learning parameters if the conditions are satisfied.
2. Update the SOC state.
3. Update the start-of-health (SOH) state and SOH learning.
4. Update the state-of-power.
5. Check for possible warnings.
6. Store the results.
7. End the iteration.
    a. At this point, the fuel gauge is ready to send the fuel gauge output.
    b. If enabled, the fuel gauge pulls the IRQ pin low.

Cell Model Selection

The fuel gauge algorithm must use a cell mathematical model that can be uploaded directly from the MPS graphical user interface (GUI). It can be selected directly from the cell database, or it can be loaded through a file provided by MPS. Customers can read back the cell model information to ensure that the correct cell model has been loaded.

Main Settings

The execution period (EXE_TIME command) defines the rate of the fuel gauge algorithm execution. The minimum execution time is 500ms, whereby the fuel gauge algorithm runs every 500ms. The rate of obtention of measurements sent to the FG must strictly follow the execution time.

For SFG operation, the user should send the execution command (EXE_CMD command) at the frequency configured in the execution period (EXE_TIME). For CFG operation, the CFG can automatically trigger iterations if the fuel gauge is enabled (FG_EN command).

The FG estimates the time-to-empty/full with each algorithm iteration when the system is charging or discharging. When not charging or discharging, the FG runs the empty/full estimations less often to minimize energy consumption based on the rest execution period multiplier (WEXE_TIME_REST register); empty/full estimations are executed one time every period of (EXE_TIME x WEXE_TIME_REST).

For example, if EXE_TIME = 4s and WEXE_TIME_REST = 4, then the FG runs full/empty estimations once every 16s.

The actual algorithm execution time depends on the specification calculations required and the fuel gauge's IC. The MPF42791, which is typically configured for 10s, only requires about 25ms to estimate the SOC only, while it requires about 400ms when making empty/full estimations. When operating at a 500ms execution period, this leaves about 100ms to reload the AFE data for the next FG iteration.

The charge/discharge current thresholds (CHG_ITH and DSG_ITH register) are the thresholds at which the battery is considered to be charging/discharging. Below these thresholds, the FG considers the battery to be at rest.

The maximum SOC slope (SOC_DMAX register) defines the maximum SOC change for each second of charge/discharge.

For example, if there is 1C discharge, the SOC slope is about 100% / 3600s = 0.028%/s. This setting should be set a bit higher than 0.028%/s for correct operation in applications up to 1C. This setting must be greater when higher C-rates can be applied to ensure that the SOC curve reported by the fuel gauge can catch the real charge/discharge curve.

The maximum resting SOC slope (SOC_DMAX_REST register) defines the maximum SOC change for each second while in rest at the current level. If users do not want SOC changes under relaxation; this value should be 0.

Battery Pack Configuration

The number of series cells (NCELLS_SER register) sets the number of series cells for the whole pack and affects the battery pack voltage.

The number of parallel cells (NCELLS_PAR register) sets the number of parallel cells for the whole pack and affects the battery pack capacity.

The number of temperature sensors (NTSS register) sets the total number of NTCs used within the battery pack. When matching to an MPS AFE, this number should match the setting of cell monitor NTCs (NUM_OF_CELL_NTCS register).

Temperature Configuration

The cells temperature selection (TSS_CELLX register) defines the assignment between each cell temperature NTC and the cell(s) it is monitoring.

The maximum charge temperature (TCHG_MAX register) sets the maximum temperature during charge according to the custom application. This is used for the predicted charge over-temperature warnings (WARN_OTCHG_CC register and WARN_OTCHG_END register).

The maximum discharge temperature (TDIS_MAX register) sets the maximum temperature during discharge according to the customer application. This is used for the predicted discharge over-temperature warnings (WARN_OTDIS register).

The pack heat transfer coefficient (HCONV_PACK register) sets the thermal transfer ability for the user's battery pack. A larger coefficient means the system has better thermal dissipation. This affects the temperature model and SOC estimation under high temperatures.

For example, if the battery pack has a heatsink or cooling system, there is less temperature rise under the same charge/discharge conditions.

Added Resistance

Due to wire or bus-bar connection reasons, there is some resistance between two connected cells. Similarly, there is also some resistance in the battery management system (BMS) board. This resistance causes errors in the cell/pack voltage monitoring, which affects SOC accuracy. An MPS fuel gauge provides registers for resistance values to compensate for these voltage drops and improve SOC accuracy. Figure 10 on page 13 shows a diagram for these resistances.

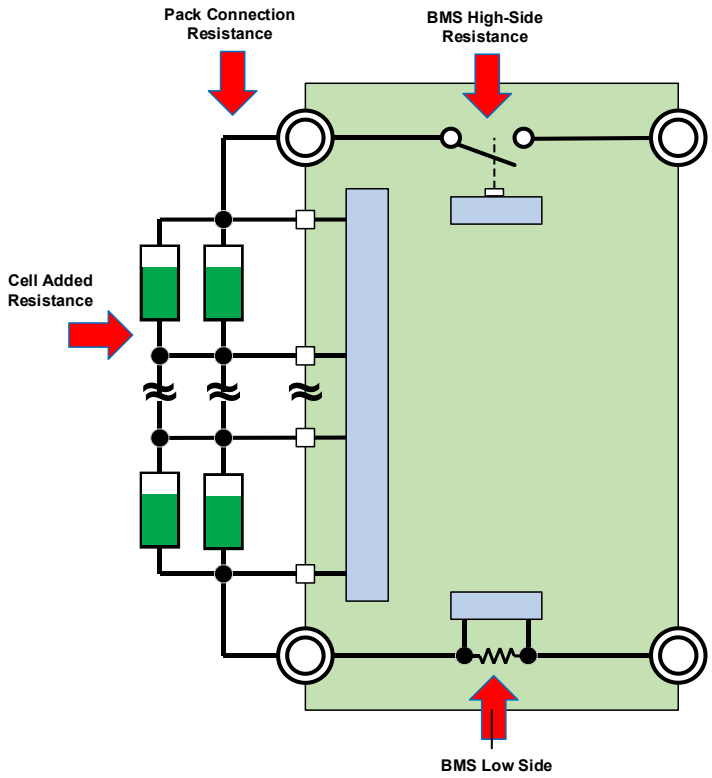

Figure 10: Resistances in the Battery

The pack connection resistance (RCXN_PACK register) is the resistance from the battery outputs to the BMS board. It comprises the resistance of both positive and negative wires/connections.

The BMS high-side resistance (RBMS_HSIDE register) is the total resistance at the BMS high side due to PCB traces and protection FETs.

The BMS low-side resistance (RBMS_LSIDE register) is the total resistance at the BMS low side due to shunt resistance and PCB tracing.

The cell added resistance (RCXN_CELLX register) is the resistance due to the cell-to-cell connection or any bus-bar between cells.

The fuel gauge's GUI provides two methods to include cell-added resistance. It can be introduced as a single value for all cells in the battery packs, or as individual values for each cell in the battery pack. If the user cannot measure these values, the default configuration can be used.

Battery Voltage Limits

These settings define the maximum and minimum voltage limitations of the cell/pack based on the custom application. These values are used during empty/full estimation, power estimation, and SOC estimation.

The maximum cell voltage (VCELL_MAX register) is the maximum cell voltage used in SOC and full estimations.

The minimum cell voltage (VCELL_MIN register) is the minimum cell voltage used in SOC and empty estimations.

The maximum pack voltage (VPACK_MAX register) is the maximum pack voltage used in SOC and full estimations.

The minimum pack voltage (VPACK_MIN register) is the minimum pack voltage used in SOC and empty estimations.

Expert Mode

The maximum cell voltage margin (VCELL_MAX_MRGN register) is the margin used to cover the deviation during maximum cell voltage sensing.

The minimum cell voltage margin (VCELL_MIN_MRGN register) is the margin used to cover the deviation during minimum cell voltage sensing.

The maximum pack voltage margin (VPACK_MAX_MRGN register) is the margin used to cover the deviation during maximum pack voltage sensing.

The minimum pack voltage margin (VPACK_MIN_MRGN register) is the margin used to cover the deviation during minimum pack voltage sensing.

The maximum/minimum cell and pack limits only define the SOC points at 100% and 0%. The fuel gauge regards (Max Voltage - Margin) as a possible full point and (Min Voltage + Margin) as the empty point. These values are not used for any protection activities and can be different from the over-voltage (OV)/under-voltage (UV) thresholds set within the AFE.

Battery Current Limits

These settings define the maximum and minimum current limits during charge and discharge, based on the custom application. The margins described are applied to cover the deviation caused by current sensing.

The maximum charge current (ICHG_MAX register) is the maximum charge current that the pack can accept.

The maximum discharge current (IDIS_MAX register) is the maximum discharge current that the pack can accept.

Expert Mode

The maximum discharge current margin (IDIS_MAX_MRGN register) is the margin used to cover the current sensing error.

The maximum charge current margin (ICHG_MAX_MRGN register) is the margin used to cover the current sensing error.

Load Setting

These registers characterize the load to improve discharge behavior predictions. These parameters are related to the custom application and are included in the empty prediction to establish the 0% pack SOC point and for time-to-empty estimation.

The nominal discharge average current (IDIS_AVG_SET register) defines the average current during discharge. This register self-updates if the learning option is enabled.

The nominal discharge termination current (IDIS_END_SET register) defines the current at which the applications will stop the discharge. This register self-updates if the learning option is enabled.

Charger Setting

These registers characterize the charger to improve charge behavior prediction. These parameters are related to the custom application and helps predict when the battery reaches the 100% SOC. Therefore these parameters are crucial for the time-to-full estimation.

The nominal charge constant current (ICHG_CC_SET register) defines the constant charge current during charging. This register will self-update if the learning option is enabled.

The nominal charge termination current (ICHG_END_SET register) defines the current at the end of charging, the cut-off current of the constant voltage phase. This register will self-update if the learning option is enabled.

The nominal charge constant voltage (VCHG_CV_SET register) defines the voltage when charging stops at the end of constant voltage (CV) phase. This register will self-update if the learning option is enabled.

Expert Mode

The charge termination current margin (ICHG_END_MRGN register) defines the margin of tolerance for current-sensing, which is used for full prediction.

To ensure accurate full/empty predictions, it is recommended that users use the most accurate charge constant voltage (CV) and termination current. This means that [CV Voltage < Max Pack Voltage < (Max Cell Voltage x Cell Number)]. For the full discharge value, it is recommended to set [(Min Cell Voltage x Cell Number) < Min Pack Voltage] to improve the limiting factor report.

Empty Setting/Full Setting

In real applications, the load and charger current may not be constant. Under these conditions, the reported remaining time-to-full/empty will always change. To reduce the impact of the transient current on the full/empty estimations, the fuel gauge includes a low-pass filter.

The remaining time-to-empty filter time constant (EMTY_RTIME_FILTER register) is used for the current measurement's low-pass filter. A higher value means more spikes will be filtered during the remaining time-to-empty prediction.

The remaining time-to-full filter time constant (FULL_RTIME_FILTER register) is used for the current measurement's low-pass filter. A higher value means more spikes will be filtered during the remaining time-to-full prediction.

The predicted temperature to empty filter time constant (EMTY_DTEMP_FILTER register) is used for the low-pass filter on the temperature. A higher value means a sudden temperature change will have less effect on thermal modeling processes and the remaining time-to-empty prediction.

Predicted temperature to full filter time constant (FULL_DTEMP_FILTER register) is used for the low-pass filter on the temperature. The higher value means a sudden temperature change will have less effect on thermal modeling process and the remaining time-to-full prediction.

Learnings

When learnings are enabled, the fuel gauge can self-update certain key parameters. The process of updating these parameters depends on specific rules for each learning section.

State-Of-Health (SOH)

The update conditions are listed below:

- Status == Discharge.
- Min Temp (SOH_LRN_TMIN) ≤ Temperature ≤ Max Temp (SOH_LRN_TMAX).
- Min Current (SOH_LRN_IMIN) ≤ Current ≤ Max Current (SOH_LRN_IMAX).
- Min Range (SOH_LRN_MIN) ≤ New SOH ≤ Max Range (SOH_LRN_MAX).
- Max increase (SOH_LRN_DMAX) defines the maximum SOH increase value during each update while min increase (SOH_LRN_DMIN) defines the minimum SOH decrease value during each update.
- Gain (SOH_LRN_K) defines the gain factor on how much capacity change can be accounted for during the SOH calculation. A smaller value means a smoother SOH change.

Heat Transfer Coefficient

The update conditions are listed below:

- Status different from REST.

- Average Temp - (Ambient Temp + Temp Difference) ≥ Temperature Derivative Threshold (HCONV_LRN_DTEMPTH). A larger value means a smoother change for the heat transfer coefficient.
- Time Differential Value ≥ Time Derivative Threshold (HCONV_LRN_DTIMETH). A larger value means a smoother change for heat transfer coefficient.
- Range (HCONV_LRN_RNG) defines the maximum change range for the heat transfer coefficient.
- Gain (HCONV_LRN_K) defines the gain factor for how much thermal change can be accounted for in the heat transfer coefficient calculation. A smaller value means a smoother coefficient change.

Charge Constant Current (CC)

The update conditions are listed below:

- Pack SOC ≤ SOC Threshold (ICHG_CC_LRN_SOCTH).
- Averaged Derivative Current ≤ Current Derivative Threshold (ICHG_CC_LRN_DITH).
- Learning will start only if pack SOC is less than or equal to ICHG_CC_LRN_SOCTHC and the current change is slower than ICHG_CC_LRN_DITH for more than one sample counter threshold (ICHG_CC_LRN_CNTRTH) time during a fuel gauge iteration.
- The updated charge constant current (ICHG_CC) value is between the nominal charge constant current (ICHG_CC_SET) minus the charge current learning range (ICHG_CC_LRN_RNG) and the nominal charge constant current (ICHG_CC_SET) plus the charge current learning range (ICHG_CC_LRN_RNG).

Charge Termination Current

The update conditions are listed below:

- Pack SOC ≥ SOC Threshold (ICHG_END_LRN_SOCTH).
- Current > Current Threshold (ICHG_END_LRN_ITH).
- Averaged Voltage Derivative ≤ Voltage Derivative Threshold (ICHG_END_LRN_DVTH).
- The updated charge termination current (ICHG_END) value is between the nominal charge termination current (ICHG_END_SET) minus the charge termination current learning range (ICHG_END_LRN_RNG) and the nominal charge termination current (ICHG_END_SET) plus the charge termination current learning range (ICHG_END_LRN_RNG).

Charge Constant Voltage (CV)

The update conditions are listed below:

- Pack SOC ≥ SOC Threshold (VCHG_CV_LRN_SOCTH).
- Current > Current Threshold (VCHG_CV_LRN_ITH).
- Averaged Voltage Derivative ≤ Voltage Derivative Threshold (VCHG_CV_LRN_DVTH).
- The updated charge constant voltage (VCHG_CV) value is between the nominal charger constant voltage (VCHG_CV_SET) minus the charge constant voltage learning range (VCHG_CV_LRN_RNG) and the nominal charger constant voltage (VCHG_CV_SET) plus the charge constant voltage learning range (VCHG_CV_LRN_RNG).

Discharge Average Current

The update conditions are listed below:

- Discharge Current > Current Threshold (IDS_AVG_LRN_ITH).

- The averaged discharge current (IDIS_AVG) learning has a time constant filter (IDIS_AVG_LRN_FILTER). A larger value means the current spikes will be filtered more for the learning cycle.
- The updated discharge average current (IDIS_AVG) is between the nominal discharge average current (IDIS_AVG_SET) minus the average discharge current learning range (IDIS_AVG_LRN_RNG) and the nominal discharge average current (IDIS_AVG_SET) plus the average discharge current learning range (IDIS_AVG_LRN_RNG)

Discharge Termination Current

The update conditions are listed below:

- Pack SOC ≤ SOC Threshold (IDIS_END_LRN_SOCTH).
- Discharge Current > Current Threshold (IDIS_END_LRN_ITH).
- The discharge termination current (IDIS_END) learning has a time constant filter (IDIS_END_LRN_FILTER). A larger value means the current spike will be filtered more during the learning cycle.
- The peak current gain (IDIS_END_LRN_KPEAK) defines the impact of the current peaks to the average discharge termination current for the discharge termination current calculation. If some higher current pulses can happen, a higher value in this register can help to better predict empty point by predicting whether an empty condition may occurs during one of these pulses rather than when the current is around the average value.
- The updated discharge termination current (IDIS_END) is between the nominal discharge termination current setting (IDIS_END_SET) minus the discharge termination learning range (IDIS_END_LRN_RNG) and the nominal discharge termination current setting (IDIS_END_SET) plus the discharge termination current learning range (IDIS_END_LRN_RNG).

Without continuous learning, the fuel gauge uses the data from the previous cycle as the discharge termination current. The fuel gauge does not update this parameter until the cycle has finished. It is recommended to use the continuous learning.

Equivalent Series Resistance (ESR)

The update conditions are listed below:

- Current Variation Rate ≥ Current Threshold (RESR_LRN_DITH).
- Maximum Variation of Resistance under Standard Mode < Max Resistance Variation Standard (RESR_LRN_DRSTD).
- Maximum Variation of Resistance under Extended Mode < Max Resistance Variation Extended (RESR_LRN_DREXT).
- The fuel gauge uses the larger limit once the number of times the maximum variation of standard mode has reached number of hits to extended (RESR_LRN_NHIT2EXT).

Added Resistance

The update conditions are listed below:

- Current Variation Rate ≥ Current Derivate Threshold (RCXN_CELLS_LRN_DITH).
- Minimum Resistance (RCXN_CELLS_LRN_RMIN) ≤ Updated Cell Connection Resistance (RCXN_CELLS_LRN_RMAX).
- RCXN_CELLS_LRN_NAVG Defines the number of times when the FG observes change for cell connection resistance. The final result will be the average between all the times the learnings occurred (RCXN_CELLS_LRN_NAVG times).

When added resistance learning is enabled, ESR learning is automatically disabled until added resistance learnings have occurred for the amount of times set via RCXN_CELLS_LRN_NAVG; once the adding resistance learning is complete, it is automatically disabled. If ESR was initially enabled, then it is enabled again.

Expert Mode

Selecting the Measurement Data for the Fuel Gauge

MPS fuel gauges can be configured to operate without certain measurement data from the AFE. Users must enable which specific measurements will be provided to the fuel gauge for every iteration. If a reading is disabled, the value is internally processed by the fuel gauge and output in the applicable register inside the fuel gauge input registers.

Table 1 shows the variables.

Table 1: FG Readings

Variables	Relevant Registers	Remarks
Pack Voltage	VRDG_PACK_EN	The FG uses the AFE's reading if this function is enabled. Otherwise, it will sum all cell voltages in total and add the voltage drop caused by configured resistances.
Pack Current	IRDG_PACK_EN	If disabled, the pack current is processed from the cell's current.
Cells Current	IRDG_CELLS_EN	Synchronized current reading with cell voltages. If disabled, this is the same as the pack current .
Coulomb Count	CCRDG_EN	If disabled, Coulomb counting is processed from the pack current and execution period.
Ambient Temperature	TRDG_AMB_EN	If the AFE has a temperature reading dedicated to the ambient temperature, it can be used as an input for the FG to help with thermal prediction. Otherwise, disable this reading so that FG can estimate ambient temperature by itself.

Enabling the Learnings for the Fuel Gauge

Enabling a learning means that the fuel gauge will update the configured value if the learning conditions are met (see Table 2).

Table 2: FG Functions

Functions	Relevant Registers
State-of-health (SOH)	SOH_LRN_EN
Equivalent series resistance (ESR)	RESR_LRN_EN
Cell connection resistance	RCXN_CELLS_LRN_EN
Charge CC current	ICHG_CC_LRN_EN
Charge end current	ICHG_END_LRN_EN
Charge CV voltage	VCHG_CV_LRN_EN
Discharge average current	IDIS_AVG_LRN_EN
Discharge end current	IDIS_END_LRN_EN
Thermal modeling	THRM_MDL_EN
Heat transfer coefficient	HCONV_LRN_EN
Max charge power	PCHG_SHW_EN
Max discharge power	PDIS_SHW_EN
Continuous discharge end current	IDIS_END_LRN_CONT_EN
Continuous heat transfer coefficient	HCONV_LRN_CONT_EN

If continuous learning is enabled, the fuel gauge always updates the new learning value to the register until the learning stops. Otherwise, the fuel gauge only updates the value at the end of the learning cycle.

Expert Mode

FG Interrupts Selections

MPS's fuel gauges incorporate an interrupt (IRQ) to notify the system of several events. IRQ is set low if an iteration starts, and then set high if certain events occur (see Table 3).

Table 3: FG Interrupts

Interrupt Type	Relevant Registers	Remark
Iteration complete	IT_DONE_INTR_EN	Indicates whether the iteration is finished.
Status change	STATUS_INTR_EN	Indicates a pack state change (REST, DSG, CHG).
Empty limiting factor	EMTY_LIM_INTR_EN	Indicates a empty limiting factor change.
Full limiting factor	FULL_LIM_INTR_EN	Indicates a full limiting factor change.
Pack SOC result changed	SOC_PACK_RSLT_INTR_EN	Indicates a pack SOC change.
OT warning	OT_WARN_INTR_EN	Indicates if an over-temperature condition occurred.
Average discharge current learning	IDIS_AVG_INTR_EN	Indicates a average discharge current value or learning status change.
Discharge end current learning	IDIS_END_INTR_EN	Indicates a discharge termination current value or learning status change.
Charge CC current learning	ICHG_CC_INTR_EN	Indicates a constant charge current value or learning status change.
Charge end current learning	ICHG_END_INTR_EN	Indicates a charge termination current value or learning status change.
Charge CV voltage learning	VCHG_CV_INTR_EN	Indicates a charge constant voltage value or learning status change.
ESR learning	SORESR_CELLS_INTR_EN	Indicates an ESR value or learning status change.
Cell connection resistance	RCXN_CELLS_INTR_EN	Indicates a cell connection resistance or learning status change.
Heat transfer coefficient	HCONV_INTR_EN	Indicates a heat transfer coefficient or learning status change.
Cells SOH learning	SOH_CELLS_INTR_EN	Indicates a SOH learning coefficient or learning status change.
Discharge power limiting factor changed	PDIS_LIM_INTR_EN	Indicates a DSG power limiting factor change.
Charge power limiting factor changed	PCHG_LIM_INTR_EN	Indicates a CHG power limiting factor change.

Performance Evaluation

Below is an example to evaluate the pack SOC performance. This process be used during development by drawing the ideal pack SOC curve in a test that goes either from 0% to 100% or from 100% to 0%. By knowing where the curve should start and end, one can calculate the capacity added to or removed from the battery during this time to draw the ideal curve between the start and end points.

1. After the test, extract the log data from the fuel gauge. Read the CCRDG register. Add a column (accumulated capacity) that sums all the previous CCRDG values up to the current row to obtain the capacity added to or removed from battery up to that row.
2. The last value of the new row defines the total test capacity.
3. The discharge reference SOC (in %) can be calculated with Equation (1):

$$\text{Discharge Reference SOC} = 100 \times (1 - (\text{Accumulated Capacity}[X] / \text{Total Test Capacity})) \quad (1)$$

The charge reference SOC (in %) can be estimated with Equation (2):

$$\text{Charge Reference SOC} = 100 \times (\text{Accumulated Capacity[X]} / \text{Total Test Capacity}) \quad (2)$$

4. The SOC Error = (Pack SOC - Reference SOC).

Figure 11 shows a reference for SOC evaluation.

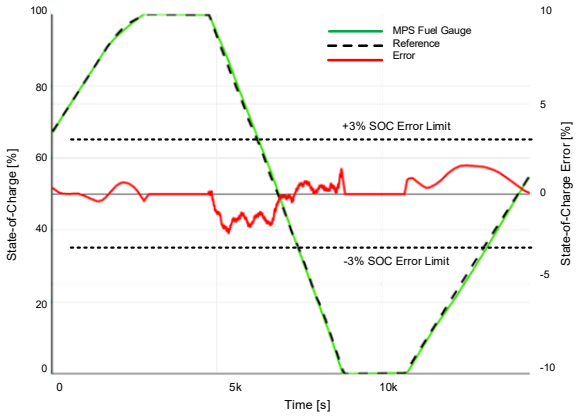

Figure 11: SOC Evaluation Reference

The fuel gauge not only supports a high-accuracy SOC calculation, but it also supports cell degradation predictions, run time estimations, available power estimations, and ESR updates. The fuel gauge can also be used to determine appropriate battery protection functions within the AFE, as well as to control balancing strategies. In summary, the MPS FG can provide an all-in-one solution and help users ease their development process and MCU resource. Contact an MPS FAE for additional information.

REVISION HISTORY

Revision #	Revision Date	Description	Pages Updated
1.0	5/19/2026	Initial Release	-