Coulomb Counting and State of Charge Estimation featuring the RAA489206/ISL94216A Battery Front End

This application note describes the implementation of coulomb counting featuring the RAA489206 Battery Front End and a Microcontroller Unit RA4W1, and its use to estimate the State of Charge of battery packs. It shows how to use the features of the RAA489206/ISL94216A to perform ampere hour counting and estimate the State of Charge using basic correction methods. Sample code and accuracy evaluation are provided.

Contents

1. Introduction

Estimating the SOC is an essential task of a Battery Management System (BMS). If the BMS can provide correct and accurate SOC estimation, the overall system can benefit in different aspects. Good SOC estimation extends the battery lifetime, maximizes the use of the pack capacity, minimizes its size, enhances the power-system reliability, and allows using the battery pack aggressively within design limits. This results in higher performance and reliability, optimized density, and lower cost. However, obtaining the SOC of a cell or battery pack is not trivial. There is presently no sensor or methodology to measure SOC directly, so it is necessary to infer or estimate the SOC based on cell terminal voltage, current and temperature measurements.

Coulomb counting is a common method to estimate the SOC in industrial applications. It offers low implementation complexity and acceptable accuracy by relying on the estimation of the initial capacity and the measurement of the current. If the capacity at a given moment is known, the remaining capacity can be estimated by adding the integrated charge current and subtracting the integrated discharging current. This method works well if charge and discharge coulombic efficiencies are 1 and the estimation of the initial capacity is accurate. These conditions are rarely met due to the complex interrelationships among factors such as usage profile, temperature, and irregular variations of cell capacities, resistances, and aging effects during the lifetime of the battery. In addition, measurement errors due to current sensors, noise, bias, leakage, and nonlinearity, accumulate and affect SOC estimation. Even if the Battery Font End (BFE) features dedicated and specialized coulomb counting hardware, it is necessary the estimation algorithm perform correction of its SOC estimations to ensure accuracy.

Dedicated hardware such as integrator ADCs are not the only way to perform coulomb counting. In industrial applications where the current variation rate is not so high, a coulomb counting ADC does not offer higher accuracy over techniques such as current sampling. This application note shows how to implement the same functionality using the measurement capabilities of the RAA489206 BFE and a Micro-Controller Unit (MCU). Moreover, it shows how to use it together with cells characterization and basic correction actions at fully charged and fully discharged points to achieve acceptable SOC estimation accuracy for industrial applications.

This application note applies to the RAA489206 and ISL94216A Renesas BFEs.

2. Definitions

A cell is fully charged when its terminal voltage during charge equals the manufacturer-specified voltage value VF and the charge current equals the manufacturer-specified End-of-Charge current I_{EOF}. The SOC of a fully charged cell is defined to be 100%. On the other hand, a cell is fully discharged when its Open-Circuit Voltage (OCV) equals the manufacturer-specified cut-off voltage ($V_{CUT\text{-}OFF}$). The SOC of a fully discharged cell is 0%.

SOC is a relative measurement of the energy available in the battery pack. It is defined as Equation 1.

(EQ. 1) $$SOC[\%] = \frac{Q_{releasable}}{Q_{nom}} \times 100$$

$Q_{releasable}$ is the charge that must be removed to bring the battery from its current state to fully discharged state. Q_{nom} is specified by the cell manufacturer and is the maximum capacity of a cell lot.

Depth of Discharge (DOD), by contrast, indicates the amount of charge that has been removed from the cell relative to the nominal capacity. It is defined as Equation 2 where $Q_{released}$ is the charge removed from the cell.

(EQ. 2) $$DOD[\%] = \frac{Q_{released}}{Q_{nom}} \times 100$$

The total capacity is the charge removed as the cell is brought from fully charged state to fully discharged state. It is expressed in terms of $Q_{releasable}$ and $Q_{released}$ as shown in Equation 3.

(EQ. 3) $$Q_{total} = Q_{releasable} + Q_{released}$$

Q_{total} is not a constant value. It fades because of side reactions and structural deterioration as the battery ages. Such decay of the total capacity results in the reduction of available charge, which should be considered to estimate the SOC of the cell. To take this into account, the State of Health (SOH) of the cell is defined as a figure of merit of the total capacity with respect to the nominal capacity. This is shown in Equation 4.

(EQ. 4) $$SOH[\%] = \frac{Q_{total}}{Q_{nom}} \times 100$$

Dividing Equation 3 by the nominal capacity Q_{nom}, the relation between SOH, SOC, and DOD is obtained using Equation 5.

(EQ. 5) $$SOH = SOC + DOD$$

3. Coulomb Counting Estimation

Coulomb counting keeps track of charge added to (Q_{in}) and removed from the battery (Q_{out}) integrating the battery current over time. The charge $Q_{in/out}$ transferred between times t_1 and t_2 ($t_1 < t_2$, in seconds) due the current $I_{meas}(t)$ (in milliamperes) is estimated using Equation 6.

(EQ. 6) $$Q_{in/out}[mAh] = \frac{1}{3600} \times \left(\int_{t_1}^{t_2} \eta \times I_{meas}(t)dt \right)$$

By convention, I_{meas} is positive for charging current to indicate that the charge Q_{in} is being added to the battery, whereas it is negative for discharging current to indicate that charge Q_{out} is being removed from the battery. η is the coulombic efficiency equals to η_c during charging, and to η_d during discharging.

Considering the sign of the current in Equation 6 and the definition of DOD in Equation 2, the change in the DOD during the interval $t_2 – t_1$ is shown in Equation 7.

(EQ. 7) $$\Delta DOD = DOD(t_2) - DOD(t_1) = \frac{-Q_{in/out}}{Q_{nom}}$$

Using Equation 7 to generalize, the accumulated DOD of a cell at time t (in seconds) is calculated using Equation 8.

(EQ. 8) $$DOD(t) = DOD(t - t_{meas}) + \Delta DOD$$

t_{meas} is the time that has elapsed since the last DOD estimation. Thus, using the DOD of a cell and considering its SOH, the SOC is estimated using Equation 9.

(EQ. 9) $$SOC(t) = SOH(t) - DOD(t)$$

This formula indicates that if the initial SOC = SOC_0 of a cell is known, its SOC can be estimated by using the following steps:

1. Calculate the initial DOD0 as $DOD_0 = SOH – SOC_0$
2. Calculate ΔDOD measuring and integrating the current through the cell over time.
3. Estimate the accumulated DOD as in Equation 8
4. Estimate the SOC subtracting DOD from the current SOH

Although this procedure may seem trivial, steps 1, 2, and 4 pose three non-trivial challenges that must be addressed to achieve accurate SOC estimation. First, determining accurately the initial SOC requires time-consuming cell characterization and data storage in the managing device (MCU), whose memory resources are limited. Second, the SOH varies along the battery life and is affected by many factors such as the usage profile, abuse events, number of charge/discharge cycles, so runtime SOH corrections are necessary. Finally, there is loss of accuracy in the current integration due to sampling and digitalization of the current measurement, and the approximation of the continuous integral as a summation of discrete areas, a method commonly implemented in discrete devices such MCUs to estimate integrals. These challenges are faced by all BMSs even if the integrated BFE features a dedicated coulomb counting ADC. The following sections describe how to address these challenges by implementing basic methods for SOC estimation and SOH correction and using the features of the RAA489206 to sample and integrate the battery pack current.

4. Initial SOC Estimation and SOH Correction

4.1 Initial SOC Estimation

The relation between the SOC and the Open-Circuit Voltage (OCV) allows estimating the SOC0 of a cell. Figure 1 shows the OCV versus SOC curve of a commercial Lithium-Ion cell at temperature room (25 °C). This curve can be characterized offline, and some points stored in the BMS memory as a lookup table. The BMS can then determine the SOC_0 measuring the cell voltage and getting the corresponding SOC from the lookup table.

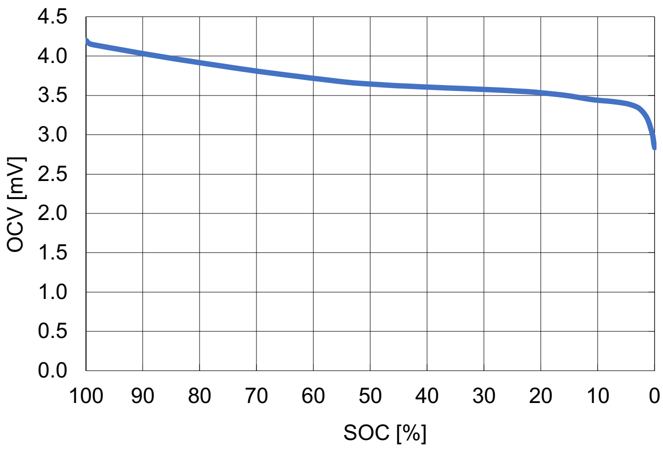


Figure 1. Relation of the SOC and Open-Circuit Voltage (OCV) of a Commercial Li-Ion Cell

Because the method relies on the measurement of the cell voltage (V_{CELL}) to determine SOC_0, the error the BFE specifies for cell voltage measurements impacts the accuracy of the estimation. The RAA489206 BFE specifies +10mV and -10mV as maximum and minimum V_{CELL} measurement errors, respectively. Assuming these two values as worst-case scenario and the OCV-SOC relation in Figure 1, Figure 2 illustrates the error (in percentage) induced by V_{CELL} measurements in the estimation of the SOC using the OCV method. The results show non uniform error along the SOC. The error for SOC ranges with low OCV/SOC slope such as between 22% and 50% exhibit higher SOC estimation error. This is because the low OCV/SOC slope implies high SOC/OCV slope, which results in high estimation deviation (between 2% and 3.5%) from the actual SOC when V_{CELL} measurements differ slightly from the actual cell voltage. Conversely, sections of OCV-SOC curve with high slope exhibit low SOC estimation error (lower than 2%). SOC regions where estimation accuracy is safety-critical, such as those close to fully charged (95%-100%) and discharged (0%-5%) states, show estimation errors below 1%. In industrial applications, such error values have low impact on SOC estimation and can be reduced to lower levels by averaging measurements. The RAA489206 features V_{CELL} measurement averaging, which reduces the maximum OCV-based SOC estimation error to 2% at most.

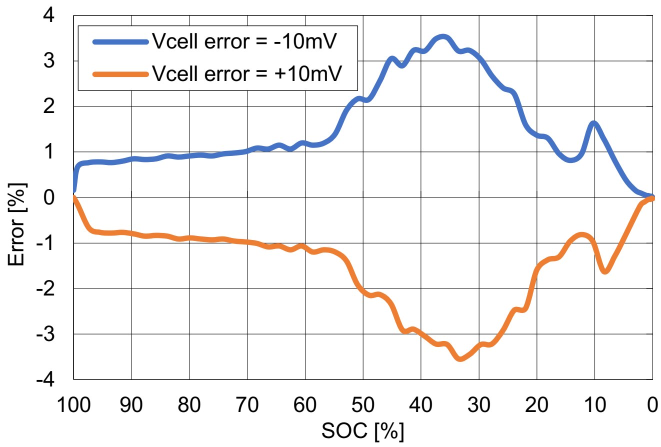


Figure 2. Figure 2. SOC Estimation Error due to V_{CELL} Measurement Error

Another factor that affects SOC estimation using cell OCV is the temperature. The OCV of a cell varies with temperature according to its Temperature Coefficient of Voltage (TCV). The TCV is expressed in µV/°C indicating how many microvolts the OCV changes per Celsius degree. It is characteristic of each cell and depends on cell structure, quality, and chemistry. For Li-Ion cells, large positive TCV values vary between 200 and 500µV/°C, whereas it is between -300 and -500µV/°C for negative TCVs. To counteract this effect, characterizations over the operational temperature range of the system might be necessary. If the cell manufacturer provides its TCV, the BMS system can correct the OCV using the temperature measured at the battery pack. However, if the system uses good quality Li-Ion cell with low TCV, the effects of the temperature for SOC_0 estimation are negligible and can be corrected at runtime. The tests performed in this application note are at room temperature (25°C), so neither temperature characterization nor TCV are considered.

The OCV method can be used to not only estimate initial SOC, but also correct the SOC at runtime when the system is in the off state. When the system is off and the cells are relaxed (dV/dt~10µV/s), the BMS can use the OCV/SOC to get and correct the current SOC estimation. The application of this method and the SOH correction described in the following section can obtain acceptable accuracy for most industrial applications. In addition, this is achieved without elaborate cell characterizations or sophisticated algorithms. If the application requires higher accuracy or off states are rare, the BMS must implement more complex methods such as on-state SOC correction or Internal Resistance (IR) estimation. These methods are out of the scope of this application note.

4.2 SOH Correction

The increase of the internal resistance of the cell during the lifetime of the battery causes the degradation of the SOH. Therefore, if SOC estimations do not re-evaluate the SOH at runtime, they may deviate significantly from the actual charge contained in the cell. A simple strategy is using fully charged and discharged states to correct the assumed SOH. If these states occur often during the system operation, the SOC accuracy can benefit from the corrections.

When the cell is fully discharged (SOC = 0), the SOH must equal the value of the accumulated DOD according to Equation 9. The SOH can be then re-evaluated as Equation 10.

(EQ. 10) $$SOH \leftarrow DOD$$

SOH is proportional to Q_{tot}, so if Q_{tot} is underestimated, DOD becomes negative at the next fully charged state. On the other hand, if Q_{max} is overestimated, DOD remains positive at the next fully charged state. [1] Therefore, the SOH of the battery can be re-evaluated at the fully charged state as Equation 11.

(EQ. 11) $$SOH \leftarrow SOC - DOD$$

Using Equation 3, the current Q_{max} can be set accordingly using the estimated SOC as in Equation 12.

(EQ. 12) $$Q_{max} \leftarrow SOH \times Q_{rated}$$

The frequency of the occurrence of fully charged and fully discharged states depends on the usage profile, so they may rarely occur. In this case, the BMS shall adopt other methodologies such the use of off states to estimate the total capacity of the cell and update its SOH. Some manufacturers may provide information about the life cycle of the cells and the degradation with respect to the number of discharge/charge cycles. If this information is available, the SOH degradation can be obtained by keeping track of the number of charge/discharge cycles. Other data that is sometimes detailed in cell specifications is the variation of the discharge capacity with respect to the temperature. This specification can also be considered to correct the SOH according to the battery pack temperature.

5. Coulomb Counting SOC Estimation featuring the RAA489206

This section describes the application implemented in the **r_soc.c** file of the sample code accompanying this application note. The application uses Coulomb Counting, SOH correction at fully charged and discharged states, and the SOC versus OCV data of Figure 1 to estimate the SOC of a battery pack. Figure 3 shows the target architecture consisting of a MCU as managing unit and the RAA489206 as BFE. The battery pack is built with four ICR18650-26J Li-Ion cells providing from 11V at fully discharged state, up to 16.8V at fully charged state.

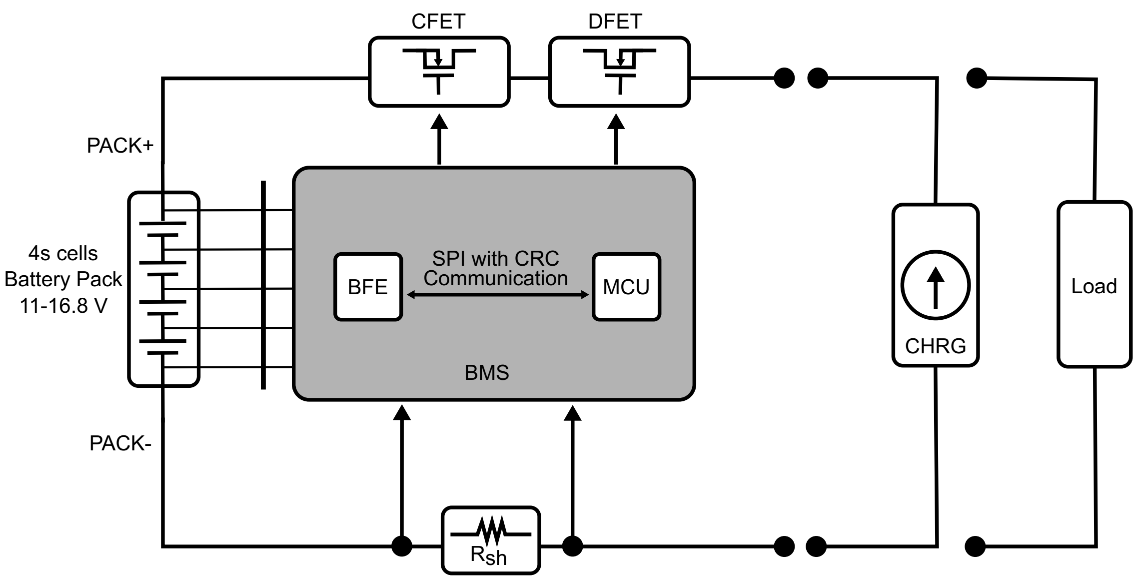


Figure 3. BMS Architecture

Table 1 lists the main cell specifications. The functions that implement the SOC estimation application are described algorithmically, so that the BMS designer can implement them regardless of the selected MCU. The

1. Ng, Kong Soon; Moo, Chin-Sien; Yi-Pin, Chen; Hsieh, Yao-Ching (2009). Enhanced Coulomb Counting Method for Estimating State-of-Charge and State-of-Health of Lithium-ion Batteries. *Journal of Applied Energy, 86* (9). 1506-1511

sample code is written for the RA4W1 MCU, but it can be ported to other MCUs of the RA family using the Renesas Flexible Software Package. See the application note *MCU Sample Code for Driving the RAA489206 16-Cell Battery Front End* for details on software and hardware requirements, code architecture and implementation of the sample code, and setup and use of the Command Line Interface (CLI) application.

Table 1. ICR18650-26J Main Specifications

Specification	Value
Typical Discharge Capacity	Min 2500mAh at1C and RT (25°) = 2600mAh
Charging Voltage	4.2V
Discharge Cut-Off Voltage ($V_{CUT-OFF}$)	2.75V
Nominal Voltage	3.63V
End-Of-Charge Current (I_{EOC})	20mA

5.1 System Overview

Figure 4 depicts the high-level flow of the SOC estimation application. When the CLI application starts, the command *init bfe* initializes the BFE device. After the successful execution of the initialization command, the SOC estimation application is started by entering the command *soc* <*soc_0*>; soc_0 is an optional parameter used to specify the initial SOC (in percentage) of the battery if the user knows it beforehand. The execution of the soc command initiates the routine *soc_start()* in the **r_soc.c** file.

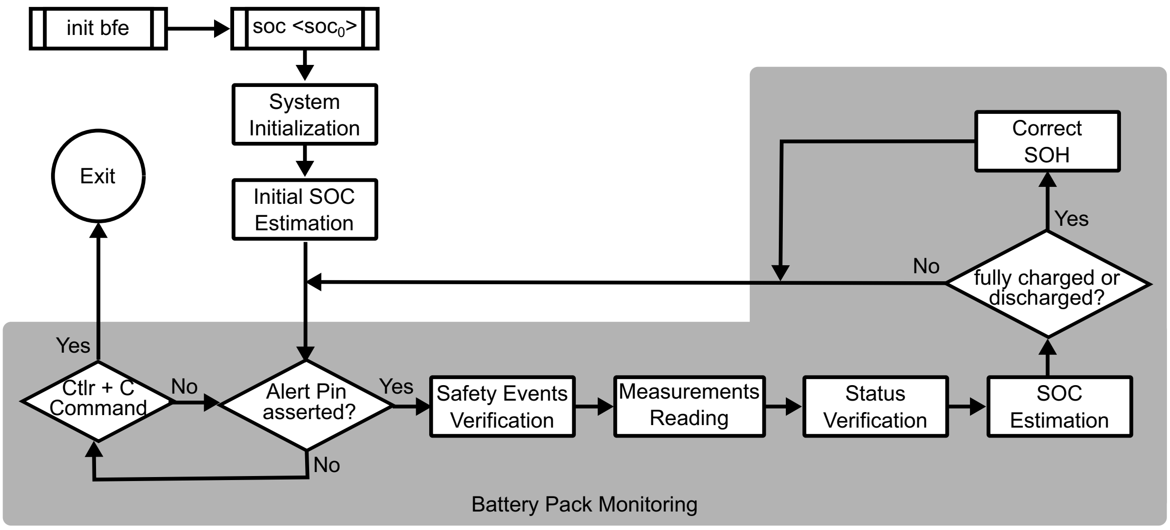


Figure 4. High Level System Flow of the SOC Estimation Application

5.1.1 soc_start()

soc_start() is the main body of the SOC estimation application. This routine sets the configuration parameters of the BFE (*init_system()*), calls the routine to estimate the initial SOC (*get_initial_soc()*), and starts the monitoring of the battery pack (*monitor_pack()*), which includes the coulomb counting implementation, SOC estimation and SOH correction. The application outputs messages according to the macro CSV_MODE defined in the **r_soc.h** file. When CSV_MODE is set to 1 (default value), the application shows data as Comma-Separated Values (CSV). This function allows logging the data received through the terminal. The log file stored as a CSV file can be imported and the data analyzed offline. Figure 5 shows the CSV output, which is the mode used in this application note to register and store the data reported by the BMS. The application enters the *monitor_pack()* routine, which returns on reception of the exit command CTRL + C.

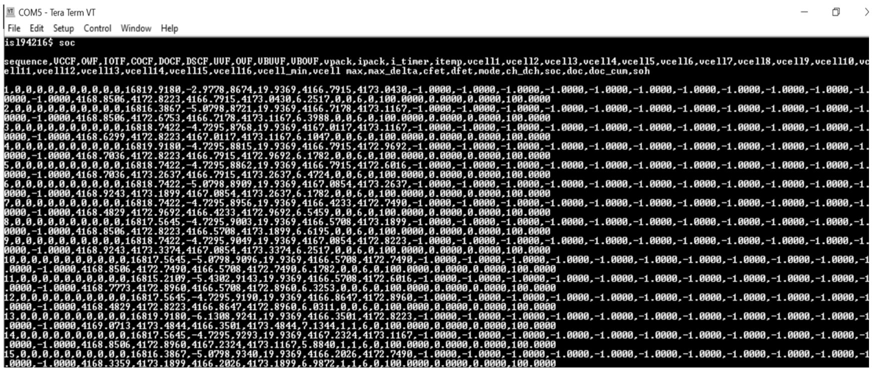


Figure 5. Comma-Separated Value Output (CSV_MODE = 1)

5.1.2 init_system()

init_system() initializes system settings executing the following actions:

1. Read all BFE registers to update the shadow register image stored in the MCU.
2. Open the Low Power Mode (LPM) module of the MCU. This module sets the MCU to LPM when neither IRQ nor CLI command is received.
3. Initializes the IRQ pin used to detect the assertion of the BFE alert pin.
4. Configures LPM operation of the BFE using the LPM configuration settings retrieved by the function *get_lpm_config()*.
5. Set the Discharge Short Circuit (DSC) threshold.
6. Enable load detection during LPM and automatic FETs shut-off upon violation of safety thresholds.
7. Disable cell voltage overvoltage fault while discharging, and cell voltage undervoltage fault while charging.
8. Set the faults that shall assert the Alert pin upon threshold violation. The masked and unmasked faults are retrieved by the function *get_masks_config()*.
9. Set safety thresholds: Discharge Short-Circuit (DSC), Discharge Overcurrent (DOC), Charge Overcurrent (COC), cell and pack overvoltage and undervoltage. The threshold values are defined in r_soc.h according to the cell specifications and system configuration. The macros can be modified to use the application in a system with different cells and meet the specifications and requirements of user’s system.
10. Select the cells that must be enabled in the BFE to monitor the battery pack. The routine uses the constant selected_cells, which is set by the macro BMS_CELLS_SELECTION(*NUM_PACK_CELLS*). This macro defined in r_soc.h sets the correct cell positions according to the number of cells specified by *NUM_PACK_CELLS*.

5.1.3 get_initial_soc()

The first action in *get_initial_soc()* is reading the current cell voltage by triggering a cells voltage measurement event. If the command is correctly executed, the routine determines the minimum and the maximum cell voltages calling the static function *get_min_max_vcell()*. This function sets the static float variables *s_min_vcell* and *s_max_vcell* with the minimum and maximum values, respectively. The routine then checks the global variable *g_initial_soc*, which contains the value of the optional soc_0 parameter of the soc command. If no soc_0 or a negative value has been entered, the initial SOC is estimated as follows:

1. Obtain the released capacity by calling the *get_val_lookup_table(uint16_t *tbl_ptr, uint16_t in_value)* routine. This routine returns the released capacity of a cell for the OCV value specified by *in_value*. **tbl_ptr* must point to an array containing the characterized $Q_{released}$ versus OVC data as tbl_qr_ovc = [n, x_1, y_1, x_2, y_2,...., x_n, y_n]; where n is the number of [x_i,y_i] (i = 1,2,…,n]) tuples in the array, x_i is the cell OCV in mV with $x_i > x_i + 1$, and y_i is the released capacity in mAh for the corresponding x_i.

In *get_initial_soc()*, **tbl_ptr* points to the constant array *g_ocv_q_released_mah_table* in the **r_icr1865026j_02.c** file. This array contains the $Q_{released}$ versus OVC data of the ICR18650-26J cell at low discharge rate (0.2C).

2. If returned value *lut_value* is less than the nominal capacity set by the macro CELL_NOMINAL_CAPACITY_MAH, *releasable_capacity* Q is calculated subtracting *lut_value* from the nominal capacity. If *lut_value* is greater, *releasable_capacity* is set 0mAh.
3. Call the routine *bfe_battery_z_k_init()* with the nominal capacity and the releasable capacity as parameters. This function defined in the **r_coulomb_counting.c** file initializes the variables used by the coulomb counting implementation to estimate the SOC of the battery pack.

5.1.4 monitor_pack()

monitor_pack() is the kernel routine of the SOC estimation application. This function monitors the safety faults reported by the BFE, verifies the status of the battery pack, and estimates its SOC. The function starts by enabling the charge pump, so that FETs can be turned on actively upon load or charger detection. Then the routine starts the continuous scan using the configuration settings stored in the static variable *s_soc_sc_config*. Once the device successfully starts continuous scan, the routine initializes the variable *s_last_ipack_timer* with the current timer. The code that implements coulomb counting uses the static variables *s_last_ipack_timer* and *s_last_ipack* to integrate the pack current.

The routine starts an infinite loop that sets the MCU to low power mode if neither Interrupt Request (IRQ) caused by the alert pin is present nor CLI command is received. If the user enters a command over the terminal, the MCU exits the sleep state and processes the command in the function *R_BFE_CLI_PROCESS()* function. The key sequence CTRL+C calls the execution of the *soc_stop()* function, which disables the IRQ interruption, stops continuous scan, turns FETs off, and reads the BFE status and data registers. When the BFE asserts the alert pin, the MCU exits the sleep state and executes the designated Interrupt Service Routine (ISR) *alert_callback()* in **r_bms.c**, which sets the global variable *g_alert_pin_asserted* to true. This leads to the execution of the code conditioned by *g_alert_pin_asserted*, which first increments log-related variables that control the console output rate, and then continues with a for loop that calls sequentially the callback functions contained in the static constant *callbacks_vector*. The *callbacks_vector* is a union of type *u_monitoring_callbacks_vector_t* defined in the **r_bms.h** file. It stores the structure *st_callbacks_prio*, which contains fields with pointers to the callback functions called in the for loop. Each callback shall return true if a critical fault or error is detected, and false if no abnormal conditions are reported by the BFE. If a callback indicates the presence of a critical condition by returning true, the routine *critical_fault_detected()* is called. This function starts a periodic timer interruption, whose ISR turns a LED on and off to notify the user of the existence of a critical fault. Callbacks are divided into three groups according to the performed tasks. Table 2 shows the groups, their associated callbacks, and a short description of the performed tasks.

5.1.4.1 Safety Thresholds Verification

In battery-powered devices, events such as over/undervoltage (OV, UV), over-temperature (OT), and overcurrent (OC) can result in fire and explosion if the cells of the battery pack are incorrectly or not managed. These callbacks aim at preventing such events by ensuring the operation of the battery pack within the safety thresholds specified by the cell manufacturer.

- *dsc_callback()*
    - *dsc_callback()* has the highest priority. The callback reads first the priority faults register (0x63), which contains the DSC fault bit (DSCF 0x63.2). This bit is set by the analog comparison between the DSC threshold and the voltage across the sense resistor, so it is independent of the measurements performed in continuous scan mode. If the DSC fault bit is set, the callback turns off both FETs. The device is set to turn off FETs off automatically upon detection of this event, so this is a redundant action to ensure both FETs are in off state.
- *read_registers_callback()*
    - *read_registers_callback()* sets the registers contained in the MCU register bank variable g_isl94216_registers to the current BFE values. Before reading all registers, the callback calls the function

wait_bfe_available(), which waits until the BFE finishes the measurements performed during the continuous scan. This ensures the BFE data registers contain updated values. If the BFE remains busy (0x01.2 Busy = 1), the function prints out a warning text indicating the registers might not contain updated values.

- *doc_callback()*
  - *doc_callback()* checks whether the DOC fault bit (0x63.3 DOCF) is set. If the fault is present, the callback turns FETs off (redundant action) and read current and DOC threshold to print out their corresponding values if the CSV_MODE is off.
- *coc_callback()*
  - *coc_callback()* checks whether the COC fault bit (0x63.4 COCF) is set. If the fault is present, the callback turns FETs off (redundant action) and read current and COC threshold to print out their corresponding values if the CSV_MODE is off.
- *vcell_ov_callback()*
  - *vcell_ov_callback()* checks whether the Overvoltage fault bit (0x63.0 OVF) is set. If the fault is present, the callback verifies if charge current is flowing through the pack. Since the presence of overvoltage together with charge current can be a safety issue, the callback turns FETs off (redundant action) if the Charge indicator (0x67.6 CHRGI) is set. If the UV event is detected, this callback sets the static variable *s_vcutoff_reached* to true, which indicates the battery pack is fully discharged. Finally, the callback reads cell voltages and prints out the maximum cell voltage and the V_{CELL} OV threshold if the CSV_MODE is off.
- *vpack_ov_callback()*
  - *vpack_ov_callback()* checks whether the V_{BAT1} Overvoltage fault bit (0x65.7 VBOVF) is set. If the fault is present, the callback verifies if charge current is flowing through the pack. As in voltage cell overvoltage, the presence of overvoltage event together with charge current may derive in a safety issue, so the callback turns FETs off (redundant action) if the Charge indicator (0x67.6 CHRGI) is set. Finally, the callback reads pack voltage and prints out its value and the V_{BAT1} OV threshold if the CSV_MODE is off.
- *vcell_uv_callback()*
  - *vcell_uv_callback()* checks whether the Undervoltage fault bit (0x63.1 UVF) is set. If the fault is present, the callback verifies if discharge current is flowing through the pack. The presence of undervoltage event together with discharge current may derive in a safety issue, so the callback turns FETs off (redundant action) if the Discharge indicator (0x67.7 DCHRGI) is set. If the UV event is detected, this callback sets the static variable *s_vcutoff_reached* to true, which indicates the battery pack is fully discharged. Finally, the callback reads cell voltages, and prints out the maximum cell voltage and the V_{CELL} UV threshold if the CSV_MODE is off.
- *vpack_uv_callback()*
  - *vpack_uv_callback()* checks whether the V_{BAT1} Undervoltage fault bit (0x65.6 VBUVF) is set. If the fault is present, the callback verifies if discharge current is flowing through the pack. The presence of undervoltage event together with discharge current may derive in a safety issue, so the callback turns FETs off (redundant action) if the Discharge indicator (0x67.7 DCHRGI) is set. Finally, the callback reads pack voltage and prints out its value and the V_{BAT1} UV threshold if the CSV_MODE is off.
- *read_other_callback()*
  - *read_other_callback()* checks whether the following faults have occurred:
    - VCC undervoltage
    - Open wire
    - External Temperature/Auxiliary threshold violation
    - Charge pump not ready
    - Oscillator frequency error or Communication Timeout
    - Regulator overcurrent while in IDLE or SCAN mode
    - Regulator overcurrent while in Low Power Mode
    - VTEMP undervoltage

- *iotf_callback()*
  - This callback checks the occurrence of internal over-temperature event, which is indicated by the Internal Over-Temperature Fault Indicator bit (0x63.5 IOTF). If IOTF is set, the callback turns FETs off (redundant action), reads the current internal temperature and fault threshold, and prints their values if the CSV_MODE is off.
- *delta_callback()*
  - This callback checks whether the maximum difference between cell voltages is larger than the maximum delta threshold set in the BFE. This is an end-of-life event issue, the callback turns FETs off (redundant action) and returns true if the event is detected. The fault presence is indicated by the Delta V_{CELL} Max Fault bit (0x66.3 DVCF).
- *iotw_callback()*
  - This callback checks the BFE reports the warning temperature threshold has been reached. This is indicated by the Internal Over-Temperature Warning Indicator bit (0x66.6 IOTW). This event is not considered a safety issue, so the callback just reads the current internal temperature and the warning threshold and prints their values if the IOTW bit is set.

Table 2. Callbacks Grouped According to Performed Tasks

Group	Callbacks	Task
Safety Thresholds Verification	dsc_callback	Check the occurrence of DSC Overcurrent event
	read_registers_callback	Update BFE registers stored in MCU memory
	doc_callback	Check the occurrence of DOC event
	coc_callback	Check the occurrence of COC event
	vcell_ov_callback	Check the occurrence of Cell Overvoltage event
	vpack_ov_callback	Check the occurrence of Pack Overvoltage event
	vcell_uv_callback	Check the occurrence of Cell Undervoltage event
	vpack_uv_callback	Check the occurrence of Pack Undervoltage event
	read_other_callback	Check the occurrence of other faults
	iotf_callback	Check the occurrence of internal over-temperature fault event
	delta_vcell_callback	Check the occurrence of maximum cell voltage delta event
	iotw_callback	Check the occurrence of internal over-temperature warning event
Battery Pack Measurements and Status	read_measurements	Read battery pack measurements: cell voltages (V_{CELLS}), pack voltage (V_{PACK}), pack current (I_{PACK}), minimum and maximum cell voltages, maximum delta voltage, and FETs status
	read_status_callback	Establish the current mode of the BFE. Determine the validity of measurements. Determine whether charge, discharge or no current is flowing. Turn FETs on upon charger or load detection and ensure monitoring by starting continuous scan mode. Clear faults to verify whether the fault condition has been removed.
SOC Estimation	get_soc	Estimate the SOC of the battery pack. Perform SOH correction at fully discharged and fully charged states

5.1.4.2 Battery Pack Measurements and Status

When the assertion of the alert pin occurs, the BFE starts a sequence of measurements that set registers with values that must be transformed into application-readable units, such as Voltage, Amperes, Celsius degrees, which enables displaying and analyzing the information retrieved by the BFE. In addition, the BFE performs a series of comparisons and verifications of thresholds that set registers whose content represent the status of the battery pack. For example, bits that indicate whether the pack is charging, discharging or no current is passing through, detection of load or charger, mode of the BFE, etc. The callbacks *read_measurements* and *read_status_callback* serve the purpose of registering the measurements reported by the BFE in physical units and determine the status of the battery pack.

- *read_measurements()*
  - This callback reads, stores in static variables, and prints out the measurements of the following physical values measured or set during the sequence executed during the continuous scan mode:
    - Battery pack Voltage in mV stored in *s_vpack_mv*
    - Battery pack Current in mA stored in *s_current_ipack_ma*
    - Cells Voltages in mV stored in *s_vcells_mv*
    - Internal Temperature in °C stored in *s_bfe_temperature*
    - Maximum and minimum cell voltages calling the routine *get_min_max_vcell()*
    - Other values: FETs status, and current value of the Ipack timer (0x54-0x57) stored in *s_current_ipack_timer*
- *read_status_callback()*
  - The r*ead_status* callback starts by initializing the boolean static variable *s_valid_measurements* to false. When the BFE is LPM, the BFE data registers do not contain valid values, so this variable indicates the validity of the measurements read in *read_measurements* callback. The callback continues with the determination of the current BFE mode. If the BFE is in IDLE or SCAN mode, the measurements are valid, so *s_valid_measurements* is set to true. Otherwise, the variable remains false. If the BFE is in LPM, the load verifies whether the bit mask that prevents a load detection event from asserting the alert pin (0x87.4) is set. If the bit was set to 1, the bit is cleared to enable the BFE to assert the alert pin when a load is connected to the battery pack. FETs are also turned off actively when in LPM to ensure consistency between their actual states (off in LPM) and the values stored in the MCU registers. The next steps in the callback keep track of the number of consecutive valid measurements. Using the variable *s_num_valid_measurements*, the callback indicates if two consecutive current samples are valid, so that the SOC estimation callback can use them to integrate the current. The callback then determines if charge or discharge current is flowing through the battery pack. If either the Load Present bit (0x67.4 LD PRESI) or the Charger Present bit (0x67.5 CH PRESI) is set, and no current is flowing (*s_current_flowing* = false), the code first masks load detection events, and then starts continuous scan mode. Successful start of the continuous mode results in turning on FETs to allow the flow of current, whereas a failure in the *startContinuousScan()* call generates a critical error. The final part of the callback verifies whether the previous callbacks have detected a critical fault by checking the value of *s_critical* fault. If the variable *s_critical_fault* is true, the routine clears all faults by calling the *p_clearAllFaults* BFE function to verify whether fault conditions detected in previous scans have been resolved.
- SOC Estimation: *get_soc()*
  - The callback *get_soc()* performs the routines and logic that estimate the current SOC of the battery pack and correct the SOH when the battery is fully charged or discharged. The first conditional block in the callback verifies whether current is flowing and at least two valid measurements have been processed. If both conditions are met, the callback calls the routine *bfe_battery_z_k_get()* defined in the r_coulomb_counting.c file. This function performs current integration by implementing the trapezoidal rule numerical method. The trapezoidal rule belongs to Quadrature rules based on interpolation. This method approximates the integral of a curve as the area of the trapezoid formed by points t_1, $f(t_1)$, t_2 and $f(t_2)$, where $t_1 < t_2$ are the time at which the samples $f(t_1)$ and $f(t_2)$ have been measured. Therefore, the routine *bfe_battery_*z_k_ge*t()* uses two consecutive valid current measurements: *s_current_ipack_ma* and *s_last_ipack;* and their corresponding timer stamps: *s_current_ipack_timer.value* and *s_last_ipack_timer.value* to estimate ∆DOD according to

Equation 7 as

(EQ. 13)

$$\Delta DOD[mAh] = (-1) \times \frac{(t_2 - t_1) \times BFE\_IPACK\_TIMER\_STEP\_S \times (I_{PACK}(t_1) + I_{PACK}(t_2))}{2} \times \frac{1}{3600}$$

The result of ΔDODs used to update DOD and the SOC.
When the *bfe_battery_z_k_get* returns, the callback determines whether the battery pack has reached its fully charged state. This is done by checking if:

- The rounded minimum cell voltage equals the manufacturer-specified voltage value V_F = FULL_VCELL_MV (macro defined in the **r_icr1865026j_02a.h** file) AND
- The current pack *s_current_ipack* is positive AND
- The current pack *s_current_ipack* is lower than I_{EOC} = FULL_I_MA (macro defined in the **r_icr1865026j_02a.h** file)

Because the measurements can fluctuate due to transients and errors inherent in current and voltage measurements, the fulfillment of these three conditions are subject to be present during at least FILL_I_COUNT times, a macro defined in the **r_soc.h** file. When FIL_I_COUNT is reached, the callback calls the routine *bfe_battery_full_correct()* to correct the SOH at fully charged state according to Equation 11. The callback then turns FETs off to avoid reaching the OV threshold.

If an UV event is present, the *vcel_uv_callback()* must have set the static variable *s_vcutoff_reached* to true to indicate the battery pack is fully discharged. Therefore, the *get_soc* callback calls the routine bfe_battery_z_empty_correct*()* to correct the SOH at fully discharged state according to Equation 10.

The last part of the *get_soc()* callback updates the static variables *s_last_ipack* and *s_last_ipack_timer* with the current values, this is, *s_current_ipack_ma* and *s_current_ipack_timer,* respectively. These values shall be used as $I_{PACK}(t_1)$, t_1, respectively, to calculate the charge transferred according to Equation 13. The callback finishes by printing out the estimated SOC and the cumulative DOD, and returns false.

The callbacks and routines explained above are implemented and run in a test environment to determine the accuracy of the coulomb counting method in the estimation of the SOC featuring the Renesas RAA489206/ISL94216A BFE. The following section presents the measurement data, results analysis and the observations that BMS designers should consider when designing and implementing a SOC estimation method for industrial applications.

5.2 Experimental Tests and Results

5.2.1 Setup Description

Figure 6 shows the test setup used to verify the implementation of the SOC estimation in BMS for a battery pack with ICR18650-26J Li-Ion cells connected in series. The BMS consists of an RA4W1 MCU evaluation kit and a BFE, which can be either the RAA489206 or the ISL94216A. The MCU evaluation board runs the CLI application with the SOC estimation command and is connected through serial interface to a PC running a terminal emulator (such as TeraTerm). The emulator displays the output of the SOC estimation command and stores it as a CSV file. A Source Measure Unit (SMU) is connected to the positive and negative pack terminals using Kelvin connection (4-Wire). The SMU is a calibrated device capable of measuring and registering accurately both voltage and current while sourcing or sinking current. The SMU is the reference and calibrated device that provides the actual measurements.

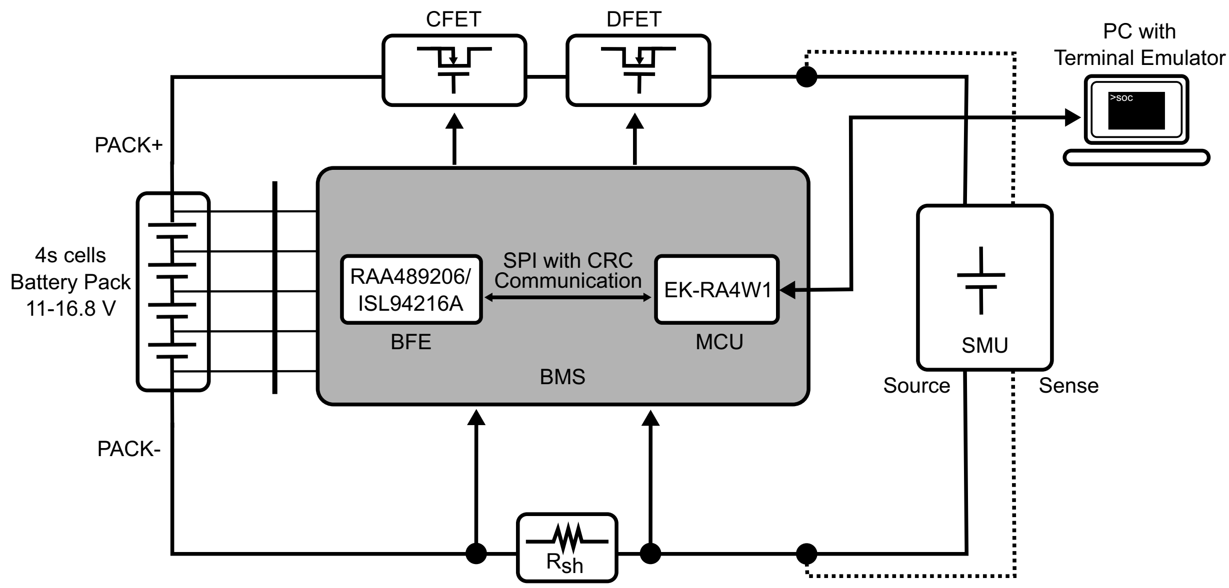


Figure 6. Test Setup for the Implementation of SOC Estimation

The test setup and the programmable SMU enable performing discharge and charge of the battery pack. To discharge the battery pack, the output voltage of the SMU is set below the battery pack voltage. This enables the sink mode of the SMU, so discharge current flows from the cells to the SMU, which measures the released charge ($Q_{released}$). On the other hand, charge is performed by setting the SMU output voltage above the battery pack voltage, which transfers charge from the SMU to the cells.

During the tests, the BMS reports measurements, internal variables, events, and estimations over the terminal emulator as shown in Figure 5. Table 3 lists the labels and the description of the values reported by the BMS. Concurrently, the SMU measures and stores battery pack voltage, current and time stamps of the battery pack, and calculates the transferred charge and the true SOC when a test finishes. To show the accuracy of the SOC estimation provided by the Coulomb Counting implementation described in this application note, two tests are performed: Current Integration and SOC Estimation.

The Current Integration test aims to determine the accuracy of the integration approximation method by discharging the battery pack using six current profiles: constant, triangular, saw, exponential, large pulse, and short pulse. These profiles intend to reproduce some current patterns that might present in industrial applications, so that the results represent the expected accuracy if the described SOC estimation is implemented in their BMS.

The SOC Estimation test produces the main results of this application note. It shows the accuracy of the SOC reported by the Coulomb Counting implementation featuring the RAA489206/ISL94216A BFE. In this test, the pack is initially brought to fully discharged state at a low rate and rests until reaching a well-relaxed state. Then, the battery is cyclically brought from fully discharged state to fully charged, and vice versa, while both BMS and SMU register and store the SOC. Charge is done according to cell manufacturer’s specifications and using the Constant Voltage / Constant Current (CC/CV) method, whereas discharge is performed by applying a constant current of 1.05 A (0.4C).

Table 3. Measurements and Status Variables Printed Out during Tests

Variable	Description	Variable	Description
Sequence	Output sequential number	itemp	Internal temperature in °C
VCCF	Vcc undervotage fault bit	vcell1…vcell16	Cell [1..16] voltage in mV
OWF	Open-wire fault bit	vcell_min	Minimum cell voltage in mV
IOTF	Internal over temperature fault bit	vcell_max	Maximum cell voltage in mV

Table 3. Measurements and Status Variables Printed Out during Tests (Cont.)

Variable	Description	Variable	Description
COCF	Charge Overcurrent Fault bit	vcell_max	Maximum cells voltage delta
DOCF	Discharge Overcurrent Fault bit	cfet	CFET state (1=on, 0=off)
DSCF	Discharge Short Circuit Fault bit	dfet	DFET state (1=0n, 0=off)
UVF	Cell Undevoltage Fault bit	mode	BFE mode
OVF	Cell Undervoltage Fault bit	ch_dch	Discharging=-1, charging=1, none=0
VBUVF	Vpack Undervoltage Fault bit	soc	State of Charge
VBOVF	Vpack Overvoltage Fault bit	doc	Depth of Discharge
vpack	Pack voltage in mV	doc_cum	Cummulative DOD in mAh
ipack	Pack current in mA	SOH	State of Health
i_time	Current timer value		

5.2.2 Current Integration Accuracy

Table 4 shows the current profiles generated by the SMU to evaluate the integration accuracy. SMU measurements (reference current) are shown using orange solid lines, whereas measurements reported by the BMS are shown using blue dashed lines. The results show the BMS can follow with high precision the reference current reported by the SMU. The small divergences are due to the offset current and the use of sample average (4 samples) in BFE measurements, which introduces some measurement delay. Although the short pulse profile exhibits the highest divergence in terms of delay from the true value, the reported magnitude if almost identical to the reference values. This profile is a worst-case scenario with short discharge pulses (~30 ms) and period (~1 s), which are assumed rare in industrial applications.

Table 4. Current Profiles Tested During Current Integration Test

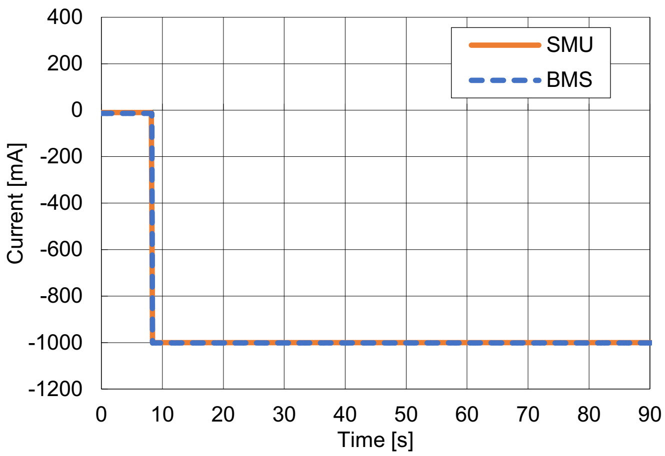
<table>
<tr><th>Current Profile</th><th>SMU (orange) and BMS (blue dashes) Measurements</th></tr>
<tr><td>Constant</td><td>
</td></tr>
</table>

Table 4. Current Profiles Tested During Current Integration Test (Cont.)

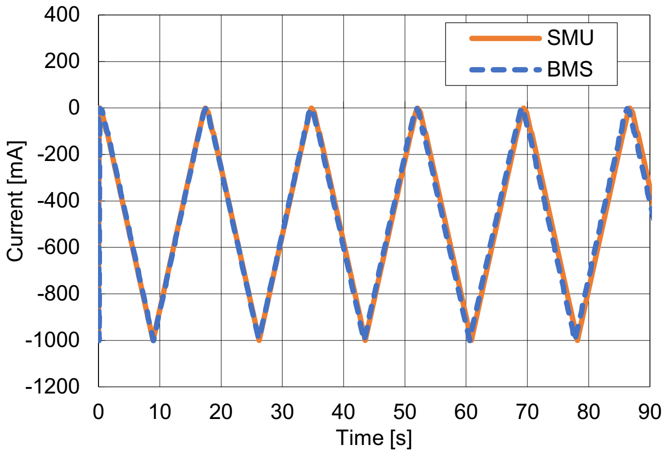
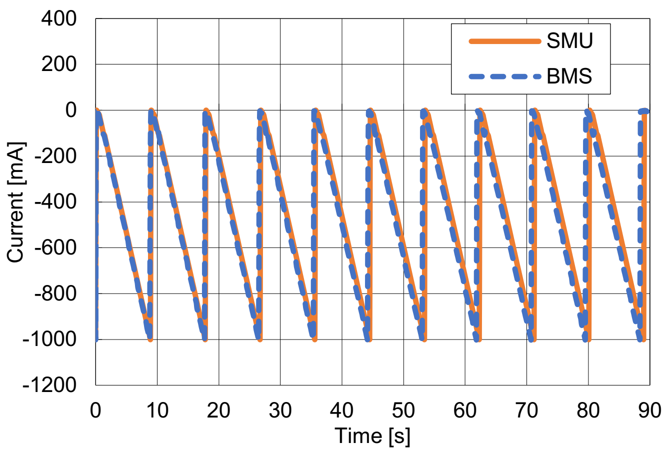
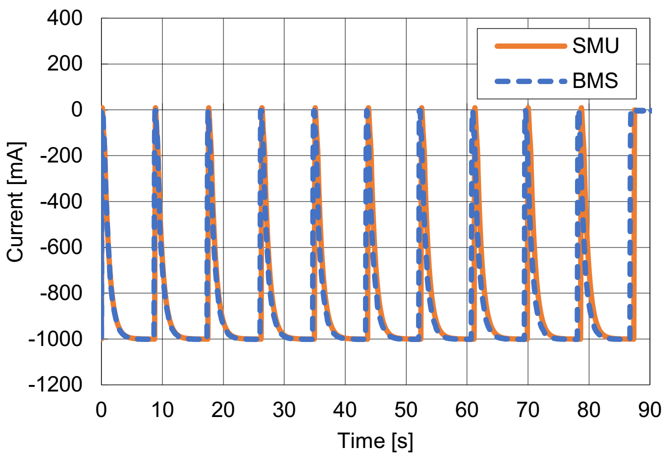
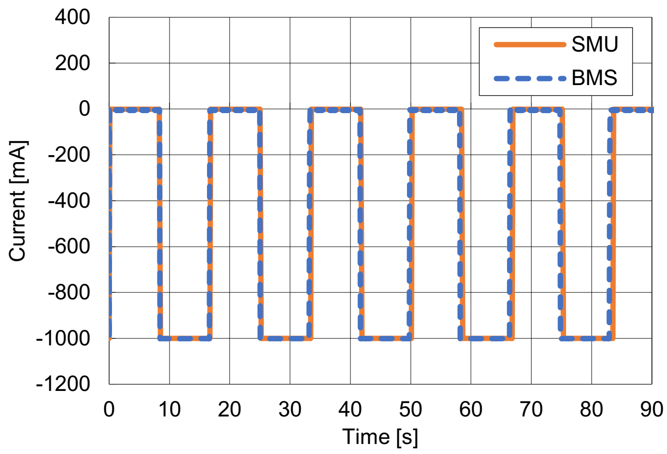
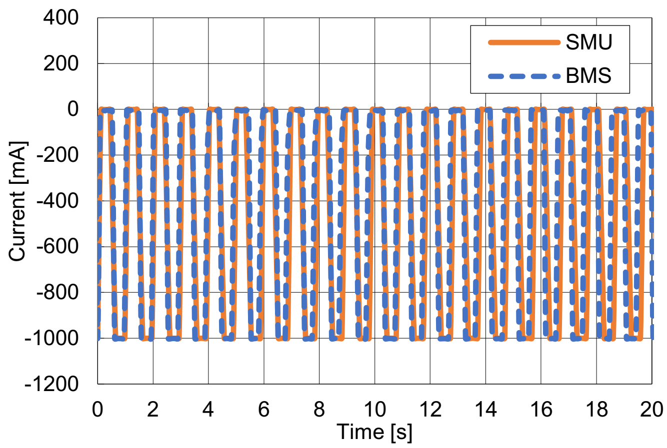
Current Profile	SMU (orange) and BMS (blue dashes) Measurements
Triangular	
Saw	
Exponential	

Table 4. Current Profiles Tested During Current Integration Test (Cont.)

Current Profile	SMU (orange) and BMS (blue dashes) Measurements
Long Pulse	
Short Pulse (scaled here to show the profile details)	

The error in the current integration reported by the BMS is calculated using the cumulative released charge reported by the BMS ($cum(Q_{released\text{-}BMS})$) and the true cumulative released charge ($cum(Q_{true})$) reported by the SMU at each current sample point as:

(EQ. 14)
$$error_{Q_{released}}[\%] = \frac{cum(Q_{true}) - cum(Q_{released\text{-}BMS})}{cum(Q_{true})}$$

Table 5 shows the cumulative charge $cum(Q_{released\text{-}BMS})$ reported by both the BMS and the SMU (reference device) and their respective errors for each current profile. The figures depicting the cumulative charge show that the BMS reports closely the released charge with respect to the reference value. This is confirmed by the figures detailing the error, which converges rapidly towards values below 2%. The significant error at the beginning is mainly due to the offset component of the measured current. When the BMS just starts counting the released charge, the cumulative charge is small, so any additional small component integrated by the BMS, such as the error due to the offset current, becomes dominant in the reported value. The resulting deviation is augmented by the reference value being closed to zero in the denominator of Equation 14. When the cumulative charge is significantly larger than the error component (starting at around 10s), the error decreases rapidly.

The error defined in Equation 14 shows the evolution of the error during the test, but does not provide a unique indicator of the integration accuracy. Table 6 shows the Normalized Root Mean Squared Error (NRMSE) of the cumulative charge reported by the BMS. These values provide a unique measure of the difference between cum($Q_{released\text{-}BMS}$) and cum(Q_{true}). Because NRMSE values are normalized, they can be compared with each other even though they are obtained from different current profiles and different total cumulative charge. The NRMSE values show that the current integration estimation exhibits less than 1% of error for all the tested current profiles. Although such accurate integration does not result in high SOC estimation accuracy, it is a determining component that does contribute to achieving it. The SOC test described in the following section shows the resulting accuracy of SOC estimations.

Note: The presented tests were conducted for discharge current profiles. However, the observations and conclusions are valid for charge currents, too. The high accuracy obtained for constant and exponential current profiles indicate that the same order of accuracy can be obtained during the CC and CV phases of the charge method.

Table 5. Cumulative $Q_{released}$ and Error of the reported $Q_{released}$ as Accuracy Indicator

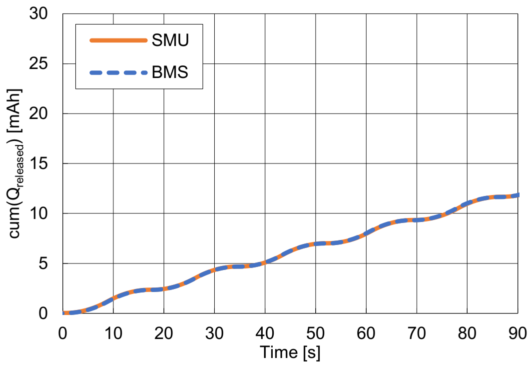
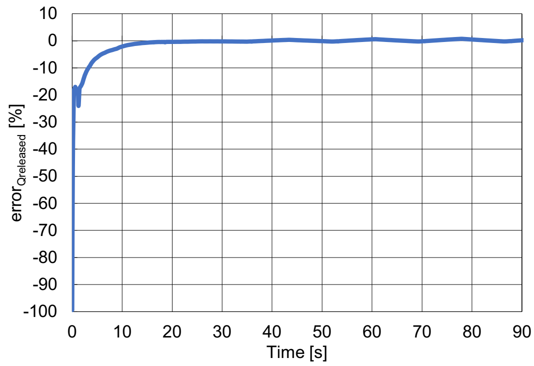
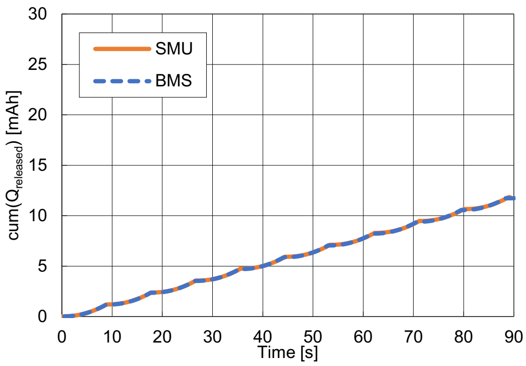
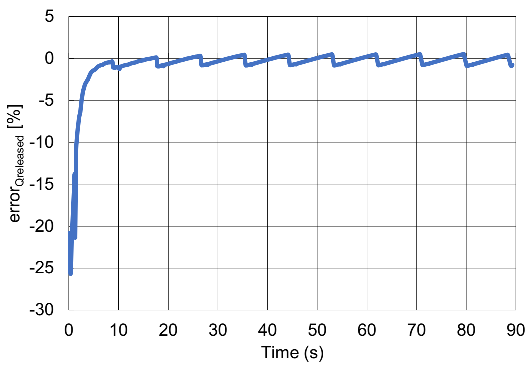
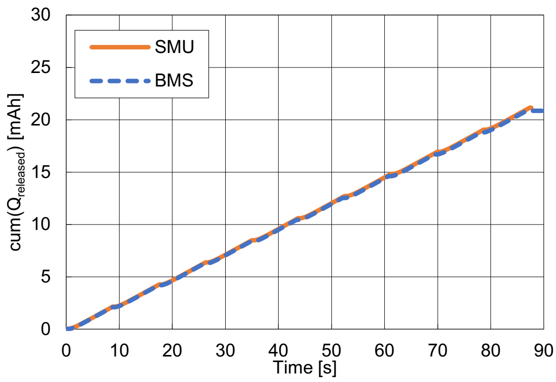
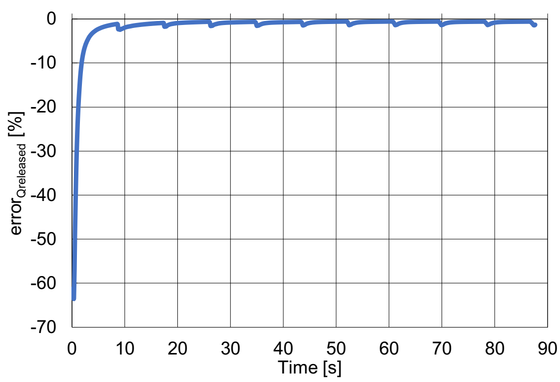
<table>
<tr><th>Current Profile</th><th>Cumulative Released Charge</th><th>Error of the $Q_{released}$ [%]</th></tr>
<tr><td>Triangular</td><td>
</td><td>
</td></tr>
<tr><td>Saw</td><td>
</td><td>
</td></tr>
<tr><td>Exponential</td><td>
</td><td>
</td></tr>
</table>

Table 5. Cumulative $Q_{released}$ and Error of the reported $Q_{released}$ as Accuracy Indicator

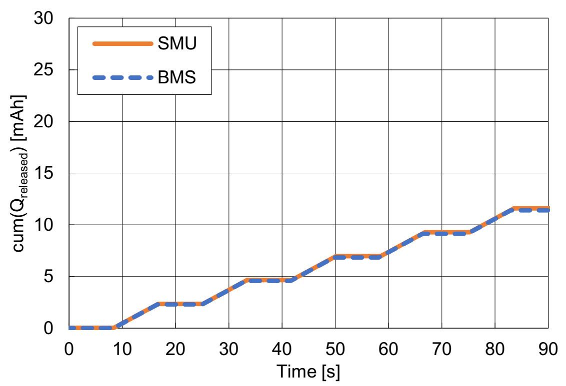
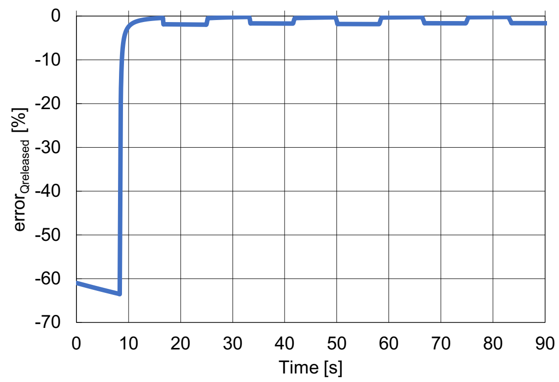
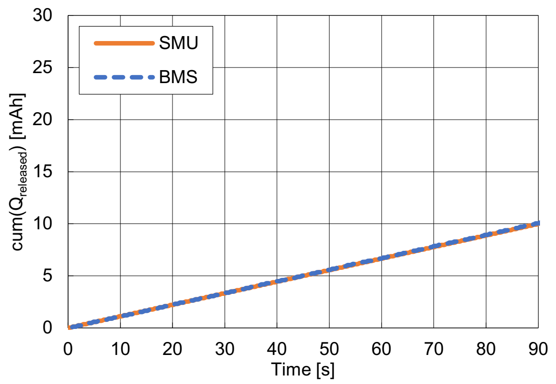
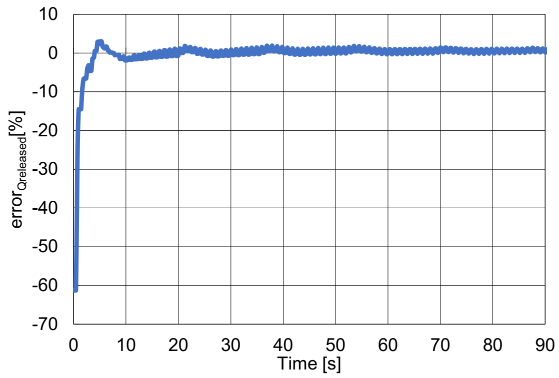
Current Profile	Cumulative Released Charge	Error of the $Q_{released}$ [%]
Long Pulse		
Short Pulse		

Table 6. Normalized Mean Squared Error (NRMSE) of the Cumulative Charge

Profile	Constant	Triangular	Saw	Exponential	Long Pulse	Short Pulse
NRMSE [%]	0.078592	0.233349	0.235651	0.004384	0.736070	0.410588

5.2.3 SOC Estimation Test

This test evaluates the accuracy of the SOC estimation provided by the Coulomb Counting implementation described in this application note. The test consists in performing full charge and full discharge cycles on the battery pack while the BMS reports its estimated SOC (SOC_{BMS}), and the SMU reports the true battery pack SOC (SOC_{true}).

To fully charge the battery pack, the SMU is set to source a maximum current of 1.05 A (0.4C) and supply voltage 16.8V. This corresponds to 4.2V/cell, which is the charge voltage specified by the cells manufacturer (Table 1). During the CC phase, the charge current is 1.05A and the battery pack voltage increases. When the battery pack voltage reaches the SMU output voltage, the CV phase starts, so the SMU keeps the output voltage at 16.8V while the current decreases exponentially. The full charge test finishes as soon as the SMU measures pack current is equal to the I_{EOC} specified by cell manufacturer (20mA).

The battery pack is brought to fully discharge state by setting the SMU to supply 11V (4 × $V_{CUT\text{-}OFF}$) and limit the current to 1.05A (0.4C). As the battery is fully charged, its terminal voltage is about 16.8V. The voltage difference between the SMU output and the battery pack enables the sink mode of the SMU, which starts discharging the battery pack at the configured limit current. The battery pack discharges at constant rate until the pack voltage reaches $V_{PACK\text{-}CUT\text{-}OFF}$ = 4 × 2.75V =11V, when the SMU finishes the test.

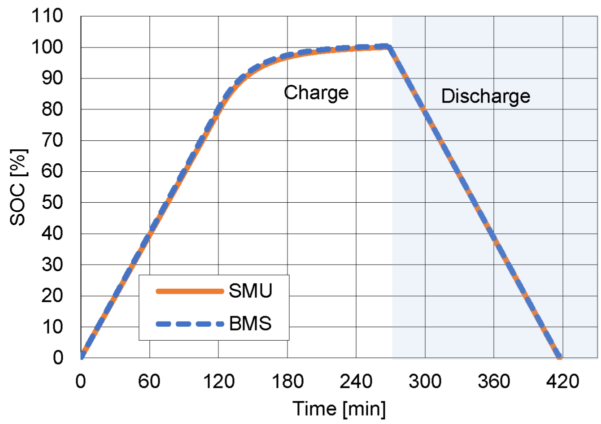
During both discharge and charge cycles, the SMU integrates and calculates the cumulative charge transferred to and from the battery pack. The cumulative charge at each measurement point corresponds to $Q_{true\text{-}released}$. Because charge/discharge cycles start from either fully charged state of fully discharged state, the total cumulative charge corresponds to the real total capacity $Q_{real\text{-}total}$ when their respective conditions are reached: $V_{CUT\text{-}OFF}$ for discharge, and V_F and I_{EOC} for charge. The SMU can then calculate the real SOC at each point as

(EQ. 15) $$SOC_{true}[\%] = \frac{Q_{real\text{-}total} - Q_{true\text{-}released}}{Q_{real\text{-}total}}$$

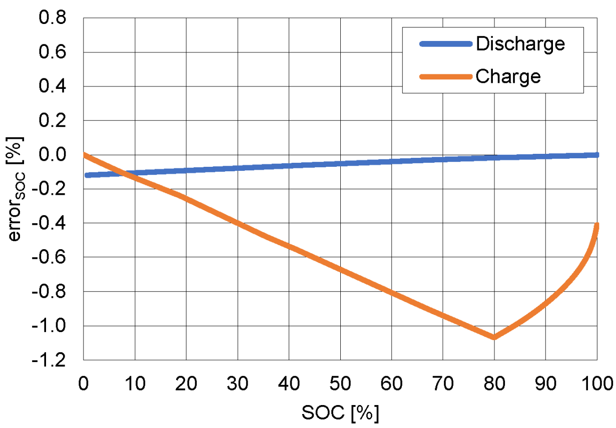
The accuracy of the SOC estimation is measured in terms of the error of SOC_{BMS} with respect to SOC_{true} as

(EQ. 16) $$error_{SOC}[\%] = SOC_{true} - SOC_{BMS}$$

Figure 7, Figure 8, and Figure 9 depict the SOC estimated by the BMS (SOC_{BMS}) and reported by the SMU (SOC_{true}) of three charge/discharge cycles, and their corresponding errors. The depicted cycles are representative of the average (Figure 7), minimum (Figure 8), and maximum (Figure 9) errors obtained during the conducted tests. The results show an accurate estimation of the reference SOC of the battery pack during both discharge and charge cycles. During charge, the error starts at 0, indicating precise estimation of the initial SOC (SOC_i). The error increases linearly until the pack reaches 80% of the SOC, which occurs 120 minutes after starting the test. This corresponds to the CC phase, so the linear error is mainly due to offset error in current measurements. The maximum error magnitude during this phase is about 1.1%. When the charge starts its CV phase, the current decreases, and so do the charge rate and the estimation error magnitude, which is 0.7% at most. When the battery pack reaches its fully charged state after 240min (4.5 hours), the implemented algorithm corrects the SOC estimation setting its value to 100%. SOH correction is performed, too, but the evaluation of its impact on SOC estimation accuracy needs many cycles to be noticeable. This is not evaluated in this application note. The discharge phase discharges the battery pack at a constant rate of 0.4 C, so it takes 150min (2.5 hours) to get the fully discharged state. The estimation error starts at 0% due to the performed SOC correction, and its magnitude increases linearly as the battery pack discharges. However, the increment slope is so low that the final error is 0.1% in average and 2% at most when the fully discharged state is achieved.

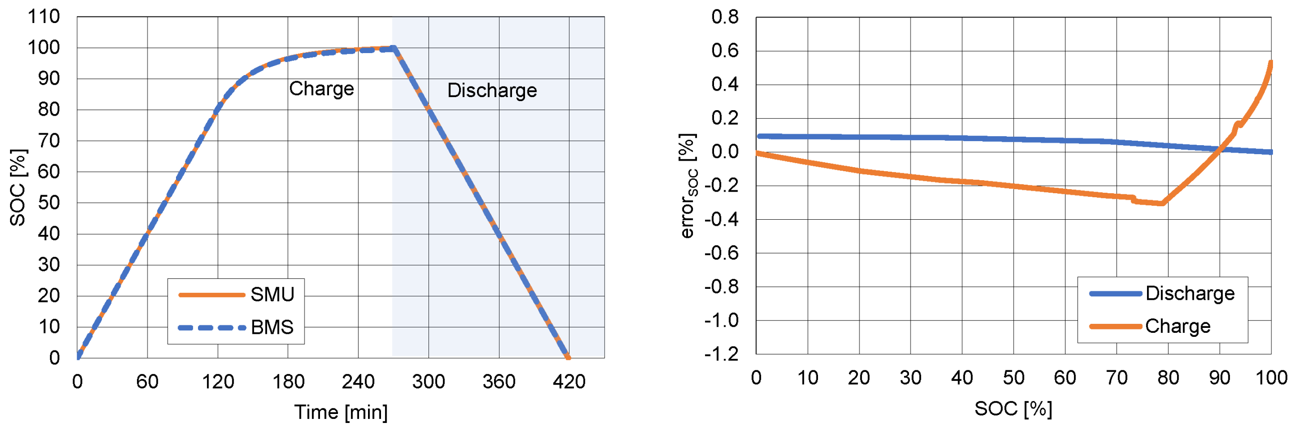


a) SOC_{BMS} and SOC_{true} during Charge and Discharge



Error of SOC Estimation with respect to SOC_{true}

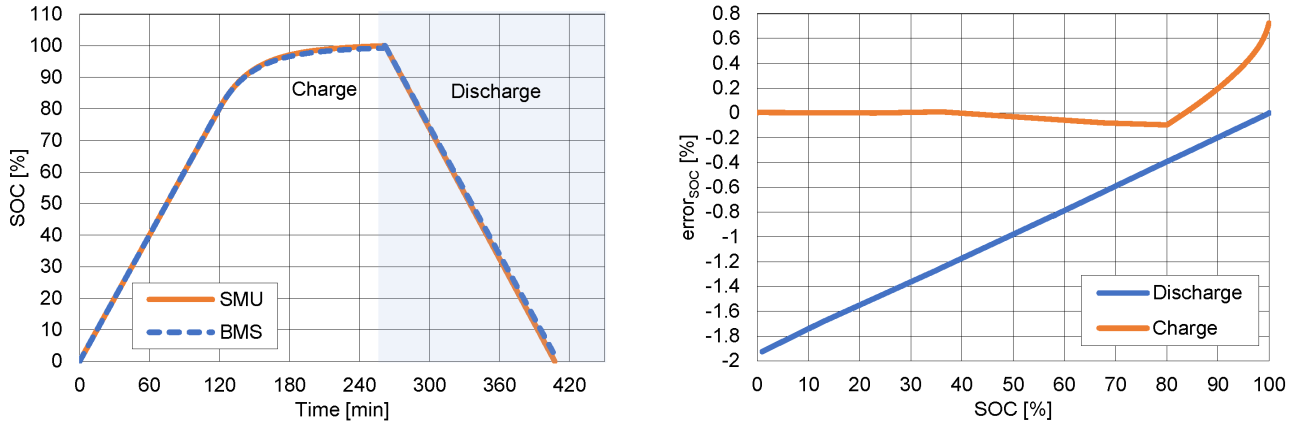
Figure 7. Cycle 1: Representative of Average SOC Estimation Error



a) SOC_{BMS} and SOC_{true} during Charge and Discharge

b) Error of SOC Estimation with respect to SOC_{true}

Figure 8. Cycle 2: Representative of Minimum SOC Estimation Error



a) SOC_{BMS} and SOC_{true} during Charge and Discharge

b) Error of SOC Estimation with respect to SOC_{true}

Figure 9. Cycle 3: Representative of Maximum SOC Estimation Error

Unlike the current integration values, the SOC estimations of each test are all within the same range (0%-100%) and already in percentage units. Therefore, normalization is not needed, and the unique indicator selected to measure SOC estimation accuracy is the Root Mean Squared Error (RMSE). Table 7 shows the RMSE calculated for the cycles representing average, minimum and maximum SOC estimation errors.

Table 7. Mean Squared Error (RMSE) of the State of Charge Estimation

Cycle	Average	Minimum	Maximum
RMSE [%]	0.521088	0.233763	0.729372

According to the RMSE indicator, the Coulomb Counting method implemented using the current measurements features of the RAA489206/ISL94216A BFE can provide SOC estimation with less than 1% of error without the need of specialized or dedicated coulomb counting ADC.

6. Conclusion

Accurate estimation of the State of Charge (SOC) of the battery pack enables high system performance and reliability, longer battery lifetime, optimized capacity, size, and density, and lower battery cost. This application note provides a detailed description of how to implement the Coulomb Counting SOC estimation method featuring the RAA489206/ISL94216A Battery Front End. This document reviews first the concepts and of SOC estimation and State of Health (SOH) correction, and then details their use and realization in the routines of the accompanying sample code.

The SOC estimation accuracy of the presented implementation is verified on a 4-cells battery pack built with commercial cells. The results show the Coulomb Counting method deployed in a BMS consisting of the RAA489206/ISL94216A BFE and RA4W1 MCU estimates the SOC with at most 1% of Root Mean Squared Error (RMSE). Such accuracy is high enough for most industrial applications. The implemented method requires neither exhaustive battery characterization nor specialized Coulomb Counting ADC. Additional characterization data and more advanced correction methodologies can be added to ensure accurate estimations during the whole lifetime of the battery and under varying temperature conditions.

7. Revision History

Revision	Date	Description
1.00	Jan 23, 2023	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks