bq27742-G1

Technical Reference Manual

Literature Number: SLUUAX0C
March 2014–Revised July 2015

Contents

List of Figures

List of Tables

Preface

This Technical Reference Manual (TRM) details the bq27742-G1 battery fuel gauge. This document is intended to complement but not supersede information in the *bq27742-G1 Single Cell Li-Ion Battery Fuel Gauge with Integrated Protection Data Sheet* (SLUSBV9).

Formatting Conventions Used in This Document

Information Type	Formatting Convention	Example
Commands	*Italics* with parentheses and no breaking spaces	*RemainingCapacity()* command
Data Flash	*Italics*, **bold**, and breaking spaces	***Design Capacity*** data
Register bits and flags	Brackets and *italics*	*[TDA]* bit
Data Flash bits	Brackets, *italics*, and **bold**	***[LED1]*** bit
Modes and states	ALL CAPITALS	UNSEALED mode

Related Documentation from Texas Instruments

Go to the TI Web site at www.ti.com for updated related documentation, including:

1. *bq27742-G1 Single Cell Li-Ion Battery Fuel Gauge with Integrated Protection Data Sheet* (SLUSBV9)
2. *Theory and Implementation of Impedance Track Battery Fuel-Gauging Algorithm in bq2750x Family Application Report* (SLUA450)
3. *How to Generate Golden Image for Single-Cell Impedance Track Devices Application Report* (SLUA544)
4. *bq27742EVM Single Cell Impedance Track Technology Evaluation Module User's Guide* (SLUUAH1)

Trademarks

Impedance Track is a trademark of Texas Instruments. All other trademarks are the property of their respective owners.

General Description

The bq27742-G1 fuel gauge accurately predicts the battery capacity and other operational characteristics of a single Li-based rechargeable cell. It can be interrogated by a system processor to provide cell information, such as state-of-charge (SOC) and time-to-empty (TTE).

Information is accessed through a series of commands, called *Standard Commands*. Further capabilities are provided by the additional *Extended Commands* set. Both sets of commands, indicated by the general format *Command()*, read and write information contained within the control and status registers, as well as its data flash locations. Commands are sent from system to fuel gauge using the serial communications engine, and can be executed during application development, pack manufacture, or end-equipment operation.

Cell information is stored in the non-volatile flash memory. Many of these data flash locations are accessible during application development. They cannot, generally, be accessed directly during end-equipment operation. Access to these locations is achieved by either use of the companion evaluation software, through individual commands, or through a sequence of data-flash-access commands. To access a desired data flash location, the correct data flash subclass and offset must be known.

The fuel gauge provides 64 bytes of user-programmable data flash memory, partitioned into two 32-byte blocks: ***Manufacturer Info Block A*** and ***Manufacturer Info Block B***. This data space is accessed through a data flash interface. For specifics on accessing the data flash, see Section 5.1.2, *Manufacturer Information Blocks*.

The key to the high-accuracy fuel gauging prediction is Texas Instruments proprietary Impedance Track™ algorithm. This algorithm uses cell measurements, characteristics, and properties to create state-of-charge predictions that can achieve less than 1% error across a wide variety of operating conditions and over the lifetime of the battery.

The fuel gauge measures charge and discharge activity by monitoring the voltage across a small-value series sense resistor (5 mΩ to 20 mΩ, typical) located between the CELL+ and PACK+ terminals of the battery pack. When a cell is attached to the fuel gauge and Impedance Track™ is enabled, cell impedance is calculated based on the selected load profile, open-circuit voltage (OCV) at present depth-of-discharge (DOD), and measured cell voltage under load. In addition, the maximum chemical capacity (Qmax) of the cell is updated after the fuel gauge records two qualified OCV measurements (taken when the battery pack is in a well relaxed state) and the accumulated charge between them is large enough. Update of these parameters allows the fuel gauge to maintain accurate capacity prediction over the life of the battery despite increasing impedance and chemical capacity loss due to Li-Ion aging effects.

External temperature sensing is supported with the use of a high-accuracy negative temperature coefficient (NTC) thermistor with R25 = 10 kΩ ± 1% and B25/85 = 3435 kΩ ± 1% (for example, Semitec 103AT-2) for measurement. The fuel gauge can also be configured to use its internal temperature sensor. The fuel gauge monitors cell temperature in order to accurately compensate open-circuit voltage and resistance values used in remaining capacity simulations as well as provide overtemperature protection for the cell and temperature-dependent charging parameter (for example, JEITA charging profile) reporting to the host system.

A full suite of configurable Li-Ion protections are available with hardware-based overvoltage (OVP), undervoltage (UVP), overcurrent in charge (OCC), overcurrent in discharge (OCD), and short-circuit in discharge (SCD) and firmware-based overvoltage, undervoltage, overtemperature in charge (OTC), overtemperature in discharge (OTD), internal short detection (ISD), tab disconnect detection (TDD), and FET body diode protection. The protection can be configured so as to support a two-level safety topology, with high-precision firmware-based OVP, UVP, OTC, OTD, ISD, and TDD offering first level safety condition checking and autonomous, fast-responding hardware-based OVP, UVP, OCC, OCD, and SCD providing exceedingly robust second level safety protection.

To minimize power consumption, the fuel gauge has three different power modes: NORMAL, SLEEP, and FULLSLEEP. The fuel gauge passes automatically between these modes, depending upon the occurrence of specific events, though a system processor can initiate entry into some of these modes directly. More details can be found in Section 2.7, *Power Modes*.

Functional Description

2.1 Fuel Gauging

The bq27742-G1 fuel gauge measures the cell voltage, current, and temperature to determine battery SOC based on outputs from the Impedance Track™ algorithm (see the *Theory and Implementation of Impedance Track Battery Fuel-Gauging Algorithm Application Report* [SLUA450] for more information). The fuel gauge monitors charge and discharge activity by sensing the voltage across a small-value resistor (5 mΩ to 20 mΩ, typical) between the SRN and SRP pins and in series with the cell. By integrating charge passing through the battery, the SOC is accurately adjusted during charge or discharge operation.

The total chemical capacity of the battery (Qmax) is found by comparing states of charge before and after applying a discharge that results in charge passed of at least 37% of ***Design Capacity***. The initial ***Qmax Cell 0*** and ***Design Capacity*** are set based on values from the applicable cell manufacturers' data sheet multiplied by the number of parallel cells. When a system load is applied, the impedance of the cell is calculated from the OCV, the measured voltage under load, and the discharge current or power as configured in ***Load Select***. The ***Load Select*** parameter can be set to various options such as average discharge current from the last cycle (***Avg I Last Run***), present average discharge current, 14-s-filtered *AverageCurrent()*, ***Design Capacity***/5 discharge rate, a user-defined discharge rate written to *AtRate()*, or the value programmed in ***User Rate-mA***. The ***Load Mode*** parameter extends this further by supporting the same options with respect to power instead of current.

During battery discharge, the fuel gauge simulates an iterative discharge based on the selected load profile and sums the resulting charge it takes to reach ***Terminate Voltage***, the charge passed since the last OCV measurement, and the starting charge passed that is required to reach the last OCV point in order to determine *FullChargeCapacity()*, which is the total capacity that can be extracted from a fully charged battery based on the present load profile and temperature. The predicted capacity left in the battery, or *RemainingCapacity()*, is simply the summed charge from present SOC to ***Terminate Voltage*** as determined by the simulation result.The fuel gauge uses the battery impedance and OCV profiles, Qmax, and present SOC to achieve this accurate prediction.

During battery charge, the fuel gauge coulomb counts up from where the last discharge ended and stores the battery voltage once charge termination is detected (***V at Chg Term***) in order to compute the depth-of-discharge (DOD) at end of charge (DOD@EOC). This provides a more accurate 100% SOC reference point for deriving starting charge passed (Qstart) since most systems do not charge the battery to absolutely full.

In addition to *RemainingCapacity()* and *FullChargeCapacity()*, the fuel gauge also reports uncompensated (that is, <C/20) versions of capacity in *NominalAvailableCapacity()* and *FullAvailableCapacity()*, respectively. At relaxation entry (|*AverageCurrent()*| < ***Quit Current*** for ***Chg Relax Time*** or ***Dsg Relax Time***, depending on previous state), the fuel gauge waits 60 seconds before beginning to take OCV measurements in order to check that the battery is in a well-relaxed state (dV/dt < 1 uV/s) and to update the reported SOC every hour (although accumulated charge can result in change to SOC between these updates). Once the dV/dt prerequisite is met, additional OCV measurements are taken every 100 seconds thereafter and can be qualified for use in a Qmax update if additional criteria are fulfilled. More specifically, measurements must have been taken within the supported temperature range between 10°C and 40°C and outside of the flat region of the battery voltage curve (varies based on chemistry). If two DOD points (that is, DOD0 and DOD1) are available from two separate relaxation cycles and the accumulated coulomb count between them is at least 37% of ***Design Capacity***, then a Qmax update is performed to update the battery chemical capacity. If the dV/dt condition is not met and the fuel gauge continues to reside in RELAXATION mode, then a forced OCV measurement and subsequent DOD computation occurs after 5 hours have elapsed; however, a Qmax update is still subject to the temperature, voltage, and minimum passed charge requirements before an update can occur. If *AverageCurrent()* > ***Deadband*** is detected during the OCV measurement, then IR correction is used to compensate the value prior to using it to compute a new DOD. The value programmed in ***Max IR Correct*** determines the maximum allowed correction voltage based on detected charge current. If discharge current is detected instead, then no cap is applied.

2.1.1 Fast Qmax Update

In certain applications, it can be especially difficult to achieve traditional Qmax updates for systems with noisy standby currents that prohibit the dV/dt requirement from being met and whose total time in relax never reaches 5 hours, or for systems that rarely enter relaxation at all. The Fast Qmax feature provides a complement to traditional Qmax updates in order to more reliably account for aging effects in such systems by allowing Qmax updates to achieved with only one or no qualified DOD points from a battery relaxation state. The feature is enabled via the ***Pack Configuration C [FastQmax]*** bit and uses several conditional checks to determine when fast Qmax-specific DOD point collection is allowed to start and when these are qualified for use in a fast Qmax update.

NOTE: The Fast Qmax Update algorithm is not used during a learning cycle if ***Update Status*** ≠ 2.

The algorithm begins taking new discharge-based fast Qmax DOD points every 30 seconds once the following conditions are detected:

- DOD > ***Fast Qmax Start DOD%*** or Voltage < ***Terminate Voltage*** + ***Fast Qm Start V Delta***, and
- Current ≤ C/***Fast Qmax Current Threshold***

The algorithm qualifies and saves the discharge-based fast Qmax DOD point when the following conditions are met:

- DOD > ***Fast Qmax End DOD%*** or Voltage ≤ ***Terminate Voltage*** + ***Fast Qmax Volt Buffer***, and
- Number of Fast Qmax measurements ≥ ***Fast Qmax Min Points***

The algorithm also qualifies and saves a charge-based fast Qmax DOD point when full charge termination is achieved (that is, *Flags()[FC]* is set).

NOTE: The ***Pack Configuration C [FastQmax]*** bit only controls enable/disable of DOD point collection in the near end of discharge region. DOD points at end of charge are always recorded for fast Qmax purposes.

As a result, it is possible to get fast Qmax updates with DOD points from relaxation, end of discharge, and end of charge. However, a fast Qmax update will not happen immediately upon end of charge, and will only be enforced if an attempt to record a new OCV has failed during battery relaxation following a full charge event. As with traditional Qmax updates, a fast Qmax update is disqualified if the intended DOD points were captured outside of the allowed 10°C to 40°C temperature range or if the passed charge accumulated between them is less than 37% × ***Design Capacity***.

2.1.2 Resistance Update Qualification

Resistance measurements are conducted over the course of a discharge cycle but are only qualified for an update if the applied discharge current exceeds a ***Design Capacity***/10 rate. An additional qualifier allows resistance updates to be permitted on the basis of IR drop between cell open-circuit voltage and measured voltage under load. The IR drop is configured in ***Res V Drop*** and allows higher potential for resistance updates in applications with low-rate discharges.

2.1.3 Fast Resistance Scaling

Fast Resistance Scaling provides improved SOC convergence to 0% (that is, ***Terminate Voltage***) by scaling resistance values used in capacity prediction simulations instead of using interpolated resistance table values as-is. The algorithm becomes active once *StateofCharge()* ≤ ***Fast Scale Start SOC*** or *Voltage()* < (***Terminate Voltage*** + ***Term V Delta***) and begins scaling resistance values every 30 seconds based on the ratio of most recent measured resistance (R_{new}) to stored resistance (R_{old}) at the present SOC. This allows the predicted remaining capacity to gradually converge to the cell's empty point and avoids potential for SOC jumps to empty near the end of discharge. The minimum and maximum scaling factors that can be employed in the Fast Resistance Scaling algorithm are stored in ***Min Res Scale*** and ***Max Res Scale***, respectively, where a value of 1000 corresponds to 1x and 200 corresponds to 0.2x. For most applications, the default value of ***Term V Delta*** and ***Fast Scale Start SOC*** are recommended. Further, it is typically best to keep (***Terminate Voltage*** + ***Term V Delta***) below 3.6 V for most battery applications. The feature itself is enabled via the ***[FConvEn]*** bit in ***Pack Configuration B***.

2.1.3.1 Load Select in Fast Resistance Scaling Mode

An independently configurable load profile selection for fast resistance scaling is supported in Fast Scaling Load Select and can be used to improve convergence to empty for systems that exhibit significant load changes near the end of discharge. If typical load behavior is consistent throughout the entirety of the battery discharge curve, then the feature can be disabled by setting ***Fast Scaling Load Select*** to the same value as ***Load Select***.

2.1.3.2 Thermal Modeling in Fast Resistance Scaling Mode

Thermal modeling is designed to estimate cell heating based on current flow through the battery and compensates the predicted *RemainingCapacity()* based on the true cell temperature output from the model. However, it is possible to overestimate cell self-heating in particular cases and this could result in overestimation of *RemainingCapacity().* As a result, thermal modeling is disabled by default in the Fast Resistance Scaling region but can be enabled for a given application using the ***Pack Configuration C [FConvTempEn]*** configuration bit.

2.1.4 StateOfCharge() Smoothing

2.1.4.1 SOC Smoothing in Charge/Discharge

It is common for sudden changes in operating conditions such as temperature and discharge load to cause drastic but legitimate changes in the amount of capacity that can be extracted from a given Li-Ion cell or pack. These changes are typically perceived as jumps in reported SOC and can sometimes be alarming to end equipment users who may not understand how environmental conditions impact available battery capacity. SOC smoothing solves this by gradually equalizing the difference in reported SOC vs "true" SOC over the present cycle. The method for accomplishing this differs depending on whether or not the current cycle is a charge or discharge. During discharge, the algorithm adds or removes delta charge (deltaQ) from the present coulomb count to accelerate or decelerate change in *RemainingCapacity()* until it is able to converge to the true value by the time the battery reaches empty. During charge, *FilteredFCC()* is modified to account for deltas in the true and reported versions of Remaining Capacity as well as the true and reported versions of Full Charge Capacity. Since *FilteredFCC()* is continuously modified to ensure SOC convergence in charge, it is not a real determinant of the total available battery capacity and *UnfilteredFCC()* should instead be referred to for this purpose. The ***[SmoothEn]*** bit in ***Pack Configuration C*** determines whether unfiltered or filtered values are mapped to *RemainingCapacity()*, *FullChargeCapacity()*, and *StateofCharge()* for reporting to the system host, as shown in the table below.

Pack Configuration C [SmoothEn]	*RemainingCapacity()*	*FullChargeCapacity()*	*StateOfCharge()*
0	*UnfilteredRM()*	*UnfilteredFCC()*	*UnfilteredRM()/UnfilteredFCC()*
1	*FilteredRM()*	*FilteredFCC()*	*FilteredRM()/FilteredFCC()*

2.1.4.2 SOC Smoothing in Relaxation

In relaxation state, temperature changes and applied currents below ***Quit Current*** can still trigger changes in the reported *StateofCharge()*. Similarly, other scenarios can cause differences in true vs reported *RemainingCapacity()* and *FullChargeCapacity()* when entering a battery relaxation state. In order to enable convergence between true and reported SOC values, a similar smoothing algorithm is also supported during cell relaxation and can be enabled with the ***[RlxSmEn]*** bit in ***Pack Configuration C***. The feature can be configured to equalize the SOC difference over time or instantly, depending on the configured setting in the ***Pack Configuration D [SMSYNEN]*** bit (1 = instant convergence, 0 = convergence over time). ***[RlxSmEn]*** must be set to 1 for instant convergence to be selectable/supported.

2.1.4.3 SOC Smoothing in Overcharged and Overdischarged Scenarios

In cases where the cell is in an overcharged or overdischarged state, *FilteredRM()* will begin decrementing or incrementing immediately when a load or charger is applied, respectively. However, this behavior can be overridden to hold *FilteredRM()* at full charge (100%) or empty (0%) until the charge surplus (overcharged state) or charge deficit (overdischarged state) is equalized, at which point it is then allowed to change. The hold at full in overcharged state and hold at empty in overdischarged state options can be enabled via the ***Pack Configuration [SOCHoldOvrChg]*** and ***Pack Configuration [SOCHoldOvrDsg]*** bits, respectively.

2.1.5 StateofCharge() Hold at 99%

The *StateofCharge()* hold at 99% feature prohibits the fuel gauge from reporting 100% until full charge termination is detected and the *Flags()[FC]* bit is set. It is enabled using the ***[SOCHold99]*** bit in Pack Configuration.

2.1.6 StateofCharge() Hold at 1%

The *StateofCharge()* hold at 1% feature prohibits the fuel gauge from reporting 0% until ***Terminate Voltage*** is reached. It is enabled using the ***[SOCHold1]*** bit in Pack Configuration.

2.1.7 Trace and Downstream Resistance Compensation

Prediction accuracy for remaining capacity simulations can be further improved in systems with excessive trace lengths between cell and fuel gauge or fuel gauge and system point of load by setting a nominal value in ***Trace Resistance*** or ***Downstream Resistance***, respectively. The fuel gauge adds the ***Trace Resistance*** and ***Downstream Resistance*** to the cell resistance values used in capacity simulations in order to obtain a more realistic voltage drop under load in simulated discharges when faced with non-trivial trace parasitics in a given pack design. Likewise, trace resistance is removed from any resistance measurements made during discharge prior to storing in data flash.

2.1.8 Imax Calculation

The Imax feature allows a system to determine how much load it can apply for ***Max Current Pulse Duration*** without causing an instant drop to ***Terminate Voltage***. It is extremely useful for systems that require intelligent load throttling at various points of operation. The fuel gauge computes the *Imax()* current based on the programmed ***Max Allowed Current*** and ***Max Current Pulse Duration*** when ***Pack Configuration D [IMAXEN]*** = 1 and triggers an interrupt on the RC2 pin if *Imax()* changes by more than ***Max Current Interrupt Step***. ***Reserve Capacity*** is factored into the *Imax()* calculation if ***Pack Configuration D [IMAXRESRVEN]*** = 1.

2.2 Li-Ion Protection

Li-Ion protection is supported in the form of several firmware and hardware-based safety features to provide an optimal balance between accuracy and flexibility as well as allow for a tiered, two-level protection scheme for increased reliability. Firmware-based safety checking includes detection of overvoltage (OVP), undervoltage (UVP), overtemperature in charge (OTC), overtemperature in discharge (OTD), internal short (ISD), and tab disconnect (TDD) conditions. Body diode protection is also supported in firmware to provide additional monitoring of current when other protections are active. Hardware-based protection includes detection of overvoltage, undervoltage, overcurrent in charge (OCC), overcurrent in discharge (OCD), and short-circuit current in discharge (SCD) conditions. The fuel gauge monitors for each of these safety events and responds with disable of the applicable FET or setting a notification flag (or both). Additionally, some of these protections, specifically OTC, OTD, ISD, and TDD, can be configured to trigger an interrupt on RC2 if the ***Pack Configuration [HOSTIE]*** bit is set and the BTP feature is not enabled (***Pack Configuration C [BTP_EN]*** = 0).

2.2.1 Firmware Protection

2.2.1.1 Overvoltage

Overvoltage protection opens the CHG FET and sets the *SafetyStatus()[OVP]* flag when measured *Voltage()* ≥ ***OV Prot Threshold*** for ***OV Prot Delay***. Recovery is achieved, protection flag cleared, and the CHG FET closed when *Voltage()* ≤ ***OV Prot Recovery***.

2.2.1.2 Undervoltage

Undervoltage protection opens the DSG FET and sets the *SafetyStatus()[UVP]* flag when measured *Voltage()* ≤ ***UV Prot Threshold*** for ***UV Prot Delay***. Recovery is achieved, protection flag cleared, and the DSG FET closed when *Voltage()* ≥ ***UV Prot Recovery***.

2.2.1.3 Overtemperature in Charge

Overtemperature in charge protection sets the *SafetyStatus()[OTC]* flag when measured *Temperature()* ≥ ***OT Chg*** for ***OT Chg Time***. If the ***Pack Configuration D [OT_FET]*** bit is set, then the CHG FET will also be opened when the condition is detected. Recovery is achieved, protection flag cleared, and the CHG FET closed when *Temperature()* ≤ ***OT Chg Recovery***. An interrupt can be configured to trigger on the RC2 pin when an overtemperature in charge condition is detected. The feature can be completely disabled by setting ***OT Chg Time*** = 0.

2.2.1.4 Overtemperature in Discharge

Overtemperature in discharge protection sets the *SafetyStatus()[OTD]* flag when measured *Temperature()* ≥ ***OT Dsg*** for ***OT Dsg Time***. If the ***Pack Configuration D [OT_FET]*** bit is set, then the DSG FET will also be opened when the condition is detected. Recovery is achieved, protection flag cleared, and the DSG FET closed when *Temperature()* ≤ ***OT Dsg Recovery***. An interrupt can be configured to trigger on the RC2 pin when an overtemperature in discharge condition is detected. The feature can be completely disabled by setting ***OT Dsg Time*** = 0.

2.2.1.5 Internal Short Detection

An internal short condition is detected and the *SafetyStatus()[ISD]* flag set when *SelfDischargeCurrent()* ≤ –***DesignCapacity***/***ISD Current Threshold***. The *SelfDischargeCurrent()* is computed based on two open-circuit voltage reads separated by a 1-hour interval. from there, the charge passed and time elapsed between the two OCV points is used to compute the amount of *SelfDischargeCurrent()*. An interrupt can be configured to trigger on the RC2 pin when an internal short condition is detected. Enable/disable of the internal short detection feature is controlled via the ***[SE_ISD]*** bit in ***Pack Configuration B***.

2.2.1.6 Tab Disconnect Detection

A tab disconnect condition is detected and the *SafetyStatus()[TDD]* flag set when the ratio of current *StateofHealth()* to previous *StateofHealth()* is less than ***TDD SOH percent***. An interrupt can be configured to trigger on the RC2 pin when a tab disconnect condition is detected. Enable/disable of the tab disconnect detection feature is controlled via the ***[SE_TDD]*** bit in ***Pack Configuration B***.

2.2.1.7 Body Diode Protection

Body diode protection temporarily re-enables the applicable FET while |*AverageCurrent()*| ≥ ***Body Diode Threshold*** and either of the FETs is disabled based on firmware instruction. The body diode protection control of the FET is removed once *AverageCurrent()* < ***Body Diode Threshold*** with the precipitating safety condition still present or if the safety condition disappears altogether.

2.2.2 Hardware Protection

2.2.2.1 Protector Configuration

Configuration of the hardware protection is performed by selecting the desired fault detection thresholds for OVP in ***Prot OV Config*** and OCC, OCD, and SCD in ***Prot OC Config***. All available options are detailed in Table 2-1, *Hardware Overvoltage Protection Configuration*, Table 2-2, *Hardware Overcurrent in Charge Protection Options*, Table 2-3, *Hardware Overcurrent in Discharge Protection Options*, and Table 2-4, *Hardware Short-Circuit in Discharge Protection Options*, respectively. In order for new protection settings to take effect, the ***Prot Checksum*** must be updated based on the sum of ***Prot OV Config*** and ***Prot OC Config***. Without proper update to the protector checksum, the CHG and DSG FETs will be held open to ensure an accidental protector mis-configuration does not result in false tripping (if set too low) or failure to trip (if set too high). The fuel gauge will indicate an invalid protector checksum by setting the *SafetyStatus()[INVPROTCHK]* flag. The checksum value can be automatically computed by the fuel gauge when the host sends the *PROTECTOR_CHKSUM* command, which returns the computed checksum with the MSB set to 1 if a mismatch was detected.

Table 2-1. Hardware Overvoltage Protection Configuration

Prot OV Config			OVERVOLTAGE (V_{OVP}) SETTING
[OVP2]	*[OVP1]*	*[OVP0]*	
0	0	0	4.275 V
0	0	1	4.300 V
0	1	0	4.325 V
0	1	1	4.350 V
1	0	0	4.375 V
1	0	1	4.400 V
1	1	0	4.425 V
1	1	1	4.450 V (default)

Table 2-2. Hardware Overcurrent in Charge Configuration

Prot OC Config		OVERCURRENT IN CHARGE (V_{OCC}) SETTING
[OCC1]	*[OCC0]*	
0	0	6 mV
0	1	13 mV
1	0	18 mV (default)
1	1	28 mV

Table 2-3. Hardware Overcurrent in Discharge Configuration

Prot OC Config			OVERCURRENT IN DISCHARGE (V_{OCD}) SETTING
[OCD1]	*[OCD0]*	*[OCD0]*	
0	0	0	14 mV
0	0	1	24 mV
0	1	0	34 mV (default)
0	1	1	44 mV
1	0	0	53 mV
1	0	1	63 mV
1	1	0	73 mV
1	1	1	83 mV

Table 2-4. Hardware Short-Circuit in Discharge Configuration

Prot OC Config [SCD]	SHORT-CIRCUIT IN DISCHARGE (V_{SCD}) SETTING
0	73 mV (default)
1	148 mV

2.2.2.2 Protector Operating Modes

2.2.2.2.1 Operating Modes

The battery protector has several operating modes depending on a variety of conditions. The various modes are described below:

- NORMAL Mode
- OVERVOLTAGE Mode
- UNDERVOLTAGE Mode
- OVERCURRENT IN DISCHARGE and SHORT-CIRCUIT IN DISCHARGE Mode
- OVERCURRENT IN CHARGE Mode
- SHUTDOWN WAIT Mode
  - ANALOG SHUTDOWN State
  - LOW VOLTAGE CHARGING State

The relationships among these modes are shown in Figure 2-1.

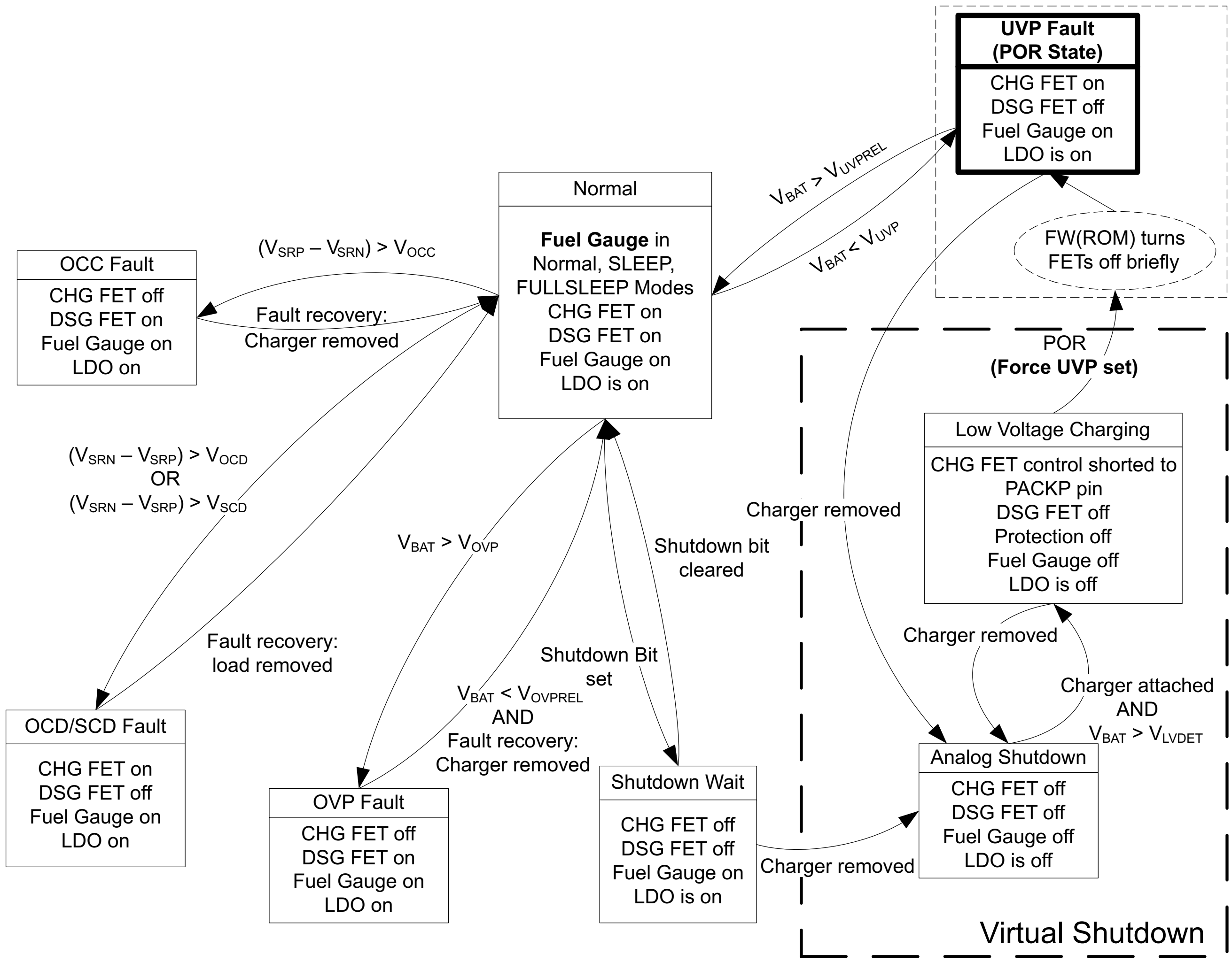


Figure 2-1. Operating Modes

2.2.2.2.1.1 NORMAL Mode

In this mode, the protector is fully powered and operational. Both CHG and DSG FETs are closed and the protector continuously checks for fault conditions.

FET enable override capability is available to the fuel gauge to force the CHG or DSG FET open based on firmware instruction. It is useful for firmware-defined safety features or other special functionality that requires one or both FETs to be opened based on specific conditions. It cannot, however, be used to enforce FET turn-on when the hardware protector is one of the protection fault modes, as the latter has ultimate authority over the FET drive control circuitry.

Firmware can also command the fuel gauge to go into SHUTDOWN mode based on a dedicated *Control()* subcommand from the host. In this case, firmware sets the Shutdown bit to indicate intent to go into SHUTDOWN mode. The fuel gauge then transitions to SHUTDOWN WAIT mode and waits for charger removal prior to disabling the internal LDO and fully powering down the entire device.

2.2.2.2.1.2 OVERVOLTAGE Mode

In this mode, an OVERVOLTAGE PROTECTION (OVP) FAULT mode is entered when the voltage on the VBAT pin continuously exceeds the V_{OVP} threshold for longer than the t_{OVP} delay time. At this point, the fuel gauge enables the fault recovery detection circuitry, which monitors the PACKP pin for charger removal. The OVP fault is cleared once the pack voltage drops below the cell voltage by more than 300 mV and the cell voltage drops below V_{OVPREL}, which causes the fuel gauge to transition to NORMAL mode.

2.2.2.2.1.3 UNDERVOLTAGE Mode

In this mode, an UNDERVOLTAGE PROTECTION (UVP) FAULT mode is entered when the voltage on the VBAT pin continuously falls below the V_{UVP} threshold for longer than the t_{UVP} delay time. The fuel gauge then enables the charger attach detection circuitry and, if no charger is found, sends the fuel gauge into ANALOG SHUTDOWN mode to minimize power consumption and avoid further discharge of the battery. The UVP fault is cleared once charger attach is detected and the cell voltage rises above V_{UVPREL}, which causes the fuel gauge to transition to NORMAL mode.

The fuel gauge can enter this mode from LOW VOLTAGE CHARGING mode when the battery pack is being charged from a deeply discharged state or from NORMAL mode when the battery pack is being discharged past the UVP threshold.

2.2.2.2.1.4 OVERCURRENT CHARGE Mode

In this mode, an OVERCURRENT IN CHARGE (OCC) FAULT mode is entered when the voltage across the sense resistor continuously exceeds the V_{OCC} threshold for longer than the configured t_{OCC} delay time. Recovery occurs when the PACKP voltage drops to more than 300 mV below the cell voltage, indicating charger removal.

2.2.2.2.1.4.1 OVERCURRENT DISCHARGE and SHORT-CIRCUIT DISCHARGE Mode

In this mode, a short-circuit discharge (SCD) or overcurrent discharge (OCD) protection fault is detected when the voltage across the sense resistor continuously exceeds the V_{OCD} or V_{SCD} thresholds for longer than the t_{OCD} or t_{SCD}delay times. Recovery occurs when the PACKP voltage rises to within 300 mV of the cell voltage, indicating load removal.

2.2.2.2.1.5 SHUTDOWN WAIT Mode

Transition to this mode occurs when the host sends the SET_SHUTDOWN command and the fuel gauge subsequently initiated the shutdown sequence.

The shutdown sequence is as follows:

1. Open both CHG and DSG FETs.
2. Determine if any faults are set. If any faults are set, then go back to NORMAL mode.
3. Wait for charger removal. Once the charger is removed, turn off the LDO, which puts the fuel gauge into ANALOG SHUTDOWN mode.

2.2.2.2.1.5.1 Analog Shutdown State

In this mode, the fuel gauge is completely powered down and no portions of the device are functional. Once the charger is connected, the fuel gauge will transition into either LOW VOLTAGE CHARGING mode (if below the power-on reset voltage) or NORMAL mode (if above the POR voltage and no faults are detected).

2.2.2.2.1.6 Low Voltage Charging State

In this mode, the fuel gauge shorts the CHG FET gate to PACKP pin if the cell voltage is above the V_{LVDET} threshold, allowing the battery to be trickle charged with the CHG FET biased in the ohmic region. If below the aforementioned threshold, low voltage charging is prohibited for safety reasons and the cell will likely be permanently unrecoverable due to being dangerously depleted.

2.3 Lifetime Data Logging Parameters

The Lifetime Data logging function helps development and diagnosis with the fuel gauge. The *IT_ENABLE* subcommand needs to be enabled (command 0x0021) for lifetime data logging functions to be active. The fuel gauge logs the lifetime data as specified in the ***Lifetime Data*** and ***Lifetime Temp Samples*** data flash subclasses. The data log recordings are controlled by the ***Lifetime Resolution*** data flash subclass.

The Lifetime Data Logging can be started by setting the *IT_ENABLE* subcommand and setting the Update Time register to a non-zero value.

Once the Lifetime Data Logging function is enabled, the measured values are compared to what is already stored in the data flash. If the measured value is higher than the maximum or lower than the minimum value stored in the data flash by more than the Resolution set for at least one parameter, the entire Data Flash Lifetime Registers are updated after at least ***LTUpdateTime***.

LTUpdateTime sets the minimum update time between DF writes. When a new maximum or minimum is detected, a LT Update window of [update time] seconds is enabled and the DF writes occur at the end of this window. Any additional maximum or minimum value detected within this window is also updated. The first new maximum or minimum value detected after this window triggers the next LT Update window.

Internal to the fuel gauge, there exists a RAM maximum/minimum table in addition to the DF maximum/minimum table. The RAM table is updated independent of the resolution parameters. The DF table is updated only if at least one of the RAM parameters exceeds the DF value by more than resolution associated with it. When DF is updated, the entire RAM table is written to DF. Consequently, it is possible to see a new maximum/minimum value for a certain parameter even if the value of this parameter never exceeds the maximum or minimum value stored in the data flash for this parameter value by the resolution amount.

The Life Time Data Logging of one or more parameters can be reset or restarted by writing new default (or starting) values to the corresponding data flash registers through SEALED or UNSEALED access as described below. However, when using UNSEALED access, new values take effect only if the device is reset within ***LT Update Time*** after the DF is loaded with new values.

The logged data in ***Lifetime Data*** subclass (subclass ID = 59) can be read and written in both SEALED and UNSEALED modes. However, in SEALED mode, access to this subclass is using a process identical to accessing ***Manufacturer Info Block B***. The *DataFlashBlock()* command code is 4. See Section 5.1.2, *Manufacturer Information Blocks*, for details of this sequence.

The subclasses ***Lifetime Resolution*** (subclass ID = 66) and ***Lifetime Temp Samples*** (subclass ID = 60) that contain settings for lifetime data logging can be configured only in UNSEALED mode using the regular DF access method.

The Lifetime resolution registers contain the parameters which set the limits related to how much a data parameter must exceed the previously logged maximum/minimum value to be updated in the lifetime log. For example, V must exceed MaxV by more than Voltage Resolution to update MaxV in the data flash.

2.4 System Control Function

The fuel gauge provides system control functions which allow the fuel gauge to enter SHUTDOWN mode for minimal power consumption or provide interrupt notifications to the host system for low capacity, overtemperature, high battery voltage, low battery voltage, internal short, or tab disconnect conditions.

2.4.1 SHUTDOWN Mode

In this mode, the gauge and protector are completely shutdown. The device automatically enters SHUTDOWN mode when the following conditions are satisfied:

- |*Current()*| < SleepCurrent
- *Voltage()* < ShutdownVoltage
- Device is in SLEEP or FULLSLEEP mode.

The device can also be instructed to enter SHUTDOWN mode by sending the *SET_SHUTDOWN* command. Once received, the fuel gauge sets the *CONTROL_STATUS[SHUTDWN]* bit and initiates the shutdown sequence, which opens both the charge and discharge FETs and waits for charger removal. Once the charger is removed, the fuel gauge will completely power down. The only way to recover from SHUTDOWN mode is to connect a charger. If the *CLEAR_SHUTDOWN* command is sent during the wait for charger disconnection phase, the fuel gauge stops the SHUTDOWN sequence and clears the *[SHUTDWN]* bit.

2.4.2 Host Interrupts

The fuel gauge has the capability to interrupt the host system when one or more of a predefined set of events occurs in the gauge. Level-based interrupts are asserted on the RC2 pin when ***Pack Configuration [HOSTIE]*** is set and at least one of the applicable events is detected. The interrupt polarity is also selectable in ***Pack Configuration [HOSTIPol]*** and an internal 1.8-V pullup can be enabled via ***Pack Configuration [HOSTIPU]***. All available interrupt pin configuration options are displayed in Table 2-5, *RC2 Interrupt Configuration*.

Table 2-5. RC2 Interrupt Configuration

Pack Configuration			**RC2 Interrupt Configuration**
[HOSTIEN]	**[HOSTIPU]**	**[HOSTIPol]**	
0	0	X	Interrupts disabled and no 1.8-V pullup
0	1	X	Interrupts disabled and internal 1.8-V pullup active
1	0	0	Active-low interrupts and no internal 1.8-V pullup
1	0	1	Active-high interrupts and no internal 1.8-V pullup
1	1	0	Active-low interrupts and internal 1.8-V pullup active
1	1	1	Active-high interrupts and internal 1.8-V pullup active

Several conditions can result in an interrupt assertion to the host system. These range from capacity and battery voltage-related conditions to more serious safety conditions. The full list of available interrupt trigger events is detailed in Table 2-6, *Interrupt Trigger Events*.

Table 2-6. Interrupt Trigger Events

Interrupt Condition	Flags() Status Bit	Enable Condition	Comment
SOC1 Set	*[SOC1]*	Always	This interrupt is raised when the *[SOC1]* flag is set.
Battery High	*[BATHI]*	Always	This interrupt is raised when the *[BATHI]* flag is set.
Battery Low	*[BATLOW]*	Always	This interrupt is raised when the *[BATLOW]* flag is set.
Over-Temperature Charge	*[OTC]*	***OT Chg Time*** ≠ 0	This interrupt is raised when the *[OTC]* flag is set.

Table 2-6. Interrupt Trigger Events (continued)

Interrupt Condition	Flags() Status Bit	Enable Condition	Comment
Over-Temperature Discharge	*[OTD]*	***OT Dsg Time*** ≠ 0	This interrupt is raised when the *[OTD]* flag is set.
Internal Short Detection	*[ISD]*	***[SE_ISD]*** = 1 in ***Pack Configuration B***	This interrupt is raised when the *[ISD]* flag is set.
Tab Disconnect Detection	*[TDD]*	***[SE_TDD]*** = 1 in ***Pack Configuration B***	This interrupt is raised when the *[TDD]* flag is set.
Imax	*[IMAX]*	***[IMAXEN]*** = 1 in ***Pack Configuration D***	This interrupt is raised when the *[IMAX]* flag is set.
Battery Trip Point (BTP)	*[SOC1]*	***[BTP_EN]*** = 1 in ***Pack Configuration C***. The BTP interrupt supersedes all other interrupt sources, which are unavailable when BTP is active.	This interrupt is raised when *RemainingCapacity()* ≤ *BTPSOC1Set()* or *RemainingCapacity()* ≥ *BTPSOC1Clear()* during battery discharge or charge, respectively. The interrupt remains asserted until new values are written to both the *BTPSOC1Set()* and *BTPSOC1Clear()* registers.

2.4.3 Low Capacity

The fuel gauge has two flags available in the *Flags()* register that warn when the SOC of the battery has fallen to critically low levels. The *Flags()[SOC1]* flag is set when *RemainingCapacity()* falls below the ***SOC1 Set Threshold*** and is cleared once *RemainingCapacity()* rises above the ***SOC1 Clear Threshold***. The *Flags()[SOCF]* flag is set when *RemainingCapacity()* falls below the ***SOCF Set Threshold*** and is cleared once *RemainingCapacity()* rises above the ***SOCF Clear Threshold***.

The *[SOC1]* flag will trigger an interrupt on RC2 if ***Pack Configuration [HOSTIE]*** = 1.

2.4.4 Battery Level

The fuel gauge can indicate when battery voltage has fallen below or risen above predefined thresholds. The *[BATHI]* of *Flags()* is set high to indicate *Voltage()* is above the ***BH Set Volt Threshold*** for a predefined duration configured in ***BH Volt Time***. This flag returns to low once battery voltage is below or equal the ***BH Clear Volt threshold***. It is recommended that the ***BH Set Volt Threshold*** is configured higher than the ***BH Clear Volt threshold*** to provide proper voltage hysteresis.

The *[BATLOW]* of *Flags()* is set high to indicate *Voltage()* is below the ***BL Set Volt Threshold*** for a predefined duration set in ***BL Volt Time***. This flag returns to low once battery voltage is above or equal the ***BL Clear Volt threshold***. It is recommended that the ***BL Set Volt Threshold*** is configured lower than the ***BL Clear Volt threshold*** to provide proper voltage hysteresis.

Both the *[BATHI]* and *[BATLOW]* flags will trigger an interrupt on RC2 if ***Pack Configuration [HOSTIE]*** = 1.

2.4.5 Safety Conditions

The fuel gauge can indicate detection of overtemperature in charge, overtemperature in discharge, internal short, and tab disconnect events. To enable overtemperature interrupts, ***OT Chg Time*** and ***OT Dsg Time*** must be configured to non-zero values and ***Pack Configuration [HOSTIE]*** should be set to 1. To enable internal short and tab disconnect interrupts, ***Pack Configuration B [SE_ISD]*** and ***Pack Configuration B [SE_TDD]*** must be set, respectively.

The *[ISD]* and *[TDD]* flags will trigger an interrupt on RC2 if ***Pack Configuration B [SE_ISD]*** or Pack Configuration B ***[SE_TDD]*** are set to 1, respectively.

2.4.6 Battery Trip Point Interrupt Function

To provide increased flexibility for capacity-based interrupts to the host, the fuel gauge incorporates a Battery Trip Point (BTP) function that allows the system to dynamically update the traditional ***SOC1 Set Threshold*** and ***SOC1 Clear Threshold*** at runtime using the *BTPSOC1Set()* and *BTPSOC1Clear()* standard commands. These thresholds are used to trigger an interrupt on the RC2 pin whenever the set or clear thresholds are crossed following update to the *BTPSOC1Set()* and *BTPSOC1Clear()* values. Configuration of the interrupt polarity and enable/disable of the feature is provided via the ***Pack Configuration [HOSTIPol]*** and ***Pack Configuration C [BTP_EN]*** bits, respectively, while initialization values for the interrupt set and clear thresholds are programmed in ***SOC1 Set Threshold*** and ***SOC1 Clear Threshold*** as normal. An internal 1.8-V pullup can be enabled on RC2 using the ***Pack Configuration [HOSTIPU]*** bit.

NOTE: Enabling of the BTP feature automatically disables all other interrupt sources on the RC2 pin, so care must be taken in configuring the fuel gauge for each particular end application, especially if non-BTP interrupts such as overtemperature, internal short detection, tab disconnect detection, and battery low and high indications are required in the system.

When BTP is enabled, the fuel gauge continuously compares *RemainingCapacity()* with the values programmed in *BTPSOC1Set()* and *BTPSOC1Clear()* to determine whether or not it has crossed below the set or above the clear threshold. Once a threshold is crossed, additional conditions are verified to guard against an unintended interrupt trigger. For the BTP set threshold, the direction of current flow is checked to confirm that a discharge event is occurring. If true, the *Flags()[SOC1]* bit is set to 1 and an interrupt asserts on the RC2 pin. For the BTP clear threshold, the device again checks the direction of current flow to ensure that a charge event is occurring. Afterwards, an internal variable is examined to determine whether or not a change in the state of *Flags()[SOC1]* has already occurred due to a prior clear threshold crossing. If true, no change is made and a new interrupt will not fire, however, it is implied that a pre-existing interrupt will still be asserted. If false, the current state of *Flags()[SOC1]* is flipped to its opposite value and an interrupt subsequently triggered on the RC2 pin. In this way, the correct behavior is guaranteed in cases where the host updates the BTP set and clear thresholds diligently based on RC2 interrupts but also when there is a failure to update the thresholds. If, at any time, new values are written to either *BTPSOC1Set()* or *BTPSOC1Clear()* then the *[SOC1]* flag automatically reinitializes to 0 and the RC2 pin de-asserts to its default state. The entire functional flow of the BTP feature is illustrated in Figure 2-2, *BTP Algorithm Flow*.

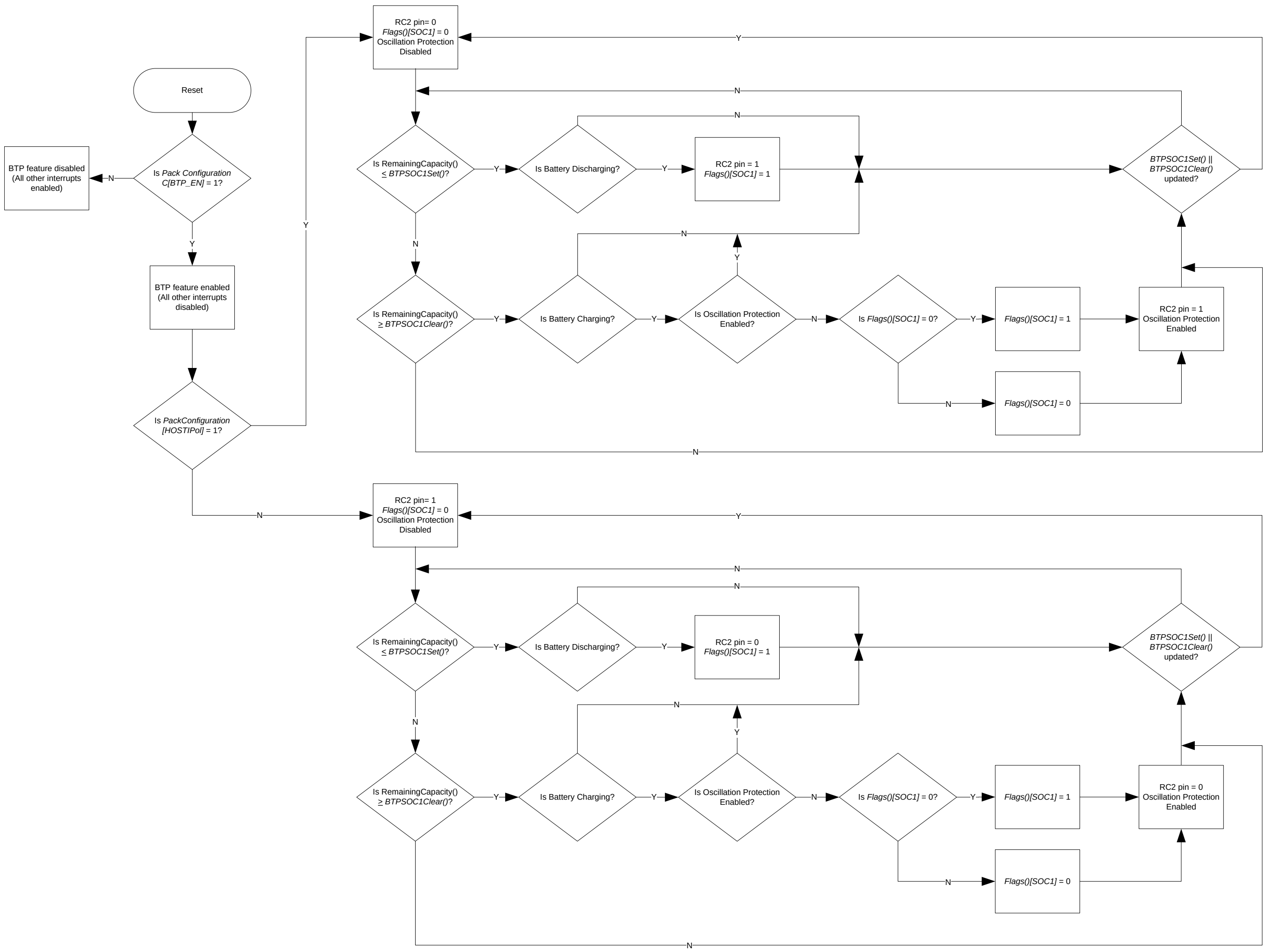


Figure 2-2. BTP Algorithm Flow

In normal usage, the BTP thresholds are continuously updated by the host system at predetermined increments, each time reinitializing the *Flags()[SOC1]* bit to 0 and waiting for the crossing of the next threshold to trigger a new interrupt. If the thresholds are always updated after each interrupt, then it is implied that the crossing of a set or clear threshold always triggers a new interrupt. This is highlighted below in Figure 2-3, *BTP Configuration with Multiple Thresholds*.

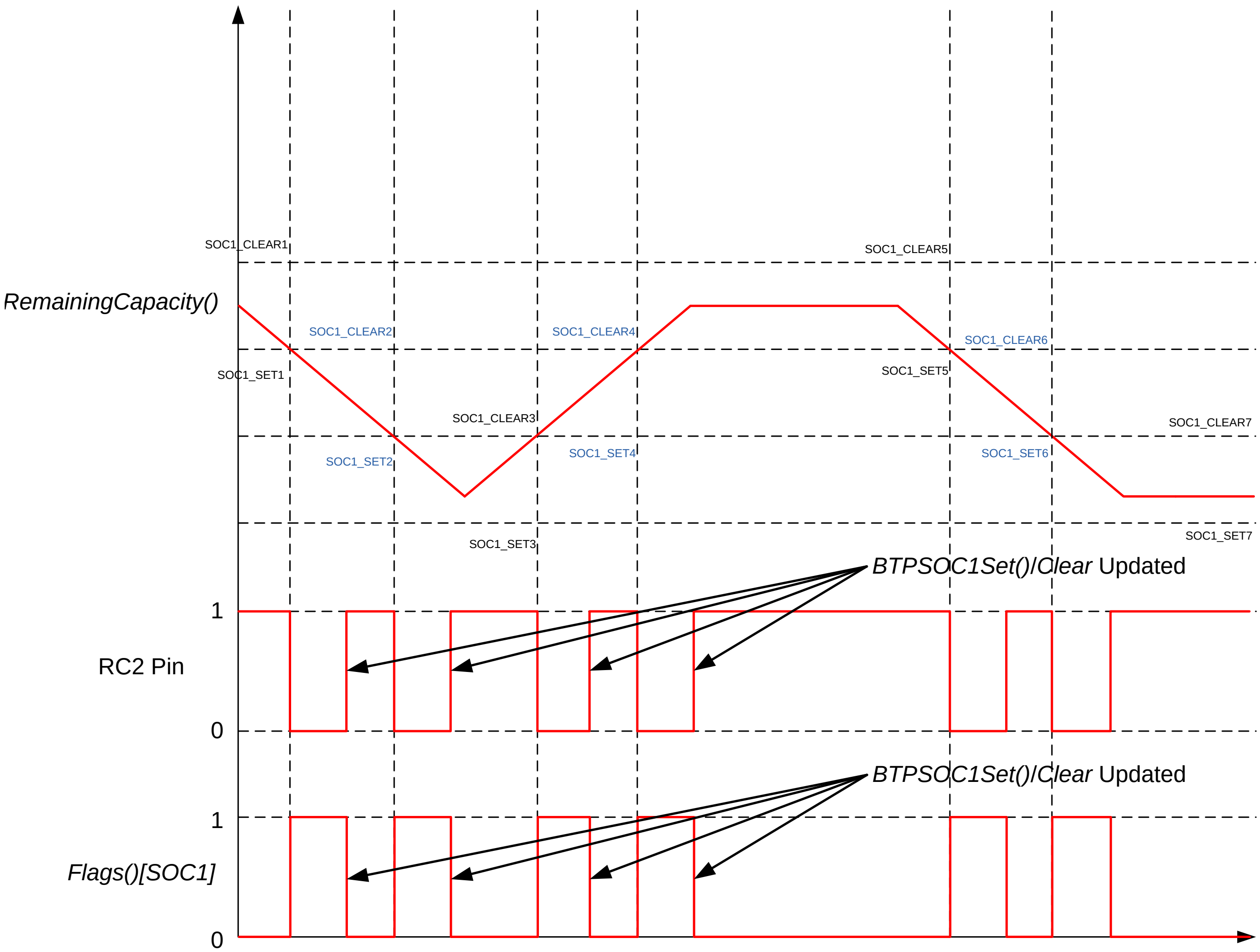


Figure 2-3. BTP Configuration with Multiple Thresholds

However, it is possible that the host may fail to write new thresholds or experience a significant delay in attempting to do so. In this case, there could be an occurrence where the clear threshold is crossed after an interrupt due to a prior set threshold crossing. Thus, the *[SOC1]* bit would experience a change but a new interrupt would not be triggered on RC2. Thus, continued crossings without updates to *BTPSOC1Set()* or *BTPSOC1Clear()* will only result in changes to *Flags()[SOC1]*. Figure 2-4, *BTP Configuration with Shared Thresholds*, shows the case where identical thresholds are written to *BTPSOC1Set()* or *BTPSOC1Clear()*. Figure 2-5, *BTP Configuration with Separate Thresholds*, shows the alternate case where unique thresholds are written to *BTPSOC1Set()* or *BTPSOC1Clear()*.

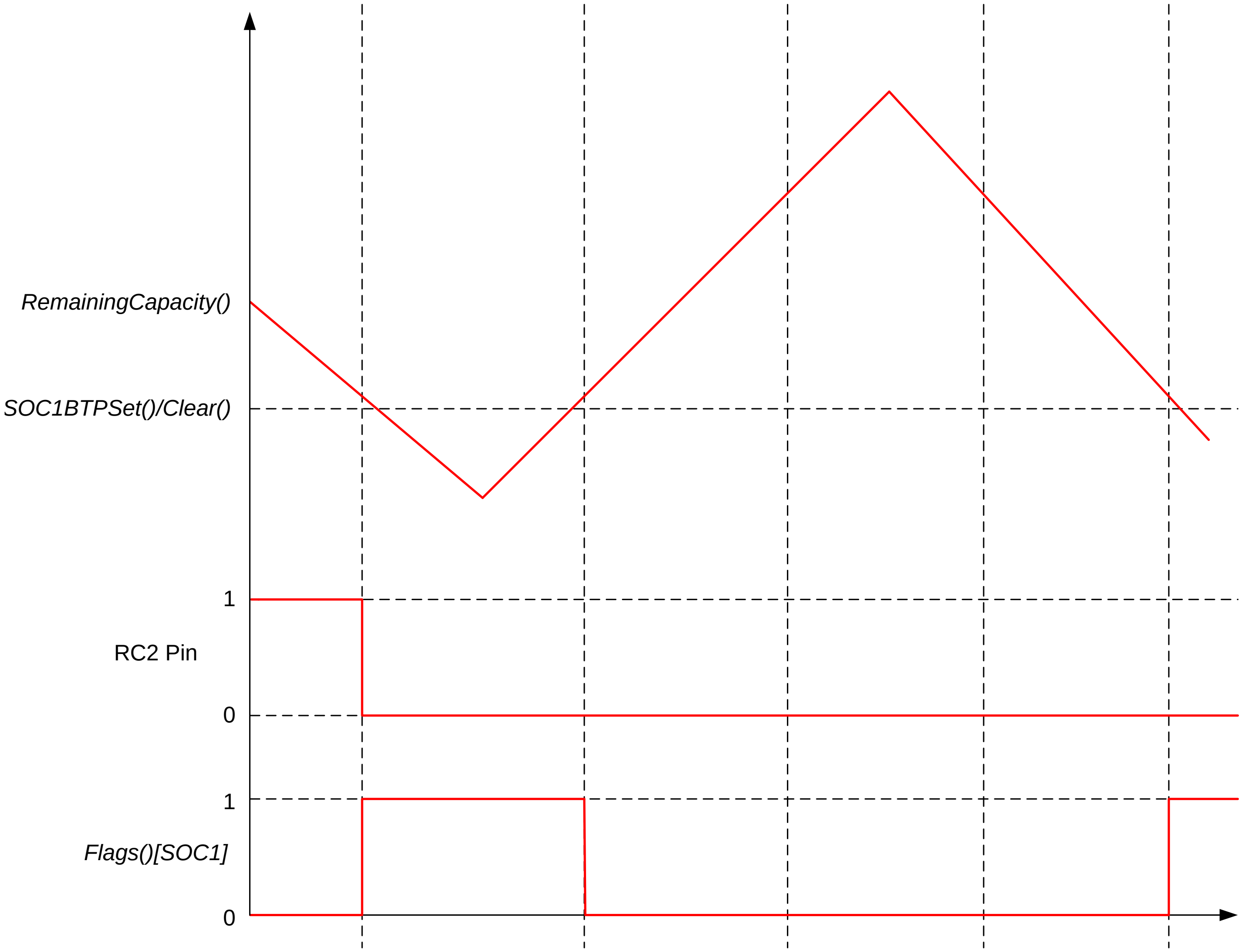


Figure 2-4. BTP Configuration with Shared Thresholds

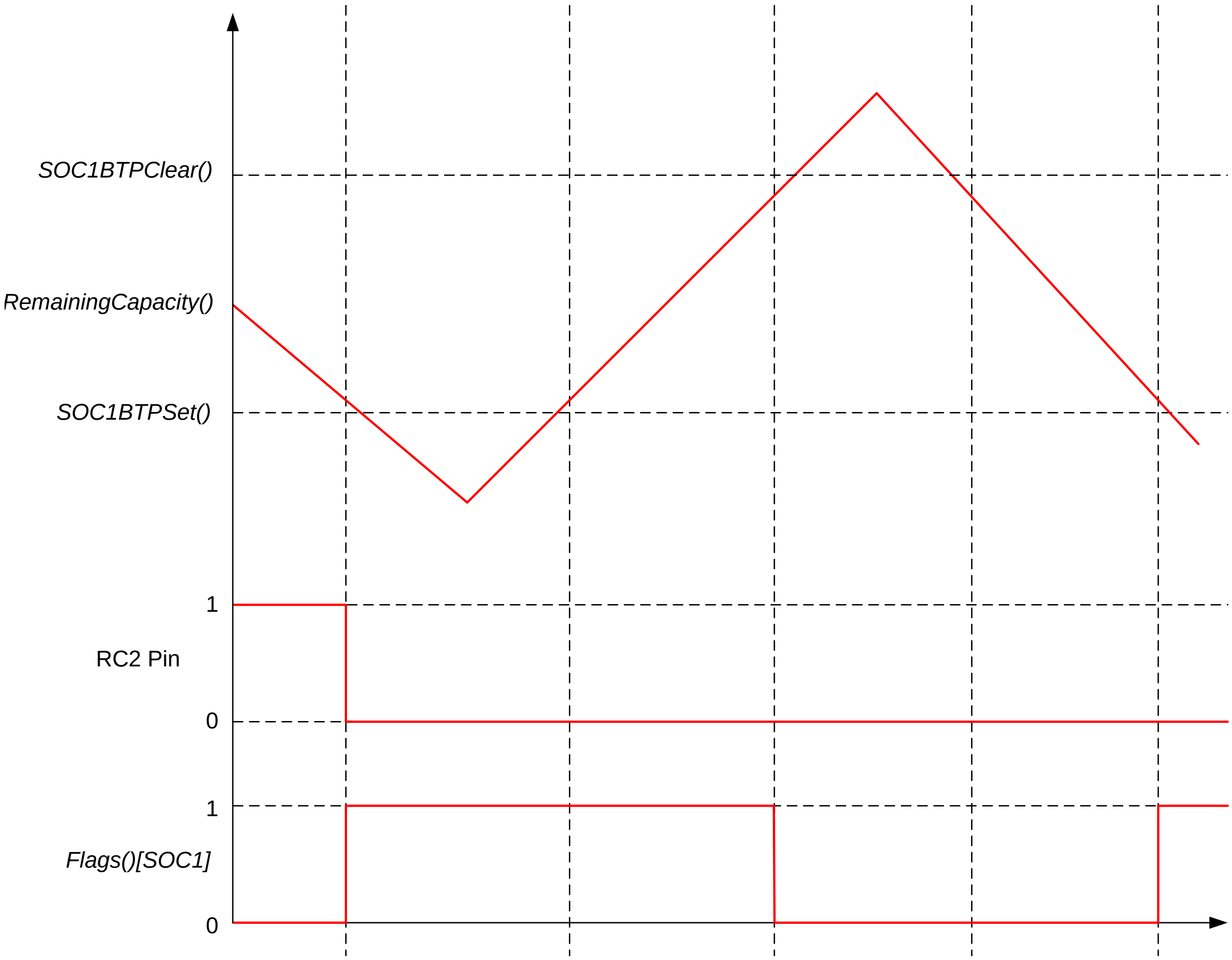


Figure 2-5. BTP Configuration with Separate Thresholds

2.5 Temperature Measurement and the TS Input

The fuel gauge measures battery temperature via the TS input to supply battery temperature status information to the fuel gauging algorithm and charger-control sections of the gauge. Alternatively, the gauge can also measure internal temperature via its on-chip temperature sensor, but only if the ***[TEMPS]*** bit of ***Pack Configuration*** is cleared.

Regardless of which sensor is used for measurement, a system processor can request the current battery temperature by reading the *Temperature()* register (see Section 4.1, *Standard Data Commands*, for specific information).

The thermistor circuit requires the use of an external negative temperature coefficient (NTC) thermistor with R25 = 10 kΩ ± 1% and B25/85 = 3435 kΩ ± 1% (such as Semitec 103AT-2) that connects between the REG25 and TS pins. Additional circuit information for connecting the thermistor to the fuel gauge is shown in Chapter 6, *Reference Schematic*.

2.6 Li-Ion Charging Features

2.6.1 JEITA Charging Profile

The fuel gauge provides full support for the JEITA charging algorithm, which employs separate constant-current constant-voltage (CCCV) charging parameters depending on the measured*Temperature()*. The allowable charging range is divided into four regions defined by ***T1 Temp***, ***T2 Temp***, ***T3 Temp***, ***T4 Temp***, and ***T5 Temp***, each with it's own dedicated *ChargingCurrent()* and *ChargingVoltage()* values.

- If *Temperature()* < ***T1 Temp***, *ChargingCurrent()* and *ChargingVoltage()* are set to 0.
- If ***T1 Temp*** ≤*Temperature()* ≤ ***T2 Temp***, ***T1-T2 Chg Current*** and ***T1-T2 Chg Voltage*** are reported.
- If ***T2 Temp*** <*Temperature()* ≤ ***T3 Temp***, ***T2-T3 Chg Current*** and ***T2-T3 Chg Voltage*** are reported.
- If ***T3 Temp*** <*Temperature()* ≤ ***T4 Temp***, ***T3-T4 Chg Current*** and ***T3-T4 Chg Voltage*** are reported.
- If ***T4 Temp*** <*Temperature()* ≤ ***T5 Temp***, ***T4-T5 Chg Current*** and ***T4-T5 Chg Voltage*** are reported.
- If *Temperature()* > ***T5 Temp***, *ChargingCurrent()* and *ChargingVoltage()* are set to 0.

The diagrams in Figure 2-6, *JEITA Charging Current Profile*, and Figure 2-7, *JEITA Charging Voltage Profile*, provide a visual depiction of the JEITA charging algorithm.

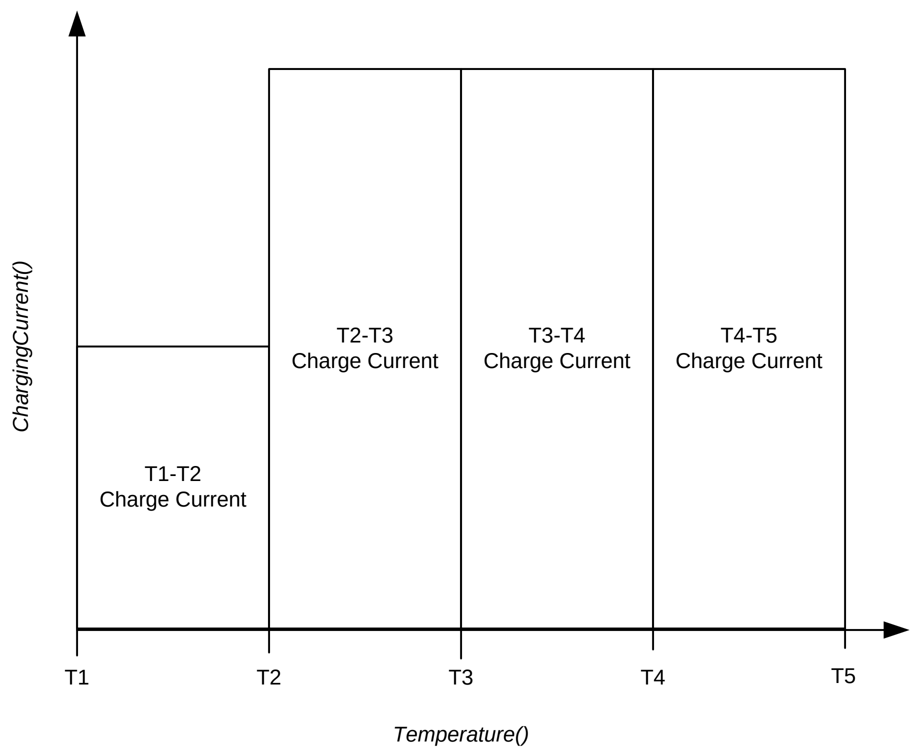


Figure 2-6. JEITA Charging Current Profile

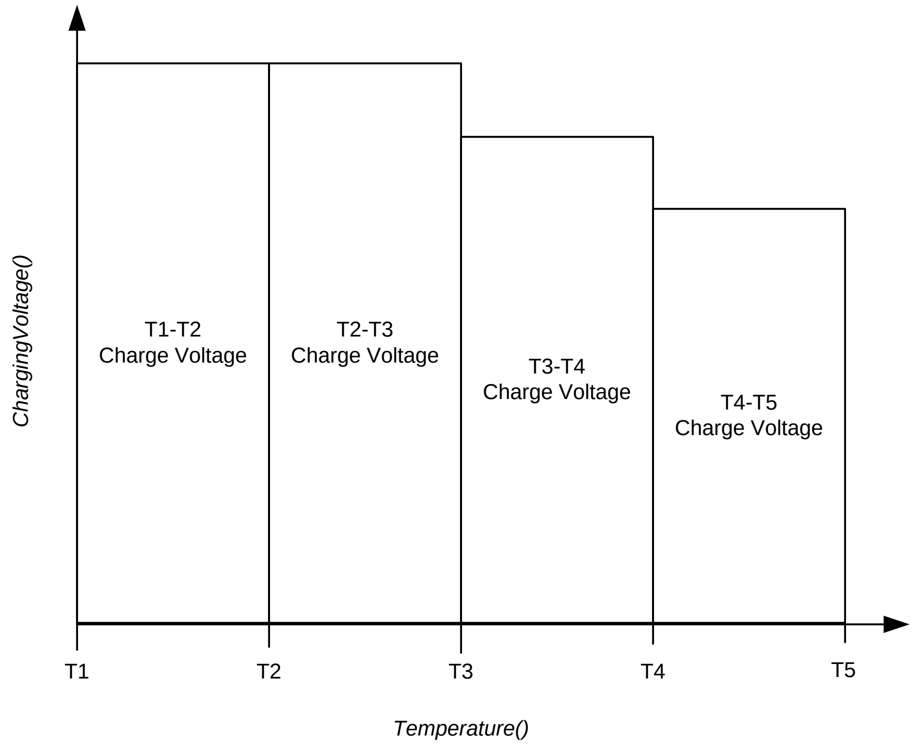


Figure 2-7. JEITA Charging Voltage Profile

Temperature hysteresis (***Temp Hys***) is also applied to movement between various ranges in order to prevent charging parameter oscillation when *Temperature()* continuously changes by a few degrees right on the edge of a temperature boundary. When moving from cooler to warmer temperatures, positive hysteresis is applied to the ***T1 Temp*** and ***T2 Temp*** thresholds. On the contrary, when moving from warmer to cooler temperatures, negative hysteresis is applied to the ***T3 Temp***, ***T4 Temp***, and ***T5 Temp*** thresholds. In order to convert the four-range JEITA profile to a classic, notebook-style three-range version, simply set T4 Temp = T3 Temp.

The diagrams in Figure 2-8, *Temperature Hysteresis for Charging Current*, and Figure 2-9, *Temperature Hysteresis for Charging Voltage*, illustrate how temperature hysteresis is applied depending on transition direction.

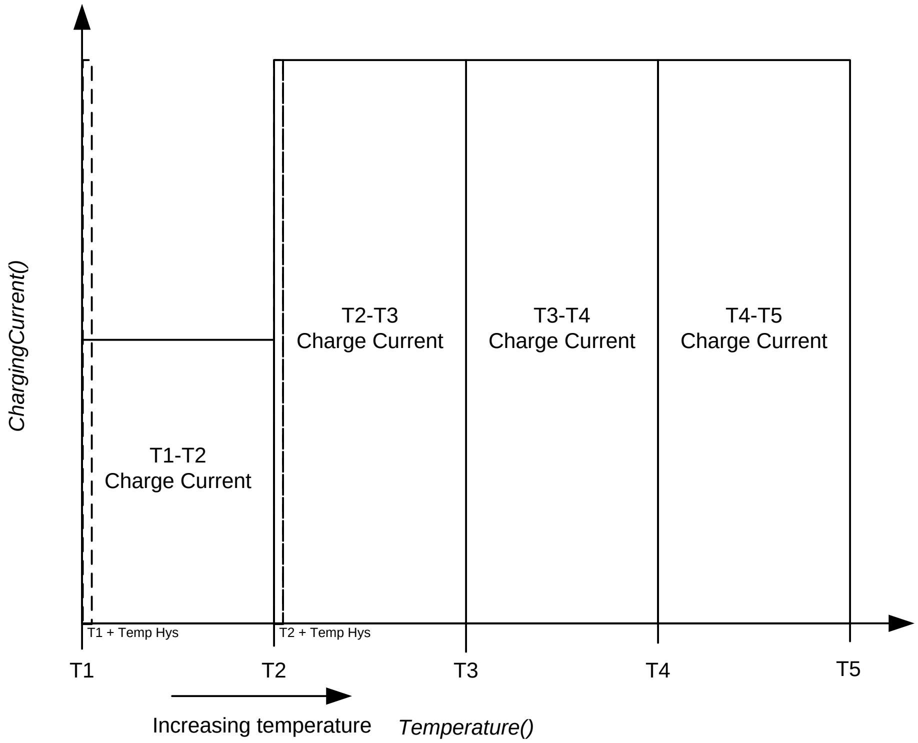


Figure 2-8. Temperature Hysteresis for Charging Current

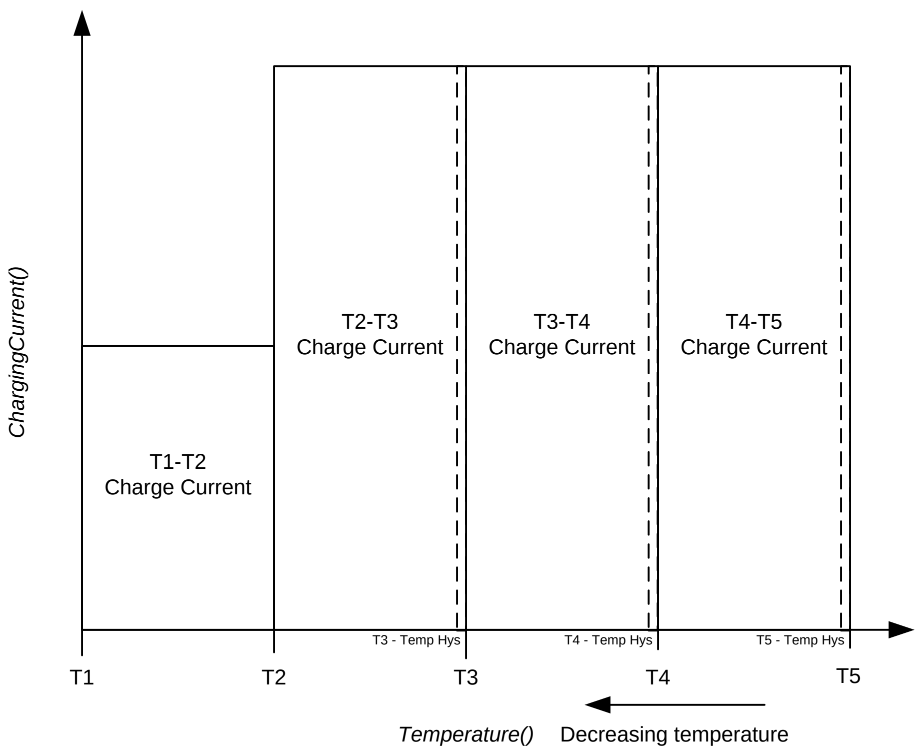


Figure 2-9. Temperature Hysteresis for Charging Voltage

2.6.2 Charge Suspend

If *Temperature()* < T1 Temp or > T5 Temp during active charging, a charge suspend condition is indicated by setting the *Flags()[CHG_SUS]* bit to 1 and clearing *ChargingCurrent()* and *ChargingVoltage()* to 0. Additionally, the fuel gauge can force the CHG FET open in charge suspend conditions if ***Pack Configuration D [CSFET]*** is set to 1.

2.6.3 Charge Inhibit

If *Temperature()* < T1 Temp or > T4 Temp without active charging, a charge inhibit condition is indicated by setting the *Flags()[CHG_INH]* bit to 1 and clearing *ChargingCurrent()* and *ChargingVoltage()* to 0. Additionally, the fuel gauge can force the CHG FET open in charge inhibit conditions if ***Pack Configuration D [CIFET]*** is set to 1.

2.6.4 Full Charge Termination Detection

Full charge termination is detected on the basis of voltage-,current-, and capacity-based conditions or SOC level, depending on the setting configured in ***FC Set %***. If set to -1, then the following conditions are used as to qualify successful full charge termination detection:

1. *Voltage()* **≥ *Charging Voltage – Taper Voltage***, and
2. During two consecutive periods of ***Current Taper Window***, the *AverageCurrent()* is **< *Taper Current***, and
3. During two consecutive periods of ***Current Taper Window***, the accumulated change in capacity > 0.25 mAh

Else, setting ***FC Set %*** to some non-zero and non-negative value will result in charge termination being detected at that *StateofCharge()*. Once full charge termination conditions are met, the *Flags()[FC]* bit is set to indicate charge termination to the host. The fuel gauge can also open the CHG FET during full charge termination if the ***Pack Configuration D [FCFET]*** bit is set to 1. Additionally, if ***Pack Configuration [RMFCC]*** = 1, then *RemainingCapacity()* is set equal to *FullChargeCapacity()* upon full charge termination. The fuel gauge exits charge termination and associated flags are cleared when SOC decreases below ***FC Clear %***. A separate *[CHG]* bit in *Flags()* can be cleared to provide an earlier nearly full charge warning to the system based on the SOC threshold configured in TCA Set. Similar to ***FC Set %***, if ***TCA Set %*** is programmed to -1, the *[CHG]* bit state will be dependent on the voltage-, current-, and capacity-based full charge termination detection conditions instead of an SOC level. Likewise, the *[CHG]* bit is re-set once SOC decreases below ***TCA Clear %***.

2.7 Power Modes

The fuel gauge has three power modes: NORMAL, SLEEP, and FULLSLEEP. In NORMAL mode, the fuel gauge is fully powered and continually refreshes its dataset every 1 second. In SLEEP mode, the fuel gauge CPU is halted and frequency of data measurement is reduced to 20-second intervals for increased power savings when the system is in a standby state. In FULLSLEEP mode, the fuel gauge disables its high frequency oscillator (HFO) for highest operating power savings. The relationship between these modes is shown in Figure 2-10. Details are described in Section 2.7.1 through Section 2.7.3.

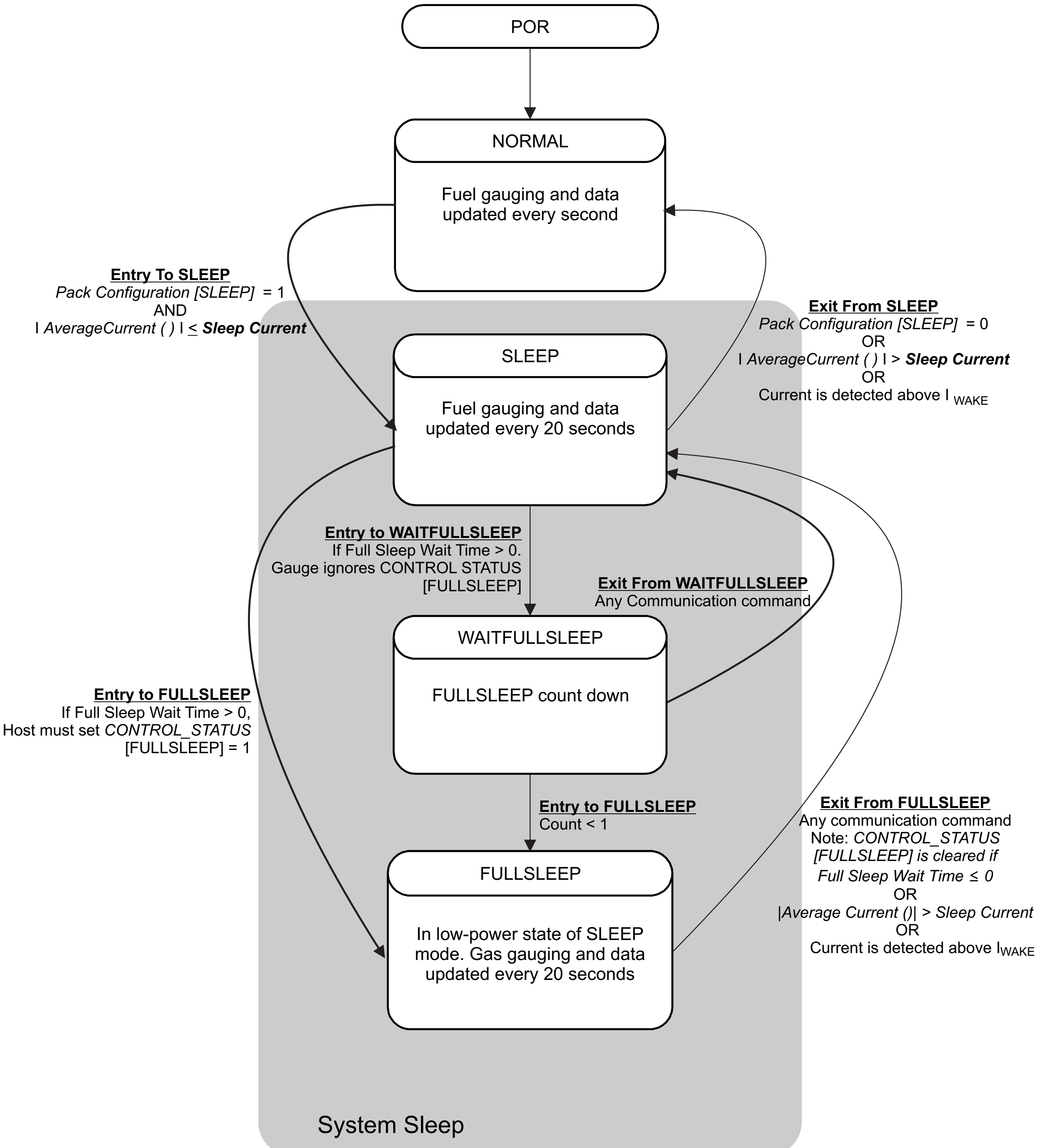


Figure 2-10. Power Mode Diagram–System Sleep

2.7.1 NORMAL Mode

NORMAL mode is the fuel gauge's standard operational mode where *Voltage()*, *AverageCurrent()*, and *Temperature()* measurements are taken and the full interface dataset is updated. Because the gauge consumes the most power in NORMAL mode, the Impedance Track algorithm minimizes the time the fuel gauge remains in this mode.

2.7.2 SLEEP Mode

SLEEP mode is entered automatically if the feature is enabled via the ***Pack Configuration [SLEEP]*** bit and *AverageCurrent()* < ***Sleep Current***. Once entry into SLEEP mode has been qualified, but prior to entering it, the fuel gauge performs an ADC auto-calibration to minimize offset. In SLEEP mode, the fuel gauge takes data measurements and updates its data set every 20 seconds. However, a majority of its time is spent in an idle condition.

The fuel gauge exits SLEEP if any of the following conditions are detected:

- *AverageCurrent()* > ***Sleep Current*** OR
- |Current| > I_{WAKE} through R_{SENSE} is detected when the I_{WAKE} comparator is enabled OR
- ***Pack Configuration [SLEEP]*** is cleared to 0.

2.7.3 FULLSLEEP Mode

If ***FS Wait*** time > 0, then FULLSLEEP mode is entered automatically when the fuel gauge is in SLEEP mode and its counter increments from 0 to ***FS Wait*** time. Manual entry into FULLSLEEP mode can be commanded from the host if ***FS Wait*** time = 0 and the *SET_FULLSLEEP* command is issued, which sets the *CONTROL_STATUS [FULLSLEEP]* bit and immediately transitions the fuel gauge into this mode. In FULLSLEEP mode, the fuel gauge also takes data measurements and updates its data set every 20 seconds, but also disabled its HFO for additional power savings. As a result, the fuel gauge may hold the serial communication lines low for as long as 4 ms when transitioning out of FULLSLEEP mode in order to allow sufficient time for its HFO frequency to stabilize.

The fuel gauge exits FULLSLEEP mode if any of the following conditions are detected:

- Any communication command OR
- *Average Current()* > ***Sleep Current*** OR
- |***Current***| > I_{WAKE} through R_{SENSE} is detected when the I_{WAKE} comparator is enabled.

2.8 Power Control

2.8.1 Reset Functions

When the fuel gauge detects a software reset by sending *Control() [RESET]* subcommand, it resets the firmware and increments the reset counter. This counter is accessible by issuing the command *Control()* function with the *RESET_DATA* subcommand.

2.8.2 Wake-Up Comparator

The wake-up comparator indicates a change in cell current while the fuel gauge is in SLEEP modes. ***Pack Configuration*** uses bits ***[RSNS1, RSNS0]*** to set the sense resistor selection. ***Pack Configuration*** also uses the ***[IWAKE]*** bit to select one of two possible voltage threshold ranges for the given sense resistor selection. An internal interrupt is generated when the threshold is breached in either charge or discharge directions. Setting both ***[RSNS1]*** and ***[RSNS0]*** to 0 disables this feature.

Table 2-7. I_{WAKE} Threshold Settings(1)

IWAKE	RSNS1	RSNS0	Vth(SRP-SRN)
0	0	0	Disabled

(1) The actual resistance value vs the setting of the sense resistor is not important just the actual voltage threshold when calculating the configuration. The voltage thresholds are typical values under room temperature.

Table 2-7. I_{WAKE} Threshold Settings[(1)] (continued)

IWAKE	RSNS1	RSNS0	Vth(SRP-SRN)
1	0	0	Disabled
0	0	1	1.0 mV or –1.0 mV
1	0	1	2.2 mV or –2.2 mV
0	1	0	2.2 mV or –2.2 mV
1	1	0	4.6 mV or –4.6 mV
0	1	1	4.6 mV or –4.6 mV
1	1	1	9.8 mV or –9.8 mV

2.8.3 Flash Updates

Data flash can only be updated if *Voltage()* ≥ ***Flash Update OK Voltage***. Flash programming current can cause an increase in LDO dropout. The value of ***Flash Update OK Voltage*** must be selected such that the V_{PWR} voltage does not fall below its minimum of 2.45 V during flash write operations.

2.9 Coulomb Counter Autocalibration

The fuel gauge provides an autocalibration feature that measures the voltage offset error across SRP and SRN from time-to-time as operating conditions change. It subtracts the resulting offset error from normal sense resistor voltage, V_{SR}, for maximum measurement accuracy.

Autocalibration of the CC begins on entry to SLEEP mode, except if *Temperature()* is ≤ 5°C or *Temperature()* ≥ 45°C, but will not occur more than once per every 10 hours.

The fuel gauge also performs autocalibration offset calibration any time the following conditions are detected:

1. The condition of *AverageCurrent()* ≤ 100 mA, and
2. Voltage change since last offset calibration ≥ 256 mV or Temperature change since last offset calibration is greater than 8°C for ≥ 60 seconds.

Capacity and current measurements continue at the last measured rate during the offset calibration when these measurements cannot be performed. If the battery voltage drops more than 32 mV during the offset calibration, the load current has likely increased considerably; therefore, the offset calibration is stopped.

Communications

3.1 Authentication

The fuel gauge supports a SHA-1-based authentication protocol that allows a host to securely verify battery pack authenticity. Sending a 160-bit random challenge initiates the authentication process wherein the fuel gauge computes a response digest using a double SHA-1 transform. The transmitted challenge is appended to a secret 128-bit authentication key and run through the transform. Afterwards, the resulting hash is then re-appended to the same key and a second hash is computed, resulting in the final 160-bit digest that is returned to the host. The host reproduces the same digest calculation on its side, using the shared key, and compares to the one read from the fuel gauge. If they match, the authentication process is successful.

3.2 Key Programming (Data Flash Key)

By default, the fuel gauge contains a default plain-text authentication key of 0x0123456789ABCDEFFEDCBA9876543210. This default key is intended for development purposes. It must be changed to a secret key and the part immediately SEALED, before putting a pack into operation. Once written, a new plain-text key cannot be read again from the fuel gauge while in SEALED mode.

Once the fuel gauge is UNSEALED, the authentication key can be changed from its default value by writing to the *Authenticate()* Extended Data Command locations. A 0x00 is written to *BlockDataControl()* to enable the authentication data commands. The *DataFlashClass()* is issued 112 (0x70) to set the Security class. Up to 32 bytes of data can be read directly from the *BlockData()* (0x40 through 0x5F) and the authentication key is located at 0x48 (0x40 + 0x08 offset) to 0x57 (0x40 + 0x17 offset). The new authentication key can be written to the corresponding locations (0x48 through 0x57) using the *BlockData()* command. The data is transferred to the data flash when the correct checksum for the whole block (0x40 through 0x5F) is written to *BlockDataChecksum()* (0x60). The checksum is (255 – x) where x is the 8-bit summation of the *BlockData()* (0x40 through 0x5F) on a byte-by-byte basis. Once the authentication key is written, the gauge can then be SEALED again.

3.3 Executing an Authentication Query

To execute an authentication query in UNSEALED mode, a host must first write 0x01 to the *BlockDataControl()* command, to enable the authentication data commands. If in SEALED mode, 0x00 must be written to *DataFlashBlock()*, instead.

Next, the host writes a 20-byte authentication challenge to the *Authenticate()* address locations (0x40 through 0x53). After a valid checksum for the challenge is written to *AuthenticateChecksum()*, the fuel gauge uses the challenge, in conjunction with the programmed authentication key, in its SHA-1 computations. After completion, the resulting digest is stored in the *Authenticate()* register, overwriting the pre-existing challenge. The host must wait at least 45 ms to read the resulting digest. The host may then read this response and compare it against the result created by its own parallel computation.

3.4 I^{2}C Interface

The fuel gauge supports the standard I^{2}C read, incremental read, one-byte write quick read, and functions. The 7-bit device address (ADDR) is the most significant 7 bits of the hex address and is fixed as 1010101, or 0x55. The 8-bit device address is therefore 0xAA or 0xAB for write or read, respectively.

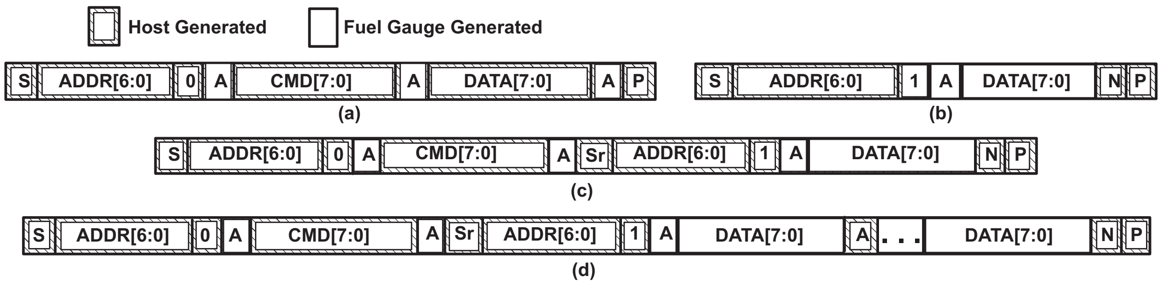


Figure 3-1. Supported I^2C Formats

(a) 1-byte write

(b) Quick read

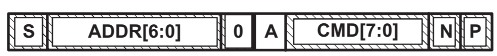
(c) 1-byte read

(d) Incremental read (S = Start, Sr = Repeated Start, A = Acknowledge, N = No Acknowledge, and P = Stop).

The quick read returns data at the address indicated by the address pointer. The address pointer, a register internal to the I^2C communication engine, increments whenever data is acknowledged by the fuel gauge or the I^2C master. Quick writes function in the same manner and are a convenient means of sending multiple bytes to consecutive command locations (such as two-byte commands that require two bytes of data).

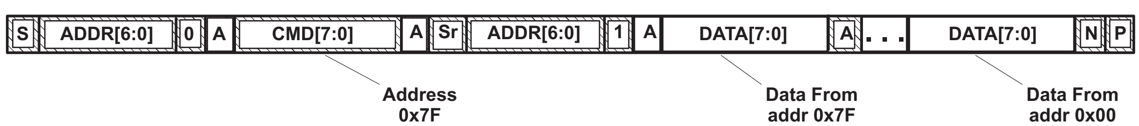
Attempt to write a read-only address (NACK after data sent by master):

Attempt to read an address above 0x7F (NACK command):



Attempt at incremental writes (NACK all extra data bytes sent):

Incremental read at the maximum allowed read address:



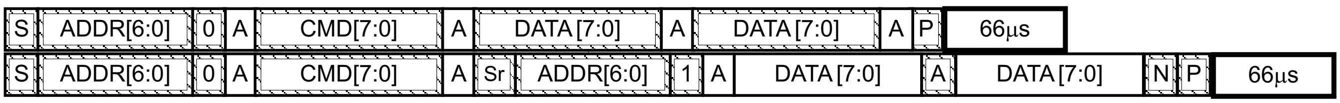
The I^2C engine releases both SDA and SCL if the I^2C bus is held low for $t_{(BUSERR)}$. If the fuel gauge was holding the lines, releasing them frees the master to drive the lines. If an external condition is holding either of the lines low, the I^2C engine enters the low-power SLEEP mode.

3.4.1 I^2C Time Out

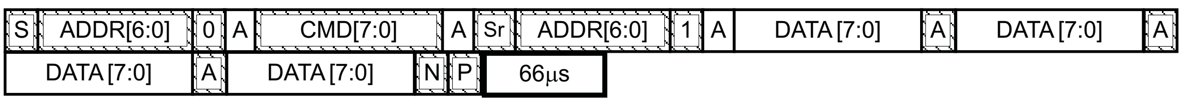
The I^2C engine releases both SDA and SCL if the I^2C bus is held low for about 2 seconds. If the fuel gauge was holding the lines, releasing them frees the master to drive the lines.

3.4.2 I^2C Command Waiting Time

To make sure the correct results of a command with the 400-kHz I^2C operation, a proper waiting time must be added between issuing command and reading results. For subcommands, the following diagram shows the waiting time required between issuing the control command the reading the status with the exception of the checksum command. A 100-ms waiting time is required between the checksum command and reading result. For read-write standard commands, a minimum of 2 seconds is required to get the result updated. For read-only standard commands, there is no waiting time required, but the host must not issue all standard commands more than two times per second. Otherwise, the gauge could result in a reset issue due to the expiration of the watchdog timer.



Waiting time between control subcommand and reading results



Waiting time between continuous reading results

3.4.3 I^2C Clock Stretching

I^2C clock stretches can occur during all modes of fuel gauge operation. In the SLEEP mode, a short clock stretch occurs on all I^2C traffic as the device must wake-up to process the packet. In NORMAL and SLEEP modes, clock stretching only occurs for packets addressed for the fuel gauge. The timing of stretches varies as interactions between the communicating host and the gauge are asynchronous. The I^2C clock stretches may occur after start bits, the ACK/NAK bit and first data bit transmit on a host read cycle. The majority of clock stretch periods are small (≤ 4 ms) as the I^2C interface peripheral and CPU firmware perform normal data flow control. However, less frequent but more significant clock stretch periods may occur when data flash (DF) is being written by the CPU to update the resistance (Ra) tables and other DF parameters such as Qmax. Due to the organization of DF, updates need to be written in data blocks consisting of multiple data bytes.

An Ra table update requires erasing a single page of DF, programming the updated Ra table and a flag. The potential I^2C clock stretching time is 24 ms maximum. This includes 20-ms page erase and 2-ms row programming time (×2 rows). The Ra table updates occur during the discharge cycle and at up to 15 resistance grid points that occur during the discharge cycle.

A DF block write typically requires a maximum of 72 ms. This includes copying data to a temporary buffer and updating DF. This temporary buffer mechanism protects from power failure during a DF update. The first part of the update requires 20 ms to erase the copy buffer page, 6 ms to write the data into the copy buffer and the program progress indicator (2 ms for each individual write). The second part of the update is writing to the DF and requires 44-ms DF block update time. This includes a 20-ms each page erase for two pages and 2 ms each row write for two rows.

In the event that a previous DF write was interrupted by a power failure or reset during the DF write, an additional 44-ms maximum DF restore time is required to recover the data from a previously interrupted DF write. In this power failure recovery case, the total I^2C clock stretching is 116 ms maximum.

Another case where I^2C clock stretches is at the end of discharge. The update to the last discharge data goes through the DF block update twice because two pages are used for the data storage. The clock stretching in this case is 144 ms maximum. This occurs if there has been an Ra table update during the discharge.

Data Commands

4.1 Standard Data Commands

The bq27742-G1 fuel gauge uses a series of 2-byte standard commands to enable system reading and writing of battery information. Each standard command has an associated command-code pair, as indicated in Table 4-1. Each protocol has specific means to access the data at each Command Code. Data RAM is updated and read by the gauge only once per second. Standard commands are accessible in the NORMAL operation mode.

Table 4-1. Standard Commands

Command Name	Command Code	Unit	SEALED Access
Control()	0x00 and 0x01	—	RW
AtRate()	0x02 and 0x03	mA	RW
UnfilteredSOC()	0x04 and 0x05	%	R
Temperature()	0x06 and 0x07	0.1°K	R
Voltage()	0x08 and 0x09	mV	R
Flags()	0x0A and 0x0B	—	R
NomAvailableCapacity()	0x0C and 0x0D	mAh	R
FullAvailableCapacity()	0x0E and 0x0F	mAh	R
RemainingCapacity()	0x10 and 0x11	mAh	R
FullChargeCapacity()	0x12 and 0x13	mAh	R
AverageCurrent()	0x14 and 0x15	mA	R
TimeToEmpty()	0x16 and 0x17	min	R
FilteredFCC()	0x18 and 0x19	mAh	R
SafetyStatus()	0x1A and 0x1B	—	R
UnfilteredFCC()	0x1C and 0x1D	mAh	R
Imax()	0x1E and 0x1F	mA	R
UnfilteredRM()	0x20 and 0x21	mAh	R
FilteredRM()	0x22 and 0x23	mAh	R
BTPSOC1Set()	0x24 and 0x25	mAh	RW
BTPSOC1Clear()	0x26 and 0x27	mAh	RW
InternalTemperature()	0x28 and 0x29	0.1°K	R
CycleCount()	0x2A and 0x2B	Counts	R
StateofCharge()	0x2C and 0x2D	%	R
StateofHealth()	0x2E and 0x2F	%/num	R
ChargingVoltage()	0x30 and 0x31	mV	R
ChargingCurrent)	0x32 and 0x33	mA	R
PassedCharge()	0x34 and 0x35	mAh	R
DOD0()	0x36 and 0x37	hex	R
SelfDischargeCurrent()	0x34 and 0x35	mA	R

4.1.1 Control(): 0x00 and 0x01

Issuing a *Control()* command requires a subsequent 2-byte subcommand. These additional bytes specify the particular control function desired. The *Control()* command allows the system to control specific features of the fuel gauge during normal operation and additional features when the fuel gauge is in different access modes, as described in Table 4-2.

Table 4-2. Control() Subcommands

Subcommand Name	Subcommand Code	SEALED Access	Description
CONTROL_STATUS	0x0000	Yes	Reports the status of DF Checksum, Impedance Track, and so on.
DEVICE_TYPE	0x0001	Yes	Reports the device type of 0x0742 (indicating bq27742-G1).
FW_VERSION	0x0002	Yes	Reports the firmware version on the device type.
HW_VERSION	0x0003	Yes	Reports the hardware version on the device type.
PROTECTOR_VERSION	0x0004	Yes	Reports the hardware protector version on the device type.
RESET_DATA	0x0005	Yes	Returns reset data.
PREV_MACWRITE	0x0007	Yes	Returns previous *Control()* subcommand code.
CHEM_ID	0x0008	Yes	Reports the chemical identifier of the Impedance Track configuration.
BOARD_OFFSET	0x0009	No	Forces the device to measure and store the board offset.
CC_OFFSET	0x000A	No	Forces the device to measure the CC offset.
DF_VERSION	0x000C	Yes	Reports the data flash version of the device.
SET_FULLSLEEP	0x0010	Yes	Sets the *CONTROL_STATUS[FULLSLEEP]* bit to 1.
SET_SHUTDOWN	0x0013	Yes	Sets the *CONTROL_STATUS[SHUTDWN]* bit to 1.
CLEAR_SHUTDOWN	0x0014	Yes	Clears the *CONTROL_STATUS[SHUTDWN]* bit to 1.
STATIC_CHEM_CHKSUM	0x0017	Yes	Calculates chemistry checksum.
ALL_DF_CHKSUM	0x0018	Yes	Reports checksum for all data flash excluding device specific variables.
STATIC_DF_CHKSUM	0x0019	Yes	Reports checksum for static data flash excluding device specific variables.
PROTECTOR_CHKSUM	0x001A	Yes	Reports checksum for protector configuration data flash excluding device specific variables.
SEALED	0x0020	No	Places the fuel gauge in SEALED access mode.
IT_ENABLE	0x0021	No	Enables the Impedance Track algorithm.
IMAX_INT_CLEAR	0x0023	Yes	Clears an Imax interrupt that is currently asserted on the RC2 pin.
START_FET_TEST	0x0024	No	Starts FET Test based on data entered in *FETTest()* register. Sets and clears the *[FETTST]* bit in *CONTROL_STATUS*.
CAL_ENABLE	0x002D	No	Toggle CALIBRATION mode.
RESET	0x0041	No	Forces a full reset of the fuel gauge.
EXIT_CAL	0x0080	No	Exit CALIBRATION mode.
ENTER_CAL	0x0081	No	Enter CALIBRATION mode.
OFFSET_CAL	0x0082	No	Reports internal CC offset in CALIBRATION mode.

4.1.1.1 CONTROL_STATUS: 0x0000

Instructs the fuel gauge to return status information to control addresses 0x00 and 0x01. The status word includes the following information.

Table 4-3. CONTROL_STATUS Flags

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	OCVTAKEN	FAS	SS	CALMODE	CCA	BCA	RSVD	RSVD
Low Byte	SHUTDN_EN	FETTST	FULLSLEEP	SLEEP	LDMD	RUP_DIS	VOK	QEN

High Byte

OCVTAKEN = Cleared on entry to RELAXATION mode and set to 1 when OCV measurement is performed in RELAXATION mode.

FAS = Status bit indicating the fuel gauge is in FULL ACCESS SEALED state. Active when set (no data flash access).

SS = Status bit indicating the fuel gauge is in the SEALED state. Active when set (no ROM access).

CALMODE = Status bit indicating the calibration function is active. True when set. Default is 0.

CCA = Status bit indicating the Coulomb Counter Calibration routine is active. The CCA routine takes place approximately 1 minute after the initialization and periodically as gauging conditions change. Active when set.

BCA = Status bit indicating the Board Calibration routine is active. Active when set.

RSVD = Reserved

RSVD = Reserved

Low Byte

SHUTDN_EN = Control bit indicating that the *SET_SHUTDOWN* subcommand has been sent and signals an external shutdown of the fuel gauge when conditions permit. See Section 2.4.1, *SHUTDOWN Mode*.

FETTST = Status bit indicating the state of the FET test. True when set. Default is 0.

FULLSLEEP = Status bit indicating the fuel gauge is in FULLSLEEP mode. True when set. The state can be detected by monitoring the power used by the fuel gauge because any communication automatically clears it.

SLEEP = Status bit indicating the fuel gauge is in SLEEP mode. True when set.

LDMD = Status bit indicating the Impedance Track algorithm is using *constant-power* model. True when set. Default is 0 (*constant-current* model).

RUP_DIS = Status bit indicating the Ra table updates are disabled. True when set.

VOK = Status bit indicating cell voltages are OK for Qmax updates. True when set.

QEN = Status bit indicating the Qmax updates are enabled. True when set.

4.1.1.2 DEVICE_TYPE: 0x0001

Instructs the fuel gauge to return the device type to addresses 0x00 and 0x01. The bq27742-G1 device type returns 0x0742.

4.1.1.3 FW_VERSION: 0x0002

Instructs the fuel gauge to return the firmware version to addresses 0x00 and 0x01. The firmware version returned is 0x0103.

4.1.1.4 HW_VERSION: 0x0003

Instructs the fuel gauge to return the hardware version to addresses 0x00 and 0x01. For bq27742-G1, 0x0000 or 0x0060 is returned.

4.1.1.5 PROTECTOR_VERSION: 0x0004

Instructs the fuel gauge to return the hardware version of the protector portion of the device to addresses 0x00 and 0x01.

4.1.1.6 RESET_DATA: 0x0005

Instructs the fuel gauge to return the number of resets performed to addresses 0x00 and 0x01.

4.1.1.7 PREV_MACWRITE: 0x0007

Instructs the fuel gauge to return the previous *Control()* subcommand written to addresses 0x00 and 0x01. The value returned is limited to less than 0x0020.

4.1.1.8 CHEM_ID: 0x0008

Instructs the fuel gauge to return the chemical identifier for the Impedance Track configuration to addresses 0x00 and 0x01.

4.1.1.9 BOARD_OFFSET: 0x0009

Instructs the fuel gauge to perform the board offset calibration. During board offset calibration the *CONTROL_STATUS [BCA]* bit is set.

4.1.1.10 CC_OFFSET: 0x000A

Instructs the fuel gauge to perform the coulomb counter offset calibration. During calibration the *CONTROL_STATUS [CCA]* bit is set.

4.1.1.11 DF_VERSION: 0x000C

Instructs the fuel gauge to return the data flash version stored in ***DF Config Version*** to addresses 0x00 and 0x01.

4.1.1.12 SET_FULLSLEEP: 0x0010

Instructs the fuel gauge to set the *CONTROL_STATUS [FULLSLEEP]* bit to 1. The gauge enters the FULLSLEEP power mode after the transition to the SLEEP power state is detected. In FULLSLEEP mode less power is consumed by disabling an oscillator circuit used by the communication engines. For I^2C communications, the first I^2C message incurs a 6- to 8-ms clock stretch while the oscillator is started and stabilized. A communication to the device in FULLSLEEP forces the part back to the SLEEP mode.

4.1.1.13 SET_SHUTDOWN: 0x0013

Sets the *CONTROL_STATUS [SHUTDN_EN]* bit to 1, thereby enabling the fuel gauge to shutdown if conditions are met.

When the *[SHUTDWN]* bit is set, the gas gauge opens both charge and discharge FETs and waits for charger removal. As soon as charger is removed, the gas gauge and protector will shutdown leaving both charger and discharge FETs open. The only way to recover from this SHUTDOWN mode is to connect a charger.

4.1.1.14 CLEAR_SHUTDOWN: 0x0014

Clears the *CONTROL_STATUS [SHUTDN_EN]* bit to 0. The gas gauge closes the charge and discharge FETs and stops the shutdown sequence.

4.1.1.15 STATIC_CHEM_CHKSUM: 0x0017

Instructs the fuel gauge to calculate chemistry checksum as a 16-bit unsigned integer sum of all static chemistry data. The most significant bit (MSB) of the checksum is masked yielding a 15-bit checksum. This checksum is compared with value stored in the data flash ***Static Chem DF Checksum***. If the value matches, the MSB is cleared to indicate pass. If it does not match, the MSB is set to indicate failure. The checksum can verify the integrity of the chemistry data stored internally.

NOTE: The ***Static Chem DF Checksum*** is programmed by the Chemistry programming tool.

4.1.1.16 ALL_DF_CHKSUM: 0x0018

Instructs the fuel gauge to calculate data flash checksum as a 16-bit unsigned integer sum of all data flash excluding device specific variables. The most significant bit (MSB) of the checksum is masked yielding a 15-bit checksum. This checksum is compared with value stored in the data flash ***All DF Checksum***. If the value matches, the MSB is cleared to indicate pass. If it does not match, the MSB is set to indicate failure. The checksum can verify the integrity of the data flash stored internally.

4.1.1.17 STATIC_DF_CHKSUM: 0x0019

Instructs the fuel gauge to calculate static data flash checksum as a 16-bit unsigned integer sum of static data flash excluding device specific variables. The most significant bit (MSB) of the checksum is masked yielding a 15-bit checksum. This checksum is compared with value stored in the data flash ***Static DF Checksum***. If the value matches, the MSB is cleared to indicate pass. If it does not match, the MSB is set to indicate failure. The checksum can verify the integrity of the static data flash stored internally.

4.1.1.18 PROTECTOR_CHKSUM: 0x001A

Instructs the fuel gauge to calculate protector checksum as a 16-bit unsigned integer sum of ***Prot OV Cfg*** and ***Prot OC Cfg*** excluding device specific variables. The most significant bit (MSB) of the checksum is masked yielding a 15-bit checksum. This checksum is compared with value stored in the data flash ***Prot Checksum***. If the value matches, the MSB is cleared to indicate pass. If it does not match, the MSB is set to indicate failure. The checksum can verify the integrity of the protector configuration.

NOTE: The fuel gauge will disable the CHG and DSG FETs while the programmed ***Prot Checksum*** is invalid (that is, does not match the computed checksum). This protects against safety events due to accidental protector misconfiguration.

4.1.1.19 SEALED: 0x0020

Instructs the fuel gauge to transition from UNSEALED state to SEALED state. The fuel gauge should always be set to SEALED state for use in customer's end equipment as it prevents spurious writes to most standard commands and blocks access to most data flash.

4.1.1.20 IT ENABLE: 0x0021

Forces the fuel gauge to begin the Impedance Track algorithm, sets bit 2 of ***UpdateStatus*** and causes the *[VOK]* and *[QEN]* flags to be set in the *CONTROL_STATUS* register. *[VOK]* is cleared if the voltages are not suitable for a Qmax update. Once set, *[QEN]* cannot be cleared. This command is only available when the fuel gauge is UNSEALED and is typically enabled at the last step of production after system test is completed.

4.1.1.21 IMAX_INT_CLEAR: 0x0023

Clears an Imax interrupt that is presently asserted on the RC2 pin. The command is only applicable if the Imax feature is enabled in ***Pack Configuration D [IMAXEN]***.

4.1.1.22 START_FET_TEST: 0x0024

In UNSEALED mode and when IT is not enabled, this command starts the FET Test based on data entered in the *FETTest()* register. On a write to this register, the *FETTest()* register is evaluated for checksum correctness (see Section 4.2.16 for details). If checksum is correct, then the *[FETTST]* bit in the *CONTROL_STATUS* register is set and the FETs selected in the *FETTest()* register are opened by using FW override of the FETs. If the *[RECEN]* bit in *FETTest()* is set, then the FW override is removed and *[FETTST]* bit is cleared after 2 seconds. If *[RECEN]* is 0, then FW override of the selected FETs is never removed unless the device is reset or a fresh *START_FET_TEST* command is sent, which allows the override to be removed.

4.1.1.23 CAL_ENABLE: 0x002D

Toggles entry into/exit out of CALIBRATION mode.

4.1.1.24 RESET: 0x0041

Instructs the fuel gauge to perform a full reset. This command is only available when the fuel gauge is UNSEALED.

4.1.1.25 EXIT_CAL: 0x0080

Instructs the fuel gauge to execute raw measurement data collection for host-managed calibration of the fuel gauge.

4.1.1.26 ENTER_CAL: 0x0081

Instructs the fuel gauge to cease raw measurement data collection for host-managed calibration of the fuel gauge.

4.1.1.27 OFFSET_CAL: 0x0082

Instructs the fuel gauge to perform offset calibration when in CALIBRATION mode (*CONTROL_STATUS[CALMODE]* = 1).

4.1.2 AtRate(): 0x02 and 0x03

The *AtRate()* is a read-write function that reads or sets the load value used in computing load-compensated capacity in the Impedance Track algorithm when ***Load Mode*** = 0 or 1 and ***Load Select*** = 5. For configurations employing ***Load Mode*** = 0, the *AtRate()* register expects the host to write values in terms of mA. With a ***Load Mode*** of 1, the fuel gauge will expect units of mWh or cWh, depending on the setting for ***Design Energy Scale***. The *AtRate()* value is a signed integer with negative values interpreted as a discharge current value.

4.1.3 UnfilteredSOC(): 0x04 and 0x05

This read-only function returns an unsigned integer value of the predicted remaining battery capacity expressed as a percentage of *UnfilteredFCC()*, with a range of 0 to 100%.

4.1.4 Temperature(): 0x06 and 0x07

This read-only function returns an unsigned integer value of the battery temperature in units of 0.1°K measured by the fuel gauge and is used for fuel gauging algorithm. It reports either the *InternalTemperature()* or the external thermistor temperature depending on the setting of the ***[TEMPS]*** bit in ***Pack Configuration***.

4.1.5 Voltage(): 0x08 and 0x09

This read-only function returns an unsigned integer value of the measured cell-pack voltage in mV with a range of 0 to 6000 mV.

4.1.6 Flags(): 0x0A and 0x0B

This read-only function returns the contents of the gas-gauge status register, depicting the current operating status.

Table 4-4. Flags Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	RSVD	RSVD	BATHI	BATLOW	CHG_INH	RSVD	FC	RSVD
Low Byte	CHG_SUS	RSVD	RSVD	IMAX	CHG	SOC1	SOCF	DSG

High Byte

RSVD = Reserved

RSVD = Reserved

BATHI = Battery High bit indicating a high battery voltage condition. See the parameters in Section 5.3.6.4, *Battery High Set Voltage Threshold, Time, and Clear,* for threshold settings.

BATLOW = Battery Low bit indicating a low battery voltage condition. See the parameters in Section 5.3.6.3, *Battery Low Set Voltage Threshold, Time, and Clear,* for threshold settings.

CHG_INH = Charge Inhibit indicates that temperature is < ***T1 Temp*** or > ***T4 Temp*** while charging is not active. True when set.

RSVD = Reserved

FC = Full-charged state is detected. FC is set when charge termination is reached and ***FC Set %*** = –1 (see Section 2.6, *Charging and Charge Termination Indication*, for details) or State of Charge is larger than ***FC Set %*** and ***FC Set %*** is not –1. True when set.

RSVD = Reserved

Low Byte

CHG_SUS = Charge Suspend indicates that temperature is < ***T1 Temp*** or > ***T5 Temp*** while charging is active. True when set.

RSVD = Reserved

RSVD = Reserved

IMAX = Indicates that the computed *Imax()* value has changed enough to signal an interrupt. True when set.

CHG = (Fast) charging allowed. True when set.

SOC1 = State-of-Charge Threshold 1 ***(SOC1 Set)*** reached. True when set.

SOCF = State-of-Charge Threshold Final ***(SOCF Set %)*** reached. True when set.

DSG = Discharging detected. True when set.

4.1.7 NomAvailableCapacity(): 0x0C and 0x0D

This read-only command pair returns the uncompensated (less than C/20 load) battery capacity remaining. Units are mAh.

4.1.8 FullAvailableCapacity(): 0x0E and 0x0F

This read-only command pair returns the uncompensated (less than C/20 load) capacity of the battery when fully charged. Units are mAh. *FullAvailableCapacity()* is updated at regular intervals, as specified by the IT algorithm.

4.1.9 RemainingCapacity(): 0x10 and 0x11

This read-only command pair returns the compensated battery capacity remaining (*UnfilteredRM()*) when the ***[SmoothEn]*** bit in ***Pack Configuration C*** is cleared or filtered compensated battery capacity remaining (*FilteredRM()*) when ***[SmoothEn]*** bit is set. Units are mAh.

4.1.10 FullChargeCapacity(): 0x12 and 0x13

This read-only command pair returns the compensated capacity of fully charged battery (*UnfilteredFCC()*) when the ***[SmoothEn]*** bit in ***Pack Configuration C*** is cleared or filtered compensated capacity of fully charged battery (*FilteredFCC()*) when ***[SmoothEn]*** bit is set. Units are mAh. *FullChargeCapacity()* is updated at regular intervals, as specified by the IT algorithm.

4.1.11 AverageCurrent(): 0x14 and 0x15

This read-only command pair returns a signed integer value that is the average current flow through the sense resistor. It is updated every second in NORMAL mode and every 20 seconds in SLEEP and FULLSLEEP modes. Units are mA.

4.1.12 TimeToEmpty(): 0x16 and 0x17

This read-only function returns an unsigned integer value of the predicted remaining battery life at the present rate of discharge, in minutes. A value of 65,535 indicates battery is not being discharged.

4.1.13 FilteredFCC(): 0x18 and 0x19

This read-only command pair returns the modified full charge capacity based on the SOC smoothing algorithm. The value is modified during battery charge and is increased or decreased to achieve SOC convergence by the time end of charge is reached. For reporting of the capacity that can be extracted from a fully charged battery, the host should always refer to *UnfilteredFCC()*. Units are mAh. *FilteredFCC()* is updated at regular intervals, as specified by the IT algorithm.

4.1.14 SafetyStatus(): 0x1A and 0x1B

This read-only function returns the status of numerous firmware-based safety protections. Internal short, tab disconnect, overtemperature in charge, overtemperature in discharge, overvoltage, and undervoltage are all supported protections. In addition, a dedicated flag indicates a protector misconfiguration when a checksum mismatch is detected in ***Prot Checksum***.

Table 4-5. Safety Status Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
Low Byte	INV_PROT_CHKSUM	RSVD	ISD	TDD	OTC	OTD	OVP	UVP

High Byte

RSVD = Bits 7:0 are reserved.

Low Byte

INV_PROT_CHKSUM = Invalid protector checksum detected. True when set.

RSVD = Reserved

ISD = Internal Short condition is detected. True when set.

TDD = Tab Disconnect condition is detected. True when set.

OTC = Overtemperature in charge condition detected. True when set.

OTD = Overtemperature in discharge condition detected. True when set.

OVP = Overvoltage condition detected. True when set.

UVP = Undervoltage condition detected. True when set.

4.1.15 UnfilteredFCC(): 0x1C and 0x1D

This read-only command pair returns the compensated capacity of the battery when fully charged. Units are mAh. *UnFilteredFCC()* is updated at regular intervals, as specified by the IT algorithm.

4.1.16 Imax(): 0x1E and 0x1F

This read-only function returns the maximum discharge current that the battery can support for ***Max Current Pulse Duration*** time without prematurely dropping to empty (that is, 0%). It is useful for systems that need to dynamically scale applied load for extended runtime at low states of charge.

4.1.17 UnfilteredRM(): 0x20 and 0x21

This read-only command pair returns the compensated battery capacity remaining. Units are mAh.

4.1.18 FilteredRM(): 0x22 and 0x23

This read-only command pair returns the filtered, compensated battery capacity remaining. Units are mAh.

4.1.19 BTPSOC1Set(): 0x24 and 0x25

This read-write function is used to dynamically update the BTP threshold for detecting *RemainingCapacity()* decreasing below the programmed value in the discharge direction.

4.1.20 BTPSOC1Clear(): 0x26 and 0x27

This read-write function is used to dynamically update the BTP threshold for detecting *RemainingCapacity()* increasing above the programmed value in the charge direction.

4.1.21 InternalTemperature(): 0x28 and 0x29

This read-only function returns an unsigned integer value of the measured internal temperature of the device in units of 0.1°K as measured by the fuel gauge.

4.1.22 CycleCount(): 0x2A and 0x2B

This read-only function returns an unsigned integer value of the number of cycles the battery has experienced with a range of 0 to 65,535. One cycle occurs when accumulated discharge ≥ ***CC Threshold***.

4.1.23 StateOfCharge(): 0x2C and 0x2D

This read-only function returns an unsigned integer value of the predicted *RemainingCapacity()* expressed as a percentage of *FullChargeCapacity()*, with a range of 0 to 100%. The *StateOfCharge()* can be filtered or unfiltered since *RemainingCapacity()* and *FullChargeCapacity()* can be filtered or unfiltered based on ***[SmoothEn]*** bit selection in ***Pack Configuration C***.

4.1.24 StateOfHealth(): 0x2E and 0x2F

0x2E SOH percentage: this read-only function returns an unsigned integer value, expressed as a percentage of the ratio of predicted *FCC(25°C, SOH Load I)* over the *DesignCapacity()*. The *FCC(25°C, SOH Load I)* is the calculated full charge capacity at 25°C and the SOH current rate which is specified by *SOH Load I*. The range of the returned SOH percentage is 0x00 to 0x64, indicating 0 to 100%, correspondingly.

4.1.25 ChargingVoltage(): 0x30 and 0x31

This read-only function returns the recommended charging voltage output from the JEITA charging profile. It is updated automatically based on the present temperature range. It is cleared to 0 when the FC bit is set.

4.1.26 ChargingCurrent(): 0x32 and 0x33

This read-only function returns the recommended charging current output from the JEITA charging profile. It is updated automatically based on the present temperature range. It is cleared to 0 when the FC bit is set.

4.1.27 PassedCharge(): 0x34 and 0x35

This signed integer indicates the amount of charge passed through the sense resistor since the last IT simulation in mAh.

4.1.28 DOD0(): 0x36 and 0x37

This unsigned integer indicates the depth of discharge during the most recent OCV reading. The reported value is scaled to an integer value per DOD0 = DOD(OCV, Temperature) × 2^{14} and has a range of 0 to 16384.

4.1.29 SelfDischargeCurrent(): 0x38 and 0x39

This read-only command pair returns the signed integer value that estimates the battery self-discharge current.

4.2 Extended Data Commands

Extended commands offer additional functionality beyond the standard set of commands. They are used in the same manner; however, unlike standard commands, extended commands are not limited to 2-byte words. The number of command bytes for a given extended command ranges in size from single to multiple bytes, as specified in Table 4-6. For details on the SEALED and UNSEALED states, see Section 5.1.3, *Access Modes*.

Table 4-6. Extended Commands

Name	Command Code	Unit	SEALED Access[1][2]	UNSEALED Access[1][2]
PackConfiguration()	0x3A and 0x3B	Hex	R	R
DesignCapacity()	0x3C and 0x3D	mAh	R	R
DataFlashClass()[2]	0x3E	NA	NA	RW
DataFlashBlock()[2]	0x3F	NA	RW	RW
BlockData()/Authenticate()[3]	0x40 to 0x53	NA	RW	RW
BlockData()/AuthenticateCheckSum()[3]	0x54	NA	RW	RW
BlockData()	0x55 to 0x5F	NA	R	RW
BlockDataCheckSum()	0x60	NA	RW	RW
BlockDataControl()	0x61	NA	NA	RW
DODatEOC()	0x62 and 0x63	NA	R	R
Qstart()	0x64 and 0x65	mAh	R	R
FastQmax()	0x66 and 0x67	mAh	R	R
Reserved	0x68 to 0x6C	NA	R	R
ProtectorStatus()	0x6D	Hex	R	R
Reserved	0x6E and 0x6F	NA	R	R
SimultaneousCurrent()	0x70 and 0x71	mA	R	R
Reserved	0x72 and 0x73	NA	R	R
FETTest()	0x74 and 0x75	Hex	R	RW
AveragePower()	0x76 and 0x77	mW or cW	R	R
ProtectorState()	0x78	Hex	R	R
AN_COUNTER	0x79			
AN_CURRENT_LSB	0x7A			
AN_CURRENT_MSB	0x7B			
AN_VCELL_LSB	0x7C			
AN_VCELL_MSB	0x7D			
AN_TEMP_LSB	0x7E			
AN_TEMP_MSB	0x7F			

(1) SEALED and UNSEALED states are entered via commands to *Control()* 0x00 and 0x01.
(2) In SEALED mode, data flash cannot be accessed through commands 0x3E and 0x3F.
(3) The *BlockData()* command area shares functionality for accessing general data flash and for using Authentication. See Section 3.1, *Authentication*, for more details.

4.2.1 PackConfiguration(): 0x3A and 0x3B

SEALED and UNSEALED Access: This command returns the value stored in ***Pack Configuration*** and is expressed in hex value.

4.2.2 DesignCapacity(): 0x3C and 0x3D

SEALED and UNSEALED Access: This command returns the value stored in ***Design Capacity*** and is expressed in mAh. This is intended to be the theoretical or nominal capacity of a new pack, but has no bearing on the operation of the fuel gauge functionality.

4.2.3 DataFlashClass(): 0x3E

This command sets the data flash class to be accessed. The subclass ID to be accessed must be entered in hexadecimal.

SEALED Access: This command is not available in SEALED mode.

4.2.4 DataFlashBlock(): 0x3F

UNSEALED Access: This command sets the data flash block to be accessed. When 0x00 is written to *BlockDataControl()*, *DataFlashBlock()* holds the block number of the data flash to be read or written. Example: writing a 0x00 to *DataFlashBlock()* specifies access to the first 32-byte block and a 0x01 specifies access to the second 32-byte block, and so on.

SEALED Access: This command directs which data flash block is accessed by the *BlockData()* command. Writing a 0x00 to *DataFlashBlock()* specifies the *BlockData()* command transfers authentication data. Issuing a 0x01 or 0x02 instructs the *BlockData()* command to transfer ***Manufacturer Info Block A or B***, respectively.

4.2.5 BlockData(): 0x40 Through 0x5F

This command range is used to transfer data for data flash class access. This command range is the 32-byte data block used to access ***Manufacturer Info Block A*** or ***B***. ***Manufacturer Info Block A*** is read-only for the SEALED access. UNSEALED access is read-write.

4.2.6 BlockDataCheckSum(): 0x60

The host system must write this value to inform the device that new data is ready for programming into the specified data flash class and block.

UNSEALED Access: This byte contains the checksum on the 32 bytes of block data read from or written to data flash. The least-significant byte of the sum of the data bytes written must be complemented ([255 – x], for x the 8-bit summation of the *BlockData()* (0x40 to 0x5F) on a byte-by-byte basis) before being written to 0x60.

SEALED Access: This byte contains the checksum for the 32 bytes of block data written to ***Manufacturer Info Block A***. The least-significant byte of the sum of the data bytes written must be complemented ([255 – x], for x the 8-bit summation of the *BlockData()* (0x40 to 0x5F) on a byte-by-byte basis) before being written to 0x60.

4.2.7 BlockDataControl(): 0x61

UNSEALED Access: This command controls data flash access mode. The value determines the data flash to be accessed. Writing 0x00 to this command enables *BlockData()* to access general data flash.

SEALED Access: This command is not available in SEALED mode.

4.2.8 DODatEOC(): 0x62 and 0x63

UNSEALED and SEALED Access: This command reports DOD at the end of charge (EOC).

4.2.9 Qstart(): 0x64 and 0x65

UNSEALED and SEALED Access: This command reports Qstart.

4.2.10 FastQmax(): 0x66 and 0x67

UNSEALED and SEALED Access: This command reports Fast Qmax.

4.2.11 Reserved – 0x68 to 0x6C

4.2.12 ProtectorStatus(): 0x6D

UNSEALED and SEALED Access: This block returns protector status register AFESTAT1.

Table 4-7. Protector Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHG_OFF	DSG_OFF	CVM	UVP	OVP	SCD	OCD	OCC

CHG_OFF = Firmware override to disable charge FET:
1 = Charge FET Off
0 = Charge FET On

DSG_OFF = Firmware override to disable discharge FET:
1 = Discharge FET Off
0 = Discharge FET On

CVM = Cell voltage monitor threshold:
1 = Cell voltage monitor threshold detected
0 = Cell voltage monitor threshold not detected

UVP = Undervoltage protection fault:
1 = Undervoltage protection fault detected
0 = Undervoltage protection fault not detected

OVP = Overvoltage protection fault:
1 = Overvoltage protection fault detected
0 = Overvoltage protection fault not detected

SCD = Short-circuit discharge fault:
1 = Short-circuit discharge fault detected
0 = Short-circuit discharge fault not detected

OCD = Overcurrent discharge fault:
1 = Overcurrent discharge fault detected
0 = Overcurrent discharge fault not detected

OCC = Overcurrent charge fault:
1 = Overcurrent charge fault detected
0 = Overcurrent charge fault not detected

4.2.13 Reserved – 0x6E and 0x6F

4.2.14 SimultaneousCurrent(): 0x70 and 0x71

UNSEALED and SEALED Access: This is the current measured across the faster 125-ms conversion window that is synchronized with the *Voltage()* measurement. It allows a more instantaneous reporting of measured current vs traditional 1-second conversions in *AverageCurrent()*.

4.2.15 Reserved – 0x72 and 0x73

4.2.16 FETTest(): 0x74 and 0x75

UNSEALED Access: This command sets up the data for the START_FET_TEST command and provides FET test status.

Table 4-8. FETTest() Register

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	STATUS							
Low Byte	CHKSUM		RSVD	RSVD	RSVD	RECEN	CHG	DSG

High Byte

STATUS = The STATUS field indicates the status of the FET Test:
0xFF = Checksum error, the *START_FET_TEST* command was unsuccessful.
2-3 = *START_FET_TEST* command was successful with RECEN = 1.
FW is currently waiting for 2 seconds to elapse to remove the FW override applied to the selected FETs.
This field works like a countdown timer. It starts at 3 and counts down by 1 each second. When it reaches 1, the FW override is removed and the selected FETs return to hardware control again.
1 = Either *START_FET_TEST* command was executed with RECEN = 1 and the selected FETs return to hardware control,
or
START_FET_TEST command was executed with RECEN = 0 (selected FETs are in FW override state and are opened).

Low Byte

CHG = If set, this results in the CHG FET being opened on executing the *START_FET_TEST* command.

DSG = If set, this results in the DSG FET being opened on executing the *START_FET_TEST* command.

RECEN = Recovery Enable. Enables recovery of the FETs 2 seconds after the *START_FET_TEST* command is executed.
If this bit is set, the selected FETs are left under hardware control after 2 seconds.
If this bit is not set, the selected FETs will remain in FW override state and remain opened.

CHKSUM = Checksum should be set to CHG + DSG + RECEN.
If the checksum does not match this value, the *START_FET_TEST* command will not have any effect and this will be indicated by setting STATUS = 0xFF.

SEALED Access: This command is not available in SEALED mode.

4.2.17 AveragePower(): 0x76 and 0x77

UNSEALED and SEALED Access: This read-word function returns an unsigned integer value of the average power of the current discharge. It is negative during discharge and positive during charge. A value of 0 indicates that the battery is not being discharged. The value is reported in units of mW (***Design Energy Scale*** = 1) or cW (***Design Energy Scale*** = 10).

4.2.18 ProtectorState(): 0x78

UNSEALED and SEALED Access: This block returns protector state machine register AFESTATE.

Table 4-9. Protector State Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSVD	RSVD	SHUTDWNW	OVP	OCD_SCD	OCC	NORMAL	HOLD

RSVD = Bits 6 and 7 are reserved.

SHUTDWNW = Protector State Machine in Shutdown Wait state
1 = Shutdown Wait State
0 = Not in Shutdown Wait State

OVP = Protector State Machine in OVP state
1 = OVP State
0 = Not in OVP State

OCD_SCD = Protector State Machine in OCD or SCD state
1 = OCD or SCD State
0 = Not in OCD or SCD State

OCC = Protector State Machine in OCC state
1 = OCC State
0 = Not in OCC State

NORMAL = Protector State Machine in NORMAL state
1 = NORMAL state
0 = Not in NORMAL state

HOLD = Protector State Machine in HOLD state
1 = HOLD state
0 = Not in HOLD state

4.2.19 AN_COUNTER: 0x79

UNSEALED and SEALED Access: This command reports AN_COUNTER.

4.2.20 AN_CURRENT_LSB: 0x7A

UNSEALED and SEALED Access: This command reports AN_CURRENT_LSB.

4.2.21 AN_CURRENT_MSB: 0x7B

UNSEALED and SEALED Access: This command reports AN_CURRENT_MSB.

4.2.22 AN_VCELL_LSB: 0x7C

UNSEALED and SEALED Access: This command reports AN_VCELL_LSB.

4.2.23 AN_VCELL_MSB: 0x7D

UNSEALED and SEALED Access: This command reports AN_VCELL_MSB.

4.2.24 AN_TEMP_LSB: 0x7E

UNSEALED and SEALED Access: This command reports AN_TEMP_LSB.

4.2.25 AN_TEMP_MSB: 0x7F

UNSEALED and SEALED Access: This command reports AN_TEMP_MSB.

Data Flash Summary

5.1 Data Flash Interface

5.1.1 Accessing The Data Flash

The data flash is a non-volatile memory that contains initialization, default, cell status, calibration, configuration, and user information. The data flash can be accessed in several different ways, depending in which mode the bq27742-G1 fuel gauge is operating and what data is being accessed.

Commonly accessed data flash memory locations, frequently read by a system, are conveniently accessed through specific instructions, already described in Chapter 4 *Data Commands*. These commands are available when the fuel gauge is either in UNSEALED or SEALED modes.

Most data flash locations, however, are only accessible in UNSEALED mode by use of the evaluation software or by data flash block transfers. These locations must be optimized and/or fixed during the development and manufacture processes. They become part of a golden image file and can then be written to multiple battery packs. Once established, the values generally remain unchanged during end-equipment operation.

To access data flash locations individually, the block containing the desired data flash location(s) must be transferred to the command register locations, where they can be read to the system or changed directly. This is accomplished by sending the set-up command *BlockDataControl()* (0x61) with data 0x00. Up to 32 bytes of data can be read directly from the *BlockData()* (0x40 to 0x5F), externally altered, then rewritten to the *BlockData()* command space. Alternatively, specific locations can be read, altered, and rewritten if their corresponding offsets are used to index into the *BlockData()* command space. Finally, the data residing in the command space is transferred to data flash, once the correct checksum for the whole block is written to *BlockDataChecksum()* (0x60).

Occasionally, a data flash class is larger than the 32-byte block size. In this case, the *DataFlashBlock()* command designates in which 32-byte block the desired locations reside. The correct command address is then given by 0x40 + offset *modulo* 32. For example, to access ***Term V Delta*** in the *IT Cfg* class, *DataFlashClass()* is issued 80 (0x50) to set the class. Because the offset is 66, it must reside in the third 32-byte block; therefore, *DataFlashBlock()* is issued 0x02 to set the block offset, and the offset used to index into the *BlockData()* memory area is 0x40 + 66 *modulo* 32 = 0x40 + 0x02 = 0x42.

Reading and writing subclass data are block operations up to 32 bytes in length. If during a write the data length exceeds the maximum block size, then the data is ignored.

None of the data written to memory are bounded by the fuel gauge—the values are not rejected by the fuel gauge. Writing an incorrect value may result in hardware failure due to firmware program interpretation of the invalid data. The written data is persistent, so a power-on reset does not resolve the fault.

5.1.2 Manufacturer Information Blocks

The fuel gauge contains 64 bytes of user programmable data flash storage: ***Manufacturer Info Block A*** and ***Manufacturer Info Block B***. The method for accessing these memory locations is slightly different, depending on whether the device is in UNSEALED or SEALED modes.

When in UNSEALED mode and when 0x00 has been written to *BlockDataControl()*, accessing the Manufacturer Info Blocks is identical to accessing general data flash locations. First, a *DataFlashClass()* command sets the subclass, then a *DataFlashBlock()* command sets the offset for the first data flash address within the subclass. The *BlockData()* command codes contain the referenced data flash data. When writing the data flash, a checksum is expected to be received by *BlockDataChecksum()*. Only when the checksum is received and verified is the data actually written to data flash.

As an example, the data flash location for ***Manufacturer Info Block B*** is defined as having a Subclass = 58 and an Offset = 32 through 63 (32-byte block). The specification of Class = System Data is not needed to address ***Manufacturer Info Block B***, but is used instead for grouping purposes when viewing data flash info in the evaluation software.

When in SEALED mode or when *BlockDataControl()* does not contain 0x00, data flash is no longer available in the manner used in UNSEALED mode. Rather than issuing subclass information, a designated Manufacturer Information Block is selected with the *DataFlashBlock()* command. Issuing a 0x01 or 0x02 with this command causes the corresponding information block (A or B, respectively) to be transferred to the command space 0x40 through 0x5F for editing or reading by the system. Upon successful writing of checksum information to *BlockDataChecksum()*, the modified block is returned to data flash.

NOTE: ***Manufacturer Info Block A*** is read-only when in SEALED mode.

5.1.3 Access Modes

The fuel gauge provides three security modes (FULL ACCESS, UNSEALED, and SEALED) that control data flash access permissions according to Table 5-1. Data Flash column refers to those data flash locations that are accessible to the user. Manufacturer Information column refers to the two 32-byte blocks.

Table 5-1. Data Flash Access

Security Mode	Data Flash	Manufacturer Information
FULL ACCESS	RW	RW
UNSEALED	RW	RW
SEALED	None	R (A); RW (B)

Although FULL ACCESS and UNSEALED modes appear identical, only the FULL ACCESS mode allows the fuel gauge to write access-mode transition keys stored in the Security class.

5.1.4 Sealing or Unsealing Data Flash

The fuel gauge implements a key-access scheme to transition between SEALED, UNSEALED, and FULL-ACCESS modes. Each transition requires that a unique set of two keys be sent to the fuel gauge via the *Control()* command. The keys must be sent consecutively, with no other data being written to the *Control()* register in between. To avoid conflict, the keys must be different from the codes presented in the CNTL DATA column of Table 4-2, *Control() Subcommands*.

When in SEALED mode the *CONTROL_STATUS [SS]* bit is set, but when the Unseal Keys are correctly received by the fuel gauge, the *[SS]* bit is cleared. When the Full-Access Keys are correctly received, the *CONTROL_STATUS [FAS]* bit is cleared.

Both ***Unseal Key*** and ***Full-Access Key*** have two words and are stored in data flash. The first word is Key 0 and the second word is Key 1. The order of the keys sent to fuel gauge are Key 1 followed by Key 0. The order of the bytes for each key entered through the *Control()* command is the reverse of what is read from the part. For an example, if the Unseal Key is 0x56781234, key 1 is 0x1234 and key 0 is 0x5678. Then *Control()* must supply 0x3412 and 0x7856 to unseal the part. The ***Unseal Key*** and the ***Full-Access Key*** can only be updated when in FULL-ACCESS mode.

5.2 Data Flash Summary Tables

Table 5-3 through Table 5-9 summarize the data flash locations available to the user, including their default, minimum, and maximum values.

Table 5-2. Data Type Decoder

Type	Min Value	Max Value
F4	$\pm 9.8603 \times 10^{-39}$	$\pm 5.707267 \times 10^{37}$
H1	0x00	0xFF
H2	0x00	0xFFFF
H4	0x00	0xFFFF FFFF
I1	–128	127
I2	–32768	32767
I4	−2,147,483,648	2,147,483,647
Sx	1-byte string	X-byte string
U1	0	255
U2	0	65535
U4	0	4,294,967,295

Table 5-3. Data Flash Summary—Configuration Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
2	Safety	0	OV Prot Threshold	I2	4200	4600	4390	mV
		2	OV Prot Delay	U1	0	5	1	s
		3	OV Prot Recovery	I2	4100	4500	4290	mV
		5	UV Prot Threshold	I2	2300	3100	2800	mV
		7	UV Prot Delay	U1	0	5	1	s
		8	UV Prot Recovery	I2	2400	3200	2900	mV
		10	Body Diode Threshold	I2	0	100	60	mA
		12	OT Chg	I2	0	1200	550	0.1°C
		14	OT Chg Time	U1	0	60	5	s
		15	OT Chg Recovery	I2	0	1200	500	0.1°C
		17	OT Dsg	I2	0	1200	600	0.1°C
		19	OT Dsg Time	U1	0	60	5	s
		20	OT Dsg Recovery	I2	0	1200	550	0.1°C
34	Charge	0	Charging Voltage	I2	4000	5000	4350	mV
36	Charge Termination	0	Taper Current	I2	0	1000	100	mA
		2	Min Taper Capacity	I2	0	1000	25	0.004 mAh
		4	Taper Voltage	I2	0	1000	100	mV
		6	Current Taper Window	U1	0	60	40	s
		7	TCA Set %	I1	–1	100	-1	Percent
		8	TCA Clear %	I1	–1	100	98	Percent
		9	FC Set %	I1	–1	100	–1	Percent
		10	FC Clear %	I1	–1	100	98	Percent
		11	DODatEOC Delta T	I2	0	1000	50	0.1°C

Table 5-3. Data Flash Summary—Configuration Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
39	JEITA	0	T1 Temp	I1	–128	127	0	°C
		1	T2 Temp	I1	–128	127	10	°C
		2	T3 Temp	I1	–128	127	45	°C
		3	T4 Temp	I1	–128	127	50	°C
		4	T5 Temp	I1	–128	127	60	°C
		5	Temp Hys	I1	–128	127	1	°C
		6	T1-T2 Chg Voltage	I2	0	4600	4350	mV
		8	T2-T3 Chg Voltage	I2	0	4600	4350	mV
		10	T3-T4 Chg Voltage	I2	0	4600	4300	mV
		12	T4-T5 Chg Voltage	I2	0	4600	4250	mV
		14	T1-T2 Chg Current	U1	0	100	50	Percent
		15	T2-T3 Chg Current	U1	0	100	80	Percent
		16	T3-T4 Chg Current	U1	0	100	80	Percent
		17	T4-T5 Chg Current	U1	0	100	80	Percent
48	Data	0	Design Voltage	I2	2000	5000	3800	mV
		8	Cycle Count	U2	0	65535	0	Count
		10	CC Threshold	I2	100	32767	900	mAh
		12	Design Capacity	I2	0	14500	1000	mAh
		14	Design Energy	I2	0	32767	3800	mWh
		16	SOH Load I	I2	–32767	0	–400	mA
		18	TDD SOH Percent	U1	0	100	80	Percent
		19	ISD Current	U2	1	32767	10	Hour Rate
		21	ISD I Filter	U1	0	255	127	Count
		22	Min ISD Time	U1	0	255	7	Hour
		23	Design Energy Scale	U1	1	10	1	Number
49	Discharge	0	SOC1 Set Threshold	U2	0	65535	150	mAh
		2	SOC1 Clear Threshold	U2	0	65535	175	mAh
		4	SOCF Set Threshold	U2	0	65535	75	mAh
		6	SOCF Clear Threshold	U2	0	65535	100	mAh
		8	BL Set Volt Threshold	I2	0	5000	2500	mV
		10	BL Set Volt Time	U1	0	60	2	s
		11	BL Clear Volt Threshold	I2	0	5000	2600	mV
		13	BH Set Volt Threshold	I2	0	5000	4500	mV
		15	BH Volt Time	U1	0	60	2	s
		16	BH Clear Volt Threshold	I2	0	5000	4400	mV
56	Manufacturer Data	0	Pack Lot Code	H2	0x00	0xFFFF	0x00	hex
		2	PCB Lot Code	H2	0x00	0xFFFF	0x00	hex
		4	Firmware Version	H2	0x00	0xFFFF	0x00	hex
		6	Hardware Revision	H2	0x00	0xFFFF	0x00	hex
		8	Cell Revision	H2	0x00	0xFFFF	0x00	hex
		10	DF Config Version	H2	0x00	0xFFFF	0x00	hex

Table 5-3. Data Flash Summary—Configuration Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
57	Integrity Data	6	All DF Checksum	H2	0x00	0x7FFF	0x00	Number
		8	Static Chem DF Checksum	H2	0x00	0x7FFF	0x7C23	Number
		10	Static DF Checksum	H2	0x00	0x7FFF	0x00	Number
		12	Prot Checksum	H2	0x00	0x7FFF	0x0011	Number
59	Lifetime Data	0	Lifetime Max Temp	I2	-600	1400	0	0.1°C
		2	Lifetime Min Temp	I2	–600	1400	500	0.1°C
		4	Lifetime Max Pack Voltage	I2	0	32767	2800	mV
		6	Lifetime Min Pack Voltage	I2	0	32767	5000	mV
		8	Lifetime Max Chg Current	I2	–32767	32767	0	mA
		10	Lifetime Max Dsg Current	I2	–32767	32767	0	mA
60	Lifetime Temp Samples	0	LT Flash Cnt	U2	0	32767	0	Count
		2	LT AFE Status	H1	0x00	0xFF	0x00	hex
64	Registers	0	Pack Configuration	H2	0x00	0xFFFF	0x097F	flags
		2	Pack Configuration B	H1	0x00	0xFF	0xA7	flags
		3	Pack Configuration C	H1	0x00	0xFF	0xB9	flags
		4	Pack Configuration D	H1	0x00	0xFF	0x83	flags
		5	Prot OC Config	H1	0x00	0xFF	0x0A	flags
		6	Prot OV Config	H1	0x00	0xFF	0x07	flags
66	Lifetime Resolution	0	LT Temp Res	U1	0	255	10	Num
		1	LT V Res	U1	0	255	25	Num
		2	LT Cur Res	U1	0	255	100	Num
		3	LT Update Time	U2	0	65535	60	Num
68	Power	0	Flash Update OK Voltage	I2	0	5000	2800	mV
		2	Sleep Current	I2	0	100	15	mA
		10	Shutdown V	I2	0	5000	0	mV
		12	FS Wait	U1	0	255	0	s

Table 5-4. Data Flash Summary—System Data Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
58	Manufacturer Info	0 through 31	Block A 0 through 31	H1	0x00	0xFF	0x00	
		32 through 63	Block B 0 through 31	H1	0x00	0xFF	0x00	

Table 5-5. Data Flash Summary—Gas (Fuel) Gauging Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	0	Load Select	U1	0	6	1	Number
		1	Load Mode	U1	0	1	1	Number
		17	Max Res Factor	U1	0	255	15	num
		18	Min Res Factor	U1	0	255	5	num
		20	Ra Filter	U2	0	1000	800	num
		22	Res V Drop	I2	0	32767	50	mV
		39	Fast Qmax Start DOD %	U1	0	100	92	%
		40	Fast Qmax End DOD %	U1	0	100	96	%
		41	Fast Qmax Start Volt Delta	I2	0	4200	200	mV
		43	Fast Qmax Current Threshold	U2	0	1000	4	HourRate
		61	Qmax Capacity Err	U1	0	100	15	0.10%
		62	Max Qmax Change	U1	0	255	30	%
		64	Terminate Voltage	I2	2800	3700	3000	mV
		66	Term V Delta	I2	0	4200	200	mV
		69	ResRelax Time	U2	0	65535	500	s
		73	User Rate-mA	I2	0	32767	0	mA
		75	User Rate-Pwr	I2	0	32767	0	cW
		77	Reserve Cap-mAh	I2	0	14500	0	mAh
		84	Max DeltaV	I2	0	32767	200	mV
		86	Min DeltaV	I2	0	32767	0	mV
		88	Max Sim Rate	U1	0	255	1	HourRate
80 (continued)	IT Cfg (continued)	89	Min Sim Rate	U1	0	255	20	HourRate
		90	Ra Max Delta	I2	0	32767	54	mΩ
		92	Trace Resistance	I2	0	32767	0	mΩ
		94	Downstream Resistance	I2	0	32767	0	mΩ
		96	Qmax Max Delta %	U1	0	100	5	mAh
		97	Qmax Bound %	U1	0	255	130	mAh
		98	DeltaV Max Delta	U2	0	65535	10	mV
		100	Max Res Scale	U2	0	32767	5000	Num
		102	Min Res Scale	U2	0	32767	200	Num
		104	Fast Scale Start SOC	U1	0	100	10	%
		105	Fast Scale Load Select	U1	0	6	3	Number
		106	Charge Hys V Shift	I2	0	2000	40	mV
		108	RaScl OCV Rst Temp Thresh	U1	0	127	15	°C
		109	Max Allowed Current	I2	0	32767	8500	mA
		111	Max Current Pulse Duration	U1	0	255	10	s
		112	Max Current Interrupt Step	I2	–32768	32767	500	mA
		114	Relax Smooth Time	U2	1	65535	1000	s

Table 5-5. Data Flash Summary—Gas (Fuel) Gauging Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value Min	Value Max	Value Default	Unit
81	Current Thresholds	0	Dsg Current Threshold	I2	0	2000	60	mA
		2	Chg Current Threshold	I2	0	2000	75	mA
		4	Quit Current	I2	0	1000	40	mA
		6	Dsg Relax Time	U2	0	65535	60	s
		8	Chg Relax Time	U1	0	255	60	s
		9	Quit Relax Time	U1	0	255	1	s
		10	Max IR Correct	I2	0	1000	400	mV
82	State	0	Qmax Cell 0	I2	0	14500	1000	mAh
		2	Update Status	H1	0x0	0x6	0x0	num
		3	V at Chg Term	I2	0	5000	4350	mV
		5	Avg I Last Run	I2	–32768	0	–299	mA
		7	Avg P Last Run	I2	–32768	0	–1131	mA
		9	Delta Voltage	I2	0	32767	2	mV
		11	T Rise	I2	0	32767	50	Num
		13	T Time Constant	I2	0	32767	1000	Num

Table 5-6. Data Flash Summary—OCV Table Class

Subclass ID	Subclass	Offset	Name	Data Type	Value Min	Value Max	Value Default	Unit
83	OCVa Table	0	Chem ID	H2	0x00	0xFFFF	0x0354	flags

Table 5-7. Data Flash Summary—Ra Table Class

Subclass ID	Subclass	Offset	Name	Data Type	Value Min	Value Max	Value Default	Unit
88	R_a0	0	Cell0 R_a flag	H2	0xFF55	0xFF55	0xFF55	
		2	Cell0 R_a 0	I2	0	32767	272	2^{-10} Ω
		4	Cell0 R_a 1	I2	0	32767	316	2^{-10} Ω
		6	Cell0 R_a 2	I2	0	32767	374	2^{-10} Ω
		8	Cell0 R_a 3	I2	0	32767	507	2^{-10} Ω
		10	Cell0 R_a 4	I2	0	32767	360	2^{-10} Ω
		12	Cell0 R_a 5	I2	0	32767	330	2^{-10} Ω
		14	Cell0 R_a 6	I2	0	32767	389	2^{-10} Ω
		16	Cell0 R_a 7	I2	0	32767	345	2^{-10} Ω
		18	Cell0 R_a 8	I2	0	32767	352	2^{-10} Ω
		20	Cell0 R_a 9	I2	0	32767	367	2^{-10} Ω
		22	Cell0 R_a 10	I2	0	32767	374	2^{-10} Ω
		24	Cell0 R_a 11	I2	0	32767	397	2^{-10} Ω
		26	Cell0 R_a 12	I2	0	32767	455	2^{-10} Ω
		28	Cell0 R_a 13	I2	0	32767	808	2^{-10} Ω
		30	Cell0 R_a 14	I2	0	32767	1182	2^{-10} Ω

Table 5-7. Data Flash Summary—Ra Table Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
89	R_a0x	0	xCell0 R_a flag	H2	0xFFFF	0xFFFF	0xFFFF	
		2	Cell0 R_a 0	I2	0	32767	272	2^{-10} Ω
		4	Cell0 R_a 1	I2	0	32767	316	2^{-10} Ω
		6	Cell0 R_a 2	I2	0	32767	374	2^{-10} Ω
		8	Cell0 R_a 3	I2	0	32767	507	2^{-10} Ω
		10	Cell0 R_a 4	I2	0	32767	360	2^{-10} Ω
		12	Cell0 R_a 5	I2	0	32767	330	2^{-10} Ω
		14	Cell0 R_a 6	I2	0	32767	389	2^{-10} Ω
		16	Cell0 R_a 7	I2	0	32767	345	2^{-10} Ω
		18	Cell0 R_a 8	I2	0	32767	352	2^{-10} Ω
		20	Cell0 R_a 9	I2	0	32767	367	2^{-10} Ω
		22	Cell0 R_a 10	I2	0	32767	374	2^{-10} Ω
		24	Cell0 R_a 11	I2	0	32767	397	2^{-10} Ω
		26	Cell0 R_a 12	I2	0	32767	455	2^{-10} Ω
		28	Cell0 R_a 13	I2	0	32767	808	2^{-10} Ω
		30	Cell0 R_a 14	I2	0	32767	1182	2^{-10} Ω

Table 5-8. Data Flash Summary—Calibration

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	0	CC Gain	F4	1.00E–01	4.00E+01	0.9536	Number
		4	CC Delta	F4	2.98E+04	1.19E+06	1119000	Number
		8	CC Offset	I2	–32768	32767	1432	mA
		10	Board Offset	I1	–128	127	88	µA
		11	Int Temp Offset	I1	–128	127	0	Number
		12	Ext Temp Offset	I1	–128	127	0	Number
		13	Pack V Offset	I1	–128	127	0	Number
107	Current	0	Filter	U1	0	255	239	Number
		1	Deadband	U1	0	255	5	mA
		2	CC Deadband	U1	0	255	34	294 nV

Table 5-9. Data Flash Summary—Security

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
112	Codes	0	Sealed to Unsealed	H4	0x00	0xFFFF FFFF	0x3672 0414	
		4	Unsealed to Full	H4	0x00	0xFFFF FFFF	0xFFFF FFFF	
		8	Authen Key3	H4	0x00	0xFFFF FFFF	0x0123 4567	
		12	Authen Key2	H4	0x00	0xFFFF FFFF	0x89AB CDEF	
		16	Authen Key1	H4	0x00	0xFFFF FFFF	0xFEDC BA98	
		20	Authen Key0	H4	0x00	0xFFFF FFFF	0x7654 3210	

5.3 Configuration Class

5.3.1 Safety Subclass

5.3.1.1 Overvoltage Protection Threshold, Delay Time, and Recovery

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
2	Safety	0	OV Prot Threshold	I2	4200	4600	4390	mV
		2	OV Prot Delay	U1	0	5	1	s
		3	OV Prot Recovery	I2	4100	4500	4290	mV

OV Prot Threshold:

When the pack voltage measured by *Voltage()* rises above the overvoltage (OV Prot Threshold) then the *SafetyStatus() [OVP]* bit is set and the CHG FET disabled after ***OV Prot Delay***.

The FET is re-enabled and the flag cleared after *Voltage()* drops back below ***OV Prot Recovery***.

OV Prot Delay:

See ***OV Prot Threshold***. This is a buffer time allotted for an overvoltage condition. The timer starts every time that *Voltage()* is greater than ***OV Prot Threshold***. When the timer expires, the fuel gauge forces an *[OVP]* in *SafetyStatus()*. Setting the ***OV Prot Delay*** to 0 disables this function.

OV Prot Recovery:

OV Prot Recovery is the voltage at which the battery recovers from an ***overvoltage*** fault.

5.3.1.2 Undervoltage Protection Threshold, Delay Time, and Recovery

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
2	Safety	5	UV Prot Threshold	I2	2300	3100	2800	mV
		7	UV Prot Delay	U1	0	5	1	s
		8	UV Prot Recovery	I2	2400	3200	2900	mV

UV Prot Threshold:

When the pack voltage measured by *Voltage()* falls below above the undervoltage (UV Prot Threshold) then the *SafetyStatus() [UVP]* bit is set and the DSG FET disabled after ***UV Prot Delay***.

The FET is re-enabled and the flag cleared after *Voltage()* rises back above ***UV Prot Recovery***.

UV Prot Delay:

See ***UV Prot Threshold***. This is a buffer time allotted for an undervoltage condition. The timer starts every time that *Voltage()* is less than ***UV Prot Threshold***. When the timer expires, the fuel gauge forces an *[UVP]* in *SafetyStatus()*. Setting the ***UV Prot Delay*** to 0 disables this function.

UV Prot Recovery:

UV Prot Recovery is the voltage at which the battery recovers from an ***undervoltage*** fault.

5.3.1.3 Body Diode Current Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
2	Safety	10	Body Diode Threshold	I2	0	100	60	mA

Body Diode Threshold:

When the pack current measured by *AverageCurrent()* rises above the Body Diode Threshold in the presence of a pre-existing FET disable condition (safety fault or charge termination, inhibit, suspend), the FET will be temporarily re-enabled to relieve thermal strain on the body diode.

The FET returns to disabled state if the current drops back below the Body Diode Threshold with the original safety conditions still detected.

5.3.1.4 Charging Overtemperature Threshold, Delay Time, and Recovery

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
2	Safety	12	OT Chg	I2	0	1200	550	0.1°C
		14	OT Chg Time	U1	0	60	5	s
		15	OT Chg Recovery	I2	0	1200	500	0.1°C

OT Chg:

When the pack temperature measured by *Temperature()* rises to or above the overtemperature charge (***OT Chg***) threshold while charging (***Current*** > ***Chg Current Threshold***), then the *Flags() [OTC]* bit is set after ***OT Chg Time***. If the OTC condition clears prior to the expiration of the ***OT Chg Time*** timer, then the *Flags() [OTC]* bit is not set.

This setting depends on the environment temperature and the battery specification. Verify battery specification allows temperatures up to this setting during a charge and that this setting is sufficient for the application. The default is 55°C, sufficient for most Li-Ion applications.

OT Chg Time:

See ***OT Chg***. This is a buffer time allotted for overtemperature in the charge direction condition. The timer starts every time that *Temperature()* is greater than ***OT Chg*** and during a charge. When the timer expires, the fuel gauge forces an *[OTC]* in *Flags()*. Setting the ***OT Chg Time*** to 0 disables this function.

Default is set to 2 seconds, sufficient for most applications. Temperature is normally a slow-varying condition that does not need high-speed triggering. It must be set long enough to prevent false triggering of the *Flags() [OTC]* bit, but short enough to prevent damage to the battery pack.

OT Chg Recovery:

OT Chg Recovery is the temperature when the battery recovers from an ***OT Chg*** fault. This is the only recovery method for an ***OT Chg*** fault.

The default is 50°C which is 5°C lower than ***OT Chg***.

5.3.1.5 Discharging Overtemperature Threshold, Delay Time, and Recovery

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
2	Safety	17	OT Dsg	I2	0	1200	600	0.1°C
		19	OT Dsg Time	U1	0	60	5	s
		20	OT Dsg Recovery	I2	0	1200	550	0.1°C

OT Dsg:

When the pack temperature measured by *Temperature()* rises to or above this threshold while discharging (***Current*** < (–)***Dsg Current Threshold***), then the *Flags() [OTD]* bit is set after ***OT Dsg Time***. If the OTD condition clears prior to the expiration of the ***OT Dsg Time*** timer, then the *[OTD]* bit is not set. If the condition does not clear, then the *[OTD]* bit is set.

This setting depends on the environment temperature and the battery specification. Verify that the battery specification allows temperatures up to this setting while discharging, and verify that these setting are sufficient for the application temperature. The default is 60°C which is sufficient for most Li-Ion applications. The default ***OT Dsg*** is higher than the default ***OT Chg*** because Li-Ion can handle a higher temperature in the discharge direction than in the charge direction.

OT Dsg Time:

See ***OT Dsg***. This is a buffer time allotted for overtemperature in the discharge direction condition. The timer starts every time that *Temperature()* measured is greater than ***OT Dsg*** during a discharge. When the timer expires, then the fuel gauge forces the *Flags() [OTD]* bit to be set. Setting the ***OT Dsg Time*** to 0 disables this feature.

This is normally set to 2 seconds which is sufficient for most applications. Temperature is normally a slow-acting condition that does not need high-speed triggering. Set ***OT Dsg Time*** long enough to prevent false triggering of the *[OTD]* bit in *Flags()*, but short enough to prevent damage to the battery pack.

OT Dsg Recovery:

OT Dsg Recovery is the temperature at which the battery recovers from an ***OT Dsg*** fault. This is the only recovery method for an ***OT Dsg*** fault.

The default is 55°C which is 5°C lower than ***OT Dsg***.

5.3.2 Charge Subclass

5.3.2.1 Charging Voltage

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
34	Charge	0	Charging Voltage	I2	4000	5000	4350	mV

The fuel gauge uses this value along with ***Taper Voltage*** to detect charge termination. During Primary Charge Termination detection, one of the three requirements is that ***Voltage*** must be above (***Charging Voltage*** – ***Taper Voltage***) for the gauge to start trying to qualify a termination. This value depends on the charger that is expected to be used for the battery pack. The default is 4.35 V.

5.3.3 Charge Termination Subclass

5.3.3.1 Taper Current, Minimum Taper Capacity, Taper Voltage, and Current Taper Window

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
36	Charge Termination	0	Taper Current	I2	0	1000	100	mA
		2	Min Taper Capacity	I2	0	1000	25	mAh
		4	Taper Voltage	I2	0	1000	100	mV
		6	Current Taper Window	U1	0	60	40	s

Taper Current is used in the Primary Charge Termination Algorithm. *AverageCurrent()* is integrated over each of the two ***Current Taper Window*** periods separately, and then they are averaged separately to give two averages (Iavg1, Iavg2). Three requirements must be met to qualify for Primary Charge Termination:

- During two consecutive periods of ***Current Taper Windows***:
  Iavg1 < ***Taper Current*** and Iavg2 < ***Taper Current***
- During the same periods: Accumulated change in capacity > ***Min Taper Capacity*** per ***Current Taper Window***
- ***Voltage*** > ***Charging Voltage*** – ***Taper Voltage***

When Primary Charge Termination conditions are met, the *Flags() [FC]* bit is set and *[CHG]* bit is cleared. Also, if the ***Pack Configuration [RMFCC]*** bit is set, then *RemainingCapacity()* is set equal to *FullChargeCapacity()*.

This register depends on battery characteristics and charger specifications, but typical values are C/10 to C/20. *AverageCurrent()* is not used for the qualification because its time constant is not the same as the ***Current Taper Window***. The reason for making two current taper qualifications is to prevent false current taper qualifications. False primary charge terminations happen with pulse charging and with random starting and stopping of the charge current. This is particularly critical at the beginning or end of the qualification period. It is important to note that as the ***Current Taper Window*** value is increased, the current range in the second requirement for primary charge termination is lowered. If the ***Current Taper Window*** is increased, then the current used to integrate to the ***Min Taper Capacity*** is decreased and this threshold becomes more sensitive.

5.3.3.2 Terminate Charge Alarm Set % and Clear %

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
36	Charge Termination	7	TCA Set %	I1	–1	100	–1	Percent
		8	TCA Clear %	I1	–1	100	98	Percent

TCA Set % is the terminate charge alarm set percentage threshold. ***TCA Set %*** sets a *StateOfCharge()* percentage threshold at which the *Flags() [CHG]* bit is cleared. When ***TCA Set %*** is set to –1, it disables the use of the charge alarm threshold. When ***TCA Set %*** is set to –1, the *[CHG]* bit is cleared when the taper condition is detected.

TCA Clear % is the terminate charge alarm clear percentage threshold. ***TCA Clear %*** sets a *StateOfCharge()* percentage level at which the *Flags() [CHG]* bit is set.

[CHG] bit is cleared:

- At taper termination if ***TCA Set %*** is –1.
- When *StateOfCharge()* ≥ ***TCA Set %*** and if ***TCA Set %*** is not –1.
- If *Flags() [OTC]* or *[CHG_INH]* is set.

[CHG] bit is set:

- When any of the conditions for *[CHG]* bit to be cleared does not exist and *StateOfCharge()* ≤ ***TCA Clear %***.

NOTE: ***TCA Set %*** and ***TCA Clear %*** only affect the *Flags() [CHG]* bit but does not affect the charge termination process or the gauging function.

5.3.3.3 Full Charge Set % and Clear %

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
36	Charge Termination	9	FC Set %	I1	–1	100	–1	Percent
		10	FC Clear %	I1	–1	100	98	Percent

FC Set %

FC Set % is the full charge set percentage threshold. ***FC Set %*** sets a *StateOfCharge()* percentage threshold at which the *Flags() [FC]* bit is set. When ***FC Set %*** is a value other than –1, the *[FC]* bit is set based on the amount of passed charge detected by the gauge and not charge termination detection. If ***FC Set %*** is set to –1, the *[FC]* bit is set based on charge termination detection (see ***Min Taper Capacity***, ***Taper Current,*** and ***Taper Voltage*** in Section 5.3.3.1).

NOTE: ***FC Set %*** only affects the *Flags() [FC]* bit which does not affect the charge termination process.

The default value is set to –1%.

FC Clear %

FC Clear % is the full charge clear percentage threshold. ***FC Clear %*** sets a *StateOfCharge()* percentage threshold at which the *Flags() [FC]* bit is cleared.

NOTE: ***FC Clear %*** only affects the *Flags() [FC]* bit which does not affect the charge termination process.

The default value is set to 98%.

5.3.3.4 DOD at EOC Delta Temperature

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
36	Charge Termination	11	DODatEOC Delta T	I2	0	1000	50	0.1°C

DODatEOC Delta T

This represents the temperature change threshold to update Q_{start} and *RemainingCapacity()* due to temperature changes. During relaxation and at the start of charging, the remaining capacity is calculated as *RemainingCapacity()* = *FullChargeCapacity()* – Q_{start}. As temperature decreases, Q_{start} can become much smaller than that of the old *FullChargeCapacity()* value, resulting in overestimation of *RemainingCapacity()*. To improve accuracy, *FullChargeCapacity()* is updated whenever the temperature change since the last *FullChargeCapacity()* update is greater than ***DODatEOC Delta T*** × 0.1ºC.

The default value is 50. Note that the units are a tenth of a °C which means a value of 50 corresponds to 5ºC.

5.3.4 JEITA

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
39	JEITA	0	T1 Temp	I1	–128	127	0	°C
		1	T2 Temp	I1	–128	127	10	°C
		2	T3 Temp	I1	–128	127	45	°C
		3	T4 Temp	I1	–128	127	50	°C
		4	T5 Temp	I1	–128	127	60	°C
		5	Temp Hys	I1	–128	127	1	°C
		6	T1-T2 Chg Voltage	I2	0	4600	4350	mV
		8	T2-T3 Chg Voltage	I2	0	4600	4350	mV
		10	T3-T4 Chg Voltage	I2	0	4600	4300	mV
		12	T4-T5 Chg Voltage	I2	0	4600	4250	mV
		14	T1-T2 Chg Current	U1	0	100	50	Percent
		15	T2-T3 Chg Current	U1	0	100	80	Percent
		16	T3-T4 Chg Current	U1	0	100	80	Percent
		17	T4-T5 Chg Current	U1	0	100	80	Percent

T1 Temp, ***T2 Temp***, ***T3 Temp***, ***T4 Temp***, and ***T5 Temp*** represent the temperature boundaries for updating the *ChargingCurrent()* and *ChargingVoltage()* values reported as part of the JEITA charging profile.

- If *Temperature()* < ***T1 Temp***, *ChargingCurrent()* and *ChargingVoltage()* are set to 0.
- If ***T1 Temp*** ≤*Temperature()* ≤ ***T2 Temp***, ***T1-T2 Chg Current*** and ***T1-T2 Chg Voltage*** are reported.
- If ***T2 Temp*** <*Temperature()* ≤ ***T3 Temp***, ***T2-T3 Chg Current*** and ***T2-T3 Chg Voltage*** are reported.
- If ***T3 Temp*** <*Temperature()* ≤ ***T4 Temp***, ***T3-T4 Chg Current*** and ***T3-T4 Chg Voltage*** are reported.
- If ***T4 Temp*** <*Temperature()* ≤ ***T5 Temp***, ***T4-T5 Chg Current*** and ***T4-T5 Chg Voltage*** are reported.
- If *Temperature()* > ***T5 Temp***, *ChargingCurrent()* and *ChargingVoltage()* are set to 0.

Positive temperature hysteresis (***Temp Hys***) is applied with increasing temperature across the ***T1 Temp*** or ***T2 Temp*** thresholds and negative temperature hysteresis is used when moving from right to left across the ***T3 Temp***, ***T4 Temp***, or ***T4 Temp*** thresholds. Programmed charging voltage parameters are in terms of mV and charging current is in terms of % ***Design Capacity***.

5.3.5 Data Subclass

5.3.5.1 Design Voltage

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	0	Design Voltage	I2	2000	5000	3800	mV

Design Voltage is the nominal voltage of the pack as specified by the battery vendor. The value should be set based on the battery specificaiton.

5.3.5.2 Cycle Count

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	8	Cycle Count	U2	0	65535	0	Count

This register records the number of cycles the battery has experienced. One cycle occurs when accumulated discharge ≥ ***CC Threshold***. The value is reported in *CycleCount()*.

5.3.5.3 Cycle Count Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	10	CC Threshold	I2	100	32767	900	mAh

This value increments *CycleCount()*. When the gauge accumulates enough discharge capacity equal to ***CC Threshold***, then it increments *CycleCount()* by 1. This discharge capacity does not have to be consecutive. The internal register that accumulates the discharge is not cleared at any time except when the internal accumulating register equals the ***CC Threshold***, and increments *CycleCount()*.

This is normally set to about 90% of the ***Design Capacity***.

5.3.5.4 Design Capacity

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	12	Design Capacity	I2	0	14500	1000	mAh

This is the original chemical capacity of the pack as specified by the battery vendor. This is used in Impedance Track algorithm in remaining and full charge capacity (RM and FCC) calculations. The value should be set based on the battery specification.

5.3.5.5 Design Energy

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	14	Design Energy	I2	0	32767	3800	mWh

Design Energy is similar to ***Design Capacity*** but represented in energy units.

Design Energy = ***Design Capacity*** × Design Voltage

The actual unit of this parameter is dependent on ***Design Energy Scale***. The default value is 3800 mWh.

5.3.5.6 State of Health Load I

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	16	SOH Load I	I2	–32767	0	–400	mA

StateOfHealth() is calculated using the ratio of *FullChargeCapacity()* (FCC) and *DesignCapacity()*. The FCC used in the SOH calculation is simulated using a fixed temperature (25ºC) and load (defined by ***SOH Load I***). The FCC value used is not necessarily the same as the *FullChargeCapacity()* data RAM register since the value reported in data RAM register changes based on current system load and temperature.

The default is –400 mA. It is recommended to set this value to a typical system current.

5.3.5.7 TDD State Of Health Percent

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	18	TDD SOH Percent	U1	0	100	80	Percent

The fuel gauge can indicate tab disconnection by detecting change of *StateOfHealth()*. This feature is enabled by setting ***[SE_TDD]*** bit in ***Pack Configuration B***. The *[TDD]* of *Flags()* is set when the ratio of current *StateOfHealth()* divided by the previous *StateOfHealth()* reported is less than ***TDD SOH Percent***. The *[TDD]* of *Flags()* can be configured to control an interrupt pin (RC2) by enabling the INTERRUPT mode. See Section 2.4.2, *Host Interrupts*, for details.

The default is 80%.

5.3.5.8 ISD Current Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	19	ISD Current	U2	1	32767	10	Hour Rate

The fuel gauge can indicate detection of an internal battery short if the ***[SE_ISD]*** bit in ***Pack Configuration B*** is set. The gauge compares the self-discharge current calculated based on RELAXATION mode to the *AverageCurrent()* measured in the system. The self-discharge rate is measured at 1 hour intervals. When battery *SelfDischargeCurrent()* is less than the predefined –***Design Capacity***/***ISD Current*** threshold, the *[ISD]* of *Flags()* is set high. The *[ISD]* of *Flags()* can be configured to control interrupt pin (RC2) by enabling the INTERRUPT mode. See Section 2.4.2, *Host Interrupts*, for details.

The default is 10 HourRate. The HourRate unit is defined as *DesignCapacity()*/[HourRate]. It is recommended to test this feature and tune this parameter to obtain the optimal value in order to avoid both false positives and false negatives.

5.3.5.9 ISD Current Filter

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	21	ISD I Filter	U1	0	255	127	Count

The ISD I Filter filters the amount of change allowed in the *SelfDischargeCurrent()* register. A large value of ***ISD I Filter*** restricts large fluctuations in the value of *SelfDischargeCurrent()* if the most recent current value read by the gauge is significantly different from the previous readings. A small value of ***ISD I Filter*** allows the value of *SelfDischargeCurrent()* to update to a value that is closer to the most recent value read by the gauge.

The default is 127.

5.3.5.10 Minimum ISD Detection Time

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	22	Min ISD Time	U1	0	255	7	Hour

This parameter defines the amount of time the gauge needs to wait after the initial DOD measurement is made in RELAXATION mode before an attempt is made to detect an internal short in the battery pack.

The default is 7 hours.

5.3.5.11 Design Energy Scale

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	23	Design Energy Scale	U1	1	10	1	Number

Design Energy Scale selects the scale and units of a set of data flash parameters. The value of ***Design Energy Scale*** can be either 1 or 10. For battery capacities larger than 6 Ahr, ***Design Energy Scale*** = 10 is recommended.

Table 5-10. Data Flash Parameter Unit/Scale Based on Design Energy Scale

Data Flash	Design Energy Scale = 1 (default)	Design Energy Scale = 10
Design Energy	mWh	cWh
Reserve Capacity (mWh)	mWh	cWh
Avg Power Last Run	mW	cW
User Rate-Pwr	mWh	cWh
T Rise	No Scale	Scaled by ×10

5.3.6 Discharge Subclass

5.3.6.1 State of Charge 1 Set and Clear Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
49	Discharge	0	SOC1 Set Threshold	U2	0	65535	150	mAh
		2	SOC1 Clear Threshold	U2	0	65535	175	mAh

SOC1 Set Threshold sets a *StateOfCharge()* percentage threshold used to indicate when *StateOfCharge()* falls to or below a defined *StateOfCharge()*. The ***SOC1 Set Threshold*** is typically used as an initial low *StateOfCharge()* warning. When *StateOfCharge()* falls below the ***SOC1 Set Threshold***, the *Flags() [SOC1]* bit is set. The *[SOC1]* bit is cleared once *StateOfCharge()* rises above the ***SOC1 Clear Threshold***. If ***SOC1 Set Threshold*** is set to (–)1, then the *[SOC1]* bit becomes inoperative.

SOC1 Set Threshold is normally set to 10% of ***Design Capacity***.

SOC1 Clear Threshold is normally set to 5% above the ***SOC1 Set Threshold***; that is, 15% of ***Design Capacity***.

5.3.6.2 State of Charge Final Set and Clear Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
49	Discharge	4	SOCF Set Threshold	U2	0	65535	75	mAh
		6	SOCF Clear Threshold	U2	0	65535	100	mAh

The ***SOCF Set Threshold*** is the *StateOfCharge()* percentage threshold used to indicate when *StateOfCharge()* falls to or below a defined *StateOfCharge()*. The ***SOCF Set Threshold*** is typically used as a final low *StateOfCharge()* warning. When *StateOfCharge()* falls below the ***SOCF Set Threshold***, the *Flags() [SOCF]* bit is set. The *[SOCF]* bit is cleared once *StateOfCharge()* rises above the ***SOCF Clear Threshold***. If ***SOCF Set Threshold*** is set to (–)1, then the *[SOCF]* bit becomes inoperative.

SOCF Set Threshold is normally set to 2% of ***Design Capacity***.

SOCF Clear Threshold is normally set to 3% above the ***SOCF Set Threshold***, that is 5% of ***Design Capacity***.

5.3.6.3 Battery Low Set Voltage Threshold, Time, and Clear

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
49	Discharge	8	BL Set Volt Threshold	I2	0	5000	2500	mV
		10	BL Set Volt Time	U1	0	60	2	s
		11	BL Clear Volt Threshold	I2	0	5000	2600	mV

BL Set Volt Threshold

BL Set Volt Threshold provides a threshold for the *Voltage()* register. Once the *Voltage()* register falls below this value for a specific time defined by ***BL Set Volt Time***, the battery low *Flags() [BATLOW]* bit is set. Fuel gauge must not be in SLEEP mode.

BL Set Volt Time

When *Voltage()* < ***BL Set Volt Threshold*** is true, ***BL Set Volt Time*** provides the time to wait before the *Flags() [BATLOW]* bit gets set. Fuel gauge must not be in SLEEP mode.

BL Clear Volt Threshold

BL Clear Volt Threshold provides a threshold for the *Voltage()* register. Once the *Voltage()* register rises above this value, the *Flags() [BATLOW]* bit is cleared immediately. The fuel gauge must not be in SLEEP mode.

5.3.6.4 Battery High Set Voltage Threshold, Time, and Clear

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
49	Discharge	13	BH Set Volt Threshold	I2	0	5000	4500	mV
		15	BH Volt Time	U1	0	60	2	s
		16	BH Clear Volt Threshold	I2	0	5000	4400	mV

BH Set Volt Threshold

BH Set Volt Threshold provides a threshold for the *Voltage()* register. Once the *Voltage()* register rises above this value for a specific time defined by ***BH Volt Time***, the battery high *Flags() [BATHI]* bit is set. The fuel gauge must not be in SLEEP mode.

BH Volt Time

When *Voltage()* < ***BH Set Volt Threshold*** is true, ***BH Volt Time*** provides the time to wait before the *Flags() [BATHI]* bit gets set. Fuel gauge must not be in SLEEP mode.

BH Clear Volt Threshold

BH Clear Volt Threshold provides a threshold for the *Voltage()* register. Once the *Voltage()* register falls above this value, the *Flags() [BATHI]* bit is cleared immediately. Fuel gauge must not be in SLEEP mode.

5.3.7 Manufacturer Data Subclass

5.3.7.1 Pack Lot Code

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
56	Manufacturer Data	0	Pack Lot Code	H2	0x00	0xFFFF	0x00	hex

The pack manufacturer can use this location to store the pack lot code.

5.3.7.2 PCB Lot Code

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
56	Manufacturer Data	2	PCB Lot Code	H2	0x00	0xFFFF	0x00	hex

The pack manufacturer can use this location to store the PCB lot code.

5.3.7.3 Firmware Version

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
56	Manufacturer Data	4	Firmware Version	H2	0x00	0xFFFF	0x00	hex

The pack manufacturer can use this location to store a firmware version number for their system or pack. This value is user-defined and is not related to the gauge's *Control(FW_VERSION)*.

5.3.7.4 Hardware Revision

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
56	Manufacturer Data	6	Hardware Revision	H2	0x00	0xFFFF	0x00	hex

The pack manufacturer can use this location to store a hardware version number for their system or pack. This value is user-defined and is not related to the gauge's *Control(HW_VERSION)*.

5.3.7.5 Cell Revision

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
56	Manufacturer Data	8	Cell Revision	H2	0x00	0xFFFF	0x00	hex

The pack manufacturer can use this location to store the version of their cell.

5.3.7.6 Data Flash Configuration Version

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
56	Manufacturer Data	10	DF Config Version	H2	0x00	0xFFFF	0x00	hex

The pack manufacturer can use this location to store the data flash configuration version. Version control of DFI files used in production is recommended.

5.3.8 *Integrity Data Subclass*

5.3.8.1 All Data Flash Checksum

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
57	Integrity Data	6	All DF Checksum	H2	0x00	0x7FFF	0x00	Number

This value is a 16-bit unsigned integer sum of each byte in the data flash. The sum is calculated on a byte-by-byte basis. The most significant bit of the checksum is masked yielding a 15-bit checksum. This checksum is compared with the value generated by command 0x1A. This checksum is intended to validate all parameters that are not pack specific. Table 5-11 shows the data flash that are excluded.

Table 5-11. All Data Flash Checksum Exclusions

Class	Subclass ID	Subclass	Comment
Configuration	56	Manufacturer Data	Pack Lot Code, PCB Lot Code
Configuration	57	Integrity Data	Reset Counter – Full (private) Reset Counter – Watch Dog (private)
Configuration	57	Integrity Data	All DF Checksum
System Data	58	Manufacturer Info	Block A Block B
Calibration	104	Data	CC Gain CC Delta CC Offset Board offset Int Temp Offset Ext Temp Offset Pack V offset

5.3.8.2 Static Chem Data Flash Checksum

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
57	Integrity Data	8	Static Chem DF Checksum	H2	0x00	0x7FFF	0x7C23	Number

This value is a 16-bit unsigned integer sum of each byte in the chemistry data flash. The sum is calculated on a byte-by-byte basis. The most significant bit of the checksum is masked yielding a 15-bit checksum. This checksum is intended to validate chemistry specific data. Table 5-12 shows the data flash that are included. This checksum is executed in conjunction with the *IT_ENABLE* subcommand. If this checksum fails, Impedance Track is not enabled.

Table 5-12. All Chemistry Data Checksum Inclusions

Class	Subclass ID	Subclass	Comment
OCV Table	83	OCV Table	ChemID (public) OCVa Table (private)
OCVb Table	84	OCVb Table	OCVb Table (private)
Rb_Hi Table	85	Rb_Hi Table	Rb Hi Table (private)
Rb_Lo Table	108	Rb_Lo Table	Rb Lo Table (private)
Gas Gauging	80	IT Cfg	Q Invalid Max V Q Invalid Min V

5.3.8.3 Static Data Flash Checksum

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
57	Integrity Data	10	Static DF Checksum	H2	0x00	0x7FFF	0x00	Number

This value is a 16-bit unsigned integer sum of each byte in the data flash. The sum is calculated on a byte-by-byte basis. The most significant bit of the checksum is masked yielding a 15-bit checksum. This checksum is compared with the value generated by command 0x1A. This checksum is intended to validate all parameters that are static. Table 5-13 shows the data flash that are excluded. The checksum execution takes approximately 5 ms and during this time, the fuel gauge does not communicate.

Table 5-13. All Static Data Flash Checksum Exclusions

Class	Subclass ID	Subclass	Comment
Configuration	48	Data	Cycle Count
Configuration	56	Manufacturer Data	Pack Lot Code, PCB Lot Code
Configuration	57	Integrity Data	Reset Counter – Full (private) Reset Counter – Watch Dog (private)
Configuration	57	Integrity Data	All DF Checksum
Configuration	57	Integrity Data	Static DF Checksum
System Data	58	Manufacturer Info	Block A Block B
LT Data	59	Lifetime Data	All Lifetime Data
LT Data	60	Lifetime Temp Samples	All Lifetime Temp Samples
Gas Gauging	82	State	Qmax Cell 0 Cycle Count Update_Status V at Chg Term Avg I Last Run Avg P Last Run Delta Voltage Max Discharge Duration (private)
Ra Tables	88	Data	Ra Table
Ra Tables	89	Data	Rax Table
Calibration	104	Data	CC Gain CC Delta CC Offset Board offset Int Temp Offset Ext Temp Offset Pack V offset

5.3.8.4 Protector Checksum

Subclass ID	Subclass	Full Name	Offset	Name	Data Type	Value			Unit
						Min	Max	Default	
57	Integrity Data	Protector Checksum	12	Prot Checksum	H2	0x00	0x7FFF	0x0011	Number

The ***Prot Checksum*** is used to guard against accidental mis-configuration of the hardware protector. It is compared against the sum of OV Prot Cfg and ***OC Prot Cfg*** every second and, if a mismatch is detected, the fuel gauge will set the *SafetyStatus()[INV_PROT_CHKSUM]* flag and disable the FETs. Once the correct checksum is programmed, the flag is cleared and the FETs are re-activated.

5.3.9 Lifetime Data Subclass, Lifetime Resolution Subclass

Lifetime data subclass contains black box data that records various data over the life of the pack. This data can be very useful for performing failure analysis on the returned packs. Lifetime data is enabled if the *CONTROL_STATUS [QEN]* bit is 1. The *[QEN]* bit is set by sending *IT_ENABLE* subcommand.

The lifetime update for the values below is throttled to not happen more than once per 60 seconds to avoid data flash wear out. The frequency of the updates will naturally slow down once pack updates the minimum and maximum values over several packs,

- ***Lifetime Max Temp***: Maximum temperature observed by the gauge. It is initialized to 300. The unit is 0.1ºC.
- ***Lifetime Min Temp***: Minimum temperature observed by the gauge. It is initialized to 200. The unit is 0.1ºC.
- ***Lifetime Max Pack Voltage***: Maximum battery voltage observed by the gauge. It is initialized to 3200. The unit is mV.
- ***Lifetime Min Pack Voltage***: Minimum battery voltage observed by the gauge. It is initialized to 4200. The unit is mV.
- ***Lifetime Max Chg Current***: Maximum charge current observed by the gauge. It is initialized to 0. The unit is mA.
- ***Lifetime Max Dsg Current***: Maximum discharge current observed by the gauge. It is initialized to 0. The unit is mA.
- ***LT Flash Cnt***: Lifetime flash page update counter keeps track of total number of updates. It is initialized to 0. The unit is counts.

5.3.9.1 Maximum Temperature, Minimum Temperature, Temperature Resolution

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
59	Lifetime Data	0	Lifetime Max Temp	I2	–600	1400	0	0.1°C
		2	Lifetime Min Temp	I2	–600	1400	500	0.1°C
66	Lifetime Resolution	0	LT Temp Res	U1	0	255	10	Num

Lifetime Max Temp value is updated if one of the following conditions is met:

- *Temperature()* – ***Lifetime Max Temp*** > ***LT Temp Res***
- *Temperature()* > ***Lifetime Max Temp*** and any other lifetime value is updated.

Lifetime Min Temp value is updated if one of the following conditions is met:

- ***Lifetime Min Temp*** – *Temperature()* > ***LT Temp Res***
- *Temperature()* < ***Lifetime Min Temp*** and any other lifetime value is updated.

5.3.9.2 Maximum Pack Voltage, Minimum Pack Voltage, Voltage Resolution

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
59	Lifetime Data	4	Lifetime Max Pack Voltage	I2	0	32767	2800	mV
		6	Lifetime Min Pack Voltage	I2	0	32767	5000	mV
66	Lifetime Resolution	1	LT V Res	U1	0	255	25	Num

Lifetime Max Pack Voltage value is updated if one of the following conditions is met:

- *Voltage()* – ***Lifetime Max Pack Voltage*** > ***LT V Res***
- *Voltage()* > ***Lifetime Max Pack Voltage*** and any other lifetime value is updated.

Lifetime Min Pack Voltage value is updated if one of the following conditions is met:

- ***Lifetime Min Pack Voltage*** – *Voltage()* > ***LT V Res***
- *Voltage()* < ***Lifetime Min Pack Voltage*** and any other lifetime value is updated.

5.3.9.3 Maximum Charge Current, Maximum Discharge Current, Current Resolution

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
59	Lifetime Data	8	Lifetime Max Chg Current	I2	–32767	32767	0	mA
		10	Lifetime Max Dsg Current	I2	–32767	32767	0	mA
66	Lifetime Resolution	2	LT Cur Res	U1	0	255	100	Num

Lifetime Max Chg Current value is updated if one of the following conditions is met:

- *Current()* – ***Lifetime Max Chg Current*** > ***LT Cur Res***
- *Current()* > ***Lifetime Max Chg Current*** and any other lifetime value is updated.

Lifetime Max Dsg Current value is updated if one of the following conditions is met:

- ***Lifetime Max Dsg Current*** – *Current()* > ***LT Cur Res***
- ***Lifetime Max Dsg Current*** > *Current()* and any other lifetime value is updated.

NOTE: During discharge, current is negative.

5.3.10 Lifetime Temp Samples Subclass

5.3.10.1 Flash Write Count

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
60	Lifetime Temp Samples	0	LT Flash Cnt	U2	0	32767	0	Count

LT Flash Cnt tracks the number of lifetime data flash updates.

5.3.10.2 AFE Status

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
60	Lifetime Temp Samples	2	LT AFE Status	H1	0x00	0xFF	0x00	hex

LT AFE Status polls *Protector Status* every 1 second and records new protector events. This register shows the cumulative status of the protector over the lifetime of the gauge. Bitmap for this register:

Table 5-14. LT AFE Status Bit Definition

Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSVD	RSVD	CVM	UVP	OVP	SCD	OCD	OCC

RSVD = Bits 6 and 7 are reserved. Do not use.

CVM = Cell voltage monitor threshold
1 = Cell voltage monitor threshold detected
0 = Cell voltage monitor threshold not detected

UVP = Undervoltage protection fault
1 = Undervoltage protection fault detected
0 = Undervoltage protection fault not detected

OVP = Overvoltage protection fault
1 = Overvoltage protection fault detected
0 = Overvoltage protection fault not detected

SCD = Short circuit discharge fault
1 = Short-circuit discharge fault detected
0 = Short-circuit discharge fault not detected

OCD = Overcurrent discharge fault
1 = Overcurrent discharge fault detected
0 = Overcurrent discharge fault not detected

OCC = Overcurrent charge fault
1 = Overcurrent charge fault detected
0 = Overcurrent charge fault not detected

5.3.11 Registers Subclass

5.3.11.1 Pack Configuration Register

Some pin configurations and algorithm settings are configured via the ***Pack Configuration*** data flash register, as indicated in Table 5-15. This register is programmed and read via the methods described in Section 5.1.1, *Accessing the Data Flash*. The register is located at subclass = 64, offset = 0.

Table 5-15. Pack Configuration Bit Definition

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	RSVD	HOSTIPU	HOSTIPol	HOSTIE	SOCHold99	IWAKE	RSNS1	RSNS0
	0	0	0	0	1	0	0	1
	0x09							
Low Byte	GNDSEL	RFACTSTEP	SLEEP	RMFCC	SOCHold1	SOCHoldOvrChg	SOCHoldOvrDsg	TEMPS
	0	1	1	1	1	1	1	1
	0x7F							

High Byte

RSVD = Bit 7 is reserved. Must be 0.

HOSTIPU = Internal 1.8 V pullup resistor (RC2 Pin).
0 = disabled
1 = enabled

HOSTIPol = Polarity for Interrupt pin.
0 = active low
1 = active high

HOSTIE = Enables RC2 pin interrupts.

SOCHold99 = The fuel gauge will prevent *StateofCharge()* from reporting 100% until *Flags()[FC]* is set. Set to 1 to enable.

IWAKE, RSNS1, RSNS0 = These bits configure the current wake function (See Section 2.8.2, *Wake-Up Comparator*).

Low Byte

GNDSEL = The ADC ground select control. The V_{SS} (pins C1 and C2) is selected as ground reference when the bit is clear. Pin A1 is selected when the bit is set.

RFACTSTEP = Enables Ra step up/down to Max/Min Res Factor before disabling Ra updates.

SLEEP = The fuel gauge can enter sleep, if operating conditions allow. True when set. (See Section 2.7.2, *SLEEP Mode*.)

RMFCC = RM is updated with the value from FCC, on valid charge termination. True when set. (See Section 2.6.4, *Charge Termination Detection*)

SOCHold1 = The fuel gauge will prevent *StateofCharge()* from reporting 0% until *Voltage()* is less than or equal to ***Terminate Voltage***. Set to 1 to enable.

SOCHoldOvrChg = The fuel gauge will hold *StateofCharge()* at 100% while in an overcharge condition and not decrement until the charge surplus is equalized. Set to 1 to enable.

SOCHoldOvrDsg = The fuel gauge will hold *StateofCharge()* at 0% while in an overdischarge condition and not decrement until the charge deficit is equalized. Set to 1 to enable.

TEMPS = Selects external thermistor for Temperature() measurements. True when set. (See Section 2.5, *Temperature Measurement and The TS Input*)

5.3.11.2 Pack Configuration B Register

Some pin configurations and algorithm settings are configured via the ***Pack Configuration B*** data flash register, as indicated in Table 5-16. This register is programmed and read via the methods described in Section 5.1.1, *Accessing the Data Flash*. The register is located at subclass = 64, offset = 2.

Table 5-16. Pack Configuration B Bit Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ChgDoDEoC	SE_TDD	VconsEN	SE_ISD	RSVD	LFPRelax	DoDWT	FConvEn
1	0	1	0	0	1	1	1
0xA7							

ChgDoDEoC = Enable DoD at EoC recalculation during charging only. True when set. Default setting is recommended.

SE_TDD = Enable Tab Disconnect Detection. True when set. (See Section 2.2.1.6, *Tab Disconnect Detection*)

VconsEN = Enable voltage consistency check. True when set. Default setting is recommended.

SE_ISD = Enable Internal Short Detection. True when set. (See Section 2.4.5, *Internal Short Detection*)

RSVD = Bit 3 is reserved. Must be 0.

LFPRelax = Enable $LiFePO_4$ long RELAXATION mode. True when set.

DoDWT = Enable DoD weighting feature of gauging algorithm. This feature can improve accuracy during relaxation in a flat portion of the voltage profile, especially when using $LiFePO_4$ chemistry. True when set.

FConvEn = Enable fast convergence algorithm. Default setting is recommended. (See Section 2.1.3, *Fast Resistance Scaling*)

5.3.11.3 Pack Configuration C Register

Some algorithm settings are configured via the ***Pack Configuration C*** data flash register, as indicated in Table 5-17. This register is programmed and read via the methods described in Section 5.1.1, *Accessing the Data Flash*. The register is located at subclass = 64, offset = 3.

Table 5-17. Pack Configuration C Bit Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FastQmax	FConvTempEn	RlxSmEn	SmoothEn	SleepWkChg	RSVD	RSVD	BTP_EN
1	0	1	1	1	0	0	1
0xB9							

FastQmax = Fast Qmax feature is enabled.

FConvTempEn = Thermal modeling is enabled while in FAST RESISTANCE SCALING mode. Set to 1 to enable. Default of 0 is recommended.

RlxSmEn = SOC smoothing is enabled while in battery relaxation state. Set to 1 to enable.

SmoothEn = Enable SOC smoothing algorithm. True when set. (See Section 2.1.4, *StateOfCharge() Smoothing*)

SleepWkChg = Enables compensation for the passed charge missed when waking from SLEEP mode.

RSVD = Bits 1 and 2 are reserved. Must be 0.

BTP_EN = BTP interrupts are enabled on the RC2 pin. When enabled, all other interrupts are disabled. Set to 1 to enable.

5.3.11.4 Pack Configuration D

Table 5-18. Pack Configuration D Bit Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OTFET	FCFET	CIFET	CSFET	SMSYNCEN	RSVD	IMAXRESRVEN	IMAXEN
1	0	0	0	0	0	1	1
0x83							

OTFET = FET disable will occur based on an overtemperature fault. *SafetyStatus()[OTC]* = 1 turns off CHG FET. *SafetyStatus()[OTD]* = 1 turns off DSG FET.

FCFET = FET disable will occur based on a charge termination. *Flags()[FC]* = 1 turns off CHG FET.

CIFET = FET disable will occur based on a charge inhibit condition. *Flags()[CHG_INH]* = 1 turns off CHG FET.

CSFET = FET disable will occur based on a charge suspend condition. *Flags()[CHG_SUS]* = 1 turns off CHG FET.

SMSYNCEN SOC smoothing in relax will immediately equalize differences in true SOC vs reported *StateofCharge()* instead of gradually converging capacity (RM and FCC) differences over time.

RSVD = Bit 2 is reserved. Must be 0.

IMAXRESRVEN = Enables usage of ***Reserve Capacity*** in the *Imax()* calculation.

IMAXEN = Enables maximum allowed discharge reporting in *Imax()*.

5.3.11.5 Protector Overcurrent Configuration

Table 5-19. Protector Overcurrent Configuration Bit Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSVD	RSVD	SCD	OCD2	OCD1	OCD0	OCC1	OCC0
0	0	0	0	1	0	1	0
0x0A							

RSVD = Bits 7 and 6 are reserved. Must be 0.
SCD = Short-circuit in Discharge Configuration
0 = 73 mV
1 = 148 mV
OCD2, OCD1, OCD0 = Overcurrent in Discharge Configuration
000 = 14 mV
001 = 24 mV
010 = 34 mV
011 = 44 mV
100 = 53 mV
101 = 63 mV
110 = 73 mV
111 = 83 mV
OCC1, OCC0 = Overcurrent in Charge Configuration
00 = 6 mV
01 = 13 mV
10 = 18 mV (default)
11 = 28 mV

5.3.11.6 Protector Overvoltage Configuration

Table 5-20. Protector Overvoltage Configuration Bit Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSVD	RSVD	RSVD	RSVD	RSVD	OVP2	OVP1	OVP0
0	0	0	0	0	1	1	1
0x07							

RSVD = Bits 7 through 3 are reserved. Must be 0.
OVP2, OVP1, OVP0 = Overvoltage Configuration
000 = 4.275 V
001 = 4.300 V
010 = 4.325 V
011 = 4.350 V
100 = 4.375 V
101 = 4.400 V
110 = 4.425 V
111 = 4.450 V

5.3.12 Lifetime Resolution Subclass

5.3.12.1 Lifetime Update Time

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
66	Lifetime Resolution	3	LT Update Time	U2	0	65535	60	Num

This parameter sets the minimum time between data flash writes to update the Lifetime Parameters. The default for this register is 60.

5.3.13 Power Subclass

5.3.13.1 Valid Update Voltage

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
68	Power	0	Flash Update OK Voltage	I2	0	5000	2800	mV

This register controls one of the several data flash protection features. It is critical that data flash is not updated when the battery voltage is too low. Data flash programming takes much more current than normal operation of the gauge, and with a depleted battery, this current can cause the battery voltage to drop dramatically, forcing the gauge into reset before completing a data flash write. The effects of an incomplete data flash write can corrupt the memory, resulting in unpredictable and extremely undesirable results. The voltage setting in ***Flash Update OK Voltage*** prevents any writes to the data flash below this value. If a charger is detected, then this register is ignored.

The default for this register is 2800 mV. Ensure that this register is set to a voltage where the battery has plenty of capacity to support data flash writes but below any normal battery operation conditions.

5.3.13.2 Sleep Current Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
68	Power	2	Sleep Current	I2	0	100	15	mA

When *AverageCurrent()* is less than ***Sleep Current*** or greater than (–)***Sleep Current***, the gauge enters SLEEP mode if the feature is enable by setting the ***Pack Configuration [SLEEP]*** bit.

This setting should be below any normal application currents.

5.3.13.3 Shutdown Voltage

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
68	Power	10	Shutdown V	I2	0	5000	0	mV

When voltage drops below the ***Shutdown V*** threshold and current is below ***Sleep Current***, then the device attempts to shutdown. If ***Shutdown V*** is set to 0 then this portion of shutdown operation is disabled. The *CLEAR_SHUTDOWN* subcommand can be used to interrupt the shutdown procedure if the charger was attached right after the conditions for automated shutdown occur.

The default value is set to 0 mV.

5.3.13.4 Full Sleep Wait Time

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
68	Power	12	FS Wait	U1	0	255	0	s

FS Wait provides the time to wait for the fuel gauge to go from SLEEP mode to FULLSLEEP mode. When the ***FS Wait*** value is 0, the gauge waits for the *SET_FULLSLEEP* subcommand, once the gauge receives this command while in SLEEP mode, it immediately goes to FULLSLEEP mode. If ***FS Wait*** is non-zero, the gauge switches to FULLSLEEP from SLEEP, once the timer expires. During the wait time, *SET_FULLSLEEP* subcommand is ignored. Note that when the gauge is in FULLSLEEP mode, any communication with the gauge triggers it to get out of FULLSLEEP mode, and the ***FS Wait*** counter is reset. If FULLSLEEP state is exited due to any other condition, the ***FS Wait*** counter is not reset. The best way to check the mode of the gauge is to monitor the drawn current out of the gauge.

The default value is 0 seconds.

5.4 System Data Class

5.4.1 Manufacturer Information Subclass

5.4.1.1 Block A and Block B

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
58	Manufacturer Info	0 through 31	Block A 0 through 31	H1	0x00	0xFF	0x00	
		32 through 63	Block B 0 through 31	H1	0x00	0xFF	0x00	

Each block can hold a maximum of 8 characters or 32 bytes of user-programmable data.

The method for accessing these memory locations is different, depending on whether the device is in UNSEALED or SEALED mode.

When in UNSEALED mode and when an 0x00 has been written to *BlockDataControl()*, accessing the ***Manufacturer Info Blocks*** is identical to accessing general data flash locations. First, a *DataFlashClass()* command sets the subclass, then a *DataFlashBlock()* command sets the offset for the first data flash address within the subclass. The *BlockData()* command codes contain the referenced data flash data. When writing the data flash, a checksum is expected to be received by *BlockDataCheckSum()*. Only when the checksum is received and verified is the data actually written to data flash.

As an example, the data flash location for ***Manufacturer Info Block B*** is defined as having a Subclass = 58 and an Offset = 32 through 63 (32-byte block). The specification of Class = System Data is not needed to address ***Manufacturer Info Block B***, but is used instead for grouping purposes when viewing data flash info in the evaluation software.

When in SEALED mode or when *BlockDataControl()* does not contain 0x00, data flash is no longer available in the manner used in UNSEALED mode. Rather than issuing subclass information, a designated ***Manufacturer Information Block*** is selected with the *DataFlashBlock()* command. Issuing a 0x01, 0x02, or 0x03 with this command causes the corresponding information block (A or B, respectively) to be transferred to the command space 0x40 through 0x5F for editing or reading by the system. Upon successful writing of checksum information to *BlockDataCheckSum()*, the modified block is returned to data flash.

NOTE: ***Manufacturer Info Block A*** is read-only when in SEALED mode.

5.5 Gas (Fuel) Gauging Class

5.5.1 IT Cfg Subclass

5.5.1.1 Load Select

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	0	Load Select	U1	0	6	1	Number

Load Select defines the type of power or current model to be used to compute load-compensated capacity in the Impedance Track algorithm. By default, ***Load Select*** is set to 1, which means the IT algorithm uses a running average of the current discharge period. Once the discharge stops, the algorithm stores the average in data flash as the ***Avg I Last Run*** and ***Avg P Last Run*** variables. For simulations during RELAXATION, CHARGE, and at the start of DISCHARGE (since a new average has not been gathered), it would use data flash values for simulations. Once the discharge lasts 500 seconds, the gauge re-simulates using the new running average. Thereafter, during discharge, it uses the continuous running average for any subsequent simulations. Re-simulations can also be triggered by a temperature change, which can cause an updated simulation to occur earlier than 500 seconds into a discharge. In this case, this simulation would use the present running average.

If ***Load Mode*** = 0 (constant-current model), then the options presented in Table 5-21 are available.

Table 5-21. Constant-Current Model Used When Load Mode = 0

Load Select Value	Current Model Used
0	Average discharge current from previous cycle: There is an internal register that records the average discharge current through each entire discharge cycle. The previous average is stored in this register.
1 (default)	Present average discharge current: This is the average discharge current from the beginning of this discharge cycle until present time.
2	Average current: based off the *AverageCurrent()*
3	Current: based off of a low-pass-filtered version of *AverageCurrent()* (τ = 14 s)
4	Design capacity/5: C Rate based off of Design Capacity /5 or a C/5 rate in mA.
5	Use the value specified by *AtRate()*
6	Use the value in ***User_Rate-mA.*** This gives a completely user-configurable method.

If ***Load Mode*** = 1 (constant-power model) then the following options are available:

Table 5-22. Constant-Power Model Used When Load Mode = 1

Load Select Value	Power Model Used
0	Average discharge power from previous cycle: There is an internal register that records the average discharge power through each entire discharge cycle. The previous average is stored in this register.
1	Present average discharge power: This is the average discharge power from the beginning of this discharge cycle until present time.
2	Average current × voltage: based off the *AverageCurrent()* and *Voltage()*.
3	Current × voltage: based off of a low-pass-filtered version of *AverageCurrent()* (τ = 14 s) and *Voltage()*
4	Design energy/5: C Rate based off of Design Energy /5 or a C/5 rate in mW or cW.
5	Use the value specified by *AtRate()*
6	Use the value in User_Rate-Pwr. This gives a completely user-configurable method.

5.5.1.2 Load Mode

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	1	Load Mode	U1	0	1	1	Number

Load Mode selects either the constant-current or constant-power model for the Impedance Track algorithm as used in ***Load Select*** (see Section 5.5.1.1, *Load Select*). When ***Load Mode*** is 0, the constant-current model is used (default). When Load Mode is 1, the constant-power model is used. The *CONTROL_STATUS [LDMD]* bit reflects the status of ***Load Mode***.

This is normally set to 0 (constant-current model) but it is application specific. If the application load profile more closely matches a constant-power model, then set to 1. This provides a better estimation of remaining run time, especially close to the end of discharge where current increases to compensate for decreasing battery voltage.

5.5.1.3 Maximum and Minimum Resistance Factor

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	17	Max Res Factor	U1	0	255	15	num
		18	Min Res Factor	U1	0	255	5	num

Max Res Factor:

Maximum allowable cumulative percentage (ratio) increase for impedance values stored in the Ra table (over 15 gridpoint updates).

For Ra_new > Ra_old,
New Ra = min(Ra_new, Ra_old × ***Max Res Factor*** ÷ 10)

The default setting is 15. The algorithm divides the value of this parameter by 10. The upper bound is determined by multiplying (***Max Res Factor***/10) by the impedance value stored in the Ra table; therefore, a value of 15 indicates resistance can only change by 50% from the current resistance value in the positive direction.

Min Res Factor:

Maximum allowable cumulative percentage (ratio) decrease for impedance values stored in the Ra table (over 15 gridpoint updates).

For Ra_new < Ra_old
New Ra = max(Ra_new, Ra_old × ***Min Res Factor*** ÷ 10)

The default setting is 5. The algorithm divides the value of this parameter by 10. The lower bound is determined by multiplying (***Min Res Factor***/10) by the impedance value stored in the Ra table. Therefore a value of 5 indicates resistance can only change by 50% from the current resistance value in the negative direction.

5.5.1.4 Ra Filter

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	20	Ra Filter	U2	0	1000	800	num

Ra table updates are filtered. This is a weighting factor which takes a certain percentage of the previous Ra table value and the remaining percentage comes from the newest calculated Ra value. This is to prevent resistances in the Ra table from changing quickly. After this filter has been applied, there is a final check to make sure that the new resistances satisfy both ***Max Res Factor*** and ***Min Res Factor***.

Ra Filter is a filter constant used to calculate the filtered Ra value that is stored into data flash from the old Ra value.

Ra = (Ra_old × ***Ra Filter*** + Ra_new × (1000 – ***Ra Filter***)) ÷ 1000

It is normally set to 800 (80% previous Ra value plus 20% learned Ra value to form new Ra value).

5.5.1.5 Resistance Update Voltage Drop

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	22	Res V Drop	I2	0	32767	50	mV

Res V Drop is used during battery discharge to qualify sufficient conditions for measuring and storing resistance values. It is useful in applications with low-rate discharge or frequent cold temperature usage that typically have trouble achieving consistent resistance updates. Even with low current, the voltage drop requirement can still be met if enough cell resistance is evident.

5.5.1.6 Fast Qmax Start DOD Percent, Fast Qmax Start Voltage Delta, Fast Qmax Current Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	39	Fast Qmax Start DOD %	U1	0	100	92	%
		41	Fast Qmax Start Volt Delta	I2	0	4200	200	mV
		43	Fast Qmax Current Threshold	U2	0	1000	4	HourRate

Fast Qmax measurement starts when the following conditions are met:

- DOD > ***Fast Qmax Start DOD%*** or
  Voltage < ***Terminate Voltage*** + ***Fast Qmax Start Volt Delta***
- Current < C/***Fast Qmax Current Threshold***

5.5.1.7 Fast Qmax End DOD Percent

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	40	Fast Qmax End DOD %	U1	0	100	96	%

Fast Qmax measurement is performed at the end of discharge when the following conditions are met:

- Number of Fast Qmax measurements > 3
- DOD > ***Fast Qmax End DOD%*** or
  Voltage < ***Terminate Voltage*** + 50 mV

5.5.1.8 Qmax Capacity Error

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	61	Qmax Capacity Err	U1	0	100	15	0.10%

Qmax Capacity Err specifies maximum capacity error allowed during Qmax update. Capacity error is estimated based on the time spent for Qmax measurement.

5.5.1.9 Maximum Qmax Change

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	62	Max Qmax Change	U1	0	255	30	%

Max Qmax Change specifies maximum allowed change in Qmax value during Qmax update. Qmax update is disqualified if change from previous Qmax value is greater than ***Max Qmax Change***.

5.5.1.10 Termination Voltage

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	64	Terminate Voltage	I2	2800	3700	3000	mV

Terminate Voltage is used in the Impedance Track algorithm to compute *RemainingCapacity()*. This is the absolute minimum voltage for end of discharge, where the remaining chemical capacity is assumed to be zero.

Terminate Voltage stores the voltage for the end of discharge where *RemainingCapacity()* is set to 0 mAh.

Set ***Terminate Voltage*** based on battery cell specifications to prevent damage to the cell or set to the absolute minimum system voltage, taking into account impedance drop from the PCB traces, FETs, and wires. The default value is set to 3000 mV.

5.5.1.11 Termination Voltage Delta and Fast Scale Start SOC

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	66	Term V Delta	I2	0	4200	200	mV
		104	Fast Scale Start SOC	U1	0	100	10	%

Fast Scale Start SOC and ***Term V Delta*** specify voltage and SOC thresholds for Fast Ra Scaling activation. Fast Ra Scaling is activated when either of the following conditions is true:

- SOC < ***Fast Scale Start SOC***
- Voltage < (***Terminate Voltage*** + ***Term V Delta***)

The default value for ***Term V Delta*** is 200 mV. For most battery applications, it is recommended to keep (***Terminate Voltage*** + ***Term V Delta***) below 3.4 V.

5.5.1.12 Simulation Res Relax Time

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	69	ResRelax Time	U2	0	65535	500	s

This value is used for Impedance Track transient modeling of effective resistance. The resistance increases from zero to final value determined by the Ra table as defined by the exponent with time constant ***Res Relax Time*** during discharge simulation. Default value has been optimized for typical cell behavior.

ResRelax Time or resistance relaxation time is used for transient modeling. It represents the time it takes for the internal resistance to be fully saturated. This way the gauge will not simulate immediate large IR drops when it calculates the instantaneous voltage from the battery under load.

The default value is 500 seconds, which is sufficient for most applications.

5.5.1.13 User-Defined Rate-Current

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	73	User Rate-mA	I2	0	32767	0	mA

This is the discharge rate used for Impedance Track simulation of voltage profile to determine discharge capacity. It is only used when ***Load Mode*** = 0 (constant-current) and ***Load Select*** = 6 (user-defined rate).

User Rate-mA is only used if Load Select is set to 6 and ***Load Mode*** = 0. If these criteria are met, then the current stored in this register is used for the *RemainingCapacity()* computation in the Impedance Track algorithm. This is the only function that uses this register.

It is unlikely that this register is used. An example application that requires this register is one that has increased predefined current at the end of discharge. With this application, it is logical to adjust the rate compensation to this period because the IR drop during this end period is affected the moment ***Terminate Voltage*** is reached. The default value is 0.

5.5.1.14 User-Defined Rate-Power

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	75	User Rate-Pwr	I2	0	32767	0	cW

This is the discharge rate used for Impedance Track simulation of voltage profile to determine discharge capacity. It is only used when ***Load Mode*** = 1 (constant-power) and ***Load Select*** = 6 (user-defined rate).

User Rate-Pwr is only used if Load Select is set to 6 and ***Load Mode*** = 1. If these criteria are met, then the power stored in this register is used for the *RemainingCapacity()* computation in the Impedance Track algorithm. This is the only function that uses this register.

It is unlikely that this register is used. An example application that requires this register is one that has increased predefined power at the end of discharge. With this application, it is logical to adjust the rate compensation to this period because the IR drop during this end period is affected the moment ***Terminate Voltage*** is reached. The actual unit of this parameter is dependent on ***Design Energy Scale***. The default value is 0.

5.5.1.15 Reserve Capacity

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	77	Reserve Cap-mAh	I2	0	14500	0	mAh

Reserve Cap-mAh determines how much actual remaining capacity exists after reaching 0 *RemainingCapacity()*, before ***Terminate Voltage*** is reached when ***Load Mode*** = 0 is selected. A loaded rate or no-load rate of compensation can be selected for *Reserve Cap* by setting the ***[RESCAP]*** bit in the ***Pack Configuration*** data flash register. This is a specialized function to allow time for a controlled shutdown after 0 *RemainingCapacity()* is reached.

Carefully select ***Reserve Cap-mAh*** based upon the system requirements. The default value is set to 0 mAh.

5.5.1.16 Maximum and Minimum Delta Voltage

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	84	Max DeltaV	I2	0	32767	200	mV
		86	Min DeltaV	I2	0	32767	0	mV

Max DeltaV:

This is the maximum ***Delta Voltage*** that is saved during discharge cycles. See Section 5.5.3.6 for the description of ***Delta Voltage***. The default is 200 mV.

Min DeltaV:

This is the minimum ***Delta Voltage*** that is saved during discharge cycles. See Section 5.5.3.6 for the description of ***Delta Voltage***. The default is 0 mV.

5.5.1.17 Maximum and Minimum Simulation Rate

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	88	Max Sim Rate	U1	0	255	1	HourRate
		89	Min Sim Rate	U1	0	255	20	HourRate

Max Sim Rate:

Maximum IT simulation rate (inversed). 2 implies C/2. This is the maximum load used in IT simulations in terms of C-rate.

This register defaults to 1.

Min Sim Rate:

Minimum IT simulation rate (inversed). 20 implies C/20. This is the minimum load used in IT simulations in terms of C-rate.

This register defaults to 20.

5.5.1.18 Ra Maximum Delta

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	90	Ra Max Delta	I2	0	32767	54	mΩ

The maximum jump allowed during updates of an Ra table grid point.

Calculate and modify ***Ra Max Delta*** when creating the golden file, set this to 15% of the grid 4 Ra value after optimization cycle is completed, the default is 54.

5.5.1.19 Trace Resistance

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	92	Trace Resistance	I2	0	32767	0	mΩ

Trace Resistance is the nominal resistance between the cell and the coulomb counter measurement point in a given application. Flex cabling and long copper traces on the PCB itself can contribute to this resistance and inject error into the SOC prediction. The fuel gauge will offset cell resistance with this value in order to improve *RemainingCapacity()* estimation.

5.5.1.20 Downstream Resistance

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	94	Downstream Resistance	I2	0	32767	0	mΩ

Downstream Resistance is the nominal resistance between the coulomb counter measurement point and the system voltage node in a given application. Long copper traces on the PCB itself can contribute to this resistance and inject error into the SOC prediction. The fuel gauge will offset cell resistance with this value in order to improve *RemainingCapacity()* estimation.

5.5.1.21 Qmax Maximum Delta Percent

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	96	Qmax Max Delta %	U1	0	100	5	mAh

This is the percent of *DesignCapacity()* to limit how much Qmax may grow or shrink during any one Qmax update

The default is 5%.

5.5.1.22 Qmax Upper Bound Percent

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	97	Qmax Bound %	U1	0	255	130	mAh

Maximum allowed Qmax increase over lifetime of the pack. It is calculated as a fraction of ***Design Capacity***.

5.5.1.23 Delta V Maximum Delta

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	98	DeltaV Max Delta	U2	0	65535	10	mV

Limits on how far ***Delta Voltage*** grows or shrinks on one grid update (in mV).

This register defaults to 10.

5.5.1.24 Maximum and Minimum Resistance Scale

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	100	Max Res Scale	U2	0	32767	5000	Num
		102	Min Res Scale	U2	0	32767	200	Num

Min Res Scale and ***Max Res Scale*** specify allowed change in Ra during Fast Ra Scaling algorithm. Value of 1000 corresponds to 1x and value of 200 corresponds to 0.2x.

5.5.1.25 Fast Scale Load Select

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	105	Fast Scale Load Select	U1	0	6	3	Number

Fast Scale Load Select is used to configure an independent load profile for use with Fast Resistance Scaling Mode. It can be set to any value supported by the standard ***Load Select*** and is useful for systems that exhibit significant load changes near the end of discharge, allowing the gauge to better predict remaining SOC in such cases. The default value for ***Fast Scale Load Select*** is set to 3 (14 s average of the current/power). This makes it more responsive to changes in load near empty, and helps it converge better to 0%. This helps in cases where the discharge is at a relatively light load during most of the discharge, but the load increases dramatically near the end.

5.5.1.26 Charge Hysteresis Voltage Shift

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	106	Charge Hys V Shift	I2	0	2000	40	mV

Charge Hys V Shift is a flash parameter that helps the gauge to avoid Qmax update in the flat region after a charge to avoid OCV hysteresis effects. If OCV (in mV) < Flat region upper bound (typically ~3800 mV) + ***Charge Hys V Shift***, then Qmax update is not allowed.

It is recommended to keep this value at the default setting of 40 mV.

5.5.1.27 Ra Scale OCV Reset Temperature Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	108	RaScl OCV Rst Temp Thresh	U1	0	127	15	°C

RaScl Ocv Rst Temp Thresh determines the temperature threshold at which the scaling factor used in FAST RESISTANCE SCALING mode is reset if a new open-circuit voltage measurement is captured.

5.5.1.28 Maximum Allowed Current

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	109	Max Allowed Current	I2	0	32767	8500	mA

Max Allowed Current is the worst-case current pulse that the system expects to impose on the battery for ***Max Current Pulse Duration***. It is used to compute the reported *Imax()*.

5.5.1.29 Maximum Current Pulse Duration

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	111	Max Current Pulse Duration	U1	0	255	10	s

Max Current Pulse Duration specifies the longest time the ***Max Allowed Current*** is expected to be applied in a given system and is used to compute *Imax()*.

5.5.1.30 Maximum Current Interrupt Step

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	112	Max Current Interrupt Step	I2	–32768	32767	500	mA

Max Current Interrupt Step determines the amount of change in reported *Imax()* required to trigger a new interrupt on the RC2 pin.

5.5.2 Current Thresholds Subclass

5.5.2.1 Discharge and Charge Detection Threshold

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	0	Dsg Current Threshold	I2	0	2000	60	mA
		2	Chg Current Threshold	I2	0	2000	75	mA

Dsg Current Threshold:

This register is used as a threshold by many functions in the fuel gauge to determine if actual discharge current is flowing out of the battery. The *[DSG]* flag in *Flags()* is the method for determining charging or discharging. If the fuel gauge detects discharge *[DSG]* is set to 1 and any other time (charging or relaxation), the *[DSG]* flag is set to 0. Discharge is detected if *AverageCurrent()* < –***Dsg Current Threshold***. Please note that current is negative while discharging.

This threshold should be set low enough to be below any normal application load current but high enough to prevent noise or drift from affecting the measurement (please note that ***Dsg Current Threshold*** is a positive value). The default is 60 mA.

Chg Current Threshold:

This register is used as a threshold by many functions in the fuel gauge to determine if actual charge current is flowing out of the battery. It is independent from the *[CHG]* bit which is used to determine charge termination. This threshold also has no effect on the *[DSG]* bit in the *Flags()* register.

Many algorithms in the fuel gauge require more definitive information about whether current is flowing in the charge or discharge direction. This is what ***Chg Current Threshold*** is used for. The default for this register is 75 mA which is sufficient for most applications.

5.5.2.2 Quit Current

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	4	Quit Current	I2	0	1000	40	mA

Quit Current sets a current threshold to determine when the fuel gauge goes into RELAXATION mode from CHARGE or DISCHARGE mode. The ***Quit Current*** parameter has units of mA. Either of the following criteria must be met to enter RELAXATION mode:

1. *AverageCurrent()* is **less than** (–)***Quit Current*** and then goes within (±)***Quit Current*** for ***Dsg Relax Time***.
2. *AverageCurrent()* is **greater than *Quit Current*** and then goes within (±)***Quit Current*** for ***Chg Relax Time***.

After 30 minutes in RELAXATION mode, the fuel gauge starts checking if the dV/dt < 1 µV/s requirement for OCV readings is satisfied. When the battery relaxes sufficiently to satisfy this criterion, the fuel gauge takes an OCV reading for updating Q_{max}. These updates are used by the Impedance Track algorithm.

It is critical that the battery voltage be relaxed during OCV readings to get the most accurate results. The quit current threshold must not be higher than ***Design Capacity***/20 when attempting to go into RELAXATION mode; however, it should not be so low as to prevent going into RELAXATION mode due to noise. The current threshold that the ***Quit Current*** parameter sets should always be less than the magnitude of the current threshold the ***Chg Current Threshold*** sets and less than the magnitude of the current threshold the ***Dsg Current Threshold*** sets. The default value is set to 40 mA.

5.5.2.3 Discharge and Charge Relax Time

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	6	Dsg Relax Time	U2	0	65535	60	s
		8	Chg Relax Time	U1	0	255	60	s

Dsg Relax Time:

The ***Dsg Relax Time*** is used in the function to determine when to go into RELAXATION mode after discharge current ceases. When *AverageCurrent()* is less than (–)***Quit Current*** and then goes within (±)***Quit Current***, the ***Dsg Relax Time*** timer is initiated. If the current stays within (±)***Quit Current*** until the ***Dsg Relax Time*** timer expires, then the fuel gauge goes into RELAXATION mode. After 30 minutes in RELAXATION mode, the fuel gauge starts checking if the dV/dt < 4 µV/s requirement for OCV readings is satisfied. When the battery relaxes sufficiently to satisfy these criteria, the fuel gauge takes OCV reading for updating Qmax and for accounting for self-discharge. These updates are used in the Impedance Track algorithms.

Be careful when interpreting discharge descriptions in this document while determining the direction and magnitude of the currents, because they are in the negative direction. This is application specific, the default is 60 seconds.

Chg Relax Time:

The ***Chg Relax Time*** is used in the function to determine when to go into RELAXATION mode after charge current ceases. When *AverageCurrent()* is greater than ***Quit Current*** and then goes within (±)***Quit Current***, the ***Chg Relax Time*** timer is initiated. If the current stays within (±)***Quit Current*** until the ***Chg Relax Time*** timer expires, then the fuel gauge goes into RELAXATION mode. After approximately 30 minutes in RELAXATION mode, the fuel gauge attempts to take accurate OCV readings. An additional requirement of dV/dt < 4 µV/s (delta voltage over delta time) is required for the fuel gauge to perform Qmax updates. These updates are used in the Impedance Track algorithms.

This is application specific. Default is 60 seconds.

5.5.2.4 Quit Relax Time

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	9	Quit Relax Time	U1	0	255	1	s

The ***Quit Relax Time*** is a delay time to exit relaxation. If current is greater than ***Chg Current Threshold*** or less than ***Dsg Current Threshold*** and this condition is maintained during ***Quit Relax Time***, then exiting relaxation is permitted.

This is particular to handheld applications in which low duty cycle dynamic loads are possible. Default is 1 second. For very short duration loads, it is permissible to consider the battery to have remained in RELAXATION mode if the loads were not extreme.

5.5.2.5 Maximum IR Correct

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	10	Max IR Correct	I2	0	1000	400	mV

The ***Max IR Correct*** is a maximum IR correction applied to OCV lookup under load. It only applies to OCV lookup after wakeup with detected charge current when gauge needs to establish capacity baseline, but the current is already flowing.

If current is flowing during a voltage measurement that is used for finding initial DOD, IR correction eliminates the effect of the IR drop across the cell impedance and obtain true OCV. ***Max IR Correct*** is the maximum value of IR correction that is used. It is to avoid artifacts due to very high resistance at low DOD values during charge.

This is particular to handheld applications. Default is 400 mV.

5.5.3 State Subclass

5.5.3.1 Qmax Cell 0

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	0	Qmax Cell 0	I2	0	14500	1000	mAh

Qmax contains the maximum chemical capacity of the cell profiles, and is determined by comparing states of charge before and after applying the load with the amount of charge passed. They also correspond to capacity at low rate of discharge, such as C/20 rate. For high accuracy, this value is periodically updated by the gauge during operation. Based on the battery cell capacity information, the initial value of the chemical capacity should be entered in Qmax filed. The Impedance Track algorithm updates this value and maintains it.

Before an optimization cycle is run, set this value to the battery cell datasheet capacity. After the optimization cycle is run and for creation of the golden settings, set it to the learned value. The default is 1000 mAh.

5.5.3.2 Update Status

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	2	Update Status	H1	0x0	0x6	0x0	num

Since this is a pack-side gauge, the Update Status register can be represented by the bits below:

x	x	x	x	x	Bit 2	Bit 1	Bit 0

Three bits in this register are important:

- Bit 2 (0x04) indicates whether the Impedance Track algorithm is enabled.
- Bit 1 (0x02) indicates that the fuel gauge learned optimized values for Qmax and the Ra tables during a learning cycle.
- Bit 0 (0x01) indicates that the fuel gauge learned an initial value for Qmax after the charging portion of a learning cycle.

At the beginning of a learning cycle when creating a golden file, Update Status starts at 0x00. When IT is enabled with the *IT_ENABLE* subcommand being sent to *Control()*, Update Status automatically changes to 0x04. After the charge and relaxation portion of the learning cycle are complete, Update Status should have become 0x05. Finally, after the discharge and relaxation portion of the learning cycle, Update Status becomes 0x06 if the learning cycle was successfully completed. A golden file can then be generated if Update Status was successfully set to 0x06 by the gauge. When the golden file is created, bit 2 is cleared, leaving Update Status = 0x02.

Do not change any of these bits manually. IT must be enabled only by sending the *IT_ENABLE* subcommand to the *Control()* register.

Bit 1 is a status flag that can be set by the fuel gauge as needed. This bit should never be modified except when creating a golden file.

5.5.3.3 Voltage at Charge Termination

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	3	V at Chg Term	I2	0	5000	4350	mV

This is the gauge recorded voltage at charge termination. It is used by the gauge to learn the depth of discharge (DoD) of a full battery for a given system. This is updated by the gauge after every charge termination to account for variations between systems and different temperatures.

V at Chg Term defaults to 4200 mV but can be initialized to the nominal charging voltage of the system.

5.5.3.4 Average Current Last Run

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	5	Avg I Last Run	I2	–32768	0	–299	mA

The fuel gauge logs the *AverageCurrent()* averaged from the beginning to the end of each discharge. It stores this average current from the previous discharge period in this register if the previous discharge lasted at least 500 s.

NOTE: It is recommended that users set ***Avg I Last Run*** to typical values for their system to correctly initialize predictions.

5.5.3.5 Average Power Last Run

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	7	Avg P Last Run	I2	–32768	0	–1131	mA

The fuel gauge logs the power averaged from the beginning to the end of each discharge. It stores this average power from the previous discharge period in this register provided the previous discharge lasted at least 500 seconds. To get a correct average power reading, the fuel gauge continuously multiplies instantaneous current with *Voltage()* to get power. It then logs this data to derive the average power.

NOTE: It is recommended that users set ***Avg P Last Run*** to typical values for their system to correctly initialize predictions.

5.5.3.6 Pulse Delta Voltage

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	9	Delta Voltage	I2	0	32767	2	mV

The maximum difference of *Voltage()* during short load spikes and normal load, so the Impedance Track algorithm can calculate remaining capacity for pulse loads. The ***Delta Voltage*** value is automatically updated by the gauge during operation as voltage spikes are detected. It can be initialized to a higher value if large spikes are typical for the system. Allowable values are limited by *Max Delta V* and *Min Delta V*. During the IT simulations, the target voltage of the empty battery is ***(Terminate Voltage*** + ***Delta Voltage***). This feature allows ***Terminate Voltage*** to be set at the minimum operating voltage of the system with confidence that the 0% point will be reached at a sufficiently high voltage to prevent voltage spikes from crashing the system while still extracting maximum run time from the battery when spikes are small.

Delta Voltage defaults to 2 mV.

5.5.3.7 Thermal Rise Factor

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	11	T Rise	I2	0	32767	50	Num

This is the thermal rise factor that is used in the single time constant heating-cooling thermal modeling. If set to 0, this feature is disabled and simulations in the IT algorithm will not account for self-heating of the battery cell. Larger values of ***T Rise*** lead to higher temperature rise estimates for the IT simulation.

T Rise defaults to 20.

5.5.3.8 Thermal Time Constant

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	13	T Time Constant	I2	0	32767	1000	Num

This is the thermal time constant that is used in single time constant heating-cooling thermal modeling. The default setting can be used, or it can be modified to improve low-temperature accuracy if testing shows the model does not match the actual performance.

T Time Constant defaults to 1000. This is sufficient for many applications. However, it can be modified if better predictive accuracy at low temperatures is desired.

5.6 OCV Table Class

5.6.1 OCVa Table Subclass

5.6.1.1 Chemistry Identification

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
83	OCVa Table	0	Chem ID	H2	0x00	0xFFFF	0x0354	flags

The ***Chem ID*** determines the type of chemistry which is programmed on the gauge. Changing this value by replacing it in data flash has no effect on what is programmed on to the gauge. In order to obtain a new chemistry you must go through an actual chemistry tool. For the fuel gauge, this can be done using the bqCONFIG tool.

It defaults to 0128 when you program the default flash image which can be obtained from the Texas Instruments website.

5.7 Ra Table Class

This data is automatically updated during device operation. Do not make changes except for reading the values from another pre-learned pack for creating *Golden Image Files*. Profiles have format *Cell0 R_a* M where M is the number indicating state of charge to which the value corresponds.

Cell0 R_a flag

xCell0 R_a flag

Each subclass (R_a0 and R_a0x) in the Ra Table class is a separate profile of resistance values normalized at 0 degrees for the cell in a design. The cell has two profiles. They are denoted by the x or absence of the x at the end of the subclass title:

R_a0 or **R_a0x**.

The purpose for two profiles for the cell is to ensure that at any given time at least one profile is enabled and is being used while attempts can be made to update the alternate profile without interference. Having two profiles also helps reduce stress on the flash memory. At the beginning of each of the two subclasses (profiles) is a flag called ***Cell0 R_a flag*** or ***xCell0 R_a flag***. This flag is a status flag that indicates the validity of the table data associated with this flag and whether this particular table is enabled or disabled.

Each flag has two bytes:

1. The least-significant byte (LSB) indicates whether the table is currently enabled or disabled. It has the following options:
   (a) 0x00: means the table had a resistance update in the past; however, it is not the currently enabled table for the cell. (The alternate table for the cell must be enabled at this time.)
   (b) 0xFF: This means that the values in this table are default values. These table resistance values have never been updated, and this table is not the currently enabled table for the cell. (The alternate table for the indicated cell must be enabled at this time.)
   (c) 0x55: This means that this table is enabled for the indicated cell. (The alternate table must be disabled at this time.)
2. The most-significant byte (MSB) indicates the status of the data in this particular table. The possible values for this byte are:
   (a) 0x00: The data associated with this flag has a resistance update and the *Qmax Pack* is updated.
   (b) 0x05: The resistance data associated with this flag is updated and the pack is no longer discharging (this is prior to a *Qmax Pack* update).
   (c) 0x55: The resistance data associated with this flag is updated and the pack is still discharging. (Qmax update attempt not possible until discharging stops.)
   (d) 0xFF: The resistance data associated with this flag is all default data.

This data is used by the fuel gauge to determine which tables need updating and which tables are being used for the Impedance Track algorithm.

This data is used by the Impedance Track algorithm. The only reason this data is displayed and accessible is to allow the resistance data on golden image files to be updated. This description of the ***xCell0 R_a flags*** are intended for information purposes only. It is not intended to give a detailed functional description for the resistance algorithms.

Cell0 R_a0 – Cell0 R_a14

xCell0 R_a0 – xCell0 R_a14

The ***Ra Table*** class has 15 values for each R_a subclass. Each of these values represent a resistance value normalized at 0°C for the associated *Qmax Pack*-based SOC grid point as found by the following rules:

For ***Cell0 R_aM*** where:

1. If 0 ≤ M ≤ 7: The data is the resistance normalized at 0° for: SOC = 100% – (M × 11.1%).
2. If 8 ≤ M ≤ 14: The data is the resistance normalized at 0° for:
   SOC = 100% – [77.7% + (M – 7) × 3.3%)].

This gives a profile of resistance throughout the entire SOC profile of the battery cells concentrating more on the values closer to 0% where resistance quickly increases.

SOC, as stated in this description is based on *Qmax Pack*. It is not derived as a function of SOC. These resistance profiles are used by the fuel gauge for the Impedance Track algorithm. The only reason this data is displayed and accessible is to allow the resistance data on golden image files to be updated. This resistance profile description is for information purposes only. It is not intended to give a detailed functional description for the resistance algorithms. It is important to note that this data is in mΩ units and is normalized to 25°C. The following are useful observations to note with this data throughout the application development cycle:

- Watch for negative values in the **Ra Table** class. Negative numbers in profiles should not be anywhere in this class.

Watch for smooth consistent transitions from one profile grid point value to the next throughout each profile. As the fuel gauge does resistance profile updates, these values should be roughly consistent from one learned update to another without huge jumps in consecutive grid points.

5.7.1 R_a0 Subclass

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
88	R_a0	0	Cell0 R_a flag	H2	0xFF55	0xFF55	0xFF55	
		2	Cell0 R_a 0	I2	0	32767	272	2^{-10} Ω
		4	Cell0 R_a 1	I2	0	32767	316	2^{-10} Ω
		6	Cell0 R_a 2	I2	0	32767	374	2^{-10} Ω
		8	Cell0 R_a 3	I2	0	32767	507	2^{-10} Ω
		10	Cell0 R_a 4	I2	0	32767	360	2^{-10} Ω
		12	Cell0 R_a 5	I2	0	32767	330	2^{-10} Ω
		14	Cell0 R_a 6	I2	0	32767	389	2^{-10} Ω
		16	Cell0 R_a 7	I2	0	32767	345	2^{-10} Ω
		18	Cell0 R_a 8	I2	0	32767	352	2^{-10} Ω
		20	Cell0 R_a 9	I2	0	32767	367	2^{-10} Ω
		22	Cell0 R_a 10	I2	0	32767	374	2^{-10} Ω
		24	Cell0 R_a 11	I2	0	32767	397	2^{-10} Ω
		26	Cell0 R_a 12	I2	0	32767	455	2^{-10} Ω
		28	Cell0 R_a 13	I2	0	32767	808	2^{-10} Ω
		30	Cell0 R_a 14	I2	0	32767	1182	2^{-10} Ω

5.7.2 R_a0x Subclass

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
89	R_a0x	0	xCell0 R_a flag	H2	0xFFFF	0xFFFF	0xFFFF	
		2	Cell0 R_a 0	I2	0	32767	272	2^{-10} Ω
		4	Cell0 R_a 1	I2	0	32767	316	2^{-10} Ω
		6	Cell0 R_a 2	I2	0	32767	374	2^{-10} Ω
		8	Cell0 R_a 3	I2	0	32767	507	2^{-10} Ω
		10	Cell0 R_a 4	I2	0	32767	360	2^{-10} Ω
		12	Cell0 R_a 5	I2	0	32767	330	2^{-10} Ω
		14	Cell0 R_a 6	I2	0	32767	389	2^{-10} Ω
		16	Cell0 R_a 7	I2	0	32767	345	2^{-10} Ω
		18	Cell0 R_a 8	I2	0	32767	352	2^{-10} Ω
		20	Cell0 R_a 9	I2	0	32767	367	2^{-10} Ω
		22	Cell0 R_a 10	I2	0	32767	374	2^{-10} Ω
		24	Cell0 R_a 11	I2	0	32767	397	2^{-10} Ω
		26	Cell0 R_a 12	I2	0	32767	455	2^{-10} Ω
		28	Cell0 R_a 13	I2	0	32767	808	2^{-10} Ω
		30	Cell0 R_a 14	I2	0	32767	1182	2^{-10} Ω

5.8 Calibration Class

5.8.1 Data Subclass

Most of the following values never require modification by the user. They are only modified by the calibration commands in CALIBRATION mode. For calibration using a host system, see Appendix A, *Factory Calibration*.

5.8.1.1 CC Sense Resistor Gain

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	0	CC Gain	F4	1.00E–01	4.00E+01	0.9536	Number

This is the gain factor for calibrating Sense Resistor, Trace, and internal Coulomb Counter (integrating ADC delta sigma) errors. It is used in the algorithm that reports charge and discharge in and out of the battery through the *RemainingCapacity()* register. The difference between ***CC Gain*** and ***CC Delta*** is that the algorithm that reports *AverageCurrent()* cancels out the time base because *AverageCurrent()* does not have a time component (it reports in mA) and ***CC Delta*** requires a time base for reporting *RemainingCapacity()* (it reports in mAh).

5.8.1.2 Coulomb Counter Delta

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	4	CC Delta	F4	2.98E+04	1.19E+06	1119000	Number

This is the gain factor for calibrating Sense Resistor, Trace, and internal Coulomb Counter (integrating ADC delta sigma) errors. It is used in the algorithm that reports charge and discharge in and out of the battery through the *RemainingCapacity()* register. The difference between ***CC Gain*** and ***CC Delta*** is that the algorithm that reports *AverageCurrent()* cancels out the time base because *AverageCurrent()* does not have a time component (it reports in mA) and ***CC Delta*** requires a time base for reporting *RemainingCapacity()* (it reports in mAh).

5.8.1.3 Coulomb Counter Offset

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	8	CC Offset	I2	–32768	32767	1432	mA

Two offsets are used for calibrating the offset of the internal Coulomb Counter, board layout, sense resistor, copper traces, and other offsets from the Coulomb Counter readings. ***CC Offset*** is the calibration value that primarily corrects for the offset error of the Coulomb Counter circuitry. The other offset calibration is ***Board Offset*** and is described next. To minimize external influences when doing ***CC Offset*** calibration by automatic ***CC Offset*** calibration or ***CC Offset*** calibration function in CALIBRATION mode, an internal short is placed across the SRP and SRN pins inside the fuel gauge. ***CC Offset*** is a correction for small noise and errors; therefore, to maximize accuracy, it takes about 20 seconds to calibrate the offset. Because it is impractical to do a 20-s offset during production, two different methods have been selected for calibrating ***CC Offset***.

- The first method is to calibrate ***CC Offset*** by putting the fuel gauge in CALIBRATION mode and initiating the ***CC Offset*** function as part of the entire calibration suite. This is a short calibration that is not as accurate as the second method mentioned below. Its primary purpose is to calibrate ***CC Offset*** enough so that it does not affect any other Coulomb Counter calibrations. This is only intended as a temporary calibration because the automatic calibration is done the first time the I^2C Data and Clock is low for more than 20 seconds, which is a much more accurate calibration.
- During normal Gas Gauge Operation when the I^2C clock and data lines are low for more than 5 seconds and *AverageCurrent()* is less than ***Sleep Current*** in mA, then an automatic ***CC Offset*** calibration is performed. This takes approximately 16 seconds and is much more accurate than the method in Calibration mode.

5.8.1.4 Board Offset

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	10	Board Offset	I1	–128	127	88	µA

Board Offset is the second offset register. Its primary purpose is to calibrate everything the ***CC Offset*** does not calibrate. This includes board layout, sense resistor, copper trace, and other offsets which are external to the chip. The simplified ground circuit design in the fuel gauge requires a separate board offset for each tested device.

5.8.1.5 Internal and External Temperature Offset

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	11	Int Temp Offset	I1	–128	127	0	Number
		12	Ext Temp Offset	I1	–128	127	0	Number

Int Temp Offset:

The fuel gauge has a temperature sensor built into the IC. The ***Int Temp Offset*** is used for calibrating offset errors in the measurement of the reported *Temperature()* if the internal temperature sensor is used. The gain of the internal temperature sensor is accurate enough that a calibration for gain is not required.

Ext Temp Offset:

Ext Temp Offset is for calibrating the offset of the thermistor connected to the TS1 pin of the fuel gauge as reported by *Temperature()*. The gain of the thermistor is accurate enough that a calibration for gain is not required.

5.8.1.6 Pack Voltage Offset

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	13	Pack V Offset	I1	–128	127	0	Number

Pack V Offset is a calibration value that is used to correct for any offset relating to the analog-to-digital converter (ADC) cell voltage measurement.

5.8.2 Current Subclass

5.8.2.1 Filter

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
107	Current	0	Filter	U1	0	255	239	Number

Filter specifies the value for *AverageCurrent()* filter.

5.8.2.2 Deadband

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
107	Current	1	Deadband	U1	0	255	5	mA

Deadband creates a filter window to the reported *AverageCurrent()* register where the current is reported as 0. Any negative current above this value or any positive current below this value is displayed as 0.

This defaults to 5 mA. Only a few reasons may require changing this value:

1. If the fuel gauge is not calibrated.
2. ***Board Offset*** has not been characterized.
3. If the PCB layout has issues that cause inconsistent board offsets from board to board.
4. An extra noisy environment along with reason 3.

5.8.2.3 CC Deadband

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
107	Current	2	CC Deadband	U1	0	255	34	294 nV

CC Deadband creates a filter window below which measured coulomb count is not accumulated. Any coulomb count below this value will be thrown away.

This parameter defaults to 34 294 nV based on a default sense resistor value of 5 mΩ. It should be scaled based on any changes to the sense resistor value in a given design per CC Deadband × (R_{old}/R_{new}).

5.9 Security Class

5.9.1 Codes Subclass

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
112	Codes	0	Sealed to Unsealed	H4	0x00	0xFFFF FFFF	0x3672 0414	
		4	Unsealed to Full	H4	0x00	0xFFFF FFFF	0xFFFF FFFF	
		8	Authen Key3	H4	0x00	0xFFFF FFFF	0x0123 4567	
		12	Authen Key2	H4	0x00	0xFFFF FFFF	0x89AB CDEF	
		16	Authen Key1	H4	0x00	0xFFFF FFFF	0xFEDC BA98	
		20	Authen Key0	H4	0x00	0xFFFF FFFF	0x7654 3210	

5.9.1.1 Sealed to Unsealed

This register contains the security code to transition the device from SEALED mode to UNSEALED mode.

The default code is set to 0x36720414.

5.9.1.2 Unsealed to Full Access

This register contains the security code to transition the device from UNSEALED mode to FULL ACCESS mode.

The default code is set to 0xFFFFFFFF.

5.9.1.3 Authentication Keys

This is the register to store the SHA-1 authentication key to allow a system to authenticate the battery pack.

The default key is set to 0x0123456789ABCDEFFEDCBA9876543210.

Reference Schematic

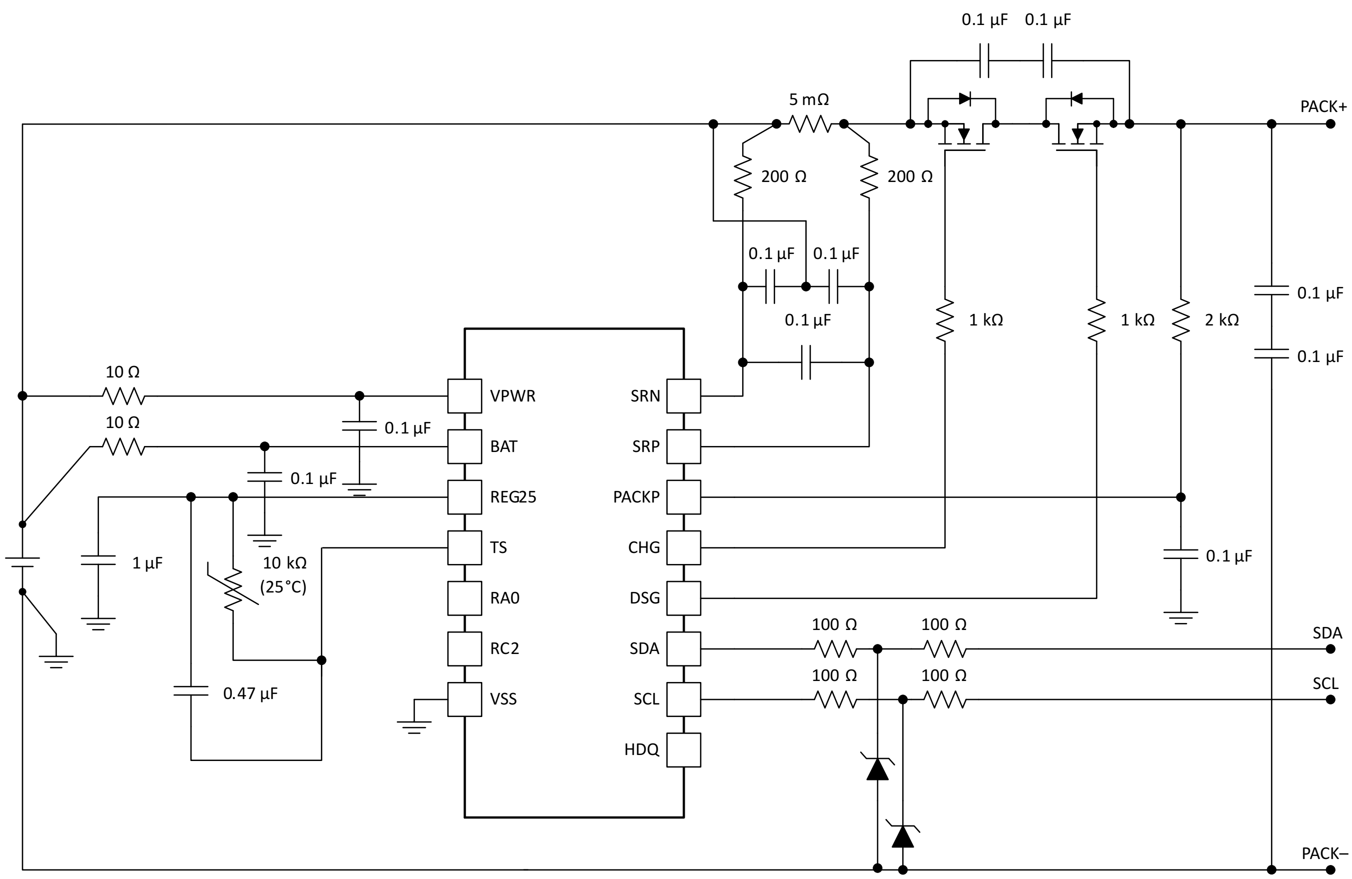


Figure 6-1. I²C Mode Schematic

Factory Calibration

The bq27742-G1 fuel gauge requires factory calibration. The gauge performs only a limited number of calibration functions. The rest must be performed by a host system using commands provided by the gauge for this purpose. The following sections give a detailed description of the various calibration sequences with the help of flow charts.

A.1 General I²C Command Information

In the following flow charts, all I²C functions take 3 arguments. Write command arguments:

- Address
- Data
- Wait time in ms

Read command arguments:

- Address
- Number of bytes read
- Wait time in ms

A.2 Calibration

A.2.1 Method

The calibration method is broken up into the following sections. The first four sequences are subroutines to be used in the main calibration sequences.

- Section A.3, *Enter Calibration Mode*
- Section A.4, *Exit Calibration Mode*
- Section A.7, *Obtain Raw Calibration Data*
- Section A.11, *Floating Point Conversion*
- Section A.5, *CC Offset*
- Section A.6, *Board Offset*
- Section A.8, *Current Calibration*
- Section A.9, *Voltage Calibration*
- Section A.10, *Temperature Calibration*

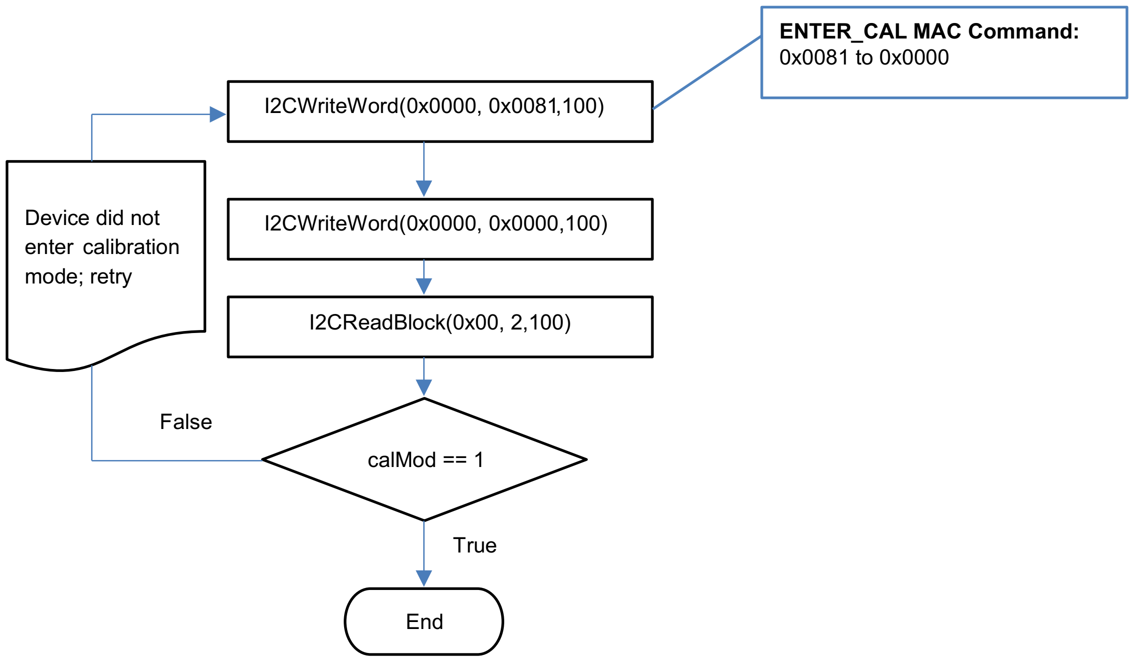
A.2.2 Sequence

Perform the following calibration sequence during battery pack manufacturing process:

1. Perform CC Offset.
2. Perform Board Offset.
3. Perform Current Calibration.
4. Perform Voltage Calibration.
5. Perform Temperature Calibration.
6. Write calibration results to data flash.

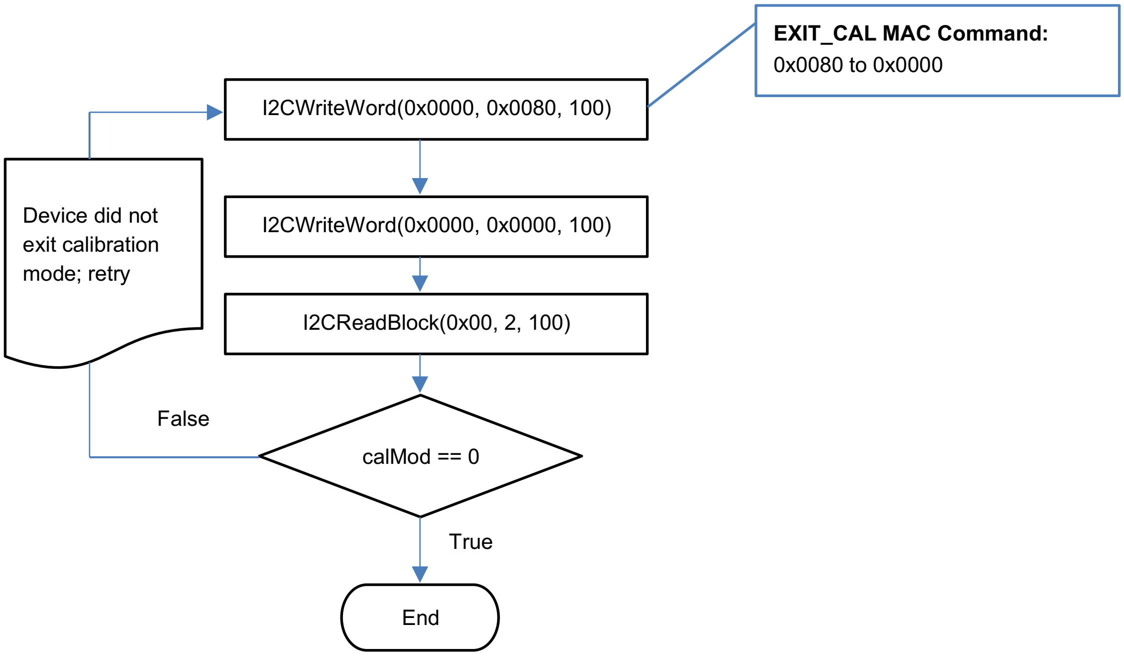
A.3 Enter Calibration Mode

This sequence puts the gauge into CALIBRATION mode. These steps must be performed when gauge is in UNSEALED mode.



A.4 Exit Calibration Mode

This sequence takes gauge out of CALIBRATION mode. These steps must be performed when gauge is in UNSEALED mode.



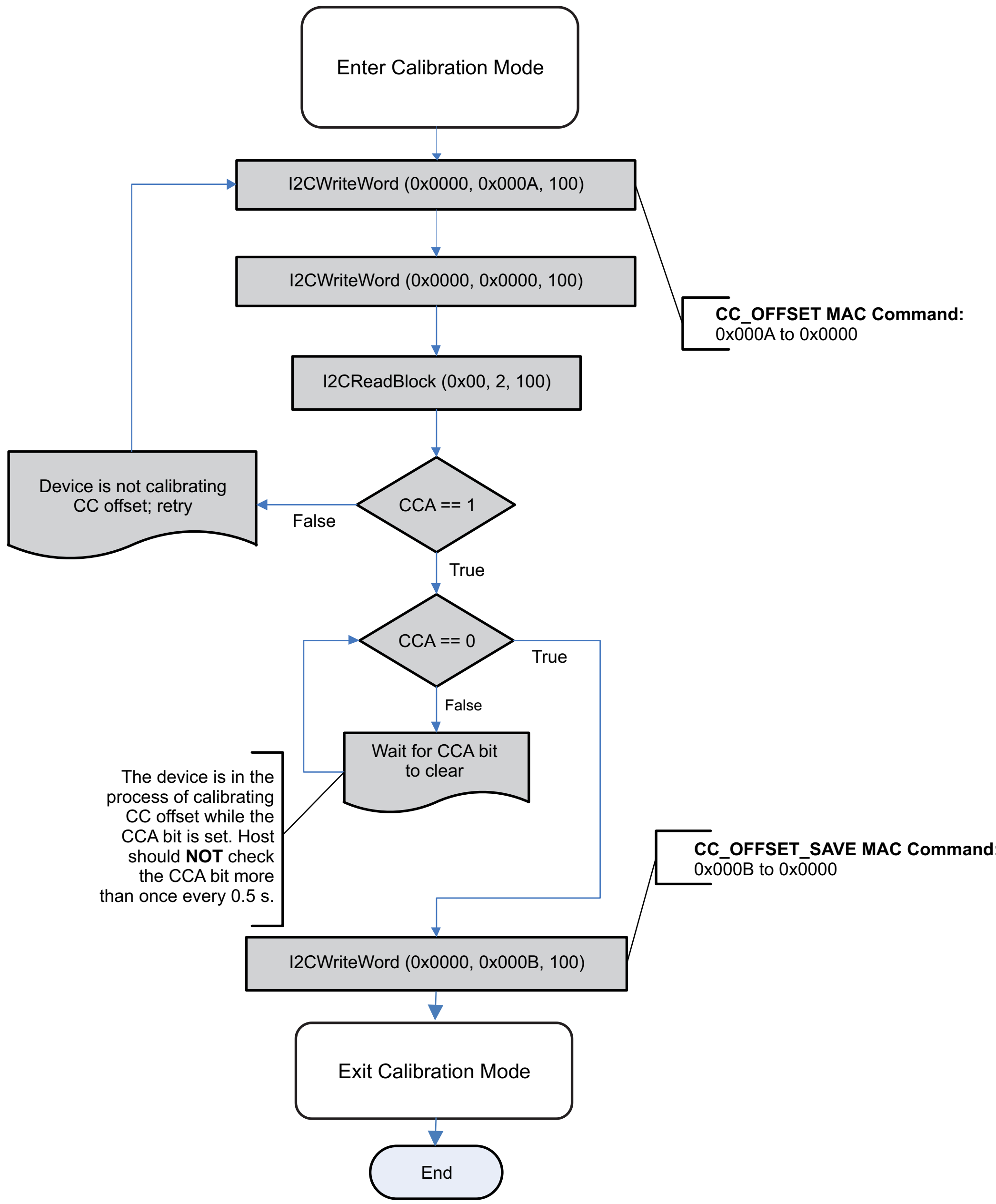
A.5 CC Offset

Use MAC commands for ***CC Offset*** calibration. The host system does not need to write information to the Data Flash (DF). See Section 4.1.1.1 for the description of the *CONTROL_STATUS[CCA]* bit. The host system needs to make sure the fuel gauge is unsealed.

NOTE: While the device is calibrating the ***CC Offset***, the host system must not read the *CONTROL_STATUS* register at a rate greater than once every 0.5 seconds.

NOTE: The step labeled **Enter Calibration Mode** refers to Section A.3, *Enter Calibration Mode*.

The step labeled **Exit Calibration Mode** refers to Section A.4, *Exit Calibration Mode*.



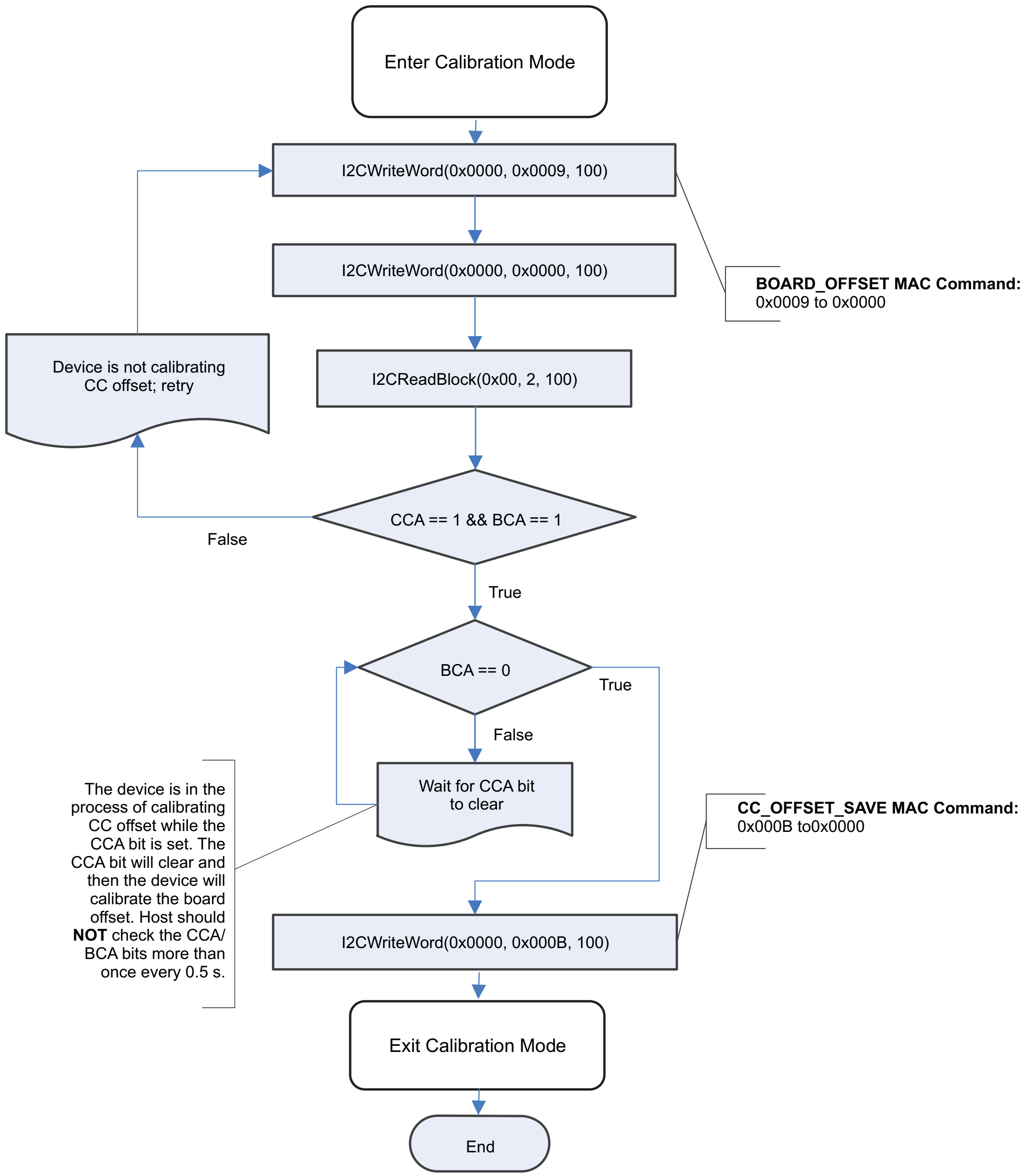
A.6 Board Offset

Use MAC commands for ***Board Offset*** calibration. The host system does not need to write information to the DF. The host system needs to make sure the fuel gauge is unsealed. See Section 4.1.1.1 for the description of the *CONTROL_STATUS[CCA]* and *[BCA]* bits. Note that calculating the ***Board Offset*** will also calculate the ***CC Offset***, therefore, it is not necessary to go through the ***CC Offset*** calibration process if the ***Board Offset*** calibration process is implemented.

NOTE: While the device is calibrating the ***CC Offset***, the host system should not read the *CONTROL_STATUS()* register at a rate greater than once every 0.5 seconds.

NOTE: The step labeled **Enter Calibration Mode** refers to Section A.3, *Enter Calibration Mode*.

The step labeled **Exit Calibration Mode** refers to Section A.4, *Exit Calibration Mode*.

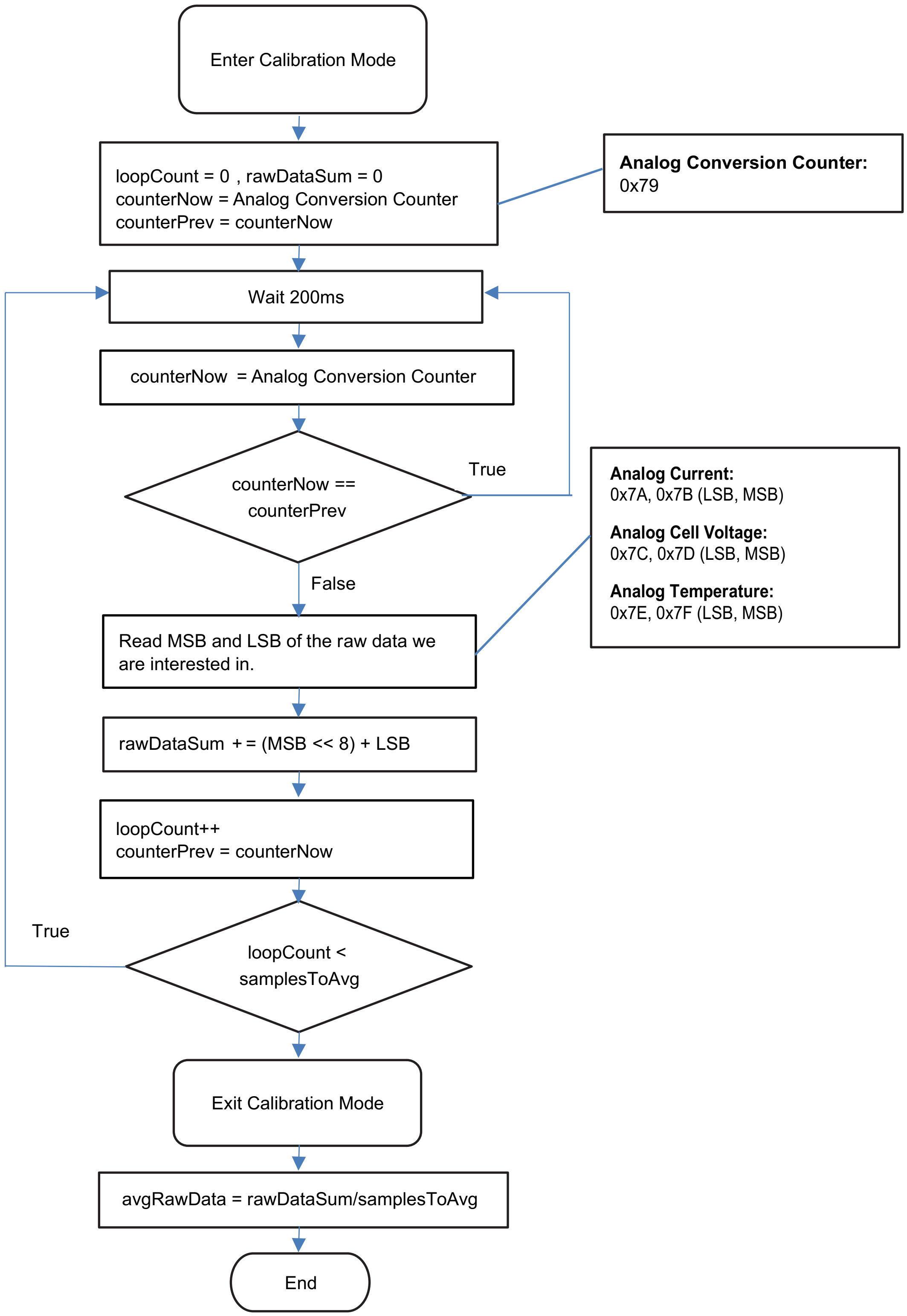

Enter Calibration Mode
I2CWriteWord(0x0000, 0x0009, 100)
I2CWriteWord(0x0000, 0x0000, 100)
BOARD_OFFSET MAC Command:
0x0009 to 0x0000
Device is not calibrating CC offset; retry
I2CReadBlock(0x00, 2, 100)
CCA == 1 && BCA == 1
False
True
BCA == 0
True
False
Wait for CCA bit to clear
The device is in the process of calibrating CC offset while the CCA bit is set. The CCA bit will clear and then the device will calibrate the board offset. Host should NOT check the CCA/ BCA bits more than once every 0.5 s.
CC_OFFSET_SAVE MAC Command:
0x000B to0x0000
I2CWriteWord(0x0000, 0x000B, 100)
Exit Calibration Mode
End

A.7 Obtain Raw Calibration Data

The following flow chart demonstrates how the host system obtains the raw data to calibrate current, voltage, and temperature. The host system uses this flow in conjunction with the Current, Voltage, and Temperature flows described in this appendix. It is recommended that the host system samples the raw data multiple times, at a rate of once per second, to obtain an average of the raw current, voltage and temperature. The host system needs to make sure the fuel gauge is UNSEALED.

NOTE: The step labeled **Enter Calibration Mode** refers to Section A.3, *Enter Calibration Mode*.

The step labeled **Exit Calibration Mode** refers to Section A.4, *Exit Calibration Mode*.

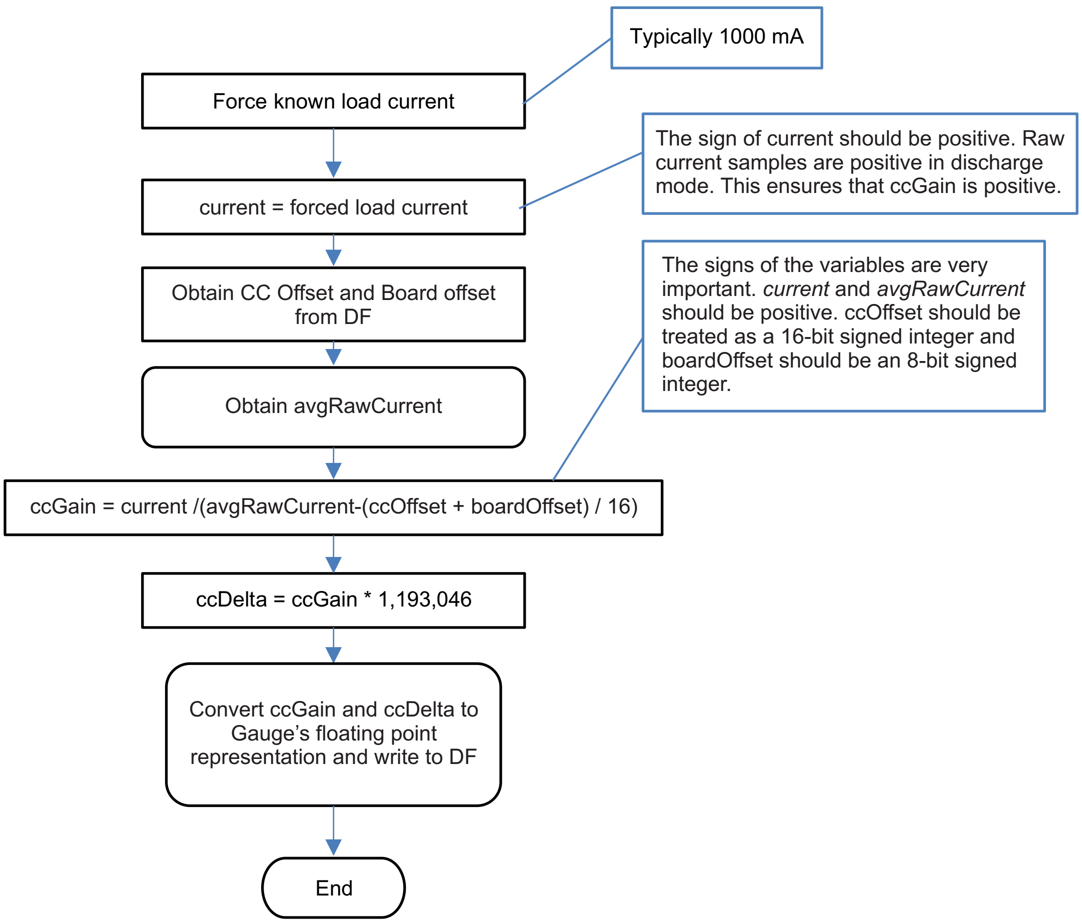

Enter Calibration Mode
loopCount = 0 , rawDataSum = 0
counterNow = Analog Conversion Counter
counterPrev = counterNow
Analog Conversion Counter:
0x79
Wait 200ms
counterNow = Analog Conversion Counter
counterNow == counterPrev
True
False
Analog Current:
0x7A, 0x7B (LSB, MSB)
Analog Cell Voltage:
0x7C, 0x7D (LSB, MSB)
Analog Temperature:
0x7E, 0x7F (LSB, MSB)
Read MSB and LSB of the raw data we are interested in.
rawDataSum += (MSB << 8) + LSB
loopCount++
counterPrev = counterNow
True
loopCount < samplesToAvg
Exit Calibration Mode
avgRawData = rawDataSum/samplesToAvg
End

A.8 Current Calibration

The ***CC Gain*** and ***CC Delta*** are two calibration parameters of concern for current calibration. A known load, typically 1000 mA, is applied to the device during this process. Details on converting the ***CC Gain*** and ***CC Delta*** to floating point format are in Section A.11, *Floating Point Conversion*. The host system needs to ensure the fuel gauge is UNSEALED.

NOTE: The step labeled **Obtain avgRawCurrent** refers to Section A.7, *Obtain Raw Calibration Data*.

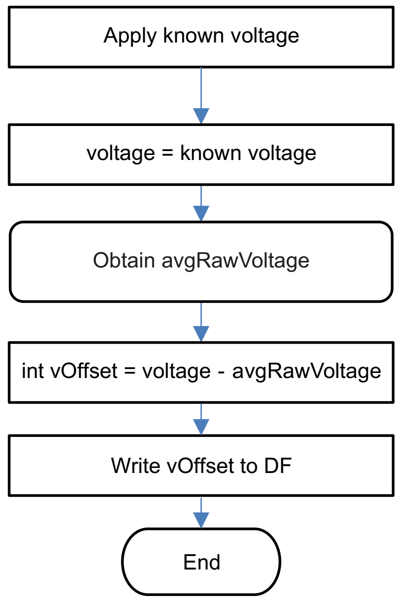
The step labeled **Convert ccGain and ccDelta to Gauge's floating point representation and write to DF** refers to Section A.11, *Floating Point Conversion*.



A.9 Voltage Calibration

A known voltage must be applied to the device for voltage calibration. The calculated voltage offset must be written to the corresponding location in DF. The voltage offset is represented by an integer that is a single byte in size and can be written to the appropriate location in DF without any intermediate steps. The host system needs to ensure the fuel gauge is UNSEALED.

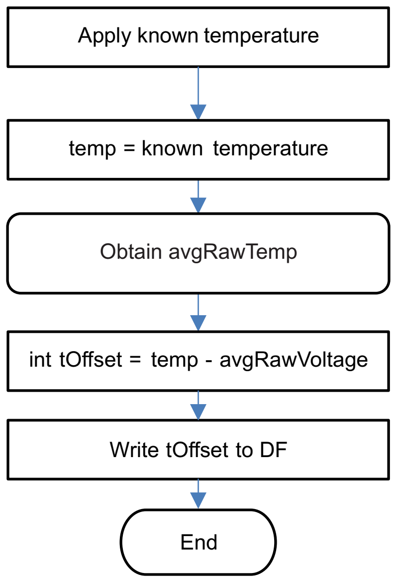
NOTE: The step labeled **Obtain avgRawVoltage** refers to Section A.7, *Obtain Raw Calibration Data*.



A.10 Temperature Calibration

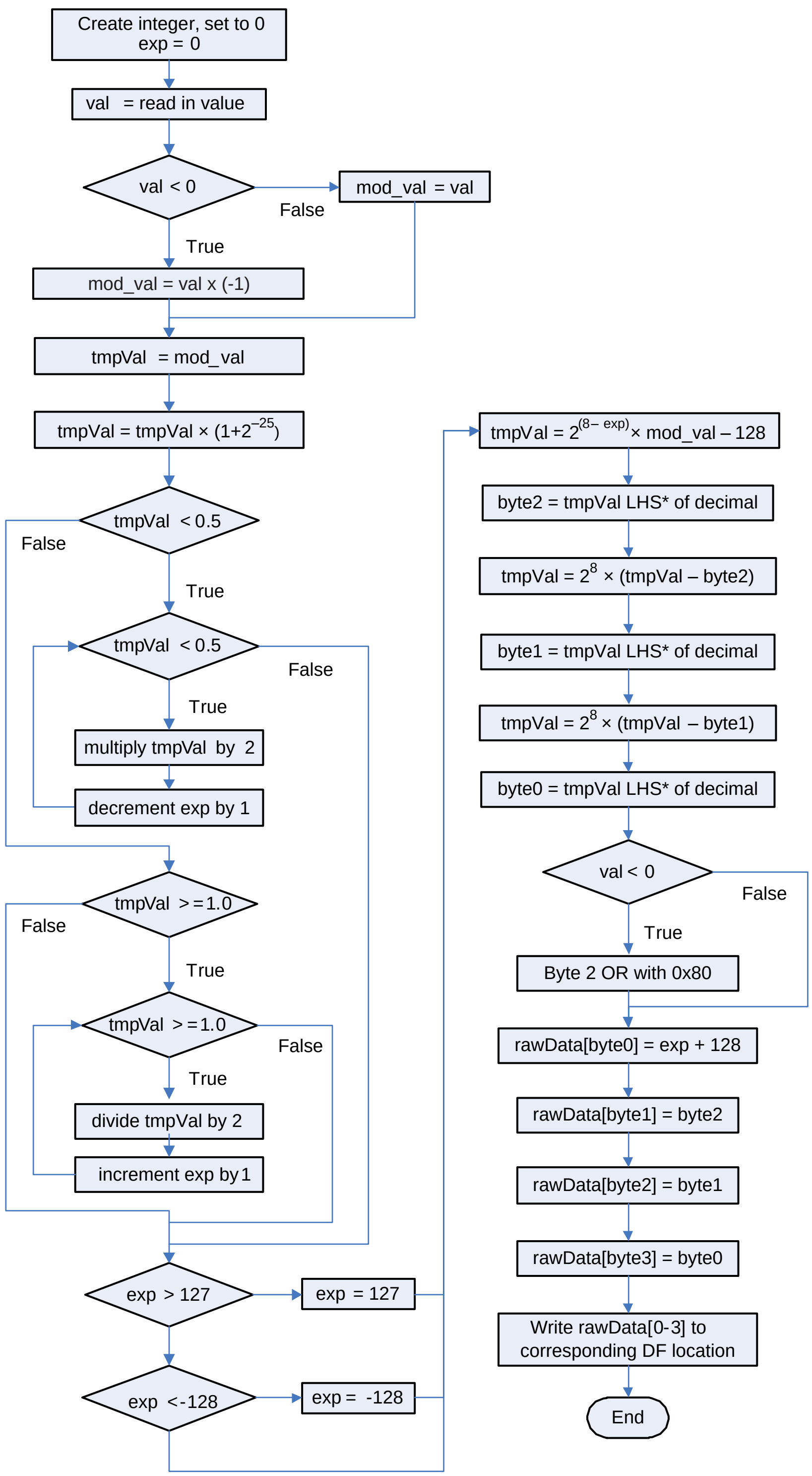
A known temperature must be applied to the device for temperature calibration. The calculated temperature offset is written to the corresponding location in DF. The temperature offset is represented by an integer that is a single byte in size and can be written to the appropriate location in DF without any intermediate steps. The host system needs to ensure the fuel gauge is unsealed.

NOTE: The step labeled **Obtain avgRawTemp** refers to Section A.7, *Obtain Raw Calibration Data*.



A.11 Floating Point Conversion

This section details how to convert the floating point ***CC Gain*** and ***CC Delta*** values to the format understood by the gauge.



* LHS is an abbreviation for Left Hand Side. This refers to truncating the floating point value by removing anything to the right of the decimal point.

Glossary

ACK	Acknowledge character
ADC	Analog-to-digital converter
BCA	Board calibration
CC	Coulomb counter
CCA	Coulomb counter calibration
CE	Chip enable
CHARGE Mode	Refers to a mode to where the gauge reads *AverageCurrent()* > ***Chg Current Threshold*** for at least 1 second.
Clear	Refers to a bit in a register becoming a logic LOW or 0. The Battery Management Studio (bqStudio) software represents a clear bit with the color green.
C Rate	C rate corresponds to discharge current that will discharge the battery in one hour, which is equal to full capacity of the battery in mAh.
cWh	Centiwatt-hour
DF	Data flash
DISCHARGE Mode	Refers to a mode where the gauge read *AverageCurrent()* < ***(–)Dsg Current Threshold*** for at least 1 second.
DOD	Depth of discharge in percent as related to Qmax. 100% corresponds to empty battery.
DOD0	Depth of discharge that was looked up in the DOD (OCV) table based on OCV measurement in relaxed state.
EOC	End of charge
FC	Fully charged
FCC	Full charge capacity. Total capacity of the battery compensated for present load current, temperature, and aging effects (reduction in chemical capacity and increase in internal impedance).
FIFO	First in, first out
Flag	This word usually represents a read-only status bit that indicates some action occurred or is occurring. This bit typically cannot be modified. The flags are set and cleared automatically by the gauge.
FVCA	Fast voltage and current acquisition
GPIO	General-purpose input output
IC	Integrated circuit
ID	Identification
IO	Input or output
IT	Impedance Track
I^2C	Inter-integrated circuit
LDO	Low dropout
LSB	Least significant bit
LT	Lifetime
MAC	Manufacturer access command or control command
mAh	Milliamp-hour
MSB	Most significant bit
mWh	Milliwatt-hour
NACK	Negative acknowledge character
NTC	Negative temperature coefficient
OCV	Open-circuit voltage. Voltage measured on fully-relaxed battery with no load applied.
OTC	Overtemperature in charge
OTD	Overtemperature in discharge
Qmax	Maximum chemical capacity
Qpass	Qmax Passed Charge. The amount of charge passed between two DOD0 points required for learning Qmax.
RDIS	Resistance update disabled

Rem Cap	Present remaining capacity in the battery compensated for present load current, temperature, and aging effects (reduction in chemical capacity and increase in internal impedance).
RM	Remaining capacity
RW	Read or write
SCL	Serial clock: programmable serial clock used in the I^2C interface
SDA	Serial data: serial data bus in the I^2C interface
SE	Shutdown enable
Set	Refers to a bit in a register becoming a logic HIGH or 1. The Battery Management Studio (bqStudio) software represents a set bit with the color red.
SOC	State-of-charge in percent related to FCC
SOC1	State-of-charge initial
SOCF	State-of-charge final
System	The word system is sometimes used in this document. When used, it always means a host system that is consuming current from the battery pack.
TCA	Terminate charge alarm
TS	Temperature status
TTE	Time-to-empty
TTF	Time-to-full
VOK	Indicates that Qmax has been saved to data flash. This bit is located on *CONTROL_STATUS* register bit 1.

Revision History

Changes from B Revision (December 2014) to C Revision **Page**