# BQ27Z561-R2

# Technical Reference

Texas Instruments

# Contents

# *Preface*

## Read this First

This manual discusses the modules and peripherals of the BQ27Z561-R2 device, and how each is used to build a complete battery pack gas gauge solution. *BQ27Z561-R2 Impedance Track™ Battery Gas Gauge Solution for 1-Series Cell Li-Ion Battery Packs Data Sheet* (SLUSE22).

## Notational Conventions

The following notation is used if standard commands and data flash (DF) values are mentioned within a text block:

- Standard commands: *italics* with parentheses and no breaking spaces; for example, *RemainingCapacity()*
- Data flash (DF): *italics*, **bold**, and breaking spaces; for example, ***Design Capacity***
- Register bits and flags: *italics* and brackets; for example, *[TDA]*
- Data flash bits: *italics* and **bold**; for example, ***[LED1]***
- Modes and states: ALL CAPITALS; for example, UNSEALED

## Trademarks

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

## Glossary

*TI Glossary* — This glossary lists and explains terms, acronyms, and definitions.

# *Introduction*

The BQ27Z561-R2 device provides a feature-rich gas gauging solution for single-cell battery pack applications. The device has extended capabilities, including:

- Fully integrated single-cell li-ion or li-polymer cell battery gauge solution
- Next-generation patented Impedance Track™ technology accurately measures available charge in li-ion and li-polymer batteries
- Power modes:
  – NORMAL mode
  – SLEEP mode
  – DEEP SLEEP mode
  – OFF mode (CE ≤ $V_{IL}$)
- Information available to external smart charger
  – JEITA
  – Charging current
  – Charging voltage
- Diagnostic lifetime data monitor
- Supports two-wire I$^2$C interface
- Supports one-wire HDQ interface
- SHA-256 authentication
- Ultra-compact package: 12-ball DSBGA

# Basic Measurement System

## 2.1 Introduction

The BQ27Z561-R2 gauge contains an integrating analog-to-digital converter (ADC) for current measurement and a second order delta-sigma ADC for cell voltage and temperature measurements.

## 2.2 Current and Coulomb Counting

The integrating delta-sigma ADC in the gauge measures the charge/discharge flow of the battery by measuring the voltage drop across a small-value sense resistor between the SRP and SRN pins. The 16-bit integrating ADC measures bipolar signals from –0.10 V to 0.10 V with 3.74-µV resolution. The gauge reports charge activity when VSR = $V_{(SRP)}$ – $V_{(SRN)}$ is positive, and discharge activity when VSR = $V_{(SRP)}$ – $V_{(SRN)}$ is negative. This data is scaled and translated into mA using **CC Gain** and reported through *Current()*. The gauge uses this information for gauging and lifetime data logging functions. In NORMAL and SLEEP modes, the gauge continuously monitors the measured current and integrates the digital signal over time using an internal coulomb counter, which is also used for gauging.

In addition, the integrating ADC is sampled during voltage measurements and converted to mA in the same manner. This data is used for gauging (but not coulomb counting) and reported through Cell 1 Current in *DAStatus1()*.

The data reported through *Current()* can also have a deadband applied to it. This removes any noise or offset that has not been calibrated out from being reported as real current or from being accumulated by the coulomb counter in DOD0 Passed Q. This deadband value is programmed in **Deadband**.

## 2.3 Voltage

The second order delta-sigma ADC in the gauge measures the cell voltage at 1-s intervals in NORMAL mode, and **Voltage Time**-second intervals in SLEEP mode. This data is scaled and translated into mV using **Cell Gain** and reported through *Voltage()*. The gauge uses this information for gauging and lifetime data logging functions.

## 2.4 Temperature

### 2.4.1 Internal Temperature

The second order delta-sigma ADC in the gauge measures internal temperature at 1-s intervals in NORMAL mode and **Voltage Time**-second intervals in SLEEP mode. This data is translated into 0.1 K using the parameters in **Internal Temp Model** and reported through *InternalTemperature()*. The internal temperature can be used for gauging and lifetime data logging functions, and reported through *Temperature()* if **Temperature Enable[TSInt]** = 1.

### 2.4.2 Cell Temperature

The second order delta-sigma ADC in the gauge measures cell temperature via an external thermistor at 1-s intervals in NORMAL mode and **Voltage Time**-second intervals in SLEEP mode. This data is translated into 0.1 K using the parameters in **Cell Temp Model** and reported through *Temperature()* if **Temperature Enable[TS1]** = 1. The cell temperature can be used for gauging and lifetime data logging functions if **Temperature Enable[TS1]** = 1.

The cell temperature measurement requires an external 10-kΩ negative temperature coefficient (NTC) thermistor, such as the Semitec 103AT–2, connected between VSS and the TS pin.

### 2.4.3 Temperature Configuration

The following data flash parameter enables/disables the available temperature sensor options.

| Class | Subclass | Name | Type | Min | Max | Default | Description |
|---|---|---|---|---|---|---|---|
| Settings | Configuration | Temperature Enable | H1 | 0x00 | 0x03 | 0x02 | Determines whether the internal temperature (TSInt) or the external temperature (TS1 on TS pin) is reported with *Temperature()*.<br>Bit 0: TSInt—Enables internal TS<br>Bit 1: TS1—Enables external TS pin<br>0, 0: Neither enabled, *Temperature()* reports −273.2K.<br>0, 1: Enables internal TS, *Temperature()* reports the internal temperature.<br>1, 0: Enables TS1, *Temperature()* reports the value determined by thermistor on the TS pin.<br>1, 1: Both TSInt and TS1 are enabled, *Temperature()* reports the higher of the two values.<br>Bits 7:2: Reserved |

# Power Modes

## 3.1 Introduction

To enhance battery life, the BQ27Z561-R2 supports several power modes to minimize power consumption during operation.

## 3.2 NORMAL Mode

In NORMAL mode (also referred to as ACTIVE mode), the device takes voltage, current, and temperature readings every 1 s, performs gauging calculations, updates data, and makes status selections at 1-s intervals. Between these periods of activity, the device is in a reduced power state.

## 3.3 SLEEP Mode

### 3.3.1 Device Sleep

When the sleep conditions are met, the device goes into SLEEP mode with periodic wake-ups to reduce power consumption. The device returns to NORMAL mode if any exit sleep condition is met.

| Status | Condition | Action |
|---|---|---|
| Activate | *DA Config[SLEEP]* = 1 AND $\|Current()\| \leq$ *Sleep Current* AND *Voltage Time* > 0 AND *OperationStatusB()[DPSLEEPM]* = 0 AND Device in RELAX mode | The device goes to SLEEP mode. The device wakes up every *Voltage Time* to measure voltage and temperature. The device continues to coulomb count and update current every 4 s. Tip: Synchronize these two times to save power. |
| Exit | *DA Config[SLEEP]* = 0 OR $\|Current()\| >$ *Sleep Current* OR *Voltage Time* = 0 OR *OperationStatusB()[DPSLEEPM]* = 1 | Return to NORMAL mode |

The configuration options for SLEEP are in the following data flash.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Power | Sleep | Sleep Current | I2 | 0 | 32767 | 15 | mA | $\|Current()\|$ threshold to enter SLEEP mode |
| Power | Sleep | Voltage Time | U1 | 1 | 20 | 5 | s | Voltage sampling period in SLEEP mode |

## 3.4 DEEP SLEEP Mode

DEEP SLEEP mode lowers power consumption and is entered through SLEEP mode. In DEEP SLEEP mode, the device takes data similar to SLEEP mode. The gauge's current consumption is reduced by measuring current periodically, instead of continuously. Exit from this mode is through a current level or a command, as indicated in the table below.

| Status | Condition | Action |
|---|---|---|
| Activate | (*Power Config[AUTO_DP_SLP_EN]* = 1 OR *OperationStatusB()[DPSLEEPM]* = 1) AND \|wake comparator current\| ≤ *Deep Sleep Current* | The device goes to DEEP SLEEP mode. The device wakes up every *Deep Sleep Voltage Time* to measure voltage, current, and temperature. The device updates *RemainingCapacity()* every 1 s, based on the last measured current. The device performs a low resolution measurement of wake comparator current every *Wake Check Time*. Wake comparator current is overwritten by the value of *Current()* when it updates at *Deep Sleep Voltage Time* (true for all modes). |
| Exit | (*Power Config[AUTO_DP_SLP_EN]* = 0 AND *OperationStatusB()[DPSLEEPM]* = 0) OR \|wake comparator current\| > *Deep Sleep Current* | Return to SLEEP mode |

The configuration options for DEEP SLEEP are in the following data flash.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Power | Sleep | Deep Sleep Voltage Time | U1 | 1 | 20 | 20 | s | Determines the wake interval to measure voltage, current, and temperature in DEEP SLEEP mode |
| Power | Sleep | Wake Check Time | U1 | 0 | 20 | 1 | s | Determines the wake interval to determine if required to return to SLEEP mode |
| Power | Sleep | Deep Sleep Current | I2 | 0 | 32767 | 10 | mA | \|Wake Comparator Current\| must be lower than this value to allow DEEP SLEEP mode entry. |

## 3.5 HIBERNATE Mode

This is the lowest power mode for the device. This mode can be entered only if the BQ27Z561-R2 has entered SLEEP mode. In this mode, all measurement and clocking circuits are turned off; thus, the mode should only be entered when no charging or discharging of the battery is expected (for instance, when the pack is out of the system). Table 3-1 and Table 3-2 show the conditions for entry and exit of the mode.

**Table 3-1. Basic Hibernate**

| Status | Condition | Action |
|---|---|---|
| Activate | (*Power Config[HIBERNATE]* = 1 AND *OperationStatusB()[SLEEP]* = 1) AND *OperationStatusB()*.HIBERNATE = 1 | The device goes to HIBERNATE mode. |
| Exit | An I2C message to the BQ27Z561-R2 is received. | The device returns to SLEEP mode. |

The BQ27Z561-R2 can be configured to reenter HIBERNATE mode in the case of errant communication or noise on the I2C communication bus. Table 3-2 shows the configuration items for the mode.

**Table 3-2. Hibernate with Automatic Reentry**

| Status | Condition | Action |
|---|---|---|
| Activate | (*Power Config[HIBERNATE]* = 1 AND *Power Config[HIB_AUTO]* = 1 AND *OperationStatusB()[SLEEP]* = 1) AND *OperationStatusB()*.HIBERNATE = 1 | The device goes to HIBERNATE mode. |
| Exit | Two I2C messages to the BQ27Z561-R2 are received within *Power.Auto Time* seconds. | The device returns to SLEEP mode. |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Power | Hibernate | Auto Time | U1 | 0 | 20 | 2 | s | The device exits HIBERNATE mode if two I2C communications are within the time window |

## 3.6   Power Modes Configuration

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Settings | Configuration | Power Configuration | H1 | 0x00 | 0xFF | 0x10 | — | Bit 0: HIB_AUTO<br>0 = Disables automatic hibernation reentry<br>1 = Enables automatic hibernation reentry<br>Bit 1: HIBERNATE<br>0 = Disables the Hibernate MAC<br>1 = Enables the Hibernate MAC<br>Bits 3:2 Reserved<br>Bit 4: SLP_ACCUM<br>0 = Disables CC accumulation during SLEEP mode<br>1 = Enables CC accumulation during SLEEP mode<br>Bit 5: Reserved<br>Bit 6: AUTO_DP_SLP_EN<br>0 = Automatic entry into DEEP SLEEP mode disabled<br>1 = Automatic entry into DEEP SLEEP mode enabled<br>Bit 7: Reserved |
| Settings | Configuration | DA Configuration | H2 | 0x0000 | 0x0010 | 0x0010 | — | Bits 3:0: Reserved<br>Bit 4: SLEEP—SLEEP mode<br>0 = Disables SLEEP mode<br>1 = Enables SLEEP mode (default)<br>Bits 15:5: Reserved |

# *Gauging*

## 4.1 Impedance Track (IT) Model

Predicting available energy from lithium-ion cells requires a number of measurable parameters. The Impedance Track (IT) model measures, records, and predicts available capacity or energy from these parameters. The model provides the means to accurately determine available energy over the life of a cell by accounting for aging and present environmental conditions. Aging can be accounted for by measuring QMax and cell impedances as the cell is cycled. QMax is the amount of charge available in a fully charged cell. The present conditions include consideration of cell and ambient temperature as well as the present or predicted power requirements. There are three major components of Impedance Track.

- One is measurement of QMax.
- Two is measurement of cell impedance.
- Three is the calculation of capacities.

All three components rely on the relationship of the open circuit voltage (OCV) versus depth of discharge (DOD). DOD is the percent of QMax removed from a cell during a no load discharge and goes to 100% when the cell is fully discharged. The OCV versus DOD relationship is fixed for each particular cell chemistry. QMax is determined by measuring the charge passed between two DOD points; the DOD points being determined from OCV measurements during Relax mode. Cell impedance is measured during constant load times during discharge from the difference of measured voltage to the OCV voltage based on the known DOD. The DOD determined from the most recent OCV reading and adjusted by the passed charge during the discharge. In general, capacities are recalculated based on changes in temperature, load or impedance. Capacity calculations use the known DOD, load, temperature, end-of-discharge voltage, *TermV*, and *QMax*.

## 4.2 Thermal Model

Because temperature changes significantly during the course of a discharge, the algorithm needs to be able to predict the future temperature. This is required for temperature correction of battery impedance during voltage simulation near the end of discharge. To achieve this, the algorithm collects temperature-time dependence data during discharge. This data is used to update parameters of a simple thermal model, including a heat exchange coefficient, *temp k*, and a thermal time constant, *temp k*. These parameters are updated at the same time as resistances. The algorithm also records the outside temperature during relaxation periods. These parameters are used to define a temperature function T(t, T_start) that calculates a temperature profile starting from the present temperature, T_start, and continuing until the end of discharge in the simulation calculation.

## 4.3 Gauging Capacities

The BQ27Z561-R2 measures individual cell voltages, pack voltage, temperature, and current. When at rest and *GaugingStatus[REST]* is set or when cell impedance is updated, the Impedance Track capacity calculation determines battery state-of-charge from DOD and temperature. The BQ27Z561-R2 measures charge and discharge activity by integrating the voltage across a small-value series sense resistor (1 mΩ typ.) between the negative terminal of the cell stack and the negative terminal of the battery pack. The battery state-of-charge is subsequently adjusted during load or charger application, using the integrated charge passed through the battery.

The device is capable of supporting a maximum battery pack capacity of 32 Ah. See the *Theory and Implementation of Impedance Track™ Battery Fuel-Gauging Algorithm in the bq20zxx Product Family Application Report* (SLUA364) for further details of the Impedance Track calculations.

The default for Impedance Track gauging is *off*. To enable the gauging function, set **Manufacturing Status[GAUGE_EN]** = 1. The gauging function will be enabled after a reset or a seal command is set. Alternatively, the MAC command *Gauging()* can be used to turn on and off the gauging function. *Gauging()* takes an immediate effect and the **[GAUGE_EN]** is also updated accordingly.

The *ITStatus1()*, *ITStatus2()*, and *ITStatus3()* commands return various gauging related information, which is useful to verify learned cell data was updated correctly, as well as problem analyses.

## 4.4 Learning Cycles

As described in Section 4.3, IT is not enabled by default. When the BQ27Z561-R2 gauge receives an enable request , the gauge is then in its learning cycle mode. It is not necessary to use the gauge in this mode for all gauges during manufacture.

This mode helps to determine the golden image, which can be programmed into all gauges during the manufacture process. Learning cycles are generally run under lab conditions. When the golden image is learned by a gauge, the gauge is left in the IT learned mode of operation. Thus, when its data flash image is transferred to another gauge during manufacture, that gauge is then in learned mode; the gauge with this update will still need to receive an IT enable request. Requiring this request prevents the gauge from any unintended updates that could result from pack activity during manufacture. It would be one of the last commands to the pack before it is prepared for shipment. For more details, see the application note *Impedance Track™ Gas Gauge for Novices – bq20z70/z90* (SLUA397), which is can be downloaded from TI.com.

## 4.5 Impedance Track Configuration

**Load Mode —** During normal operation, the battery-impedance profile compensation of the Impedance Track algorithm can provide more accurate full-charge and remaining state-of-charge information if the typical load type is known. The two selectable options are constant current (**Load Mode** = 0) and constant power (**Load Mode** = 1).

**Load Select —** To compensate for the I × R drop near the end of discharge, the BQ27Z561-R2 must be configured for whatever current (or power) will flow in the future. While it cannot be exactly known, the BQ27Z561-R2 can use load history, such as the average current of the present discharge to make a sufficiently accurate prediction.

The BQ27Z561-R2 can be configured to use several methods of this prediction by setting the **Load Select** value. Because this estimate has only a second-order effect on remaining capacity accuracy, different measurement-based methods (methods 0–3, and method 7) result in only minor differences in accuracy. However, methods 4–6, where an estimate is arbitrarily assigned by the user, can result in a significant error if a fixed estimate is far from the actual load.

|  | **Constant Current** (**Load Mode** = 0) | **Constant Power** (**Load Mode** = 1) |
|---|---|---|
| 0 = | **Avg I Last Run** | **Avg P Last Run** |
| 1 = | Present average discharge current | Present average discharge power |
| 2 = | *Current()* | *Current()* × *Voltage()* |
| 3 = | *AverageCurrent()* | *AverageCurrent()* × *Voltage()* |
| 4 = | **Design Capacity mAh**/5 | **Design Capacity cWh**/5 |
| 5 = | *AtRate()* (mA) | N/A |
| 6 = | **User Rate-mA** | **User Rate-cW** |
| 7 = | **Max Avg I Last Run** | **Max Avg P Last Run** |

**Pulsed Load Compensation and Termination Voltage —** To take into account pulsed loads while calculating remaining capacity until **Term Voltage** threshold is reached, the BQ27Z561-R2 monitors not only average load, but also short load spikes. The maximum voltage deviation during a load spike is continuously updated during discharge and stored in **Delta Voltage**, with the minimum value allowed set in **Min Delta Voltage**. **Delta Voltage** is added to **Term Voltage** to determine end-of-discharge voltage (EDV).

**Reserve Battery Capacity —** The BQ27Z561-R2 allows an amount of capacity to be reserved in both mAh (***Reserve Cap-mAh***) and cWh (***Reserve Cap-cWh***) units between the point where the *RemainingCapacity()* function reports zero capacity and the absolute minimum pack voltage, ***Term Voltage***. This enables a system to report zero capacity/energy, but still have enough reserve capacity/energy to perform a controlled shutdown or provide an extended sleep period for the host system.

**Pack Based Termination Voltage —** The BQ27Z561-R2 forces *RemainingCapacity()* to 0 mAh when the battery voltage discharges to ***Term Voltage*** for a period of ***Term V Hold Time***.

A portion of the configuration options for Impedance Track is in the following data flash. Additional configuration options are located in the sections in which they are discussed.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Gas Gauging | IT Cfg | Pack Resistance | I2 | 0 | 32767 | 0 | mΩ | Value of the battery pack serial resistance that is NOT along the cell voltage measurement path of the gauge. This can include the resistance associated with FETs, traces, sense resistors, and any additional resistance in the path. |
| Gas Gauging | IT Cfg | System Resistance | I2 | 0 | 32767 | 0 | mΩ | Value of the system serial resistance that is along the path from the battery pack to the system power converter input. This can include FETs, traces, sense resistors and any additional resistance in the path. |
| Gas Gauging | IT Cfg | Cell 1 Interconnect Resistance | I2 | 0 | 32767 | 0 | $2^{-10}$ Ω | Value of the battery pack serial resistance that is along the cell voltage measurement path of the gauge; that is, the negative rail to the bottom of the cell and the top of the cell to the gauge. This can include the resistance associated with FETs, traces, sense resistors, and any additional resistance in the path. The gauge will offset the measured resistance with this value to improve Ra accuracy. |
| Gas Gauging | IT Cfg | Term Voltage | I2 | 0 | 32767 | 3000 | mV | Minimum pack voltage used in IT simulations. ***Term Voltage*** should be set to the minimum allowable voltage at the system power converter input. If it is desired to report 0% *RelativeStateOfCharge()* before the pack voltage reaches ***Term Voltage***, it is recommended to increase ***Reserve Cap-mAh*** and ***Reserve Cap-cWh***. |
| Gas Gauging | IT Cfg | Term V Hold Time | U1 | 0 | 255 | 2 | s | When ***Term Voltage*** is met for ***Term V Hold Time***, the *RemainingCapacity()* is set to 0 mAh. |
| Gas Gauging | IT Cfg | Max Simulation Iterations | U1 | 20 | 50 | 50 | — | Enables the user to set the max number of simulation iterations IT is allowed to do. If the user finds that the watchdog is tripping, this number can be lowered. |
| Gas Gauging | IT Cfg | Simulation Near Term Delta | I2 | 0 | 32767 | 250 | mV | Voltage delta from ***Term Voltage***, which defines "near EDV" for IT simulations. If ***Term Voltage*** is increased, ***Simulation Near Term Delta*** should be decreased to keep ***Term Voltage*** + ***Simulation Near Term Delta*** around 3.2 V–3.5 V, the knee of the discharge curve. |
| Gas Gauging | IT Cfg | Min Delta Voltage | I2 | –32768 | 32767 | 0 | mV | The minimum ***Delta Voltage*** that is saved during discharge cycles |
| Gas Gauging | IT Cfg | Load Select | U1 | 0 | 7 | 1 | — | Defines load compensation mode used by the gauging algorithm |
| Gas Gauging | IT Cfg | Load Mode | U1 | 0 | 1 | 1 | — | Defines unit used by the gauging algorithm: 0 = Constant Current 1 = Constant Power |
| Gas Gauging | IT Cfg | User Rate-mA | I2 | –9000 | 0 | 0 | mA | Discharge rate used for capacity calculation selected by ***Load Select*** = 6 |
| Gas Gauging | IT Cfg | User Rate-cW | I2 | –32768 | 0 | 0 | cW | Discharge rate used for capacity calculation selected by ***Load Select*** = 6 |
| Gas Gauging | IT Cfg | Reserve Cap-mAh | I2 | 0 | 9000 | 0 | mAh | Capacity reserved available when the gauging algorithm reports 0% *RelativeStateOfCharge()* |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | IT Cfg | Reserve Cap-cWh | I2 | 0 | 32000 | 0 | cWh | Capacity reserved available when the gauging algorithm reports 0% *RelativeStateOfCharge()* |
| Gas Gauging | IT Cfg | Predict Ambient Time | U2 | 0 | 65535 | 2000 | s | Wait time before the gauging algorithm starts to predict the ambient temperature during CHARGE and DISCHARGE modes (if *[AMB_PRED]* = 1). It is recommended to be set to 2× the value of *Temp a* to represent system thermal equilibrium. |

The parameters updated by AND used for Impedance Track are in the following data flash.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | State | Update Status | H1 | 0x00 | 0x0E | 0x00 | — | Bit 1:0: UPDATE1, UPDATE0 Update Status 0,0 = QMax NOT updated, Ra NOT updated 0,1 = QMax updated, Ra NOT updated 1,0 = QMax updated, Ra updated 1,1 = Unused Bit 2: Enables—Impedance Track gauging is enabled. This bit can only be changed via the *Gauging()* command. 0 = Disabled 1 = Enabled Bit 3: is_QMax_Field_Updated—QMax has updated in the field. 0 = Not updated 1 = Updated Bit 7:4: Reserved |
| Gas Gauging | State | Avg I Last Run | I2 | –32768 | 32767 | –500 | mA | Average current last discharge cycle |
| Gas Gauging | State | Avg P Last Run | I2 | –32768 | 32767 | –192 | cW | Average power last discharge cycle |
| Gas Gauging | State | Delta Voltage | I2 | –32768 | 32767 | 0 | mV | *Voltage()* delta between normal and short load spikes to optimize runtime calculation |
| Gas Gauging | State | Temp k | I2 | 0 | 32767 | 200 | 0.1°C/ 256 cW | Thermal model temperature factor |
| Gas Gauging | State | Temp a | I2 | 0 | 32767 | 1000 | s | Thermal model temperature time constant |
| Gas Gauging | State | Max I Last Run | I2 | –32768 | 32767 | –500 | mA | Max current last discharge cycle |
| Gas Gauging | State | Max P Last Run | I2 | –32768 | 32767 | –192 | cW | Max power last discharge cycle |

## 4.6 Gas Gauge Modes

Resistance updates take place only in DISCHARGE mode, while OCV and QMax updates only take place in RELAX mode. Fast QMax will update at the end of discharge (see Section 4.7.2 for details). Entry and exit of each mode is controlled by data flash parameters in the subclass **Gas Gauging: Current Thresholds** section. When the device is determined to be in RELAX mode and OCV is taken, the *GaugingStatus()[REST]* flag is set. In RELAX mode or DISCHARGE mode, *BatteryStatus()[DSG]* is set. Figure 4-1 shows the flow.
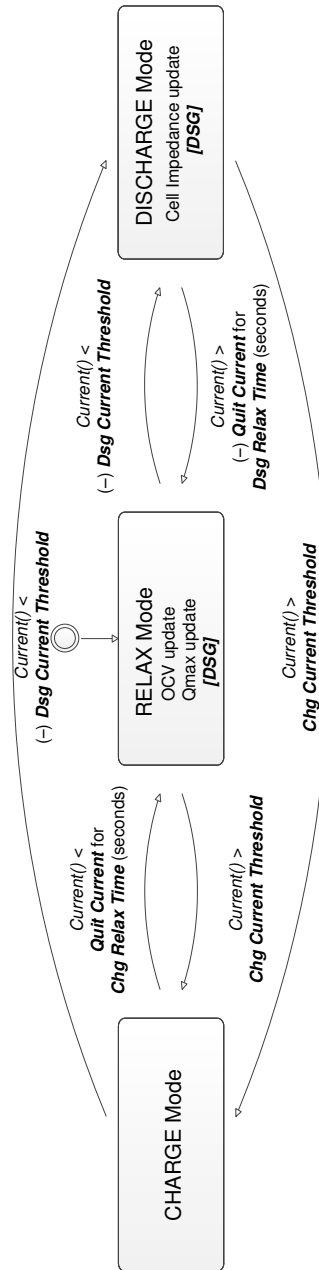
**Figure 4-1. Gas Gauge Operating Modes**

CHARGE mode is exited and RELAX mode is entered when *Current* goes below **Quit Current** for a period of **Chg Relax Time**. DISCHARGE mode is entered when *Current* goes below **(–)Dsg Current Threshold**. DISCHARGE mode is exited and RELAX mode is entered when *Current* goes above **(–)Quit Current** threshold for a period of **Dsg Relax Time**. CHARGE mode is entered when *Current* goes above **Chg Current Threshold**. Figure 4-2 shows an example of the operating modes.
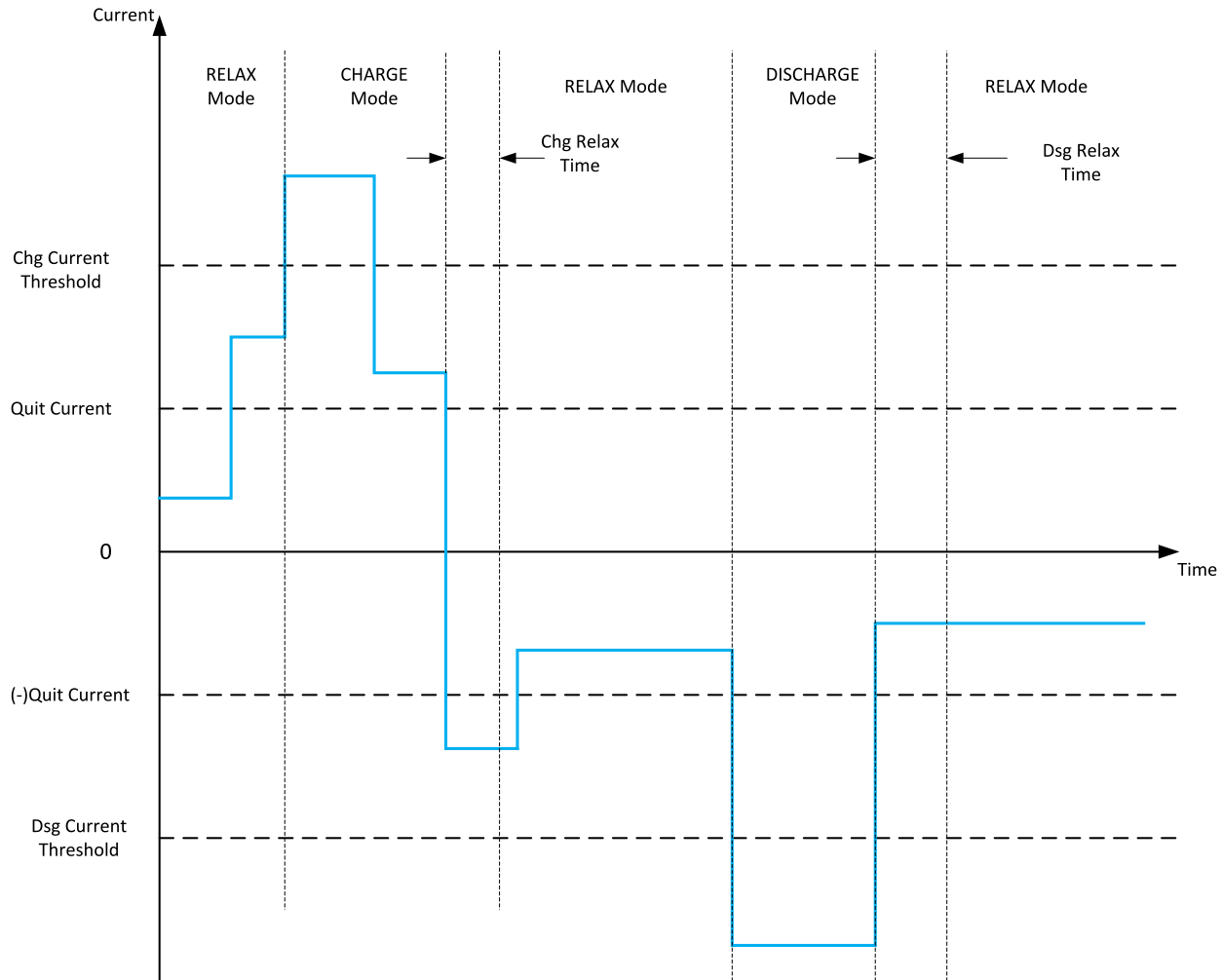


**Figure 4-2. Gas Gauge Operating Mode Example**

The configuration options for the gas gauge modes are in the following data flash.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Gas Gauging | Current Thresholds | Dsg Current Threshold | I2 | –32768 | 32767 | 60 | mA | DISCHARGE mode *Current()* threshold |
| Gas Gauging | Current Thresholds | Chg Current Threshold | I2 | –32768 | 32767 | 75 | mA | CHARGE mode *Current()* threshold |
| Gas Gauging | Current Thresholds | Quit Current | I2 | 0 | 32767 | 40 | mA | |*Current()*| threshold to enter RELAX mode |
| Gas Gauging | Current Thresholds | Dsg Relax Time | U1 | 0 | 255 | 60 | s | DISCHARGE to RELAX timeout |
| Gas Gauging | Current Thresholds | Chg Relax Time | U1 | 0 | 255 | 60 | s | CHARGE to RELAX timeout |

## 4.7 QMax and Ra

The total battery capacity is found by comparing states-of-charge before and after applying the load with the amount of charge passed. When an applications load is applied, the impedance of the cell is measured by comparing the open circuit voltage (OCV) obtained from a predefined function for the present state-of-charge with the measured voltage under load.

Measurements of OCV and charge integration determine the chemical state-of-charge (SOC) and chemical capacity (QMax). Ra is the impedance of the cell at 25°C at various states-of-charge.

The BQ27Z561-R2 acquires and updates the battery-impedance profile during normal battery usage. It uses this profile, along with the state-of-charge and QMax value, to determine *FullChargeCapacity()* and *RelativeStateOfCharge()* specifically for the configured load and present temperature. *FullChargeCapacity()* reports a capacity or energy available from a fully charged battery reduced by **Reserve Cap-mAh** or **Reserve Cap-cWh** under the configured load and present temperature until *Voltage()* reaches end-of-discharge voltage (EDV).

### 4.7.1 QMax Initial Values

The initial **QMax Cell 1** value should be taken from the cell manufacturers' data sheet multiplied by the number of parallel cells, and are also used for the **Design Capacity mAh** data flash value.

See the "Theory and Implementation of Impedance Track Battery Fuel-Gauging Algorithm" section in the *BQ20zxx Product Family Application Report* (SLUA364) for further details.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | State | QMax Cell 1 | I2 | 0 | 32767 | 5359 | mAh | QMax Cell 1 |
| Gas Gauging | Design | Design Capacity mAh | I2 | 0 | 32767 | 5300 | mAh | Design Capacity in mAh |
| Gas Gauging | Design | Design Capacity cWh | I2 | 0 | 32767 | 2040 | cWh | Design Capacity in cWh |
| Gas Gauging | Design | Design Voltage | I2 | 0 | 32767 | 3850 | mV | Design Voltage |

### 4.7.2 QMax Update Conditions

#### 4.7.2.1 Base Required Conditions

QMax learning/updates do not occur until the *GaugingStatus()[QEN]* flag is set by sending the *Gauging()* (0x0021) command to *AltManufacturerAccess()*.

The following conditions will prevent a QMax update when an OCV is taken.

**Temperature —** If *Temperature()* is outside of the range 10°C to 40°C

**Delta Capacity —** If the capacity change between suitable battery rest periods is less than 37%

**Voltage —** If *Voltage()* is inside a flat voltage region. (See the *Support of Multiple Li-Ion Chemistries with Impedance Track Gas Gauges Application Report* (SLUA372) for the voltage ranges of other chemistries.) This flat region varies with different chemistries. The *GaugingStatus()[OCVFR]* flag indicates if the cell voltage is inside this flat region when *GaugingStatus()[REST]* is set. Prior to an exit from CHARGE or DISCHARGE mode, the *GaugingStatus()[QMAXDODOK]* flag can be checked to determine if the cell is in the flat zone.

**Offset Error —** If the offset error accumulated during the time passed from the previous OCV reading exceeds 1.5% of **Design Capacity mAh**, the update is disqualified. Offset error current is calculated as a function of **Coulomb Counter Deadband** and the sense resistor value. For applications with small battery capacities, an offset error of 1.5% of **Design Capacity mAh** can be reached in a short period of time, especially with a small sense resistor. Therefore, a minimum time of 11 hours from the last qualified OCV is enforced before disqualification, which facilitates QMax updates in such applications. Enforcing this minimum time can increase an offset error to as high as 5.5% of **Design Capacity mAh**.

### 4.7.2.2 Tracking QMax Updates

Several flags in *GaugingStatus()* are helpful to track QMax update conditions. The *[REST]* flag indicates an OCV is taken in RELAX mode. *[REST]* sets when the OCV is taken and clears when exiting RELAX mode. If the firmware has successfully made an OCV prediction, the *GaugingStatus[OCVPRED]* bit is set. The bit will clear if *GaugingStatus[REST]* gets set or when RELAX mode is exited. The *[VOK]* flag indicates the last OCV reading is qualified for the QMax update. *[VOK]* sets when exiting RELAX mode after an OCV that qualifies for QMax updates (including those that update QMax). *[VOK]* clears when a QMax update occurs, when a QMax update is disqualified based on an offset error (as described above), or upon a full reset. The *[QMAX]* flag toggles when a QMax update occurs. *ITStatus2()* and *ITStatus3()* return the QMax and DOD (the depth of discharge that corresponds to the OCV reading) data.

### 4.7.2.3 Relaxed Cell OCV

The BQ27Z561-R2 updates the no-load full capacity (QMax) when two open circuit voltage (OCV) readings are taken. These OCV readings are taken when the battery is in RELAX mode before and after a minimum charge or discharge activity. A relaxed state is achieved if the battery voltage has a dV/dt of < 1 µV/s. Typically, it can take up to two hours after charging and five hours after discharging to ensure that the dV/dt condition is satisfied. If five hours are exceeded, a reading is taken even if the dV/dt condition was not satisfied. This relaxed cell state is indicated when the *[REST]* flag is set.

### 4.7.2.4 OCV Prediction

Another method available in the BQ27Z561-R2 gauge is to estimate an accurate OCV reading. After a set wait time (**OCV Pred Transient T**) in RELAX mode, the gauge begins to accumulate voltage readings. Once **OCV Pred Measure Time** has passed, the gauge uses a fast OCV algorithm to predict the final OCV value. This fast OCV method is enabled by setting **IT Gauging Ext[FOCV_EN]** = 1. This method provides the benefit of reduced relaxation requirements for QMax updates. If at any time the requirements for the conventional OCV method are achieved (dV/dt of < 1 µV/s requirement) after a fast OCV estimation, the device updates the OCV measurement accordingly. For a fast OCV estimate, entry into RELAX mode must be preceded by at least **OCV Pred Active T Limit** of a charge or discharge current large enough for the BQ27Z561-R2 to exit RELAX mode.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | IT Cfg | OCV Pred Active T Limit | U2 | 100 | 65535 | 200 | s | This is the minimum time the gauge must be in CHARGE or DISCHARGE mode before entering into RELAX mode for a fast OCV estimate. |
| Gas Gauging | IT Cfg | OCV Pred Transient T | U2 | 100 | 65535 | 300 | s | This is the minimum time the gauge must be in RELAX mode before fast OCV voltage readings start to accumulate. |
| Gas Gauging | IT Cfg | OCV Pred Measure Time | U2 | 0 | 65535 | 200 | s | This is the time in RELAX mode when fast OCV voltage readings are accumulated and fast OCV is predicted. |

### 4.7.2.5 Cycle Count Based QMax Degradation

For a gauging situation that never or rarely meets the condition for any of the previously described QMax updates, QMax will update and diminish based on a cycle count. The gauge tracks full discharge cycles since the last QMax update was determined via one of the above methods. When this cycle count reaches **Cycle Adjust Threshold**, QMax updates as QMax = QMax × (1 − (**Cycle Adj Incr** × **Cycle Adjust Threshold** /10000)).

> **NOTE:** When there is a QMax update, the discharge cycle count since the last QMax update is set to 0. The discharge cycle count since the last QMax update is incremented each time *CycleCount()* is incremented.

### 4.7.2.6 Fast QMax Update Conditions

The Fast QMax update conditions are very similar to the QMax update conditions with the following differences:

- Instead of taking two OCV readings for a QMax update, a Fast QMax update requires only one OCV reading AND
- The battery pack should discharge below 10% RSOC.

The differences in requirements enable the Fast QMax feature to have a QMax update at the end of discharge (given one OCV reading is already available and discharge < 10% RSOC) without a longer relax time after a discharge event. The temperature, delta capacity, voltage, and offset error requirements for a QMax update are still required for the Fast QMax update.

This feature is particularly useful for reducing production QMax learning cycle time or for an application that is mostly in CHARGE or DISCHARGE state with infrequent relaxation. Setting *IT Gauging Configuration[FAST_QMAX_LRN]* = 1 enables Fast QMax during production learning only (that is, *Update Status* = 06). When setting *IT Gauging Configuration[FAST_QMAX_FLD]* = 1, Fast QMax is enabled when Impedance Track is enabled and *Update Status* = 06 or 0E.

The DOD is taken for QMax at a high state-of-charge, then during discharge when at the steep portion of the voltage curve (92–96% DOD). This enables a good estimation of DOD. This DOD estimation during discharge is used to update QMax immediately instead of first needing a rest period. The actual update to QMax occurs when the discharge stops after verifying that no conditions are present to cause the QMax update to fail. The BQ27Z561-R2 devoce has an option to qualify DODEOC for QMax updates, which means it is possible to update QMax with no rest periods (by having a charge termination followed by a full discharge). Fast QMax can be enabled in either LEARN mode (*[FAST_QMAX_LRN]*) or FIELD mode (*[FAST_QMAX_FLD]*). The LEARN mode flag means Fast QMax is only enabled with *Update Status* = 06, and will be disabled when the status = 0x0E. FIELD mode means Fast QMax is enabled with *Update Status* = 06 and 0E.

### 4.7.2.7 QMax and Fast QMax Update Boundary Check

The BQ27Z561-R2 implements a QMax and Fast QMax boundary check prior to saving the value to data flash. This improves the robustness of the QMax update in case of potential QMax corruption during the update process.

The verifications are as follows:

1. Verify that the updating QMax or Fast QMax value is within *Qmax Delta* of the present QMax (above or below). *QMax Delta* is the maximum allowed QMax change for each update, expressed as a percentage of *Design Capacity mAh*. If the updating value is outside of this range, the QMax change is capped to (*QMax Delta* × *Design Capacity mAh*) / 100.
2. Bound the QMax value to a maximum of *Qmax Upper Bound*. *Qmax Upper Bound* is expressed as a percentage of *Design Capacity mAh*. If the updating value is above this maximum, the QMax update is capped to (*Qmax Upper Bound* × *Design Capacity mAh*) / 100.
3. Ensure that QMax is greater than 0 mAh.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Gas Gauging | IT Cfg | QMax Delta | U1 | 3 | 100 | 5 | % | Maximum allowed QMax change from its previous value. The QMax change will be capped by this setting if the delta from the previous QMax is larger than *QMax Delta*. *QMax Delta* is a percentage of *Design Capacity mAh*. |
| Gas Gauging | IT Cfg | *Qmax Upper Bound* | U1 | 100 | 255 | 130 | % | Maximum QMax value over the lifetime of the pack. If the updated QMax value is larger than this setting, the updated QMax will be capped to *Qmax Upper Bound*. *Qmax Upper Bound* is a percentage of *Design Capacity mAh*. |

### 4.7.3 Ra Table Initial Values

The Ra table is part of the impedance profile that updates during discharge when gauging is enabled. The initial **Cell0 R_a0...14** values should be programmed by selecting the correct chemistry data during data flash configuration. A chemistry database is constantly updating, and can be downloaded from the Gas Gauge Chemistry Updater product web page (http://www.ti.com/tool/gasgaugechem-sw). The initial **xCell0 R_a0...14** values are a copy of the non-x data set. Two sets of Ra tables are used alternatively when gauging is enabled to prevent wearing out the data flash.

The **Cell0 R_a flag** and the **xCell0 R_a flag** indicate the validity of the cell impedance table for the cell.

---

**NOTE:** Firmware updates these values: It is not recommended to change them manually.

---

#### 4.7.3.1 R_a0 Table

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Ra Table | R_a0 | Cell0 R_a flag | H2 | 0x0000 | 0xFFFF | 0xFF55 | — | High Byte:<br>0x00: Cell Impedance and QMax updated<br>0x05: RELAX mode and QMax update in progress<br>0x55: DISCHARGE mode and cell updated<br>0xFF: Cell impedance never updated<br>Low-Byte:<br>0x00: Table not used and QMax updated<br>0x55: Table being used<br>0xFF: Table never used, no QMax or cell impedance update |
| Ra Table | R_a0 | Cell0 R_a 0 | I2 | 0 | 32767 | 159 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 0 |
| Ra Table | R_a0 | Cell0 R_a 1 | I2 | 0 | 32767 | 58 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 1 |
| Ra Table | R_a0 | Cell0 R_a 2 | I2 | 0 | 32767 | 65 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 2 |
| Ra Table | R_a0 | Cell0 R_a 3 | I2 | 0 | 32767 | 79 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 3 |
| Ra Table | R_a0 | Cell0 R_a 4 | I2 | 0 | 32767 | 90 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 4 |
| Ra Table | R_a0 | Cell0 R_a 5 | I2 | 0 | 32767 | 63 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 5 |
| Ra Table | R_a0 | Cell0 R_a 6 | I2 | 0 | 32767 | 76 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 6 |
| Ra Table | R_a0 | Cell0 R_a 7 | I2 | 0 | 32767 | 82 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 7 |
| Ra Table | R_a0 | Cell0 R_a 8 | I2 | 0 | 32767 | 82 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 8 |
| Ra Table | R_a0 | Cell0 R_a 9 | I2 | 0 | 32767 | 87 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 9 |
| Ra Table | R_a0 | Cell0 R_a 10 | I2 | 0 | 32767 | 87 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 10 |
| Ra Table | R_a0 | Cell0 R_a 11 | I2 | 0 | 32767 | 101 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 11 |
| Ra Table | R_a0 | Cell0 R_a 12 | I2 | 0 | 32767 | 118 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 12 |
| Ra Table | R_a0 | Cell0 R_a 13 | I2 | 0 | 32767 | 191 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 13 |
| Ra Table | R_a0 | Cell0 R_a 14 | I2 | 0 | 32767 | 282 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 14 |

#### 4.7.3.2 R_a0x Table

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Ra Table | R_a0x | xCell0 R_a flag | H2 | 0x0000 | 0xFFFF | 0xFFFF | — | High-Byte:<br>0x00: Cell Impedance and QMax updated<br>0x05: RELAX mode and QMax update in progress<br>0x55: DISCHARGE mode and cell updated 0xFF: Cell impedance never updated.<br>Low-Byte:<br>0x00: Table is not used and QMax updated.<br>0x55: Table is being used.<br>0xFF: Table was never used and no QMax or cell impedance was updated. |
| Ra Table | R_a0x | xCell0 R_a 0 | I2 | 0 | 32767 | 159 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 0 |
| Ra Table | R_a0x | xCell0 R_a 1 | I2 | 0 | 32767 | 58 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 1 |
| Ra Table | R_a0x | xCell0 R_a 2 | I2 | 0 | 32767 | 65 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 2 |
| Ra Table | R_a0x | xCell0 R_a 3 | I2 | 0 | 32767 | 79 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 3 |
| Ra Table | R_a0x | xCell0 R_a 4 | I2 | 0 | 32767 | 90 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 4 |
| Ra Table | R_a0x | xCell0 R_a 5 | I2 | 0 | 32767 | 63 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 5 |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Ra Table | R_a0x | xCell0 R_a 6 | I2 | 0 | 32767 | 76 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 6 |
| Ra Table | R_a0x | xCell0 R_a 7 | I2 | 0 | 32767 | 82 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 7 |
| Ra Table | R_a0x | xCell0 R_a 8 | I2 | 0 | 32767 | 82 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 8 |
| Ra Table | R_a0x | xCell0 R_a 9 | I2 | 0 | 32767 | 87 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 9 |
| Ra Table | R_a0x | xCell0 R_a 10 | I2 | 0 | 32767 | 87 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 10 |
| Ra Table | R_a0x | xCell0 R_a 11 | I2 | 0 | 32767 | 101 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 11 |
| Ra Table | R_a0x | xCell0 R_a 12 | I2 | 0 | 32767 | 118 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 12 |
| Ra Table | R_a0x | xCell0 R_a 13 | I2 | 0 | 32767 | 191 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 13 |
| Ra Table | R_a0x | xCell0 R_a 14 | I2 | 0 | 32767 | 282 | $2^{-10}\ \Omega$ | Cell 0 resistance at grid point 14 |

### 4.7.4 Ra Table Update Conditions

The impedance is different across different DOD states. The cell has 15 Ra grid points representing the impedance from 0%–100% DOD. In general, the Ra table is updated during discharge. The *GaugingStatus()[RX]* flag will toggle when the Ra grid point is updated. The Ra update is disabled if any of the following conditions are met. The *GaugingStatus()[RDIS]* is set to indicate the Ra update is disabled.

- During the optimization cycle, the Ra update is disabled until QMax is updated (that is, Ra will not be updated if **Update Status** = 4).
- The Ra update is disabled if the charge accumulation error > 5% of **Design Capacity mAh**.
- During a discharge, a negative resistance is measured.

A valid OCV reading during RELAX mode or a Fast QMax update without an OCV read will clear the *[RDIS]* flag.

The configuration options related to Ra table updates are in the following data flash.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | IT Cfg | Resistance Parameter Filter | U2 | 1 | 65535 | 65142 | — | This is the filter coefficient on voltage, current, and temperature for resistance updates. Reducing this filter setting can improve low temperature performance at high rates. Examining the **Term Voltage Delta** setting and **Fast Scale Start SOC** should be done prior to adjusting this parameter when trying to improve the RSOC performance. The following is the formula to convert the DF setting into actual filter time constant: Filter time constant = [1/(1 – (DF_Value /65536))] – 1. |
| Gas Gauging | IT Cfg | Max Charge Current % | U1 | 0 | 100 | 10 | % | If *AverageCurrent()* changes by more than this threshold from the previous resistance measurement to the present, the present resistance measurement and Ra scale update are not used. This protects resistance measurements from voltage transients after step changes in current, especially transitions to relaxation. |
| Gas Gauging | IT Cfg | Resistance Update Voltage | I2 | 0 | 32767 | 50 | mV | The difference between the open circuit voltage based on DOD and the measured voltage is estimated as the IR drop. If this IR drop is greater than **Resistance Update Voltage**, then the resistance calculation is allowed. This can be helpful in applications with small battery packs and large resistances that do not meet the C/10 minimum current requirement for resistance updates. |
| Gas Gauging | IT Cfg | Ra Filter | U2 | 0 | 999 | 800 | 0.1% | Filter value used in Ra updates that specifies what percentage of the Ra update is from the new value (100%—setting) versus the old value (setting). The recommended setting is 80% if fast resistance scaling is enabled. Otherwise, the setting should be 50% as the default. |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | IT Cfg | Ra Max Delta | U1 | 0 | 255 | 15 | % of Design Resistance | Maximum value of allowed Ra change |
| Gas Gauging | IT Cfg | Design Resistance | I2 | 1 | 32767 | 90 | mΩ | Cell resistance at grid point 4. Automatically updated during learning when **Update Status** is set by the gauge to 06 and 0E. To automatically update again, set **Update Status** to 04. **Design Resistance** could be set manually when **Update Status** = 0E. |

### 4.7.5 Fast Resistance Scaling

At low temperatures and very high discharge rates, using cell resistance directly from interpolating Ra table grids can lead to higher errors in RSOC. This is especially true near the end of discharge when cell resistance is changing quickly, which could cause undesirable drops to 0% RSOC. Fast resistance scaling is an algorithm that improves RSOC convergence to 0% by scaling cell resistance values from Ra table interpolation to reflect the most recent measured resistance.

The following terms are defined to understand the operation of fast resistance scaling:

- True RSOC: The true relative state-of-charge of the battery pack, as determined by the gauge. It is computed as True Rem Q / True Full Chg Q.
- Resistance voltage: A filtered cell voltage used for resistance measurements with a time constant defined by **Near EDV Ra Param Filter** (or **Resistance parameter filter** if **IT Gauging Configuration[FF_NEAR_EDV]** = 0).
- Average voltage: A filtered pack voltage with the same time constant as *AverageCurrent()*.

#### 4.7.5.1 Calculation of Resistance Scale

When enabled with **IT Gauging Configuration[RSOC_CONV]** = 1, fast resistance scaling becomes active once one of the following is true:

- True RSOC ≤ **Fast Scale Start SOC** OR
- Resistance voltage < **Term Voltage** + **Term Voltage Delta**.

For most applications, the default values of **Term Voltage Delta** and **Fast Scale Start SOC** are recommended. It is typically best to keep **Term Voltage** + **Term Voltage Delta** around 3.2 V–3.5 V, the knee of the discharge curve, for most battery applications.

Once active, every 30 seconds the algorithm calculates a new cell resistance scale as ($R_{new}$ / $R_{old}$) × 1000, where $R_{new}$ is the most recent measured cell resistance and $R_{old}$ is the interpolated cell resistance at the present DOD. The raw scale is reported on Cell 1 RaScale. After a scale is calculated, an IT simulation is triggered to utilize the new resistance information. In addition, the scale is used for ambient temperature prediction (if enabled).

As a safeguard, the scale is limited to a minimum of 200 and a maximum of 5000 when applied, but Cell 1 RaScale maintains the raw scale. For clarity, a scale of 200 corresponds to 0.2 (reducing cell resistance) and 5000 corresponds to 5.0 (increasing cell resistance).

#### 4.7.5.2 Negative Resistance Scale

If a negative scale is calculated, it usually indicates significant error in QMax or an incorrect DOD0, which could be from system loading during initialization. When this occurs, Cell 1 RaScale resets to 1000 and *GaugingStatus()[NSFM]* sets to 1. The gauge attempts to correct the error in DOD by adjusting DOD0 such that DOD0 + (DOD0 Passed Q / QMax) equals the DOD computed from voltage, current, and temperature used for resistance.

#### 4.7.5.3 Reset of Resistance Scale

Cell 1 RaScale resets to 1000 under the following conditions:

- Initialization of the device
- IT is enabled (*GaugingStatus()[QEN]* changes to 1).

- In CHARGE mode when average voltage > **Term Voltage** + **Term V Delta** AND True RSOC > **Fast Scale Start SOC**.
- In RELAX mode when an OCV reading occurs with *Temperature()* > 15.0°C.
- In DISCHARGE mode when a negative resistance scale is calculated.

#### 4.7.5.4 Application of Resistance Scale

The scale can be applied in two ways:

- If **IT Gauging Configuration[DOD_RSCALE_EN]** = 0, the resistance scale is applied across all DODs.
- If **IT Gauging Configuration[DOD_RSCALE_EN]** = 1, the resistance scale is only applied to DODs greater than or equal to the DOD at which it was calculated.

Setting **IT Gauging Configuration[DOD_RSCALE_EN]** = 1 is recommended because it reduces the sensitivity of FCC to large changes in the resistance scale.

#### 4.7.5.5 Fast Resistance Scaling Configuration

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | IT Cfg | Term Voltage Delta | I2 | 0 | 32767 | 200 | mV | Controls when fast resistance scaling becomes active when *[RSOC_CONV]* = 1. If **Term Voltage** is increased, **Term Voltage Delta** should be decreased to keep **Term Voltage** + **Term Voltage Delta** around 3.2 V–3.5 V, the knee of the discharge curve. |
| Gas Gauging | IT Cfg | Fast Scale Start SOC | U1 | 0 | 100 | 10 | % | Controls the RSOC % that fast resistance scaling becomes active when *[RSOC_CONV]* = 1. The default value is recommended, however, it may be appropriate to adjust this value in certain applications to be around the knee of the discharge curve. |
| Gas Gauging | IT Cfg | Near EDV Ra Param Filter | U2 | 1 | 65535 | 62184 | — | Ra filter used in the fast resistance scaling region if *[FF_NEAR_EDV]* = 1. The default value should be used. |

### 4.8 FullChargeCapacity (FCC), RemainingCapacity (RemCap), and RelativeStateOfCharge (RSOC)

The Impedance Track algorithm applies QMax, impedance, temperature, voltage, and current data to predict the runtime *FullChargeCapacity()*, *RemainingCapacity()*, and *RelativeStateOfCharge()*. These values are updated if any of the following conditions are met, reflecting the battery capacity in real time:

- QMax update occurs
- Ra update occurs
- At onset of charge and discharge
- Valid charge termination
- Every 1 hour in RELAX mode after an OCV has been taken
- If temperature changes more than 5°C
- Every second when current is measured and passed charge is accumulated

#### 4.8.1 Smoothing Engine

The gauge provides an engine that ensures *FullChargeCapacity()*, *RemainingCapacity()*, and *RelativeStateOfCharge()* change smoothly, which can be desirable in certain system applications. This smoothing engine is enabled when **IT Gauging Configuration[SMOOTH]** = 1. The true and smooth versions of FCC, RemCap, and RSOC are always available, but **IT Gauging Configuration[SMOOTH]** selects which version is reported on the standard commands.

When the smoothing engine is enabled, there are specific cases that can be handled differently based on user preference.

### 4.8.1.1 Smoothing to 0% in DISCHARGE Mode

When **IT Gauging Configuration[DSG_0_SMOOTH_OK]** = 1, smoothing to 0% in DISCHARGE mode is enabled. For preventing jumps to 0% in DISCHARGE mode, two data flash parameters are used: **Term Smooth Start Cell V Delta** and **Term Smooth Time**. If *Voltage()* is less than or equal to **Term Voltage** + **Term Smooth Start Cell V Delta** while in DISCHARGE mode, time-based smoothing to 0% is initiated. *RemainingCapacity()* smoothly converges to 0 mAh over the next **Term Smooth Time** seconds. Time-based smoothing to 0% will continue unless CHARGE mode is entered.

**Term Smooth Final Cell V Delta** ensures that the gauge reports 0% before voltage drops too low. If *Voltage()* is less than or equal to **Term Voltage** – **Term Smooth Final Cell V Delta** while in DISCHARGE mode, *RemainingCapacity()* will be forced to 0 mAh, even if time-based smoothing to 0% was active.

---

> **NOTE:** **Term Smooth Final Cell V Delta** can be disabled by setting equal to **Term Voltage**, but is typically expected to be low enough to enable the system to shut down properly (without brownout).

---

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | Smoothing | Term Smooth Start Cell V Delta | I2 | 0 | 32767 | 150 | mV | If **[DSG_0_SMOOTH_OK]** = 1, then during DISCHARGE mode and once *Voltage()* is less than or equal to **Term Voltage** + **Term Smooth Start Cell V Delta**, time-based smoothing to 0% is initiated. This will smooth *RemainingCapacity()* to 0 mAh over the next **Term Smooth Time** seconds. |
| Gas Gauging | Smoothing | Term Smooth Final Cell V Delta | I2 | 0 | 32767 | 100 | mV | If **[DSG_0_SMOOTH_OK]** = 1, then during DISCHARGE mode and once *Voltage()* is less than or equal to **Term Voltage** + **Term Smooth Final Cell V Delta**, *RemainingCapacity()* is forced to 0 mAh. |
| Gas Gauging | Smoothing | Term Smooth Time | U1 | 1 | 32767 | 20 | s | If **[DSG_0_SMOOTH_OK]** = 1, then during DISCHARGE mode and once *Voltage()* is less than or equal to **Term Voltage** + **Term Smooth Start Cell V Delta**, time-based smoothing to 0% is initiated. This will smooth *RemainingCapacity()* to 0 mAh over the next **Term Smooth Time** seconds. |

### 4.8.1.2 Smoothing to 100% in CHARGE Mode

When **IT Gauging Configuration[CHG_100_SMOOTH_OK]** = 1, smoothing to 100% in CHARGE mode is enabled. For preventing jumps to 100% in CHARGE mode, this feature uses the valid charge termination logic to predict when charging will complete. Refer to *Valid Charge Termination* for the full set of details. The valid charge termination logic requires two consecutive 40-s windows that meet all taper conditions. After the first 40-s window is satisfied, time-based smoothing to 100% will be initiated. *RemainingCapacity()* smoothly converges to *FullChargeCapacity()* over the next 40-s window.

### 4.8.1.3 Smoothing in RELAX Mode

**IT Gauging Configuration[RELAX_JUMP_OK]** and **IT Gauging Configuration[RELAX_SMOOTH_OK]** determine the behavior of *RelativeStateOfCharge()* during RELAX mode.

When **IT Gauging Configuration[RELAX_JUMP_OK]** = 1, *RelativeStateOfCharge()* is allowed to jump during RELAX mode. When **IT Gauging Configuration[RELAX_JUMP_OK]** = 0, *RelativeStateOfCharge()* holds constant during RELAX mode and any RSOC jump will be passed onto CHARGE and DISCHARGE mode.

When **IT Gauging Configuration[RELAX_SMOOTH_OK]** = 1, *RelativeStateOfCharge()* is NOT allowed to jump during RELAX mode. Any RSOC jumps during RELAX mode are smoothed out over a period of **Smooth Relax Time**. If **Smooth Relax Time** is not reached, the remaining portion of the RSOC jump will be passed onto CHARGE and DISCHARGE mode.

If both bits are set to 1, **IT Gauging Configuration[RELAX_JUMP_OK]** takes higher priority and *RelativeStateOfCharge()* is allowed to jump during RELAX mode.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Gas Gauging | Smoothing | Smooth Relax Time | I2 | 1 | 32767 | 1000 | s | If **[RELAX_SMOOTH_OK]** = 1, the delta remaining capacity and full charge capacity are smoothed over this set period of time. It is recommended to use the default setting. |

## 4.9 Impedance Track (IT) Configuration Options

The BQ27Z561-R2 provides several Impedance Track configuration options to fine-tune the gauging performance. These configurations can be turned on or off through the corresponding flags in **Settings**: **IT Gauging Configuration** and **I2C Gauging Configuration**.

**[LOCK0]**: After a discharge event, cell voltage usually recovers to a slightly higher voltage during RELAX mode. A new OCV reading during this time can result in a slightly higher state-of-charge. This flag provides an option to keep *RemainingCapacity()* and *RelativeStateOfCharge()* locked during relaxation after 0% and *GaugingStatus()[FD]* are reached during discharge. The lock is removed once CHARGE mode is entered.

**[1PERCENT_HOLD]**: When enabled, holds the RSOC value at 1% until **Term Voltage** has been reached during discharge.

**[RSOC_HOLD]**: An IT simulation will run at the onset of discharge. If charge terminates at a low temperature and discharge occurs at a higher temperature, the difference in temperature could cause a small rise of RSOC for a short period of time at the beginning of discharge. This flag option prevents RSOC rises during discharge. RSOC will be held until the calculated value falls below the actual state.

**[RSOCL]**: When set, RSOC will be held to 99% until charge termination is detected.

**[RFACTSTEP]**: The gauge keeps track of the change in Ra over 15 updates. It is limited to 1.5 max. During an Ra update, if (new Ra)/(old Ra) > 1.5 or < 0.5, the gauge will take different actions based on the setting of this flag.

If the flag is set to 1 (default), the gauge allows Ra to update once using the max factor of 1.5 or min factor of 0.5, then disables the Ra update. If this flag is set to 0, the gauge will not update Ra, but will disable it. It is recommended to keep the default setting. In both cases, *GaugingStatus()[RDIS]* is set.

**[OCVFR]**: An OCV reading is taken when a dV/dt condition is met. This is not the case if charging stops within or below the flat voltage region, which varies with ChemID. The change of cell voltage in this region is very small; therefore, the same voltage error can correspond to a larger DOD error. This flag is set by default. The device waits 48 hours before taking an OCV reading if charging stops prematurely. A short discharge will not cancel this 48-hour wait. An OCV reading is taken when the dV/dt condition is met. To reduce test time during evaluation, it is helpful to remove the 48-hour wait-time requirement.

**[DOD0EW]**: DOD0 readings have an associated error based on the elapsed time since the reading, the conditions at the time of the reading (reset, charge termination, and so on), the temperature, and the amount of relax time at the time of the reading, among others. This flag provides an option to take into account both the previous and new calculated DOD0, which are weighted according to their respective accuracies. This can result in improved accuracy and in reduction of RSOC jumps after relaxation.

**[LFP_RELAX]**: When enabled, this bit activates an additional functionality only if the chemistry ID programmed indicates a LiFePO$_4$ chemistry (ID 0x4xxx or 0x04xx). Having this bit enabled with a non-LiFePO$_4$ cell type does not introduce any behavioral modifications.

LiFePO$_4$ has a unique, slow configuration relaxation near full charge. The slow decaying voltage causes RSOC to continue to drop every 5 hours. Depending on the full charge taper current, the fully settled voltage could be close to or even below **FlatVoltMax** in some cases. If **[LFP_RELAX]** is enabled and the LiFePO$_4$ chemID is selected, **[OCVFR]** will be set on exit from CHARGE mode regardless of voltage or state of charge. For **[OCVFR]** to clear, a relaxation of 48 hours or non-trivial discharge must occur. A non-trivial discharge is indicated when the cell voltage in relaxation falls below **FlatVoltMin**. The QMax update is unlikely disabled because DOD will not update due to **[OCVFR]** forcing the 48-hour timeout, and voltage likely relaxing into the flat zone. Therefore, the QMax update takes an alternative approach: Once full charge occurs (**[FC]** bit set), DOD0 = Dod_at_EOC is automatically assigned and valid for a QMax update. **[VOK]** is set if there is no QMax update. If QMax is updated, **[VOK]** is cleared. The DOD error, as a result of this action, is zero or negligible because in the LiFePO$_4$ table, the OCV voltage corresponding to DOD = 0 is much lower. **[RSOC_CONV]**: This function is also called fast resistance scaling. It is an option to address the convergence of RSOC to 0% at a low temperature and a very high rate of discharge. See *Fast Resistance Scaling* for more details.

**[FAST_QMAX_LRN]** and **[FAST_QMAX_FLD]**: The first flag enables Fast QMax during the learning cycle when **Update Status** = 06. The second bit enables Fast QMax in the field when **Update Status** ≥ 06. See *Fast QMax Update Conditions* for more details on Fast QMax.

**[FF_NEAR_EDV]**: Fast filter near EDV. If this flag is set to 1, the gauge applies an alternative filter, **Near EDV Ra Param Filter**, for an Ra update in the fast scaling region (starting around 10% RSOC). This flag should be kept to 1 as a default. If this flag is set to 0, the gauge uses the regular Ra filter, **Resistance Parameter Filter**. Both DF filters should not be changed from the default value.

**[SMOOTH]**: If this bit is set to 1, the smoothing engine is enabled. For more details, see *Smoothing Engine*, which covers **[RELAX_JUMP_OK]**, **[RELAX_SMOOTH_OK]**, **[CHG_100_SMOOTH_OK]**, and **[DSG_0_SMOOTH_OK]** as further configuration to the smoothing engine.

**[TDELTAV]**: This bit determines how the **Delta Voltage** is calculated. By setting this bit, the gauge calculates **Delta Voltage** that corresponds to the power spike defined in **Min Turbo Power**. This bit must be set to 1 if TURBO BOOST mode is used. Otherwise, leaving this bit cleared as a default enables the gauge to calculate **Delta Voltage** by using the maximum difference between instantaneous and average voltages.

**[CSYNC]**: If this bit is set to 1, the gauge synchronizes *RemainingCapacity()* to *FullChargeCapacity()* at valid charge termination.

**[CCT]**: This bit provides an option to use *FullChargeCapacity()* (when **[CCT]** = 1) or *DesignCapacity()* (when **[CCT]** = 0) for cycle count threshold calculation. If *FullChargeCapacity()* is selected for cycle count threshold calculation, the minimum cycle count threshold is always 10% of *DesignCapacity()*. This helps to avoid any erroneous cycle count increment caused by an extremely low *FullChargeCapacity()*.

**[TAMB_SYNC_SIM]**: If this bit is set to 1, the gauge performs an IT simulation after the *TambientSync()* command is received to use the newly recorded ambient temperature. This IT simulation could produce a change in *RelativeStateOfCharge()* if **[SMOOTH]** = 0. If **[SMOOTH]** = 1, *RelativeStateOfCharge()* behaves according to the configuration of the smoothing engine.

**[AMB_PRED]**: If this bit is set to 1, ambient temperature can be predicted during DISCHARGE and CHARGE modes. This can be useful in system applications that experience large increases in ambient temperature during discharges and/or charges without entry to RELAX mode. Ambient temperature prediction is triggered after being in DISCHARGE/CHARGE for **Predict Ambient Time**, when the system is assumed to be at thermal equilibrium.

**[THERM_IV]**: If this bit is set to 1, battery heat is held constant near the end of an IT simulation. This helps prevent overestimation of temperature towards the end of discharge, and applies to SOH simulations.

**[THERM_SAT]**: If this bit is set to 1, IT simulations occurring near termination in a sustained discharge (when thermal saturation is reached) assume simulated temperature to be equal to measured cell temperature. This behavior does NOT apply to SOH simulations.

**[FOCV_EN]**: If this bit is set to 1, the gauge enables a fast OCV algorithm to predict the final OCV value, which reduces relaxation requirements for QMax updates.

| Class | Subclass | Name | Type | Min | Max | Default | Description |
|---|---|---|---|---|---|---|---|
| Settings | Configuration | I2C Gauging Configuration | H1 | 0x00 | 0x0F | 0x08 | Bit 0: RSOCL—*RelativeStateOfCharge()* and *RemainingCapacity()* behavior at end of charge<br>0 = Actual value shown (default)<br>1 = Held at 99% until valid charge termination. On entering valid charge termination, updates to 100%<br>Bit 1: RSOC_HOLD—Prevents RSOC from increasing during discharge<br>0 = RSOC is not limited.<br>1 = RSOC is not allowed to increase during discharge.<br>Bit 2: LOCK0—Keeps *RemainingCapacity()* and *RelativeStateOfCharge()* from jumping back during relaxation after 0 was reached during discharge<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 3:RSVD<br>BIT 4: 1PERCENT_HOLD<br>0 = Disabled<br>1= Enabled<br>Bit 5: TAMB_SYNC_SIM—Ambient Temperature Synchronization Simulation<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 7:6: Reserved |
| Settings | Configuration | IT Gauging Configuration | H2 | 0x0000 | 0xFFFF | 0xD5FE | Bit 0: CCT—Cycle count threshold<br>0 = Use *DesignCapacity()* for cycle count threshold (default)<br>1 = Use *FullChargeCapacity()* for cycle count threshold<br>Bit 1: CSYNC—Syncs *RemainingCapacity()* with *FullChargeCapacity()* at valid charge termination<br>0 = Not synchronized<br>1 = Synchronized (default)<br>Bit 2: RFACTSTEP—Ra factor step<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 3: OCVFR—Open circuit voltage flat region<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 4: DOD0EW—DOD0 error weighting<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 5: TDELTAV—Use Turbo power to calculate TermV Delta<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 6: RSOC_CONV—RSOC convergence (fast resistance scaling)<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 7: FAST_QMAX_LRN—Fast QMax LEARN mode<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 8: FAST_QMAX_FLD—Fast QMax FIELD mode<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 9: RSVD<br>Bit 10: FF_NEAR_EDV—Fast filter near EDV<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 11: RELAX_JUMP_OK—Enables RSOC Jumps in RELAX mode<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 12: SMOOTH—Smoothing engine<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 13: LFP_RELAX—Lithium iron phosphate relax<br>0 = Disabled<br>1 = Enabled<br>Bit 14: RELAX_SMOOTH_OK—Smoothing in RELAX<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 15: DOD_RSCALE_EN—Configures which DOD the new RaScale is to be applied.<br>0 = The RaScale is applied to all DODs during IT simulations.<br>1 = The RaScale is only applied to DODs higher than the DOD where the RaScale is calculated. (default) |
| Settings | Configuration | IT Gauging Ext | H2 | 0x0000 | 0x003F | 0x003B | Bit 0: DSG_0_SMOOTH_OK—Smoothing to 0%<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 1: CHG_100_SMOOTH_OK—Smoothing to 100%<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 2: AMB_PRED—Ambient temperature prediction<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 3: THERM_IV—Thermal model IV heat<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 4: THERM_SAT—Thermal model saturation<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 5: FOCV_EN—Enables fast OCV feature<br>0 = Disabled<br>1 = Enabled (default)<br>Bits 15:6: RSVD |

## 4.10 State-Of-Health (SOH)

The BQ27Z561-R2 device implements an updated version of the battery state-of-health (SOH), which is reported by the *StateOfHealth()* command. In the previous version (in the BQ27Z561 device), the *StateOfHealth()* was calculated using a special 25°C version of *FullChargeCapacity()* as a fraction of **Design Capacity**. However, this calculation still included some parameters that were updated during operation based on the loading and use of the battery, separate from the battery status itself. This could lead to varying calculations of state-of-health for the same battery, depending on how the battery was used in the system. The updated version of *StateOfHealth()* addresses this by calculating a special *FCC_SOH()*, with the initial ambient temperature fixed at 25°C, using a current load specified by **SOH Load Rate**, and using the thermal model parameters, **SOH Temp a** and **SOH Temp k**.

The **SOH Load Rate** can be set to the typical current of the application, and it is specified in units of hour-rate (that is, **Design Capacity** / **SOH Load Rate** will be the current used for the SOH simulation). The **SOH Temp a** and **SOH Temp k** can be set to expected values for the system during operation. These data flash settings are used only for the *StateOfHealth()* calculation. The SOH full charge capacity is updated whenever QMax or the impedance table is updated. Because this implementation removes the variation of load current, temperature, and learned parameters, it is a better representation of the battery's state-of-health. The SOH full charge capacity is available on the MAC command *FCC_SOH()*.

The configuration options for state-of-health (SOH) are in the following data flash.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Gas Gauging | SoH | SOH Load Rate | U1 | 1 | 255 | 50 | 0.1 Hr rate | Rate used in SOH simulation |
| Advanced Charge Algorithm | Rec Temp Charging | Voltage | I2 | 0 | 32767 | 4400 | mV | Constant value for charge terminate voltage for the SOH calculation only |
| Gas Gauging | State | SOH Temp k | I2 | 0 | 32767 | 1 | 0.1°C/ 256 cW | Thermal model temperature factor for SOH |
| Gas Gauging | State | SOH Temp a | I2 | 0 | 32767 | 1000 | s | Thermal model temperature factor for SOH |
| Gas Gauging | It Cfg | Min Delta Voltage | I2 | −32768 | 32767 | 0 | mV | Constant added to terminate voltage for the SOH simulation |

## 4.11 TURBO Mode 3.0

The BQ27Z561-R2 TURBO Mode 3.0 helps the system to adjust the power level by providing information about maximum power, depending on the battery state-of-charge, temperature, and present battery impedance. The gauge informs the system about the power level above which would cause the system voltage to drop below termination after the 10-s pulse, which is called the sustained peak power (SPP). In addition, the gauge also reports the maximum power for the combined 10-s and 10-ms pulses called the maximum peak power (MPP).

A system with TURBO Mode 3.0 applies short high-power load pulses (for example, up to 4 C-rate for as long as 10 ms). In addition, 10-s load pulses of 2 C-rate can occur in some cases prior to 10-ms pulses, resulting in a combined effect during the turbo boost operation. The 10-s pulse support is new in TURBO Mode 3.0. Additionally, TURBO Mode 3.0 provides Rhf effective and Vload parameters for the host to use to make power-level decisions.

These high-power pulses may drop down battery voltage. If the battery voltage drops below **Shutdown Voltage**, the system will shut down. To avoid shutting down the system during turbo boost operation, the system should never apply a pulse that would cause the system voltage to drop below the termination voltage (or exceed the recommended current threshold) that could result in a shutdown, reducing the total available run time.

The SPP is computed using a 10-s effective resistance that is temperature- and DOD-dependent. The computation of MPP uses the high-frequency resistance along with the 10-s effective resistance. Both of these resistances are chemistry-specific. In addition, the **Pack Resistance** and **System Resistance** are important parameters used in the calculation of these two powers. The computed TURBO mode currents, the sustained peak current, and the maximum peak current are capped to their respective maximum discharge rates. Depending on how often the system polls the peak power data and how fast the system can switch to a lower power mode, it is possible to exceed the reported peak power levels during the

present power consumption. To avoid any system shutdown, the gauge provides a *Reserve Energy %* setting that can serve as a buffer to ensure there is available energy at the present average discharge rate. These calculations occur on the cell level using **Term Min Cell V**, on the pack level using **Term Voltage**, and on the system level using **Min System Voltage**, **Pack Resistance**, and **System Resistance** with the most conservative prediction reported.

---

**NOTE:** *Min System Voltage* should be set lower than *Term Voltage*.

---

**Table 4-1. TURBO Mode Configuration**

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Gas Gauging | Turbo Cfg | Min System Voltage | I2 | 0 | 32567 | 3000 | mV | The minimum voltage at the pack terminal to be used for TURBO mode. |
| Gas Gauging | Turbo Cfg | Ten Second Max C Rate | I1 | 0 | 1127 | –20 | .1 hr | A 10-second load value to be used by the TURBO Mode calculation |
| Gas Gauging | Turbo Cfg | Ten Millisecond Max C Rate | I2 | 0 | –127 | –40 | .1 hr | A 10-ms load value to be used by the TURBO Mode calculation |
| Gas Gauging | Turbo Cfg | High Frequency Resistance | I2 | 0 | 32767 | 36 | mΩ | The high-frequency resistance related to the specific cell chemistry and pack configuration |
| Gas Gauging | Turbo Cfg | Reserve Energy % | U1 | 0 | 100 | 0 | % | This sets the % of remaining energy at the present average discharge rate (as defined in **Load Select**) until the maximum peak power reaches the value reported by *MaxPeakPower()*. |
| Gas Gauging | Turbo Cfg | Turbo Adjustment Factor | U1 | 0 | 255 | 100 | — | This is a resistance correction factor that, if used, would be a one-time adjustment the user computes from a 10-s pulse test. |

## 4.12 Battery Trip Point (BTP)

Used with WIN8 OS and later, the battery trip point (BTP) feature indicates when the RSOC of a battery pack has depleted to a certain value set in a DF register.

The BTP feature enables a host to program two capacity- or RSOC-based thresholds that govern triggering a BTP interrupt on the BTP_INT pin, and setting or clearing of the *OperationStatus()[BTP_INT]* on the basis of *RemainingCapacity()* or *RelativeStateOfCharge()*. The source of programmed thresholds (capacity or RSOC) is selected via **Settings.Configuration.IO Config[BTP_SRC]**. The interrupt is enabled or disabled via **Settings.Configuration.IO Config[BTP_EN]**. Similarly, the polarity of the interrupt is configurable based on the value set in **Settings.Configuration.IO Config[INT_POL]**.

- *OperationStatus()[BTP_INT]* is set when:
  - Current > 0 and RemCap/RSOC > "clear" threshold as set by *BTP Chg Set()* This threshold is initialized at reset from **Settings.BTP.Init Charge Set** or **Settings.BTP.Init Charge Set RSOC**.
  - Current ≤ 0 and RemCap < "set" threshold as set by *BTP Dsg Set()*. This threshold is initialized at reset from **Settings.BTP.Init Discharge Set** or **Settings.BTP.Init Discharge Set RSOC**.
- When *OperationStatus()[BTP_INT]* is set and if **Settings.Configuration.IO Config[BTP_EN]** is set, then the BTP_INT pin output is asserted.
  - If **Settings.Configuration.IO Config[INT_POL]** is set, it will assert high; otherwise, it will assert low.
- When either *BTPDischargeSet()* or *BTPChargeSet()* commands are received, *OperationStatus()[BTP_INT]* will clear and the pin will be deasserted. The new threshold is written to either *BTPDischargeSet()* or *BTPChargeSet()*.
- At reset, the pin is set to the deasserted state.
  - If *[INT_POL]* is changed, one of the BTP commands must be reset or sent to "clear" the state.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Settings | BTP | Init Discharge Set | I2 | 0 | 32767 | 150 | mAh | Initial value for *BTPDischargeSet()*, for ***Settings.Configuration.IO Config[BTP_SRC]*** = 0 |
| Settings | BTP | Init Charge Set | I2 | 0 | 32767 | 175 | mAh | Initial value for *BTPChargeSet()*, for ***Settings.Configuration.IO Config[BTP_SRC]*** = 0 |
| Settings | BTP | Init Discharge Set RSOC | U1 | 0 | 255 | 10 | % | Initial value for *BTPDischargeSet()*, for ***Settings.Configuration.IO Config[BTP_SRC]*** = 1 |
| Settings | BTP | Init Charge Set RSOC | U1 | 0 | 255 | 15 | mAh | Initial value for *BTPChargeSet()*, for ***Settings.Configuration.IO Config[BTP_SRC]*** = 1 |

# Lifetime Data Collection

## 5.1 Description

The device has the capability to log events over the life of the battery, which is useful for analysis. Lifetime data collection is enabled by setting **ManufacturingStatus[LF_EN]** = 1. The data is collected in RAM and only written to data flash under the following conditions to avoid wear out of the data flash:

- Every 10 hours if RAM content is different from data flash.
- A reset counter increments. The lifetime RAM data is reset; therefore, only the reset counters are updated to data flash.

The lifetime data stops collecting under the following conditions:

- Lifetime data collection is disabled by setting **ManufacturingStatus[LF_EN]** = 0.

When the gauge is unsealed, the following *ManufacturingStatus()* can be used for testing lifetime data.

- *LifetimeDataReset()* can reset the lifetime data (RAM and data flash) to the default values.
- *LifetimeDataFlush()* can force an update of the RAM lifetime data to data flash.
- *LifetimeDataSpeedupMode()* can increase the rate the lifetime data is incremented.

The following lifetime data is collected when **ManufacturingStatus[LF_EN]** = 1.

- Voltage
  - Maximum/minimum cell voltage
- Current
  - Max charge/discharge current
  - Maximum average discharge current
- Temperature
  - Max/min cell temp
- Charging Events
  - Number of valid charge terminations (That is, the number of times *[VCT]* is set.)
  - Cycle count at last charge termination
- Gauging events
  - Number of QMax updates
  - Cycle Count at Last QMax update
  - Number of RA updates and disable
  - Cycle Count at Last RA update and disable
- Time based data collection feature. (This data is stored with a resolution of 1 second up to over 100 years.)
  - Total runtime
  - Time spent in different *RelativeStateOfCharge()* versus *Temperature()* ranges. shows the organization of the binned data that is collected by this Lifetimes feature.
    - Eight *RelativeStateOfCharge()* columns for each of the seven charge temperature ranges
    - Seven rows of *Temperature()* runtime values

**Table 5-1. Time Spent in** *RelativeStateOfCharge()* **–** *Temperature()* **Ranges**

|  | RSOC ≥ 95% [default] | RSOC ≥ 90% | RSOC ≥ 80% | RSOC ≥ 60% | RSOC ≥ 40% | RSOC ≥ 20% | RSOC ≥ 10% | RSOC ≥ 0% |
|---|---|---|---|---|---|---|---|---|
| Undertemperature |  |  |  |  |  |  |  |  |
| Low Temperature |  |  |  |  |  |  |  |  |
| Standard Temperature Low |  |  |  |  |  |  |  |  |
| Recommended Temperature |  |  |  |  |  |  |  |  |
| Standard Temperature High |  |  |  |  |  |  |  |  |
| High Temperature |  |  |  |  |  |  |  |  |
| Over Temperature |  |  |  |  |  |  |  |  |

## 5.2 Reset

In addition to the *ManufacturerAccess() 0x0028 Lifetime Data Reset*, **Lifetime Data Collection** can also be reset when **[SEALED_RESET]** is set using a two-word MAC sequence available in SEALED and UNSEALED modes. The two-word key is programmable using *ManufacturerAccess() 0x0035 Security Keys*. Both keys must be sent within 4 seconds of each other for **LIfetimes** data to reset.

# *Host Interrupts*

## 6.1 Description

The device can be enabled to generate an interrupt to the host processor on the INT pin under various voltage, temperature, and RSOC conditions. The **IO Config[INT_EN]** bit is used to enable or disable the interrupt function, and its polarity can be set to active high (**IO Config[INT_POL]** = 1) or active low (**IO Config[INT_POL]** = 0). The **IO Config[INT_TYPE]** bit is used to select if interrupt is a pulse on the INT pin with an approximate width of 1 ms (**IO Config[INT_TYPE]** = 1) or level-based interrupt (**IO Config[INT_TYPE]** = 0). The **IO Config[INT_PUP]** bit is to select if INT output is open-drain (**IO Config[INT_PUP]** = 0) or push-pull (**IO Config[INT_PUP]** = 1).

## 6.2 Voltage and Temperature Interrupts

The device can be enabled to generate an interrupt on the INT pin to indicate an overvoltage, undervoltage, overtemperature, or undertemperature condition. The thresholds to set and clear each event are configurable through data flash for the initial values, and later by writing to the standard commands referenced below:

- The voltage for the overvoltage set interrupt threshold is *VoltHiSetThreshold()*.
- The voltage for the overvoltage clear interrupt threshold is *VoltHiClearThreshold()*.
- The voltage for the undervoltage set interrupt threshold is *VoltLoSetThreshold()*.
- The voltage for the undervoltage clear interrupt threshold is *VoltLoClearThreshold()*.
- The temperature for the overtemperature set interrupt threshold is *TempHiSetThreshold()*.
- The temperature for the overtemperature clear interrupt threshold is *TempHiClearThreshold()*.
- The temperature for the undertemperature set interrupt threshold is *TempLoSetThreshold()*.
- The temperature for the undertemperature clear interrupt threshold is *TempLoClearThreshold()*.

The interrupts are indicated in *InterruptStatus()*.

Table 6-1 summarizes the interrupt flags, which are *[VOLT_HI]*, *[VOLT_LO]*, *[TEMP_HI]*, and *[TEMP_LO]*.

**Table 6-1. Interrupt Flags (Interrupt Status)**

| Flag | Set Criteria | Clear Criteria |
|------|--------------|----------------|
| *[VOLT_HI]* | *Voltage() > VoltHiSetThreshold()* | *Voltage() < VoltHiClearThreshold()* |
| *[VOLT_LO]* | *Voltage() < VoltLoSetThreshold()* | *Voltage() > VoltLoClearThreshold()* |
| *[TEMP_HI]* | *Temperature() > TempHiSetThreshold()* | *Temperature() < TempHiClearThreshold()* |
| *[TEMP_LO]* | *Temperature() < TempLoSetThreshold()* | *Temperature() > TempLoClearThreshold()* |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Settings | Configuration | Init Voltage High Set | I2 | 0 | 5000 | 4500 | mV | Initial value for *VoltHiSetThreshold()*, the high cell voltage set threshold |
| Settings | Configuration | Init Voltage High Clear | I2 | 0 | 5000 | 4400 | mV | Initial value for *VoltHiClearThreshold()*, the high cell voltage clear threshold |
| Settings | Configuration | Init Voltage Low Set | I2 | 0 | 5000 | 2500 | mV | Initial value for *VoltLoSetThreshold()*, the low cell voltage set threshold |
| Settings | Configuration | Init Voltage Low Clear | I2 | 0 | 5000 | 2600 | mV | Initial value for *VoltLoClearThreshold()*, the low cell voltage clear threshold |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Settings | Configuration | Init Temperature High Set | I1 | −128 | 127 | 60 | °C | Initial value for *TempHiSetThreshold()*, the high temperature set threshold |
| Settings | Configuration | Init Temperature High Clear | I1 | −128 | 127 | 55 | °C | Initial value for *TempHiClearThreshold()*, the high temperature clear threshold |
| Settings | Configuration | Init Temperature Low Set | I1 | −128 | 127 | 0 | °C | Initial value for *TempLoSetThreshold()*, the low temperature set threshold |
| Settings | Configuration | Init Temperature Low Clear | I1 | −128 | 127 | 5 | °C | Initial value for *TempLoClearThreshold()*, the low temperature clear threshold |

## 6.3 RSOC Interrupts

The device can be enabled to generate an interrupt on the INT pin at configurable RSOC intervals. The threshold setting is configurable through **SOC Delta** for the initial value, and later by writing to the standard command *SOCSetDeltaThreshold()*. Setting **SOC Delta** to 0 disables RSOC interrupts; otherwise, interrupts will be generated at the following RSOC points:

- 100%
- 100% – n × **SOC Delta**, where n is an integer from 1 to INT(100 / **SOC Delta**)
- 0%

The RSOC interrupt is indicated in *InterruptStatus()* by *[SOC_DELTA]* flag.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Settings | Configuration | SOC Delta | U1 | 0 | 100 | 1 | % | Initial value for *SOCSetDeltaThreshold()*, RSOC delta set threshold. It is the interval for generating RSOC-based interrupts on the INT pin. Set to 0 to disable these RSOC-based interrupts. Otherwise, interrupts are generated at the following RSOC points: 0%, 100%, 100% – n × **SOC Delta**, where n is an integer from 1 to INT(100 / **SOC Delta**). |

## 6.4    Host Interrupts Configuration

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Settings | Configuration | IO Config | H1 | 0x00 | 0x3C | 0x00 | — | Bit 0: BTP_EN<br>0 = Disables assertion of the INT pin when BTP is triggered<br>1 = Enables assertion of the INT pin when BTP is triggered<br>Bit 1: = BTP_SRC<br>0 = BTP interrupts use capacity-based thresholds<br>1 = BTP interrupts use RSOC-based thresholds<br>Bit 2: INT_EN<br>0 = Disables interrupt function on pin INT<br>1 = Enables interrupt function on pin INT<br>Bit 3: INT_POL<br>0 = Interrupt (INT) is active LOW.<br>1 = Interrupt (INT) is active HIGH.<br>Bit 4: INT_PUP<br>0 = Interrupts on the INT pin are open-drain type (pull up required),<br>1 = Interrupts on the INT pin are push-pull type (internal pull up used),<br>Bit 5: INT_TYPE<br>0 = Interrupts enabled on INT pin as level-type<br>1 = Interrupts enabled on INT pin as pulse-type<br>Bit 6: GPIO_POL<br>0 = Pulse (PULS) is active LOW.<br>1 = Pulse (PULS) is active HIGH.<br>Bit 7: GPIO_LEVEL_EN<br>0 = Pulse (PULS) is pulse only.<br>1 = Pulse (PULS) is level set. |

# *Programmable Pulse Width*

## 7.1 Description

The device can be enabled to generate a single pulse on the PULS pin with a programmable width. The polarity can also be set for active high or active low with **IO Config[GPIO_POL]**. The PULS pin can be used to signal other system components, via a command to the gauge, without a connection to the host processor.

The PULS pin is open-drain so a resistor must be tied to a pullup voltage. The pulse width can be set in **GPIO Pulse Width** in steps of 25 ms. Setting the width to 0 effectively disables the pulse. A single pulse is generated each time the *PulseGPIO()* command is sent.

Additionally, the PULS pin can be programmed to a level (high or low) with the **IO Config[GPIO_LEVEL_EN]** bit. When **IO Config[GPIO_LEVEL_EN]** is set, the *ClearGPIO()* command will clear the state of the PULS pin and *SetGPIO()* will set the state of the PULS pin.



**Figure 1. PULS Open-Drain Configuration**



$t_1$ - $t_0$ = GPIO Pulse Width
Active High / Low set with [GPIO_POL]
**Figure 2. Pulse Width and Polarity**
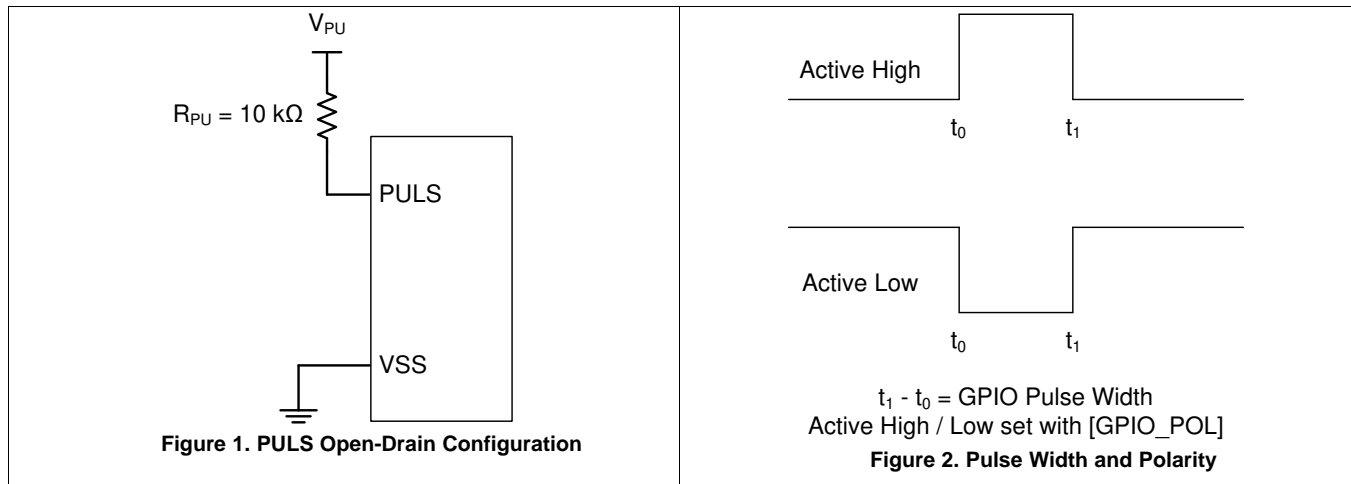
## 7.2 Programmable Pulse Width Configuration

| Class | Subclass | Name | Type | Min | Max | Default | Description |
|-------|----------|------|------|-----|-----|---------|-------------|
| Settings | Configuration | IO Config | H1 | 0x0 | 0x3C | 0x00 | Bit 0: BTP_EN<br>0 = Disables assertion of the INT pin when BTP is triggered.<br>1 = Enables assertion of the INT pin when BTP is triggered.<br>Bit 1: = BTP_SRC<br>0 = BTP interrupts use capacity-based thresholds.<br>1 = BTP interrupts use RSOC-based thresholds.<br>Bit 2: INT_EN<br>0 = Disables the interrupt function on the INT pin<br>1 = Enables the interrupt function on the INT pin<br>Bit 3: INT_POL<br>0 = Interrupt (INT) is active LOW.<br>1 = Interrupt (INT) is active HIGH.<br>Bit 4: INT_PUP<br>0 = Interrupts on the INT pin are open-drain type (pull up required).<br>1 = Interrupts on the INT pin are push-pull type (internal pull up used).<br>Bit 5: INT_TYPE<br>0 = Interrupts enabled on INT pin as level-type<br>1 = Interrupts enabled on INT pin as pulse-type<br>Bit 6: GPIO_POL<br>0 = Pulse (PULS) is active LOW.<br>1 = Pulse (PULS) is active HIGH.<br>Bit 7: GPIO_LEVEL_EN<br>0 = Pulse (PULS) is pulse only.<br>1 = Pulse (PULS) is level set. |
| Settings | Configuration | GPIO Pulse Width | U1 | — | 255 | 7 | Pulse width of PULS (7 indicates 7 × 25 ms = 175 ms) |

# Device Security

## 8.1 Description

There are three levels of secured operation within the device: SEALED, UNSEALED, and FULL ACCESS. To switch between the levels, different operations are needed with different access keys. The device also supports SHA-256 HMAC authentication with the host system. Additionally, lifetime data is protected by an access key.

## 8.2 SHA-256 Authentication

The BQ27Z561-R2 uses SHA-256 for authentication.

- SHA-256 preprocessing is described in the *SHA-1 Preprocessing* section in FIPS 180-4.
- The hash computation is described in the *SHA-1 Hash Computation* section in FIPS 180-4.
- The HMAC implementation is described in FIPS 198-1.

Detailed information about the SHA-256 algorithm can be found here:

- http://www.nist.gov/itl/
- http://csrc.nist.gov/publications/fips
- www.faqs.org/rfcs/rfc3174.html

The random number should be a 32-byte random number generated from the host processor system. Once this number is generated, it is used to generate the HMAC value using the random number as the message and the secure key as the secret key for SHA-256.

### 8.2.1 Secure Key

The secure key is stored in the secured memory of the BQ27Z561-R2 device. It is located in secure memory, which can store up to four secure keys.

*Auth Config* selects which of the four keys to be used with KEY_SEL0 and KEY_SEL1.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Settings | Configuration | Auth Config | H1 | 0x0 | 0x03 | 0x00 | — | Bit 0: KEY_SEL0<br>Bit 1: KEY_SEL1<br>00 = Enables KEY 1<br>01 = Enables KEY 2<br>10 = Enables KEY 3<br>11 = Enables KEY 4 |

### 8.2.2 Authentication Flow

The authentication procedure is described in the following steps:

1. The host writes 0x00 to 0x3E.
2. The host writes 0x00 to 0x3F.
3. Write the random challenge should be written in a 32-byte block to address 0x40–0x5F.
4. Write the checksum (1's complement sum of (1), (2), and (3)) to address 0x60.
5. Write the length 0x24 to address 0x61.
6. The gauge will compute the HMAC using SHA-256 after step (5). Wait 200 ms. (60 ms is how long authentication takes, but extra margin is added).

7. The *MACData()* command will contain the computed HMAC result using SHA-256.

8. *[AUTH]* = 1 in *OperationStatusB()*.

## 8.3 Security Modes

As shipped, the device is in FULL ACCESS mode. For device security after manufacture, the device should be set to the SEALED access operational mode. To do this, send the MAC seal 0x0030 to the gauge. After the device is set to the SEALED mode, any device resets will cause the device to enter the SEALED mode. See the following sections for instructions on how to return to the other access modes.

For security mode access keys, the device ships with factory defaults. The manufacturer should change the keys to their proprietary keys using the MAC 0x0035. See Section 13.8.19.

### 8.3.1 SEALING and UNSEALING Data Flash

The gas gauge has 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 gas gauge via the *AltManufacturerAccess()* command. The keys must be sent consecutively, with no other data being written to the *AltManufacturerAccess()* register. The *Seal Device* command instructs the device to limit access to the registers, functions, and data flash space, and sets the *[SEC1][SEC0]* flags. In SEALED mode, standard register information is accessible. Extended MAC Command functions and data flash are not accessible. Once in SEALED mode, the device can never permanently return to UNSEALED or FULL ACCESS modes. Section 13.8.33, the *OperationStatusB()* register, shows the status of the device using the *[SEC1][SEC0]* bits.

### 8.3.2 SEALED to UNSEALED

SEALED to UNSEALED instructs the device to extend access to the standard and extended registers and data flash space and clears the *[SEC1][SEC0]* flags. In UNSEALED mode, all data, standard and extended registers and DF have read/write access. Unsealing is a two-step command performed by writing the first word of the unseal key to *AltManufacturerAccess() (MAC)*, followed by the second word of the unseal key to *AltManufacturerAccess()*. The unseal key can be read and changed via the *MAC SecurityKey()* command when in FULL ACCESS mode. To return to the SEALED mode, either a hardware reset is needed, or the *MAC Seal Device()* command is needed to transit from FULL ACCESS or UNSEALED to SEALED.

### 8.3.3 UNSEALED to FULL ACCESS

UNSEALED to FULL ACCESS instructs the device to allow full access to all standard and extended registers and data flash. The device is shipped from TI in this mode. The keys for UNSEALED to FULL ACCESS can be read and changed via the MAC command *SecurityKey()* when in FULL ACCESS mode. Changing from UNSEALED to FULL ACCESS is performed by using the *AltManufacturerAccess()* command, by writing the first word of the Full Access Key to *AltManufacturerAccess()*, followed by the second word of the Full Access Key to *AltManufacturerAccess()*. In FULL ACCESS mode, the command to go to boot ROM can be sent.

# Advanced Charge Algorithm

## 9.1 Introduction

The BQ27Z561-R2 device does not control the charge directly. It can interact with a smart charger to enhance its performance. The device can change the values of *ChargingVoltage()* and *ChargingCurrent()* based on *Temperature()* and *Voltage()*. Its flexible charging algorithm is JEITA-compatible and can also meet other specific cell manufacturer charge requirements. The *ChargingStatus()* register shows the state of the charging algorithm.
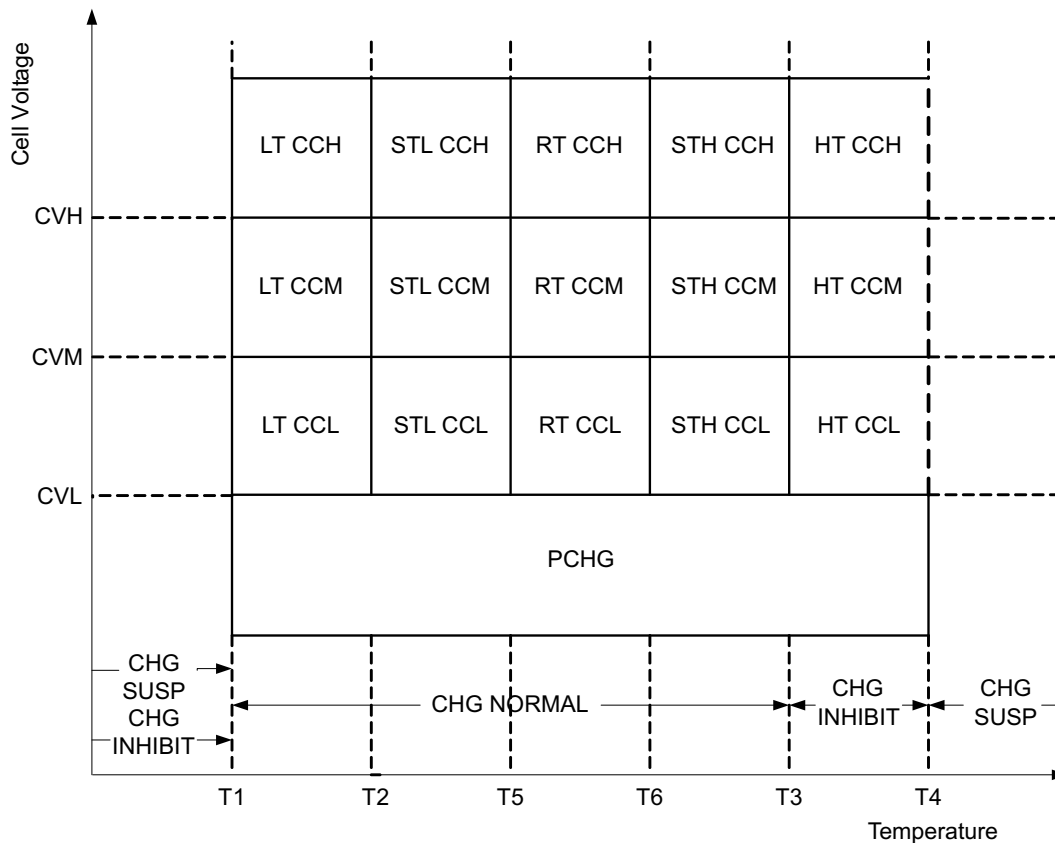


**Figure 9-1. Charging Matrix**

## 9.2 Charge Temperature Ranges

The measured temperature is segmented into several temperature ranges. The charging algorithm adjusts *ChargingCurrent()* and *ChargingVoltage()* according to the temperature range. The temperature ranges set in data flash should adhere to the following format:

*T1 ≤ T2 ≤ T5 ≤ T6 ≤ T3 ≤ T4*

*Temp Range* bit definitions can be found in *ChargingStatus()*.
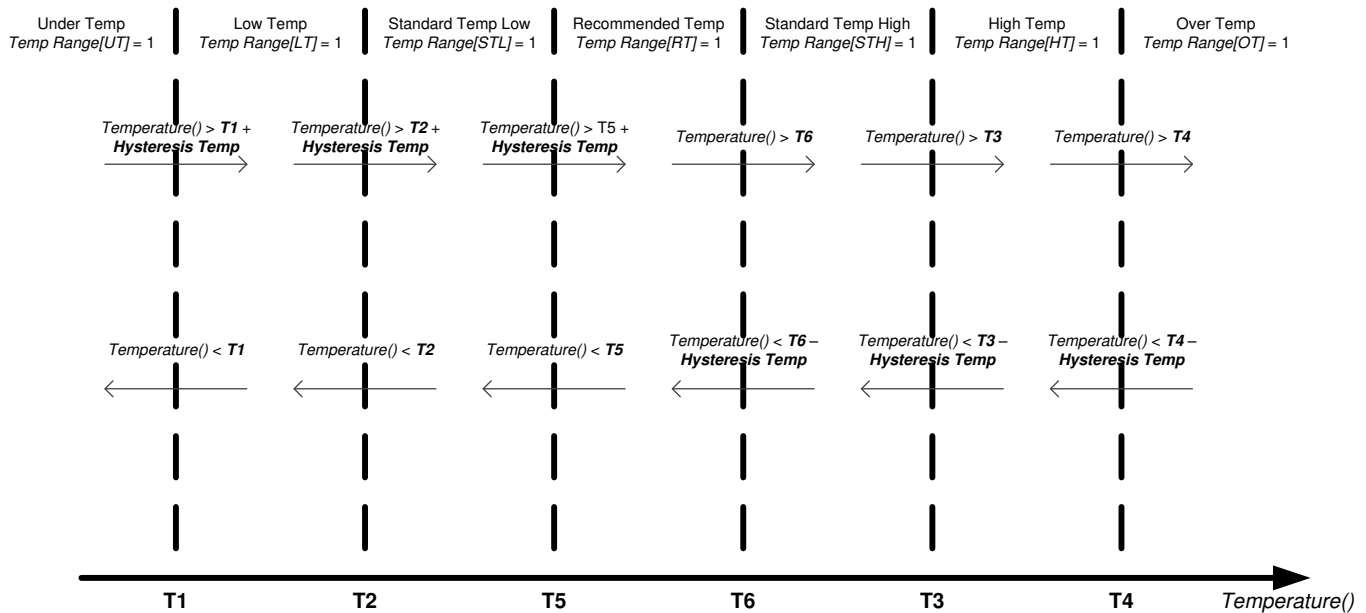
**Figure 9-2. Charging Across Temperature**

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Advanced Charge Algorithms | Temperature Ranges | T1 | I1 | −128 | 127 | 0 | °C | T1 low temperature range lower limit |
| Advanced Charge Algorithms | Temperature Ranges | T2 | I1 | −128 | 127 | 10 | °C | T2 low temperature range to standard temperature range |
| Advanced Charge Algorithms | Temperature Ranges | T5 | I1 | −128 | 127 | 20 | °C | T5 recommended temperature range lower limit |
| Advanced Charge Algorithms | Temperature Ranges | T6 | I1 | −128 | 127 | 25 | °C | T6 recommended temperature range upper limit |
| Advanced Charge Algorithms | Temperature Ranges | T3 | I1 | −128 | 127 | 45 | °C | T3 standard temperature range to high temperature range |
| Advanced Charge Algorithms | Temperature Ranges | T4 | I1 | −128 | 127 | 55 | °C | T4 high temperature range upper limit |
| Advanced Charge Algorithms | Temperature Ranges | Hysteresis | I1 | −128 | 127 | 1 | °C | Temperature hysteresis, applied when temperature is decreasing. |

## 9.3   Voltage Range

The measured cell voltage is segmented into several voltage ranges. The charging algorithm adjusts *ChargingCurrent()* according to the temperature range and voltage range. The voltage ranges set in data flash should adhere to the following format:

> *Charging Voltage Low ≤ Charging Voltage Med ≤ Charging Voltage High ≤ [Standard or Rec] Temp Charging:Voltage*

Depending on the specific charging profile, the **Low Temp Charging:Voltage** and **High Temp Charging:Voltage** settings do not necessarily have the highest setting values.
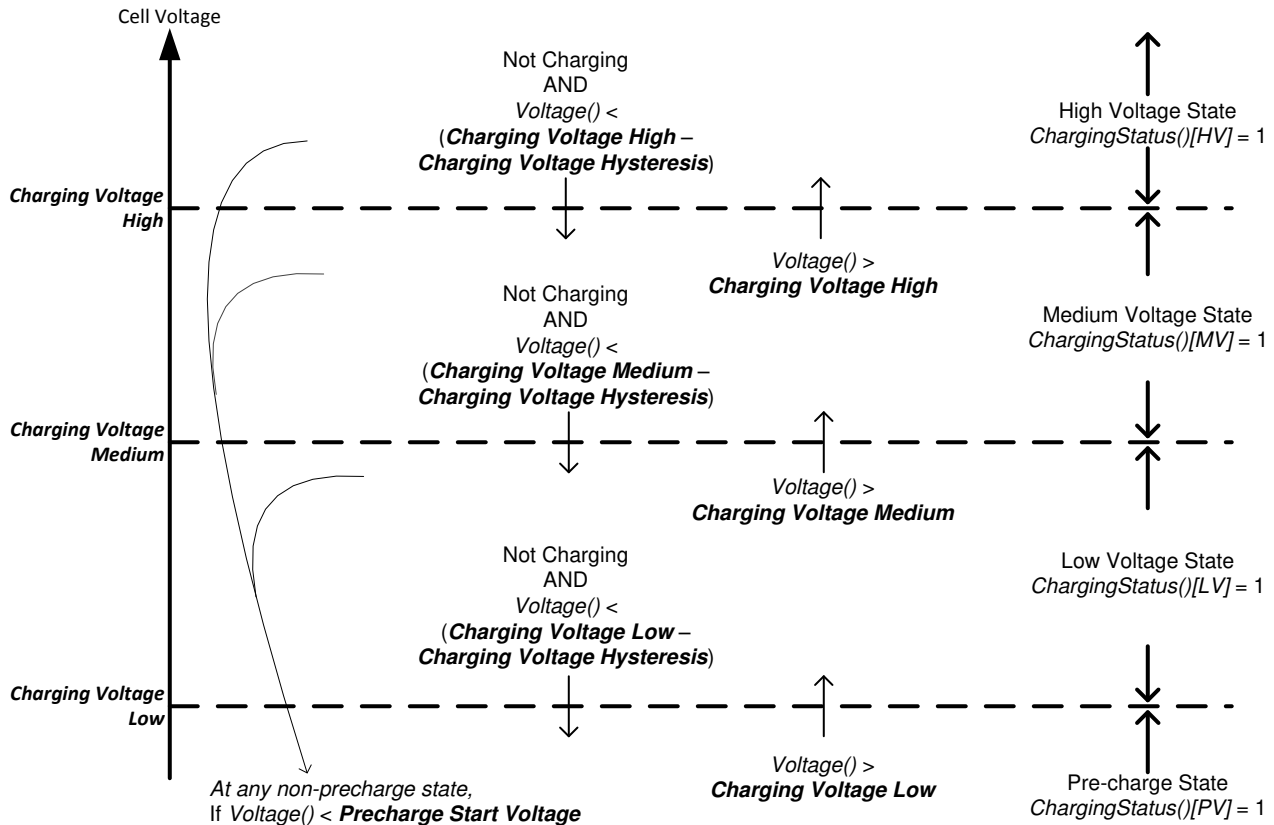


**Figure 9-3. Charging Voltage Ranges**

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Advanced Charging Algorithms | Voltage Range | Precharge Start Voltage | I2 | 0 | 32767 | 2500 | mV | Minimum cell voltage to enter PRECHARGE mode |
| Advanced Charging Algorithms | Voltage Range | Charging Voltage Low | I2 | 0 | 32767 | 2900 | mV | **Precharge Start Voltage** range to **Charging Voltage Low** range |
| Advanced Charging Algorithms | Voltage Range | Charging Voltage Med | I2 | 0 | 32767 | 3600 | mV | **Charging Voltage Low** range to **Charging Voltage Med** range |
| Advanced Charging Algorithms | Voltage Range | Charging Voltage High | I2 | 0 | 32767 | 4000 | mV | **Charging Voltage Med** to **Charging Voltage High** range |
| Advanced Charging Algorithms | Voltage Range | Charging Voltage Hysteresis | U1 | 0 | 255 | 0 | mV | **Charging Voltage Hysteresis** is applied when voltage is decreasing. |

### 9.3.1 *RelativeStateofCharge() Range*

If **[SOC_CHARGE]** in **Charging Configuration** is set, then the voltage-based threshold control, as described in Section 9.3, is replaced with *RelativeStateOfCharge()* control.

With this method, the following changes in the control transitions occur:

a.  [LV] state and *RelativeStateOfCharge()* > **Charging SOC Mid**; move to [MV].

b.  [MV] state and *RelativeStateOfCharge()* > **Charging SOC High**; move to [HV].

c. [MV] state [DSG] = 1, and *RelativeStateOfCharge()* < **Charging SOC Mid − SOC Hysteresis**; move to [LV].

d. [HV] state [DSG] = 1, and *RelativeStateOfCharge()* < **Charging SOC High − Charging SOC Hysteresis**; move to [MV].

**Table 9-1.** *RelativeStateofCharge()* **Range**

| Class | Subclass | Name | Type | Min Value | Max Value | Default Value | Unit |
|---|---|---|---|---|---|---|---|
| Advanced Charge Algorithm | SOC Range | Charging SOC Mid | U1 | 0 | 100 | 50 | % |
| Advanced Charge Algorithm | SOC Range | Charging SOC High | U1 | 0 | 100 | 75 | % |
| Advanced Charge Algorithm | SOC Range | Charging SOC Hysteresis | U1 | 0 | 100 | 1 | % |

## 9.4 Charging Current

The *ChargingCurrent()* value changes depending on the detected temperature and voltage per the charging algorithm.

The **Charging Configuration[CRATE]** bit provides an option to adjust the *ChargingCurrent()* based on *FullChargeCapacity()/DesignCapacity()*.

For example, with **[CRATE]** = 1, if *FullChargeCapacity()*/*DesignCapacity()* = 90% and **Rec Temp Charging: Current Med** is active per the charging algorithm, the *ChargingCurrent()* = **Rec Temp Charging: Current Med** × 90%.

**NOTE:** Table priority is top to bottom.

| Temp Range | Voltage Range | Condition | Action |
|---|---|---|---|
| UT or OT | Any | — | *ChargingCurrent()* = 0 |
| Any | PV | — | *ChargingCurrent()* = **Pre-Charging:Current** |
| Any | LV, MV, or HV | *ChargingStatus()[MCHG]* = 1 | *ChargingCurrent()* = **Maintenance Charging:Current** |
| LT | LV | — | *ChargingCurrent()* = **Low Temp Charging:Current Low** |
| LT | MV | — | *ChargingCurrent()* = **Low Temp Charging:Current Med** |
| LT | HV | — | *ChargingCurrent()* = **Low Temp Charging:Current High** |
| STL | LV | — | *ChargingCurrent()* = **Standard Temp Low Charging:Current Low** |
| STL | MV | — | *ChargingCurrent()* = **Standard Temp Low Charging:Current Med** |
| STL | HV | — | *ChargingCurrent()* = **Standard Temp Low Charging:Current High** |
| STH | LV | — | *ChargingCurrent()* = **Standard Temp High Charging:Current Low** |
| STH | MV | — | *ChargingCurrent()* = **Standard Temp High Charging:Current Med** |
| STH | HV | — | *ChargingCurrent()* = **Standard Temp High Charging:Current High** |
| RT | LV | — | *ChargingCurrent()* = **Rec Temp Charging:Current Low** |
| RT | MV | — | *ChargingCurrent()* = **Rec Temp Charging:Current Med** |
| RT | HV | — | *ChargingCurrent()* = **Rec Temp Charging:Current High** |
| HT | LV | — | *ChargingCurrent()* = **High Temp Charging:Current Low** |
| HT | MV | — | *ChargingCurrent()* = **High Temp Charging:Current Med** |
| HT | HV | — | *ChargingCurrent()* = **High Temp Charging:Current High** |

The configuration options for charging current are in the following data flash.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Settings | Configuration | Charging Configuration | H2 | 0x00 | 0x3F85 | 0x00 | — | Bit 0: CRATE—*ChargingCurrent()* rate<br>0 = No adjustment to *ChargingCurrent()* (default)<br>1 = *ChargingCurrent()* adjusted based on *FullChargeCapacity()*/*DesignCapacity()*<br>Bit 1: Reserved<br>Bit 2: SOC_CHARGE<br>0 = Uses voltage thresholds (CVL, CVM, and CVH) in **Advanced Charging Algorithms**<br>1 = Enables the SOC threshold to replace voltage thresholds (CVL, CVM, and CVH) in **Advanced Charging Algorithms**<br>Bits 6:3: Reserved<br>Bit 7: CS_CV<br>0 = Cell swelling voltage degradation disabled<br>1 = Cell swelling voltage degradation enabled<br>Bit 8: COMP_IR<br>0 = Disables voltage compensation for system resistance<br>1 = Enables voltage compensation for system resistance<br>:Bit 9: DEGRADE_CC<br>0 = Disables *Charging Current()* degradation based on the charging degradation mode<br>1 = Enables *Charging Current()* degradation based on the charging degradation mode<br>bIT 10: CHGV_DEGRADE<br>0 = Disables *Charging Voltage()* degradation based on the charging degradation mode<br>1 = Enables *Charging Voltage()* degradation based on the charging degradation mode<br>Bit 11: HIBAT_CHG<br>0 = Disables the high battery voltage feature<br>= Enables the high battery voltage feature<br>Bit 12: SLEEPCHG<br>0 = *ChargingCurrent()* = 0 and *ChargingVoltage()* = 0 in SLEEP mode<br>1 = *ChargingCurrent()* and *ChargingVoltage()* are determined by **Advanced Charging Algorithms** in SLEEP mode.<br>Bit 13: TAPER_VOLT<br>0 = Uses *ChargingVoltage()* for charge termination detection<br>1 = Uses fixed **Charge Term Charging Voltage** for charge termination<br>Bits 14:15<br>RSVD |
| Advanced Charging Algorithms | Pre-Charging | Current | I2 | 0 | 32767 | 88 | mA | Precharge *ChargingCurrent()* |
| Advanced Charging Algorithms | Maintenance Charging | Current | I2 | 0 | 32767 | 44 | mA | Maintenance *ChargingCurrent()* |
| Advanced Charging Algorithms | Low Temp Charging | Current Low | I2 | 0 | 32767 | 132 | mA | Low temperature range low voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Low Temp Charging | Current Med | I2 | 0 | 32767 | 352 | mA | Low temperature range medium voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Low Temp Charging | Current High | I2 | 0 | 32767 | 264 | mA | Low temperature range high voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Standard Temp Low Charging | Current Low | I2 | 0 | 32767 | 1980 | mA | Standard temperature low range low voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Standard Temp Low Charging | Current Med | I2 | 0 | 32767 | 4004 | mA | Standard temperature low range medium voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Standard Temp Low Charging | Current High | I2 | 0 | 32767 | 2992 | mA | Standard temperature low range high voltage range *ChargingCurrent()* |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Advanced Charging Algorithms | Standard Temp High Charging | Current Low | I2 | 0 | 32767 | 1980 | mA | Standard temperature high range low voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Standard Temp High Charging | Current Med | I2 | 0 | 32767 | 4004 | mA | Standard temperature high range medium voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Standard Temp High Charging | Current High | I2 | 0 | 32767 | 2992 | mA | Standard temperature high range high voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | High Temp Charging | Current Low | I2 | 0 | 32767 | 1012 | mA | High temperature range low voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | High Temp Charging | Current Med | I2 | 0 | 32767 | 1980 | mA | High temperature range medium voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | High Temp Charging | Current High | I2 | 0 | 32767 | 1496 | mA | High temperature range high voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Rec Temp Charging | Current Low | I2 | 0 | 32767 | 2508 | mA | Recommended temperature range low voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Rec Temp Charging | Current Med | I2 | 0 | 32767 | 4488 | mA | Recommended temperature range medium voltage range *ChargingCurrent()* |
| Advanced Charging Algorithms | Rec Temp Charging | Current High | I2 | 0 | 32767 | 3520 | mA | Recommended temperature range high voltage range *ChargingCurrent()* |

## 9.5 Charging Voltage

The *ChargingVoltage()* changes depending on the detected temperature per the charge algorithm. If cell temperature reduces *ChargingVoltage()* below the cell voltage, setting **[HIBAT_CHG]** can hold it unchanged while *ChargingCurrent()* is held at 0. This action continues until the desired *ChargingVoltage()* is above the cell voltage.

> **NOTE:** Table priority is top to bottom.

| Temp Range | Condition | Action |
|---|---|---|
| UT or OT | — | *ChargingVoltage()* |
| LT | — | *ChargingVoltage()* = **Low Temp Charging:Voltage** |
| STL | — | *ChargingVoltage()* = **Standard Temp Low Charging:Voltage** |
| STH | — | *ChargingVoltage()* = **Standard Temp High Charging:Voltage** |
| RT | — | *ChargingVoltage()* = **Rec Temp Charging:Voltage** |
| HT | — | *ChargingVoltage()* = **High Temp Charging:Voltage** |

Table 9-2 shows the configurations for the charging voltages. They can be changed with the *ChargingVoltageOverride* MAC.

**Table 9-2. JEITA Charging Voltages**

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Advanced Charging Algorithms | Low Temp Charging | Voltage | I2 | 0 | 32767 | 4400 | mV | Low temperature range *ChargingVoltage()* |
| Advanced Charging Algorithms | Standard Temp Low Charging | Voltage | I2 | 0 | 32767 | 4400 | mV | Standard temperature low range *ChargingVoltage()* |
| Advanced Charging Algorithms | Standard Temp High Charging | Voltage | I2 | 0 | 32767 | 4400 | mV | Standard temperature high range *ChargingVoltage()* |

**Table 9-2. JEITA Charging Voltages (continued)**

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Advanced Charging Algorithms | High Temp Charging | Voltage | I2 | 0 | 32767 | 4100 | mV | High temperature range *ChargingVoltage()* |
| Advanced Charging Algorithms | Rec Temp Charging | Voltage | I2 | 0 | 32767 | 4400 | mV | Recommended temperature range *ChargingVoltage()* |

### 9.5.1 Charge Voltage Compensation for System Impedance

The design of some battery charging systems may have a not insignificant impedance between the charger and battery terminals. In this case, a voltage compensation feature handles system level IR drops to ensure the correct charging voltage is supplied at the battery terminals. Program the **System Resistance** register with the measured resistance in milliohms (m$\Omega$) between the battery terminals and charger terminals. To enable this feature, set the configuration bit **[COMP_IR]** in (default 0) the **Charging Configuration** register.

This feature works as follows:

*ChargingVoltage()* = charging_voltage + *ChargingCurrent()* × **System Resistance**

where charging_voltage was computed as a result of a selected configuration; from the JEITA table and any applied degradations.

## 9.6 Charging Degradation Modes

This optional feature can be used to reduce charging voltages and/or currents based on measured parameters that reflect the aging of cells.

Here is an example of charging degradation using **Cycle Count** (although it is the same for the other aging parameters as itemized below):

In NORMAL mode (there is no reduction of *ChargingCurrent()* or *ChargingVoltage()*), *ChargingCurrent()* is reduced by CC Degradation Mode 1 (assuming the Cycle Count 1 entry conditions are met), then moving to Cycle Count Mode 2, *ChargingCurrent()* is further reduced by CC Degradation Mode 2 (assuming Cycle Count 2 entry conditions are met). This is similar for Cycle Count Mode 3.

Care should be taken to set the degradation mode thresholds such that each threshold is increased for each mode. The fact that any of the other measured aging parameters have not passed a degradation mode threshold would not cause the lower mode to be used.

This charging current degradation scheme (if enabled) works in conjunction with any other existing degradation/increments (such as charge loss compensation).

The following table shows the default values for how charging voltage and charging current are degraded for each degradation mode:

| **[CHGV_DEGRADE] is set.)** | **Charging Voltage (CV)** (CV degradation is available if enabled.) | **Charging Current (CC)** (CC degradation is available if enabled [DEGRADE_CC]).[1] |
|---|---|---|
| Normal | *No CV Degradation* | *No CC Degradation* |
| Mode 1 | *CV Degradation* (default 10 mV / cell) | *CC Degradation* (default 10%) |
| Mode 2 | *CV Degradation* (default 40 mV / cell) | *CC Degradation* (default 20%) |
| Mode 3 | *CV Degradation* (default 70 mV / cell) | *CC Degradation* (default 40%) |

(1) Only **[DEGRADE_CC]** or **[CRATE]** can be used at a time. Both must not be enabled together.

A degrade mode is entered when any measured parameter exceeds the thresholds set in data flash. The following thresholds are defined for each mode (see *Data Flash Summary* for threshold value settings and addresses).

- **Cycle Threshold** checked against *CycleCount()*
- **SOH Threshold** checked against *StateOfHealth()*

- **Runtime Threshold** checked against **Total FW Runtime**
- **High SOC Threshold** checked against the sum of all the high SOC times recorded in **LIfetimes**
- **High Temp Threshold** checked against the sum of all the high temperature times recorded in **LIfetimes**
- **High Temp SOC Threshold** checked against **Time Spent in HT RSOC A**

*CycleCount()* must be greater than **Cycle Count Start Runtime** before any degradation modes are detected.

The current degradation mode can be determined from the value of *[CV_DGRD]*.

**Total FW Runtime** is updated to data flash every **LF Update Time**.

---

**NOTE:** A threshold can be set to an unobtainable value for a particular aging parameter to be disabled.

---

## 9.7 Cell Swelling Control (via Charging Voltage Degradation)

Cell swelling can occur when the cell temperature and cell voltage are above certain thresholds. In these situations, the charging voltage can be stepped down gradually until the cell temperature moves back down.

This scheme works (as shown in Figure 9-4) when enabled by setting *[CS_CV]* (default is cleared) in the **Charging Configuration** register. When the max cell voltage1..4 and cell temperature are above the **Voltage Threshold** and **Temperature Threshold**, respectively, for the period defined by **Time Interval**, then the charging voltage is stepped down by **Delta Voltage**. This step down continues until either the max cell voltage1..4 and cell temperature conditions go away (that is, cell swelling reduces) or the step down reaches **Min CV**.

The charging voltage reduction/degradation resulting from this feature is reset when exiting CHARGE mode.

---

**NOTE:** This degradation works in conjunction with other degradation features; therefore, use with care.

---
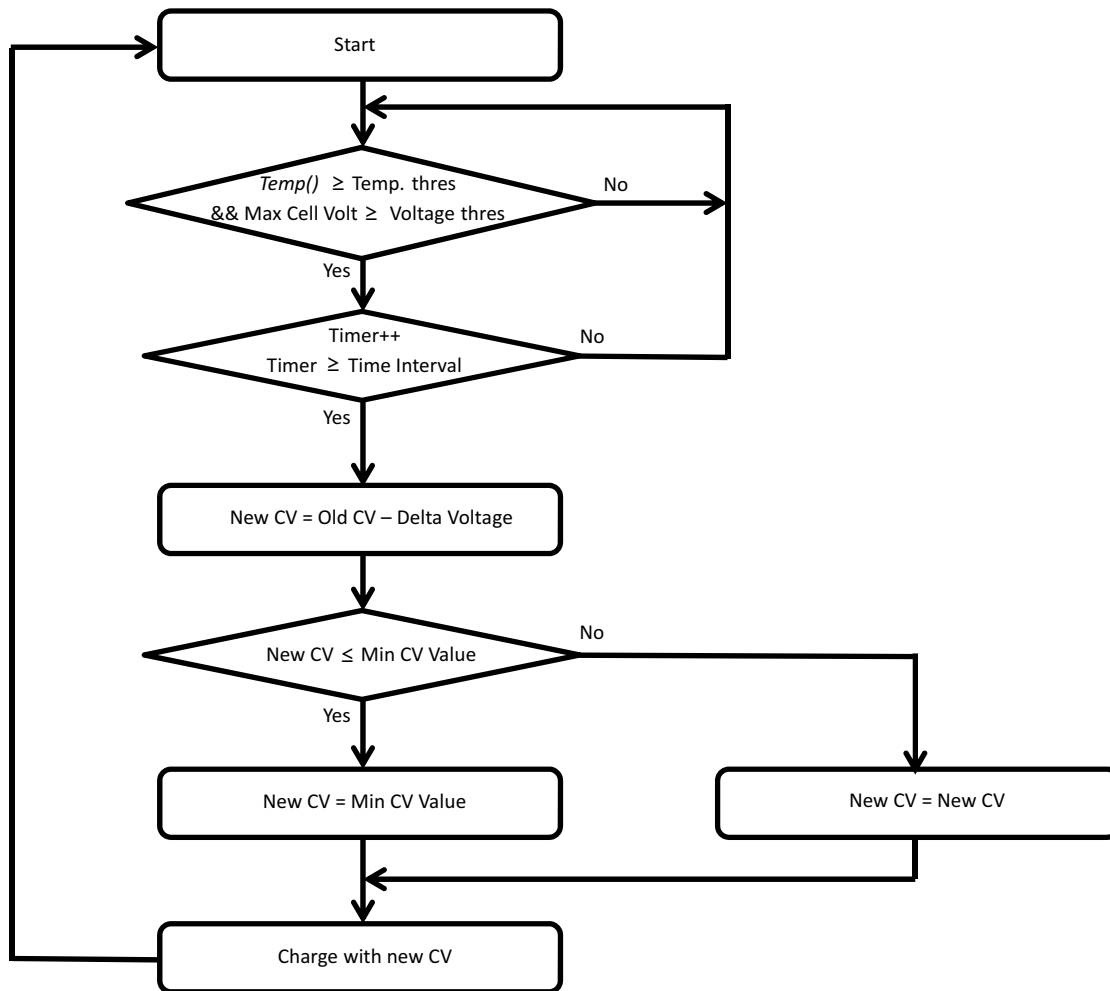
**Figure 9-4. Cell Swelling Control**

## 9.8 Valid Charge Termination

The charge termination condition must be met to enable valid charge termination. The device has the following actions at charge termination, based on the bits settings:

- If **IT Gauging Configuration[CSYNC]** = 1, *RemainingCapacity() = FullChargeCapacity()*.
- If **I2C Gauging Configuration[RSOCL]** = 1, *RelativeStateOfCharge()* and *RemainingCapacity()* are held at 99% until charge termination occurs. Only on entering charge termination is 100% displayed.
- If **I2C Gauging Configuration[RSOCL]** = 0, *RelativeStateOfCharge()* and *RemainingCapacity()* are not held at 99% until charge termination occurs. Fractions of percentages greater than 99% are rounded up to display 100%.

| Status | Condition | Action |
|---|---|---|
| Charging | *GaugingStatus()[REST]* = 0 AND<br>*BatteryStatus()[DSG]* = 0 | Charge Algorithm active |
| Valid Charge Termination | All of the following conditions must occur for two consecutive 40-s periods:<br>Charging (that is, *BatteryStatus()[DSG]* = 0) AND<br>*AverageCurrent()* < **Charge Term Taper Current** AND<br>*Voltage()* + **Charge Term Voltage** ≥ *ChargingVoltage()* AND<br>**[TAPER_VOLT]** = 0 AND<br>The accumulated change in capacity > 0.25 mAh | *ChargingStatus()[VCT]* = 1<br>*ChargingStatus()[MCHG]* = 1<br>*ChargingVoltage()* = Charging Algorithm<br>*ChargingCurrent()* = Charging Algorithm<br>*BatteryStatus()[FC]* = 1 and *GaugingStatus()[FC]* = 1 if **SOC Flag Config A[FCSETVCT]** = 1<br>*BatteryStatus()[TCA]* = 1 and *GaugingStatus()[TC]* = 1 if **SOC Flag Config A[TCSETVCT]** = 1<br>**Cell 1 Chg Voltage at EoC** = *Voltage()*<br>**Current at EoC** = *Current()*<br>Calculate Cell 1 DODEOC<br>Trigger IT simulation |

> **NOTE:** After valid charge termination, *RemainingCapacity()* and *FullChargeCapacity()* will reflect any change in capacity due to any difference in *ChargingVoltage()* from the previous valid charge termination.

> **NOTE:** Setting **[TAPER_VOLT]** causes **Charge Term Charging Voltage** to be used in place of *ChargingVoltage()*.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Advanced Charging Algorithms | Termination Config | Charge Term Taper Current | I2 | 0 | 32767 | 100 | mA | Valid charge termination taper current qualifier threshold |
| Advanced Charging Algorithms | Termination Config | Charge Term Voltage | I2 | 0 | 32767 | 100 | mV | Valid charge termination delta voltage qualifier, max cell based |
| Gas Gauging | State | Cell 1 Chg Voltage at EoC | I2 | 0 | 32767 | 4375 | mV | Cell 1 Voltage value at end of charge |
| Gas Gauging | State | Current at EoC | I2 | — | 32767 | 100 | mA | Current at end of charge |

## 9.9 Charge and Discharge Alarms

The *[TC]* and *[FC]* bits in *GaugingStatus()* can be set at charge termination and based on RSOC or cell voltage. If multiple set and clear conditions are selected, then the corresponding flag is set whenever a valid set or clear condition is met. If the set and clear conditions are true at the same time, the flag will clear. The same functionality is applied to the *[TD]* and *[FD]* bits in *GaugingStatus()*.

> **NOTE:** *GaugingStatus()[TC][TD][FC][FD]* are the status flags based on the gauging conditions only. These flags are set and cleared based on **SOC Flag Config A** and **SOC Flag Config B**.
>
> The *BatteryStatus()[TCA][TDA][FC][FD]* flags are set and cleared according to the *GaugingStatus()[TC][TD][FC][FD]* flags.

The table below summarizes the options to set and clear the *[TC]* and *[FC]* flags in *GaugingStatus()*.

| Flag | Set Criteria | Set Condition | Enable |
|---|---|---|---|
| [TC] | Cell Voltage | *Voltage()* > **TC: Set Voltage Threshold** | **SOC Flag Config A[TCSETV]** = 1 |
| | RSOC | *RelativeStateOfCharge()* > **TC: Set % RSOC Threshold** | **SOC Flag Config A[TCSETRSOC]** = 1 |
| | Valid Charge Termination (enable by default) | When *ChargingStatus[VCT]* = 1 | **SOC Flag Config A[TCSETVCT]** = 1 |

| Flag | Set Criteria | Set Condition | Enable |
|------|-------------|---------------|--------|
| *[FC]* | Cell Voltage | *Voltage() >* **FC: Set Voltage Threshold** | **SOC Flag Config B[FCSETV]** = 1 |
| | RSOC | *RelativeStateOfCharge() >* **FC: Set % RSOC Threshold** | **SOC Flag Config B[FCSETRSOC]** = 1 |
| | Valid Charge Termination (enable by default) | When *ChargingStatus[VCT]* = 1 | **SOC Flag Config A[FCSETVCT]** = 1 |

| Flag | Clear Criteria | Clear Condition | Enable |
|------|---------------|-----------------|--------|
| *[TC]* | Cell Voltage | *Voltage() ≤* **TC: Clear Voltage Threshold** | **SOC Flag Config A[TCCLEARV]** = 1 |
| | RSOC (enable by default) | *RelativeStateOfCharge() ≤* **TC: Clear % RSOC Threshold** | **SOC Flag Config A[TCCLEARRSOC]** = 1 |
| *[FC]* | Cell Voltage | *Voltage() ≤* **FC: Clear Voltage Threshold** | **SOC Flag Config B[FCCLEARV]** = 1 |
| | RSOC (enable by default) | *RelativeStateOfCharge() ≤* **FC: Clear % RSOC Threshold** | **SOC Flag Config B[FCCLEARRSOC]** = 1 |

The tables below summarize the various options to set and clear the *[TD]* and *[FD]* flags in *GaugingStatus()*.

| Flag | Set Criteria | Set Condition | Enable |
|------|-------------|---------------|--------|
| *[TD]* | Cell Voltage | *Voltage() <* **TD: Set Voltage Threshold** | **SOC Flag Config A[TDSETV]** = 1 |
| | RSOC (enable by default) | *RelativeStateOfCharge() <* **TD: Set % RSOC Threshold** | **SOC Flag Config A[TDSETRSOC]** = 1 |
| *[FD]* | Cell Voltage | *Voltage() <* **FD: Set Voltage Threshold** | **SOC Flag Config B[FDSETV]** = 1 |
| | RSOC (enable by default) | *RelativeStateOfCharge() <* **FD: Set % RSOC Threshold** | **SOC Flag Config B[FDSETRSOC]** = 1 |

| Flag | Clear Criteria | Clear Condition | Enable |
|------|---------------|-----------------|--------|
| *[TD]* | Cell Voltage | *Voltage() ≥* **TD: Clear Voltage Threshold** | **SOC Flag Config A[TDCLEARV]** = 1 |
| | RSOC (enable by default) | *RelativeStateOfCharge() ≥* **TD: Clear % RSOC Threshold** | **SOC Flag Config A[TDCLEARRSOC]** = 1 |
| *[FD]* | Cell Voltage | *Voltage() ≥* **FD: Clear Voltage Threshold** | **SOC Flag Config B[FDCLEARV]** = 1 |
| | RSOC (enable by default) | *RelativeStateOfCharge() ≥* **FD: Clear % RSOC Threshold** | **SOC Flag Config B[FDCLEARRSOC]** = 1 |

The configuration options for the charge and discharge alarms are in the following data flash.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Settings | Configuration | SOC Flag Config A | H2 | 0x0000 | 0x0FFF | 0x0C8C | — | Bit 0: TDSETV—Enables the TD flag set by cell voltage threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 1: TDCLEARV—Enables the TD flag cleared by the cell voltage threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 2: TDSETRSOC—Enables the TD flag set by the RSOC threshold<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 3: TDCLEARRSOC—Enables the TD flag cleared by the RSOC threshold<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 4: TCSETV—Enables the TC flag set by the cell voltage threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 5: TCCLEARV—Enables the TC flag cleared by the cell voltage threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 6: TCSETRSOC—Enables the TC flag set by the RSOC threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 7: TCCLEARRSOC—Enables the TC flag cleared by the RSOC threshold<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 8: Reserved<br>Bit 9: Reserved<br>Bit 10: FCSETVCT—Enables the FC flag set by valid charge termination<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 11: TCSETVCT—Enables the TC flag set by valid charge termination<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 15: 12: Reserved |
| Settings | Configuration | SOC Flag Config B | H1 | 0x00 | 0xFF | 0x8C | — | Bit 0: FDSETV—Enables the FD flag set by the cell voltage threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 1: FDCLEARV—Enables the FD flag cleared by the cell voltage threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 2: FDSETRSOC—Enables the FD flag set by the RSOC threshold<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 3: FDCLEARRSOC—Enables the FD flag cleared by the RSOC threshold<br>0 = Disabled<br>1 = Enabled (default)<br>Bit 4: FCSETV—Enables the FC flag set by the cell voltage threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 5: FCCLEARV—Enables the FC flag cleared by the cell voltage threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 6: FCSETRSOC—Enables the FC flag set by the RSOC threshold<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 7: FCCLEARRSOC—Enables the FC flag cleared by the RSOC threshold<br>0 = Disabled<br>1 = Enabled (default) |
| Gas Gauging | FD | Set Voltage Threshold | I2 | 0 | 5000 | 3000 | mV | *GaugingStatus()[FD]* and *BatteryStatus()[FD]* cell voltage set threshold |
| Gas Gauging | FD | Clear Voltage Threshold | I2 | 0 | 5000 | 3100 | mV | *GaugingStatus()[FD]* and *BatteryStatus()[FD]* cell voltage clear threshold |
| Gas Gauging | FD | Set RSOC % Threshold | U1 | 0 | 100 | 0 | % | *GaugingStatus()[FD]* and *BatteryStatus()[FD] RelativeStateOfCharge()* set threshold |
| Gas Gauging | FD | Clear RSOC % Threshold | U1 | 0 | 100 | 5 | % | *GaugingStatus()[FD]* and *BatteryStatus()[FD] RelativeStateOfCharge()* clear threshold |
| Gas Gauging | FC | Set Voltage Threshold | I2 | 0 | 5000 | 4400 | mV | *GaugingStatus()[FC]* and *BatteryStatus()[FC]* cell voltage set threshold |
| Gas Gauging | FC | Clear Voltage Threshold | I2 | 0 | 5000 | 4300 | mV | *GaugingStatus()[FC]* and *BatteryStatus()[FC]* cell voltage clear threshold |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | FC | Set RSOC % Threshold | U1 | 0 | 100 | 100 | % | *GaugingStatus()[FC]* and *BatteryStatus()[FC] RelativeStateOfCharge()* set threshold |
| Gas Gauging | FC | Clear RSOC % Threshold | U1 | 0 | 100 | 95 | % | *GaugingStatus()[FC]* and *BatteryStatus()[FC] RelativeStateOfCharge()* clear threshold |
| Gas Gauging | TD | Set Voltage Threshold | I2 | 0 | 5000 | 3200 | mV | *GaugingStatus()[TD]* cell voltage set threshold |
| Gas Gauging | TD | Clear Voltage Threshold | I2 | 0 | 5000 | 3300 | mV | *GaugingStatus()[TD]* cell voltage clear threshold |
| Gas Gauging | TD | Set RSOC % Threshold | U1 | 0 | 100 | 6 | % | *GaugingStatus[()TD] RelativeStateOfCharge()* set threshold |
| Gas Gauging | TD | Clear RSOC % Threshold | U1 | 0 | 100 | 8 | % | *GaugingStatus()[TD] RelativeStateOfCharge()* clear threshold |
| Gas Gauging | TC | Set Voltage Threshold | I2 | 0 | 5000 | 4400 | mV | *GaugingStatus()[TC]* cell voltage set threshold |
| Gas Gauging | TC | Clear Voltage Threshold | I2 | 0 | 5000 | 4300 | mV | *GaugingStatus()[TC]* cell voltage clear threshold |
| Gas Gauging | TC | Set RSOC % Threshold | U1 | 0 | 100 | 100 | % | *GaugingStatus()[TC] RelativeStateOfCharge()* set threshold |
| Gas Gauging | TC | Clear RSOC % Threshold | U1 | 0 | 100 | 95 | % | *GaugingStatus()[TC] RelativeStateOfCharge()* clear threshold |

## 9.10 Terminate Charge and Discharge Alarms

The *BatteryStatus()[TCA][TDA][[FC]FD]* flags are set according to *GaugingStatus()*. The following is a summary of the set conditions and their various alarm flags:

*[TCA]* = 1
- *GaugingStatus()[TC]* = 1 AND in CHARGE mode

*[FC]* = 1
- *GaugingStatus()[FC]* = 1

*[TDA]* = 1 if
- *GaugingStatus()[TD]* = 1 AND in DISCHARGE mode

*[FD]* = 1 if
- *GaugingStatus()[FD]* = 1

## 9.11 Precharge

The device enters PRECHARGE mode if the cell voltage falls below **Precharge Start Voltage**.

## 9.12 Maintenance Charge

Maintenance charge *[MCHG]* can allow charge after termination has been reached. This is only possible if the *GaugingStatus()[TC]* flag is not set. This means to use maintenance charge. *[TCSETVCT]* and *[TCSETSOC]* should not be enabled, and instead *[TCSETV]* can be used to stop maintenance charge based on voltage.

| Status | Condition | Action |
|--------|-----------|--------|
| Set | *ChargingStatus()[IN]* = 0 AND *ChargingStatus()[SU]* = 0 AND *ChargingStatus()[PV]* = 0 AND *GaugingStatus()[TCA]* = 1 | *ChargingStatus()[MCHG]* = 1 *ChargingVoltage()* = Charging Algorithm *ChargingCurrent()* = Charging Algorithm |
| Clear | *ChargingStatus()[IN]* = 1 OR *ChargingStatus()[SU]* = 1 OR *ChargingStatus()[PV]* = 1 OR *GaugingStatus()[TCA]* = 0 | *ChargingStatus()[MCHG]* = 0 *ChargingVoltage()* = Charging Algorithm *ChargingCurrent()* = Charging Algorithm |

## 9.13 Charge Inhibit

The BQ27Z561-R2 device can provide information to the smart charger to enable it to inhibit the start of charging at high and low temperatures to prevent damage of the cells. This feature is intended to enable the charger to prevent the start of charging when the temperature is at the inhibit range; therefore, if charging is in progress, the charger must determine its actions. *Temp Range* bit definitions can be found in *ChargingStatus()*.

| Status | Condition | Action |
|--------|-----------|--------|
| Normal | *Temp Range[LT]* = 1 OR<br>*Temp Range[STL]* = 1 OR<br>*Temp Range[RT]* = 1 OR<br>*Temp Range[STH]* = 1 | *ChargingStatus()[IN]* = 0<br>*ChargingVoltage()* = charging algorithm<br>*ChargingCurrent()* = charging algorithm |
| Trip | Not charging AND *Temp Range[HT]* = 1 | *ChargingStatus()[IN]* = 1<br>*ChargingStatus()[SU]* = 0<br>*ChargingVoltage()* = 0<br>*ChargingCurrent()* = 0 |

# *Communications*

## 10.1 HDQ Single-Pin Serial Interface

The HDQ interface is an asynchronous return-to-one protocol where a processor sends the command code to the fuel gauge. With HDQ, the least significant bit (LSB) of a data byte (command) or word (data) is transmitted first.

> **NOTE:** The HDQ pin (A3) is open-drain and requires an external pullup resistor.

The 8-bit command code consists of two fields: the 7-bit HDQ command code (bits 0:6) and the 1-bit RW field (MSB bit 7). The RW field directs the fuel gauge either to:

• Store the next 8 or 16 bits of data to the specified register or
• Output 8 bits of data from the specified register.

The HDQ peripheral can transmit and receive data as either an HDQ master or slave.

The host processor initiates HDQ serial communication by sending a break command to the fuel gauge. A break is detected when the DATA pin is driven to a logic-low state for a time $t_{(B)}$ or greater. The DATA pin should then be returned to its normal ready high logic state for a time $t_{(BR)}$. The fuel gauge is now ready to receive information from the host processor.

The *HDQ Communication Basics Application Report* (SLUA408) provides details of HDQ communication basics, including an alternative method to use a standard two-wire UART for single-wire HDQ communication.

The BQ27Z561-R2 gauge is shipped in I²C mode, but TI provides a method to switch to HDQ mode via *SwitchToHDQ()* (see *AltManufacturerAccess() 0x7C40* for details).

## 10.2 I²C Interface

The gauge supports the standard I²C 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.
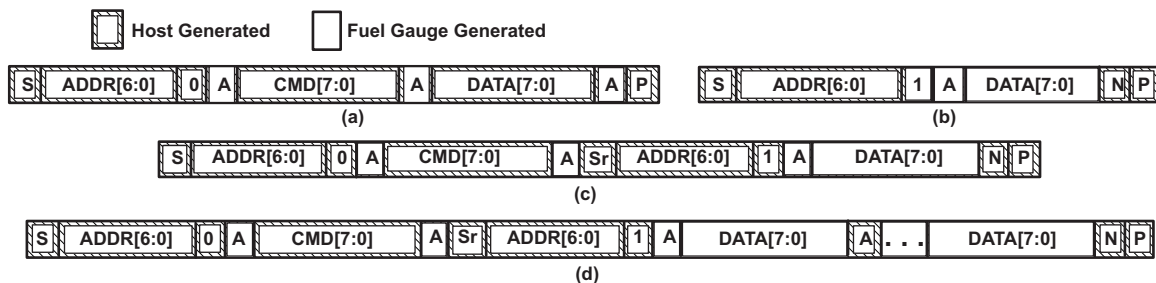


**Figure 10-1. Supported I²C Formats**

(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 I²C communication engine, increments when data is acknowledged by the fuel gauge or the I²C master. Quick writes function in the same way 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):

| S | ADDR[6:0] | 0 | A | CMD[7:0] | A | DATA[7:0] | A | P |

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

| S | ADDR[6:0] | 0 | A | CMD[7:0] | N | P |

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

| S | ADDR[6:0] | 0 | A | CMD[7:0] | A | DATA[7:0] | A | DATA[7:0] | N | . . . | N | P |

Incremental read at the maximum allowed read address:

| S | ADDR[6:0] | 0 | A | CMD[7:0] | A | Sr | ADDR[6:0] | 1 | A | DATA[7:0] | A | . . . | DATA[7:0] | N | P |

Address 0x7F — Data From addr 0x7F — Data From addr 0x00
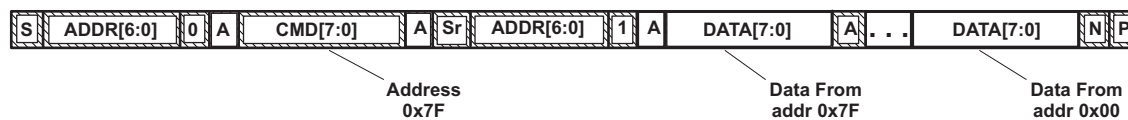
## 10.2.1 *I²C Clock Frequency*

The gauge supports different maximum clock frequencies for the I²C engine. The desired maximum clock frequency can be configured via **Settings.Configuration.I2C Configuration**.

## 10.2.2 *I²C Time Out*

The I²C engine releases SDA and SCL if the I²C bus is held low for about two seconds. If the fuel gauge were holding the lines, releasing them frees the master to drive the lines.

## 10.2.3 *I²C Command Waiting Time*

To ensure the correct results of a command with the 400-kHz I²C operation, there must be a proper waiting time between issuing the command and reading the results. For *AltManufacturerAccess()* commands, the following diagram shows the waiting time required between issuing the *AltManufacturerAccess()* command and reading the status. 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. If thousands of I²C transactions are sent to the gauge in one second, then it could adversely impact the CPU and cause a watchdog reset.

| S | ADDR[6:0] | 0 | A | CMD[7:0] | A | DATA[7:0] | A | DATA[7:0] | A | P | $t_{BUF}$ |
| S | ADDR[6:0] | 0 | A | CMD[7:0] | A | Sr | ADDR[6:0] | 1 | A | DATA[7:0] | A | DATA[7:0] | N | P | 66 μs |
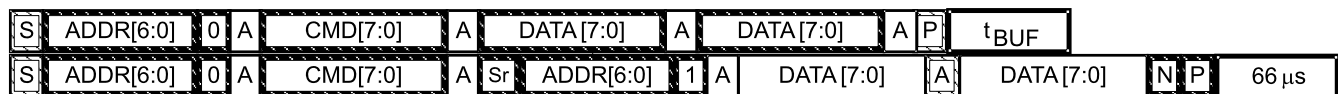
Waiting time between control subcommand and reading results

| S | ADDR[6:0] | 0 | A | CMD[7:0] | A | Sr | ADDR[6:0] | 1 | A | DATA[7:0] | A | DATA[7:0] | A |
| DATA[7:0] | A | DATA[7:0] | N | P | 66 μs |

Waiting time between continuous reading results

**Table 10-1. Command Waiting Times**

| Command | $t_{BUF}$ |
|---|---|
| Data flash signature commands: *StaticDFSignature()*, *StaticChemDFSignature()*, or *AllDFSignature()* | 250 ms |
| All other writes to *AltManufacturerAccess()* | 66 µs |

### 10.2.4 $I^2C$ Clock Stretching

$I^2C$ clock stretches can occur during all modes of fuel gauge operation. In 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/NACK 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 is written by the CPU to update the Ra tables and other data flash parameters, such as QMax. Due to the organization of data flash, updates need to be written in data blocks consisting of multiple data bytes.

For example, an Ra table update requires erasing a single page of data flash and programming the updated Ra table. The potential $I^2C$ clock stretching time is 40.08 ms maximum. This includes a 40-ms page erase and 40-µs 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.

# Manufacture Production

## 11.1 Manufacture Testing

To improve the manufacture testing flow, the gas gauge device allows certain features to be toggled on or off through *AltManufacturerAccess()* commands (for example, *LifetimeDataCollection()*, *CalibrationMode()*, and *Gauging()*). Enabling only the feature under test can simplify the test flow in production by avoiding any feature interference. The *AltManufacturerAccess()* command that toggles *ManufacturingStatus()[CAL_EN]*, *[LT_TEST]* will only set the RAM data, meaning the conditions set by this command will be cleared if a reset or seal is issued to the gauge. The *AltManufacturerAccess()* commands that toggle *ManufacturingStatus()[LF_EN], [GAUGE_EN]* will be updated to data flash and synchronized between *ManufacturingStatus()* and **Mfg Status Init**. *ManufacturingStatus()* keeps track of the status (enabled or disabled) of each feature.

**Mfg Status Init** provides the option to enable or disable individual features for normal operation. Upon a reset or a seal command, *ManufacturingStatus()* will be reloaded from **Mfg Status Init**. This means if an update is made to *Mfg Status Init* to enable or disable a feature, the gauge will only take the new setting if a reset or seal command is sent.

### 11.1.1 Manufacturing Status Configuration

| Class | Subclass | Name | Type | Min | Max | Default | Description |
|-------|----------|------|------|-----|-----|---------|-------------|
| Settings | Manufacturing | Mfg Status Init | H2 | 0x0000 | 0xFFFF | 0x0000 | Bit 0: Reserved<br>Bit 1: Reserved<br>Bit 2: Reserved<br>Bit 3: GAUGE_EN—Gauging<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 4: Reserved<br>Bit 5: LF_EN—Lifetime data collection<br>0 = Disabled (default)<br>1 = Enabled<br>Bit 15:6: Reserved |

## 11.2 I²C Configuration

The following table describes the bit settings for I²C communication options.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Settings | Configuration | I2C Configuration | H1 | 0x00 | 0xFF | 0x08 | — | Bit 0: Reserved<br>Bit 1: MFG_C_SEALED—Set to allow *AltManufacturerAccess() 0x007A ManufacturerInfoB* to write data flash when the gauge is sealed.<br>Bit 2: Reserved<br>Bit 3: XL—Enables 400-kHz I²C max clock frequency<br>0 = 100-kHz max clock frequency<br>1 = 400-kHz max clock frequency (default)<br>Bits 6–4: Reserved<br>Bit 7: FLASH_BUSY_WAIT—Set this bit to avoid NACKs on legal commands. Note there will be long clock stretches if the gauge is updating data flash. |

## 11.3 Manufacturer Information Blocks

There are three 32-byte manufacturer information blocks in data flash that record details related to the manufacture of a battery pack. The gauge does not use this data; it is available for the manufacturer to use and interpret the data. No formatting of data written in these blocks is imposed by the gauge firmware. These blocks can be accessed by either the regular data flash access methods or by using *ManufacturerAccess()* commands. See the entries for **Manufacturer Data**, **Manufacturer Info B**, and **Manufacturer Info C** in the Section 15.1 to *AltManufacturerAccess() 0x007A ManufacturerInfoB* access by Section 13.8.53.

The manufacturer information blocks can also be read or written when the gauge is in the UNSEALED mode, using the associated MACs *AltManufacturerAccess() 0x0070 ManufacturerInfo* and *AltManufacturerAccess() 0x007B ManufacturerInfoC*. *AltManufacturerAccess() 0x007B ManufacturerInfoC* can be used to write to manufacturer information block c when the gauge is sealed if **[MFG_C_SEALED]** is set in **Settings.Configuration.I²C Configuration**.

## 11.4 Verification of Data Flash

During manufacture, the configuration of the gauge in data flash will be updated. Typically, a known good data flash image is loaded to the device. To avoid the time needed to read back to verify the data flash, a number of data flash signature Manufacturer Access commands are available (see Section 13.8). The BQ27Z561-R2 has both static and non-static sections in the data flash memory. The static memory areas are only changed by host writes to data flash commands. Non-static memory may be modified by the gauge firmware.

## 11.5 Calibration

The device has integrated routines that support calibration of current and voltage readings, accessible after writing 0xF081 to *AltManufacturerAccess()* when the *ManufacturingStatus()[CAL_EN]* bit is ON. While the calibration is active, the raw ADC data is available on *MACData()*. The device stops reporting calibration data on *MACData()* if any other MAC commands are sent or the device is reset or sealed.

---

**NOTE:** The *ManufacturingStatus()[CAL_EN]* bit must be turned OFF after calibration is completed. The *ManufacturingStatus()[CAL_EN]* bit is set by default when **Mfg Status Init** is set to 0. This bit is cleared at reset or after sealing.

---

**Table 11-1. AltManufacturerAccess() Descriptions**

| AltManufacturerAccess() | Description |
|---|---|
| 0x002D | Enables/disables *ManufacturingStatus()[CAL_EN]* |
| 0xF080 | Disables raw ADC data output on *MACData()* |
| 0xF081 | Outputs raw ADC data of voltage and current *MACData()* |

The *MACData()* output format is: ZZYYaaAAbbBBccCCddDDeeEEffFFggGGhhHHiiIIjjJJkkKK,

where:

**Table 11-2. MACData() Format Descriptions**

| Value | Format | Description |
|---|---|---|
| ZZ | byte | 8-bit counter, increments when raw ADC values are refreshed (every 1000 ms). |
| YY | byte | Output status *AltManufacturerAccess()* = 0xF081: 1 |
| AAaa | 2's comp | Current (coulomb counter) |
| BBbb | 2's comp | Cell Voltage |
| CCcc | 2's comp | Reserved |
| DDdd | 2's comp | Reserved |
| EEee | 2's comp | Reserved |

**Table 11-2. MACData() Format Descriptions (continued)**

| Value | Format | Description |
|-------|--------|-------------|
| FFff | 2's comp | Reserved |
| GGgg | 2's comp | Reserved |
| HHhh | 2's comp | Cell Current |
| IIii | 2's comp | Reserved |
| JJjj | 2's comp | Reserved |
| KKkk | 2's comp | Reserved |

# *Calibration*

---

> **NOTE:** All calibration steps that require accessing data on *MACData()* must begin read operations starting at *AltManufacturerAccess()* to ensure the data portion of the block is properly refreshed per the intended *AltManufacturerAccess()* subcommand (in this case 0xF081). The first two bytes returned are the *AltManufacturerAccess()* subcommand followed by the counter, status, and raw ADC values, as shown in Table 11-1.

## 12.1 Cell (BAT) Voltage Calibration
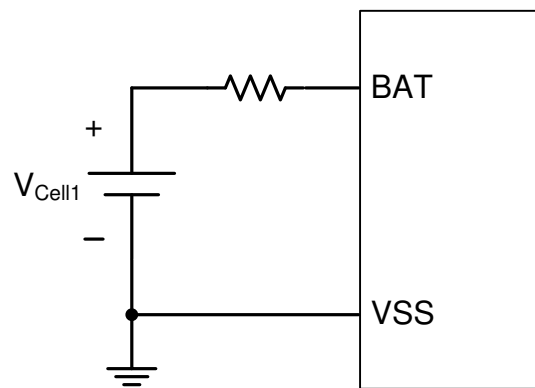


**Figure 12-1. Cell Voltage Calibration**

1. Apply known voltage in mV to the cell voltage inputs:
   - $V_{CELL1}$ between BAT terminal and VSS terminal
2. If *ManufacturingStatus()[CAL_EN]* = 0, send 0x002D to *AltManufacturerAccess()* to enable the *[CAL_EN]* flag.
3. Send 0xF081 to *AltManufacturerAccess()* to enable raw cell voltage output on *MACData()*.
4. Poll *MACData()* until the 8-bit counter value increments by 2 before reading data.
5. Read the ADC conversion readings of cell voltages from *MACData()*:
   - $ADC_{CELL1}$ = BBbb of *MACData()*
6. Average several readings for higher accuracy. Poll *MACData()* until ZZ increments, to indicate that updated values are available:
   - $ADC_{CELL1}$ = [$ADC_{CELL1}$(reading n) + ...+ $ADC_{CELL1}$(reading 1)]/n

---

7. Calculate gain value:

$$Cell\ Gain = \frac{V_{Cell1}}{ADC_{Cell1}}\ x\ 2^{16}$$

8. Write the new **Cell Gain** value to data flash.
9. Recheck the voltage reading and if it is not accurate, repeat Steps 5 and 6.
10. Send 0x002D to *AltManufacturerAccess()* to clear the *[CAL_EN]* flag if all calibration is complete.

### 12.1.1 *Voltage Calibration Data Flash*

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Calibration | Voltage | Cell Gain | I2 | −32768 | 32767 | 12101 | — | BAT – VSS gain |

## 12.2 **Current Calibration**
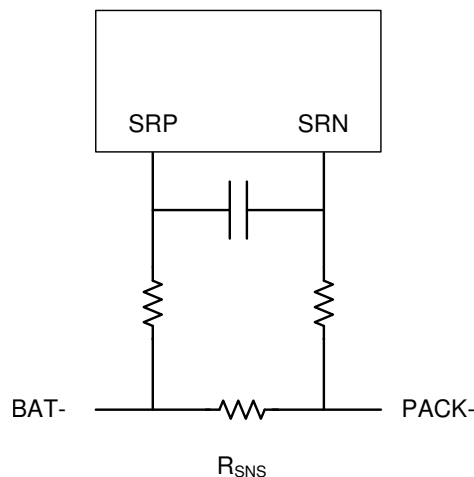


**Figure 12-2. Current Calibration (Charge Current Flow Is BAT– to PACK–)**

> **NOTE:** CC offset and board offset are no longer calibrations required by the BQ27Z561-R2 due to hardware improvements.

### 12.2.1 *CC Gain/Capacity Gain Calibration*

1. Apply a known current (typically 1 A to 2 A), and ensure $I_{CC}$ is flowing through the sense resistor connected between the SRP and SRN pins.
2. If *ManufacturingStatus()[CAL_EN]* = 0, send 0x002D to *AltManufacturerAccess()* to enable the *[CAL_EN]* flag.
3. Send 0xF081 to *AltManufacturerAccess()* to enable raw CC output on *MACData()*.
4. Poll *MACData()* until ZZ increments by 2 before reading data.
5. Check the ADC conversion current output from *MACData()*:
   - $ADC_{CC}$ = AAaa of *MACData()*

     Is $ADC_{CC}$ < 0x8000? If yes, use $ADC_{CC}$; otherwise, $ADC_{CC}$ = −(0xFFFF − AAaa + 0x0001).
6. Average several readings for higher accuracy. Poll *MACData()* until ZZ increments to indicate that updated values are available:
   - $ADC_{CC}$ = [$ADC_{CC}$(reading n) + … + $ADC_{CC}$(reading 1)]/n

7. Calculate gain values:

$$CC\ Gain = I_{CC}\ /\ ADC_{CC}$$
$$Capacity\ Gain = CC\ Gain \times 1193046.4712$$

8. Write the new **CC Gain** and **Capacity Gain** values to data flash.

9. Recheck the current reading. If the reading is not accurate, repeat the steps.

10. Send 0x002D to *AltManufacturerAccess()* to clear the *[CAL_EN]* flag if all calibration is complete.

### 12.2.2 Deadbands

The gauge can be configured to ignore current and coulomb measurements below individually programmable levels.

#### 12.2.2.1 Current Deadband

When current measures to a value less than the value programmed in **Deadband**, *Current()* will report 0. This has no effect on the coulomb counting for the gas gauging functionality. The value of **Deadband** should be selected based on the characterization of the battery electronics design combined with the environment in which the battery will be used. If the PCB senses noise causing a real no-current condition to report a non-zero value, then **Deadband** could be adjusted accordingly.

#### 12.2.2.2 Coulomb Counter Deadband

During normal operation, there could be noise generated in the battery electronics environment that could cause the gauge to accumulate incorrectly (positively or negatively). To filter out this noise, the **Coulomb Counter Deadband** setting is used. Any input below this threshold is not accumulated.

### 12.2.3 Current Calibration Data Flash

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Calibration | Current | CC Gain | F4 | 1.00E−001 | 4.00E+000 | 3.68 | — | Coulomb Counter Gain |
| Calibration | Current | Capacity Gain | F4 | 2.98262E+004 | 1.193046E+006 | 4390411.01 | — | Capacity Gain |
| Calibration | Current Deadband | Deadband | U1 | 0 | 255 | 3 | mA | Cell-based deadband to report 0 mA |
| Calibration | Current Deadband | Coulomb Counter Deadband | U1 | 0 | 255 | 9 | 116 nV | Coulomb counter deadband to report 0 charge (This setting should not be modified.) |

## 12.3 Temperature Calibration

**NOTE:** For temperature calibration, only an offset is determined and then applied to the gauge's measured temperatures.
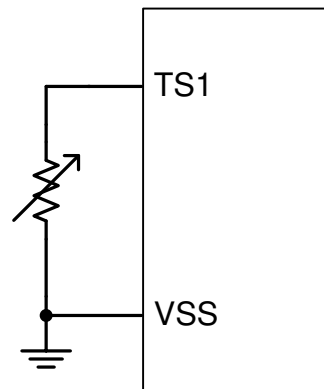
**Figure 12-3. Temperature Calibration**

### 12.3.1 *Internal Temperature Sensor Calibration*

1. Apply a known temperature in 0.1°C, and ensure that temperature Temp$_{TINT}$ is applied to the device.
2. Read the TINT offset$_{old}$ from **Internal Temp Offset**.
3. Read the reported temperature from *DAStatus2()*:
   - TINT = AAaa of *DAStatus2()*

   Is TINT > 0? If yes, TINT = AAaa − 2732.
4. Calculate temperature offset:

$$TINT\, offset = TEMP_{TINT} - TINT + TINT\, offset_{old}$$

5. Write the new **Internal Temp Offset** value to data flash.
6. Recheck the *DAStatus2()* reading. If the reading is not accurate, repeat the steps.

### 12.3.2 *TS Calibration*

1. Apply a known temperature in 0.1°C, and ensure the temperature TEMP$_{TS}$ is applied to the thermistor connected to the TS terminal.
2. Read the TS offset$_{old}$ from **External1 Temp Offset**.
3. Read the appropriate temperature from the *DAStatus2()* block as TS.
4. Calculate the temperature offset:

$$TS\, offset = \ TEMP_{TS} - TS + TS\, offset_{old}$$

5. Write the new **External1 Temp Offset** value to data flash.
6. Recheck the *DAStatus2()* reading. If the reading is not accurate, repeat the steps.

### 12.3.3 *Temperature Calibration Data Flash*

It is usually not necessary to adjust these offsets. These offsets can be used to improve temperature accuracy at temperatures away from the calibration temperature of the models below; owever, they can reduce accuracy across the range of temperatures.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Calibration | Temperature | Internal Temp Offset | I1 | −128 | 127 | 0 | 0.1 °C | Internal temperature sensor reading offset |
| Calibration | Temperature | External1 Temp Offset | I1 | −128 | 127 | 0 | 0.1 °C | TS pin temperature sensor reading offset |

### 12.3.4 Cell Temp Model

The parameters in the following table are used for the Semitec 103AT-2 thermistor. These parameters will only need to be updated if a different thermistor is used. See the application report to calculate coefficients for other thermistors.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Calibration | Cell Temp Model | Coefficient a1 | I2 | −32768 | 32767 | −17447 | — | Cell Temperature calculation polynomial a1 |
| Calibration | Cell Temp Model | Coefficient a2 | I2 | −32768 | 32767 | 29332 | — | Cell Temperature calculation polynomial a2 |
| Calibration | Cell Temp Model | Coefficient a3 | I2 | −32768 | 32767 | −25430 | — | Cell Temperature calculation polynomial a3 |
| Calibration | Cell Temp Model | Coefficient a4 | I2 | −32768 | 32767 | 29836 | — | Cell Temperature calculation polynomial a4 |
| Calibration | Cell Temp Model | Coefficient a5 | I2 | −32768 | 32767 | 1200 | — | Cell Temperature calculation polynomial a5 |
| Calibration | Cell Temp Model | Coefficient b1 | I2 | −32768 | 32767 | −293 | — | Cell Temperature calculation polynomial b1 |
| Calibration | Cell Temp Model | Coefficient b2 | I2 | −32768 | 32767 | 552 | — | Cell Temperature calculation polynomial b2 |
| Calibration | Cell Temp Model | Coefficient b3 | I2 | −32768 | 32767 | −2887 | — | Cell Temperature calculation polynomial b3 |
| Calibration | Cell Temp Model | Coefficient b4 | I2 | −32768 | 32767 | 4591 | — | Cell Temperature calculation polynomial b4 |
| Calibration | Cell Temp Model | Rc0 | I2 | −32768 | 32767 | 11703 | — | Resistance at 25°C |
| Calibration | Cell Temp Model | Adc0 | I2 | −32768 | 32767 | 11703 | — | ADC reading at 25°C |
| Calibration | Cell Temp Model | Rpad | I2 | −32768 | 32767 | 0[1] | Ω | Pad Resistance (0 to use factory calibration) |
| Calibration | Cell Temp Model | Rint | I2 | −32768 | 32767 | 0[1] | Ω | Pullup resistor resistance (0 to use factory calibration) |

[1] Setting this value to 0 causes the gauge to use the internal factory calibration default.

### 12.3.5 Internal Temp Model

Values in this table should not be modified, as the voltage-based sensor in the device was factory trimmed for these model parameters.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| Calibration | Internal Temp Model | Int Gain | I2 | −32768 | 32767 | −13908 | — | Internal temperature gain |
| Calibration | Internal Temp Model | Int Base Offset | I2 | −32768 | 32767 | 6959 | — | Internal temperature base offset |
| Calibration | Internal Temp Model | Int Minimum AD | I2 | −32768 | 32767 | 0 | — | Minimum AD count used for calculation |
| Calibration | Internal Temp Model | Int Maximum Temp | I2 | −32768 | 32767 | 6959 | 0.1 K | Maximum Temperature boundary |

# Data Commands

## 13.1 Standard Data Commands

The BQ27Z561-R2 device 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 13-1, *Standard Commands*. The LSB of the command code corresponds to the LSB register data. The LSB should be read first to keep the register data synchronized. For some host systems, the host should change the byte order so that the order corresponds to the host's native endianness. Both the I$^2$C and HDQ protocols have methods to access the data for each command code. The gauge updates and/or reads the standard command registers once per second.

**Table 13-1. Standard Commands**

| Name | | Register Code (LSB/MSB) | Description |
|---|---|---|---|
| *ManufacturerAccess/ControlStatus()* | CNTL | 0x00/0x01 | Control Register (See Section 13.1.1.) |
| *AtRate()* | AR | 0x02/0x03 | Read/write. The value is a signed integer with the negative value indicating a discharge current value. The default value is 0 and forces *AtRateTimeToEmpty()* to return 65,535. |
| *AtRateTimeToEmpty()* | ARTTE | 0x04/0x05 | This read-only function returns an unsigned integer value to predict remaining operating time based on battery discharge at the *AtRate()* value in minutes with a range of 0 to 65,534. A value of 65,535 indicates *AtRate()* = 0. The gas gauge updates the *AtRateTimeToEmpty()* within 1 s after the system sets the *AtRate()* value. The gas gauge updates these parameters every 1 s. The commands are used in NORMAL mode. |
| *Temperature()* | TEMP | 0x06/0x07 | This read-only function returns an unsigned integer value of temperature measured by the gas gauge and is used for the gauging algorithm. Values are reported in units 0.1 K. It reports either internal temperature or external thermistor temperature, depending on the setting of the **[TS1]** and **[TSInt]** bits in **Temperature Enable**. |
| *Voltage()* | VOLT | 0x08/0x09 | This read-only function returns an unsigned integer value of the measured cell pack in mV with a range of 0 to 6000 mV. |
| *BatteryStatus()* | FLAGS | 0x0A/0x0B | This read-only function returns various battery status information. |
| *Current()* | INSTCURR | 0x0C/0x0D | This read-only function returns a signed integer value that is the instantaneous current flow through the sense resistor. The value is updated every 1 s. Units are mA. |
| *RemainingCapacity()* | RM | 0x10/0x11 | This read-only command returns the predicted remaining capacity, based on rate (per configured **Load Select**), temperature, present depth-of-discharge, and stored impedance. Values are reported in mAh. |
| *FullChargeCapacity()* | FCC | 0x12/0x13 | This read-only command returns the predicted capacity of the battery at full charge, based on rate (per configured **Load Select**), temperature, present depth-of-discharge, and stored impedance. Values are reported in mAh. |
| *AverageCurrent()* | AI | 0x14/0x15 | This read-only function returns a signed integer value that is the average current flow through the sense resistor. The value is updated every 1 s. Units are mA. |
| *AverageTimeToEmpty()* | TTE | 0x16/0x17 | Uses average current value with a time constant of 15 s for this method. A value of 65,535 means the battery is not being discharged. |
| *AverageTimeToFull()* | TTF | 0x18/0x19 | This read-only function returns a unsigned integer value, predicting time to reach full charge for the battery in units of minutes based on *AverageCurrent()*. The computation accounts for the taper current time extension from a linear TTF computation based on a fixed *AverageCurrent()* rate of charge accumulation. A value of 65,535 indicates the battery is not being charged. |
| *MaxLoadCurrent()* | MLI | 0x1E/0x1F | This read-only function returns a signed integer value in units of mA of the maximum load current. The *MaxLoadCurrent()* is an adaptive measurement which is initially reported as the maximum load current programmed in **Max Load Current**. If the measured current is ever greater than **Max Load Current**, then the *MaxLoadCurrent()* updates to the new current. *MaxLoadCurrent()* is reduced to the average of the previous value and **Max Load Current** whenever the battery is charged to full after a previous discharge to an RSOC of less than **Max Load Rsoc**. This will prevent the reported value from maintaining an unusually high value. |
| *MaxLoadTimeToEmpty()* | MLTTE | 0x20/0x21 | This read-only function returns a unsigned integer value, predicting remaining battery life at the maximum discharge load current rate in units of minutes. A value of 65,535 indicates that the battery is not being discharged. |
| *AveragePower()* | AP | 0x22/0x23 | This read-only function returns a signed integer value of average power during battery charging and discharging. 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. |
| *BTPDischargeSet()* | BTP | 0x24/0x25 | This read-/write-word command updates the BTP set threshold for DISCHARGE mode for the next BTP interrupt, deasserts the present BTP interrupt, and clears the *OperationStatus()[BTP_INT]* bit. |
| *BTPChargeSet()* | BTP | 0x26/0x27 | The read-/write-word command updates the BTP set threshold for CHARGE mode for the next BTP interrupt, deasserts the present BTP interrupt, and clears the *OperationStatus()[BTP_INT]* bit. |

## Table 13-1. Standard Commands (continued)

| Name | | Register Code (LSB/MSB) | Description |
|---|---|---|---|
| *InternalTemperature()* | INT_TEMP | 0x28/0x29 | This read-only function returns an unsigned integer value of the measured internal temperature of the device in 0.1 K units measured by the gas gauge. |
| *CycleCount()* | CC | 0x2A/0x2B | This read-only function returns an unsigned integer value of the number of cycles the battery has experienced a discharge (range 0 to 65,535). One cycle occurs when accumulated discharge greater than or equal to the cycle count threshold. |
| *RelativeStateOfCharge()* | RSOC | 0x2C/0x2D | This read-only function returns an unsigned integer value of the predicted remaining battery capacity expressed as percentage of *FullChargeCapacity()* with a range of 0% to 100%. |
| *StateOfHealth()* | SOH | 0x2E/0x2F | This read-only function returns an unsigned integer value of the battery state-of-health expressed as a percentage of the ratio of predicted FCC (simulated with **SOH Load Rate** at 25°C) over the *DesignCapacity()*. The range is 0% to 100%. |
| *ChargingVoltage()* | CV | 0x30/0x31 | Returns the desired charging voltage in mV to the charger |
| *ChargingCurrent()* | CC | 0x32/0x33 | Returns the desired charging current in mA to the charger |
| *TerminateVoltage()* | | 0x34/0x35 | Returns or updates the value of **Terminate Voltage** |
| *TimeStampUpper()* | | 0x36/0x37 | Returns or updates the upper 16 bits of the time stamp in seconds |
| *TimeStampLower()* | | 0x38/0x39 | Returns or updates the lower 16 bits of the time stamp in seconds |
| *QmaxCycles()* | | 0x3A/0x3B | Returns cycle count at the last QMax update |
| *DesignCapacity()* | | 0x3C/0x3D | In SEALED and UNSEALED access: This command returns the value stored in **Design Capacity mAh**. This is intended to be a theoretical or nominal capacity of a new pack, but should have no bearing on the operation of the gas gauge functionality. |
| *AltManufacturerAccess()* | | 0x3E/0x3F | MAC data block command |
| *MACData()* | | 0x40/0x5F | MAC data block |
| *MACDataSum()* | | 0x60 | MAC data block checksum |
| *MACDataLen()* | | 0x61 | MAC data block length |
| *VoltHiSetThreshold()* | | 0x62/0x63 | This read/write function is a signed integer in units of mV for the high voltage level threshold, which sets *InterruptStatus()[VOLT_HI]*. |
| *VoltHiClearThreshold()* | | 0x64/0x65 | This read/write function is a signed integer in units of mV for the high voltage level threshold, which clears *InterruptStatus()[VOLT_HI]*. |
| *VoltLoSetThreshold()* | | 0x66/0x67 | This read/write function is a signed integer in units of mV for the low voltage level threshold, which sets *InterruptStatus()[VOLT_LO]*. |
| *VoltLoClearThreshold()* | | 0x68/0x69 | This read/write function is a signed integer in units of mV for the low voltage level threshold, which clears *InterruptStatus()[VOLT_LO]*. |
| *TempHiSetThreshold()* | | 0x6A | This read/write function is a signed integer in units of °C for the high temperature level threshold, which sets *InterruptStatus()[TEMP_HI]*. |
| *TempHiClearThreshold()* | | 0x6B | This read/write function is a signed integer in units of °C for the high temperature level threshold, which clears *InterruptStatus()[TEMP_HI]*. |
| *TempLoSetThreshold()* | | 0x6C | This read/write function is a signed integer in units of °C for the low temperature level threshold, which sets *InterruptStatus()[TEMP_LO]*. |
| *TempLoClearThreshold()* | | 0x6D | This read/write function is a signed integer in units of °C for the low temperature level threshold, which clears *InterruptStatus()[TEMP_LO]*. |
| *InterruptStatus()* | | 0x6E | This read-only function returns the status of the interrupt including cause of interrupt. |
| *SOCDeltaSetThreshold()* | | 0x6F | This read/write function is an unsigned char in units of % for the SOC delta level threshold, which clears *InterruptStatus()SOC_DELTA*. |

### 13.1.1  0x00/01 ManufacturerAccess()/ControlStatus()

This read/write word function returns the control bits when read and is an interface to the manufacturer access system (MAC) when written. This command is provided for backward compatibility/ease of use, as all of the control bits except *ControlStatus()[CheckSumValid]* are available in *OperationStatusA()* and *OperationStatusB()* in addition to *AltManufacturerAccess()* being the recommended interface for MAC (as discussed in Section 13.8).

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | FAS | SS | RSVD | RSVD | RSVD | Check Sum Valid | RSVD | RSVD | RSVD | RSVD | RSVD | LDMD | RDIS | VOK | QEN |

**RSVD (Bit 15)**: Reserved

**FAS, SS (Bits 14–13)**: Legacy SECURITY mode

           0, 0 =  Full Access

           0, 1 =  Reserved

1, 0 = Unsealed

1, 1 = Sealed

**RSVD (Bit 12)**: Reserved

**RSVD (Bits 11–10)**: Reserved

**CheckSumValid (Bit 9)**: Checksum valid

1 = Flash Writes are enabled.

0 = Flash Writes are disabled due to low voltage or PF condition.

**RSVD (Bits 8–4)**: Reserved

**LDMD (Bit 3)**: LOAD mode

1 = Constant power

0 = Constant current

**RDIS (Bit 2)**: Resistance updates

1 = Disabled

0 = Enabled

**VOK (Bit 1)**: Voltage OK for QMax update

1 = Detected

0 = Not detected

**QEN (Bit 0)**: Impedance Track Gauging (Ra and QMax updates are enabled.)

### 13.1.2 0x02/03 AtRate()

This read/write word function sets the value used in calculating *AtRateTimeToEmpty().*

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Default | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | | |
| 0x02/03 | *AtRate()* | | R/W | | Word | I2 | –32768 | 32767 | 0 | mA |

### 13.1.3 0x04/05 AtRateTimeToEmpty()

This read-word function returns the remaining time to fully discharge the battery based on *AtRate()*.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | | |
| 0x04/05 | *AtRateTimeToEmpty()* | | R | | Word | U2 | 0 | 65535 | min | 65535 indicates not being charged |

### 13.1.4 0x06/07 Temperature()

This read-word function returns the temperature in units 0.1 K.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | |
| 0x06/07 | *Temperature()* | | R | | Word | I2 | 0 | 32767 | 0.1 K |

### 13.1.5 0x08/09 Voltage()

This read-word function returns the measured cell voltage.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | |
| 0x08/09 | *Voltage()* | | R | | Word | I2 | 0 | 32767 | mV |

### 13.1.6 *0x0A/0B BatteryStatus()*

This read-word function returns various battery status information.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| RSVD | TCA | RSVD | RSVD | TDA | RSVD | RCA | RSVD | INIT | DSG | FC | FD | RSVD | RSVD | RSVD | RSVD |

**RSVD (Bit 15)**: Reserved

**TCA (Bit 14)**: Terminate Charge Alarm

  0 =  Inactive

  1 =  Active

**RSVD (Bits 13–12)**: Reserved

**TDA (Bit 11)**: Terminate Discharge Alarm

  0 =  Inactive

  1 =  Active

**RSVD (Bit 10)**: Reserved

**RCA (Bit 9)**: Remaining Capacity Alarm

  0 =  Inactive

  1 =  Active

**RSVD (Bit 8)**: Reserved

**INIT (Bit 7)**: Initialization

  0 =  Complete

  1 =  Active

**DSG (Bit 6)**: Discharging

  0 =  The battery is charging.

  1 =  The battery is discharging.

**FC (Bit 5)**: Fully Charged

  0 =  The battery is not fully charged.

  1 =  The battery is fully charged.

**FD (Bit 4)**: Fully Discharged

  0 =  The battery is okay.

  1 =  The battery is fully depleted.

**RSVD (Bits 3–0)**: Reserved

### 13.1.7 *0x0C/0D Current()*

This read-word function returns the measured current from the coulomb counter.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|------|------|-----------|------|------|------|------|
| | | SE | US | FA | | | | | |
| 0x0C/0D | *Current()* | | R | | Word | I2 | −32768 | 32767 | mA |

### 13.1.8 *0x10/11 RemainingCapacity()*

This read-word function returns the predicted remaining battery capacity.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|------|------|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x10/11 | *RemainingCapacity()* | | R | | Word | U2 | 0 | 32767 | mAh |

### 13.1.9  0x12/13 FullChargeCapacity()

This read-word function returns the predicted battery capacity when fully charged.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|------|------|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x12/13 | *FullChargeCapacity()* | | R | | Word | I2 | 0 | 32767 | mAh |

### 13.1.10  0x14/15 AverageCurrent()

This read-word function provides the filtered/average current.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|------|------|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x14/15 | *AverageCurrent()* | | R | | Word | I2 | −32767 | 32768 | mA |

### 13.1.11  0x16/17 AverageTimeToEmpty()

This read-word function returns the predicted remaining time to fully discharge the battery based on *AverageCurrent()*.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---------|------|--------|------|------|-----------|------|-----|-----|------|------|
| | | SE | US | FA | | | | | | |
| 0x16/17 | *AverageTimeToEmpty()* | | R | | Word | U2 | 0 | 65535 | min | 65535 = The battery is not being discharged. |

### 13.1.12  0x18/19 AverageTimeToFull()

This read-word function returns the predicted remaining time to achieve full charge based on *AverageCurrent()*.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---------|------|--------|------|------|-----------|------|-----|-----|------|------|
| | | SE | US | FA | | | | | | |
| 0x18/19 | *AverageTimeToFull()* | | R | | Word | U2 | 0 | 65535 | min | |

### 13.1.13  0x1E/1F MaxLoadCurrent()

This read-word function returns the maximum load current.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|------|------|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x1E/1F | *MaxLoadCurrent()* | | R | | Word | I2 | 0 | 65535 | mA |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | Max Load | Max Load Current | I2 | −32768 | 0 | −500 | mA | Initial value reported on *MaxLoadCurrent()* |
| Gas Gauging | Max Load | Max Load Rsoc | U1 | 0 | 100 | 50 | % | Threshold that RSOC must fall below to average *MaxLoadCurrent()* with **Max Load Current** on the next valid charge termination |

### 13.1.14  0x20/21 MaxLoadTimeToEmpty()

This read-word function returns the remaining time to fully discharge the battery based on *MaxLoadCurrent()*.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|---|---|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x20/21 | *MaxLoadTimeToEmpty()* | | R | | Word | I2 | 0 | 65535 | min |

### 13.1.15  0x22/23 AveragePower()

This read-word function returns the average power [*Voltage()* × *AverageCurrent()*] during battery charging or discharging. It is negative due to discharge and positive due to charge. A 0 value indicates the battery is not being discharged.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|---|---|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x22/23 | *AveragePower()* | | R | | Word | I2 | −32768 | 32767 | mW |

### 13.1.16  0x28/29 InternalTemperature()

This read-word function returns the internal die temperature in units 0.1 K.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|---|---|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x28/29 | *InternalTemperature()* | | R | | Word | I2 | 0 | 32767 | 0.1 K |

### 13.1.17  0x2A/2B CycleCount()

This read-word function returns the number of discharge cycles the battery has experienced.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|---|---|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x2A/2B | *CycleCount()* | R | R | | Word | U2 | 0 | 65535 | cycles |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Gas Gauging | Cycle | Cycle Count Percentage | U1 | 0 | 100 | 90 | % | Percentage of *DesignCapacity()* or *FullChargeCapacity()* (determined by **IT Gauging Configuration[CCT]**) to be used for the cycle count threshold |
| Gas Gauging | State | Cycle Count | U2 | 0 | 65535 | 0 | — | Value reported by *CycleCount()*. Updated by the gauge automatically based on **Cycle Count Percentage** |
| Gas Gauging | State | QMax Cycle Count | U2 | 0 | 65535 | 0 | — | The *CycleCount()* when QMax last updated |

### 13.1.18  0x2C/2D RelativeStateOfCharge()—RSOC

This read-word function returns the predicted remaining battery capacity as a percentage of *FullChargeCapacity()*.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|---|---|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x2C/2D | *RelativeStateOfCharge()* | | R | | Word | U1 | 0 | 100 | % |

### 13.1.19 0x2E/2F StateOfHealth()

This command returns the state-of-health (SOH) information of the battery in percentage of **Design Capacity mAh**. It is a read-only command.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|----|----|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x2E/2F | *StateOfHealth()* | | R | | Word | U1 | 0 | 100 | % |

### 13.1.20 0x30/31 ChargingVoltage()

This read-word function returns the desired charging voltage.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---------|------|--------|----|----|-----------|------|-----|-----|------|------|
| | | SE | US | FA | | | | | | |
| 0x30/31 | *ChargingVoltage()* | R | R | R | Word | I2 | 0 | 32767 | mV | 32767 = Request maximum voltage |

### 13.1.21 0x32/33 ChargingCurrent()

This read-word function returns the desired charging current.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---------|------|--------|----|----|-----------|------|-----|-----|------|------|
| | | SE | US | FA | | | | | | |
| 0x32/33 | *ChargingCurrent()* | R | R | R | Word | I2 | 0 | 32767 | mA | 32767 = Request maximum current |

### 13.1.22 0x34/35 TerminateVoltage()

This reads or writes the voltage level at which the state-of-charge goes to 0. See **Term Voltage** in Section 4.5.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---------|------|--------|----|----|-----------|------|-----|-----|------|------|
| | | SE | US | FA | | | | | | |
| 0x34/35 | *TerminateVoltage()* | RW | RW | RW | Word | I2 | 0 | 32767 | mV | |

### 13.1.23 0x36/0x37/0x38/0x39 Timestamp()

This 4-byte value, which is the system runtime in seconds, can be read or written. Always begin the read operation at 0x36 (the LSB) to keep the bytes synchronized during the read by copying the gauge clock to the time stamp registers. The host should read bytes into its 32-byte unsigned integer format in the appropriate endianness. When written by the host, the gauge will update its internal clock on the write to register 0x39.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---------|------|--------|----|----|-----------|------|-----|-----|------|------|
| | | SE | US | FA | | | | | | |
| 0x36/37 | *TimeStampUpper()* | R | RW | RW | Word | U2 | 0 | 65535 | s | |

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---------|------|--------|----|----|-----------|------|-----|-----|------|------|
| | | SE | US | FA | | | | | | |
| 0x38/39 | *TimeStampLower()* | R | RW | RW | Word | U2 | 0 | 65535 | s | |

### 13.1.24 QMax Cycles() 0x3A/3B

This returns the cycle count when QMax was last updated. It is helpful for a host system to compare this to the present *CycleCount()*.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit | Note |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | | |
| 0x3A/3B | QmaxCycles() | R | R | R | Word | U2 | 0 | 65535 | — | |

## 13.1.25  0x3C/3D DesignCapacity()

This read-word function returns the theoretical or nominal maximum pack capacity.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Default | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | | |
| 0x3C/3D | DesignCapacity() | R | R/W | R/W | Word | I2 | 0 | 32767 | 5300 | mAh |
| | | | | | | | | | 2040 | cWh |

## 13.1.26  0x3E/3F AltManufacturerAccess()

Writes to this command are interchangeable with *AltManufacturerAccess()*. This command is provided to enable an easy way to verify the active MAC command while reading the *MACData()* returned by the MAC. The host may simply read from *AltManufacturerAccess()* to *MACDataLength()* with one block read. For a description of returned data values, see the *AltManufacturerAccess()* version of same command in Section 13.8.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Default | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | | |
| 0x3E/3F | AltManufacturerAccess() | R | R | R | Word | — | — | — | — | — |

## 13.1.27  0x40/0x5F MACData()

This is the data block for *AltManufacturerAccess()* or *AltManufacturerAccess()* commands.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Default | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | | |
| 0x40/5F | MACData () | R | R | R | Block | — | — | — | — | — |

## 13.1.28  0x60 MACDataChecksum()

This is the checksum of the *AltManufacturerAccess()* and *MACData()* bytes.

| I²C Cmd | Name | Access | | | Protocol | Type | Min | Max | Default | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | | |
| 0x60 | MACDataChecksum() | R | R | R | Byte | — | — | — | — | — |

The checksum is the 8-bit sum of the MSB and LSB of the command + the (command length) bytes in the buffer. The final sum is the bitwise inversion of the result. Because the length is part of the checksum, the verification cannot take place until the length is written. The checksum and length must be written together as a word to be valid.

## 13.1.29  0x61 MACDataLength()

This is the length for *AltManufacturerAccess()* and *MACData()*.

| I²C Cmd | Name | Access | | | Protocol | Type | Min | Max | Default | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | SE | US | FA | | | | | | |
| 0x61 | MACDataLength() | R | R | R | Byte | — | — | — | — | — |

The length byte for all MAC commands includes the 2-byte command, the 1-byte checksum, the 1-byte length, and 1 to 32 bytes of data. This means the minimum length value is 5 for a valid block (no length or checksum is used for command only writes, so a block of 0 size is not necessary). For proper write command validation, the checksum and length must be written in order (word access triggered).

### 13.1.30  0x62/63 VoltHiSetThreshold

This read/write-word function returns the desired interrupt set threshold for a high battery voltage.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|----|----|-----------|------|-----|-----|------|
|         |      | SE | US | FA |           |      |     |     |      |
| 0x62/63 | *VoltHiSetThreshold()* | R/W | R/W | R/W | Word | I2 | 0 | 5000 | mV |

### 13.1.31  0x64/65 VoltHiClearThreshold

This read/write-word function returns the desired interrupt clear threshold for a high battery voltage.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|----|----|-----------|------|-----|-----|------|
|         |      | SE | US | FA |           |      |     |     |      |
| 0x64/65 | *VoltHiClearThreshold()* | R/W | R/W | R/W | Word | I2 | 0 | 5000 | mV |

### 13.1.32  0x66/67 VoltLoSetThreshold

This read/write-word function returns the desired interrupt set threshold for a low battery voltage.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|----|----|-----------|------|-----|-----|------|
|         |      | SE | US | FA |           |      |     |     |      |
| 0x66/67 | *VoltLoSetThreshold()* | R/W | R/W | R/W | Word | I2 | 0 | 5000 | mV |

### 13.1.33  0x68/69 VoltLoClearThreshold

This read/write-word function returns the desired interrupt clear threshold for a low battery voltage.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|----|----|-----------|------|-----|-----|------|
|         |      | SE | US | FA |           |      |     |     |      |
| 0x68/69 | *VoltLoClearThreshold()* | R/W | R/W | R/W | Word | I2 | 0 | 5000 | mV |

### 13.1.34  0x6A TempHiSetThreshold

This read/write-byte function returns the desired interrupt set threshold for a high battery temperature.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|----|----|-----------|------|-----|-----|------|
|         |      | SE | US | FA |           |      |     |     |      |
| 0x6A | *TempHiSetThreshold()* | R/W | R/W | R/W | Byte | I1 | −128 | 127 | °C |

### 13.1.35  0x6B TempHiClearThreshold

This read/write-byte function returns the desired interrupt clear threshold for a high battery temperature.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|----|----|-----------|------|-----|-----|------|
|         |      | SE | US | FA |           |      |     |     |      |
| 0x6B | *TempHiClearThreshold()* | R/W | R/W | R/W | Byte | I1 | −128 | 127 | °C |

### 13.1.36  0X6C TempLoSetThreshold

This read/write-byte function returns the desired interrupt set threshold for a low battery temperature.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|--|--|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x6C | *TempLoSetThreshold()* | R/W | R/W | R/W | Byte | I1 | −128 | 127 | °C |

### 13.1.37  0x6D TempLoClearThreshold

This read/write-byte function returns the desired interrupt clear threshold for a low battery temperature.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|--|--|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x6D | *TempLoClearThreshold()* | R/W | R/W | R/W | Byte | I1 | −128 | 127 | °C |

### 13.1.38  0x6E InterruptStatus()

This read-byte function returns the status of the interrupt bits for high/low voltage and temperature.

| Status | Condition | Action |
|--------|-----------|--------|
| Activate | 0x0057 to *AltManufacturerAccess()* | Outputs *ManufacturingStatus()* flags on *MACData()* |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RSVD | RSVD | RSVD | SOC_DELTA | TEMP_LO | TEMP_HI | VOLT_LO | VOLT_HI |

**RSVD (Bits 7–5)**: Reserved

**SOC_DELTA (Bit 4)**: RSOC interrupt

       1 =  RSOC interrupt set

       0 =  RSOC interrupt cleared

**TEMP_LO (Bit 3)**: Low temperature interrupt

       1 =  Low temperature interrupt set

       0 =  Low temperature interrupt cleared

**TEMP_HI (Bit 2)**: High temperature interrupt

       1 =  High temperature interrupt set

       0 =  High temperature interrupt cleared

**VOLT_LO (Bit 1)**: Low voltage interrupt

       1 =  Low voltage interrupt set

       0 =  Low voltage interrupt cleared

**VOLT_HI (Bit 0)**: High voltage interrupt

       1 =  High voltage interrupt set

       0 =  High voltage interrupt cleared

### 13.1.39  0x6F SOCSetDeltaThreshold

This read/write-byte function returns the desired interrupt set threshold for battery RSOC delta.

| I²C Cmd | Name | Access | | | Proto-col | Type | Min | Max | Unit |
|---------|------|--------|--|--|-----------|------|-----|-----|------|
| | | SE | US | FA | | | | | |
| 0x6F | *SOCSetDeltaThreshold()* | R/W | R/W | R/W | Byte | U1 | 0 | 100 | % |

## 13.2  0x70, 0x71 SusTurboPwr()

This command reads the maximum peak power value for 10-s pulse, sustained turbo power, in cW.

| I²C Cmd | Name | Access | | | Protocol | Type | Min | Max | Default | Unit |
|---------|------|--------|--------|--------|----------|------|-----|-----|---------|------|
| | | SE | US | FA | | | | | | |
| 0x70/71 | *SusTurboPwr()* | R/W | R/W | R/W | Word | I2 | 0 | 32767 | na | cW |

## 13.3  0x72, 0x73 MaxTurboPwr()

This command reads the maximum peak power value for 10-ms pulse occurring on top of 10-s 2 C-rate pulse.

| I²C Cmd | Name | Access | | | Protocol | Type | Min | Max | Default | Unit |
|---------|------|--------|--------|--------|----------|------|-----|-----|---------|------|
| | | SE | US | FA | | | | | | |
| 0x72/73 | *MaxTurboPwr()* | R/W | R/W | R/W | Word | I2 | 0 | 32767 | na | cW |

## 13.4  0x74, 0x75 SusTurboCurr()

This command reads the sustained peak current value, sustained turbo current, in mA. The gauge computes a new RAM value sustained turbo current every second. Sustained turbo current is initialized to the present value of max turbo current on reset or power up.

| I²C Cmd | Name | Access | | | Protocol | Type | Min | Max | Default | Unit |
|---------|------|--------|--------|--------|----------|------|-----|-----|---------|------|
| | | SE | US | FA | | | | | | |
| 0x74/75 | *SusTurboCurr()* | — | R/W | R/W | Word | I2 | 0 | 32767 | — | mA |

## 13.5  0x76, 0x77 MaxTurboCurr()

This command reads the maximum peak current value, max turbo current, in mA. The gauge computes a new RAM value of max turbo current every second. Max turbo current is initialized to present the value of max turbo current on reset or power up.

| I²C Cmd | Name | Access | | | Protocol | Type | Min | Max | Default | Unit |
|---------|------|--------|--------|--------|----------|------|-----|-----|---------|------|
| | | SE | US | FA | | | | | | |
| 0x76/77 | *MaxTurboCurr()* | R/W | R/W | R/W | Word | I2 | 0 | 32767 | — | mA |

## 13.6  0x78, 0x79 TurboSysR()

This command sets the **System Resistance** value of the system serial resistance along the path from the battery to the system power converter input that includes FETs, traces, sense resistors, and so on.

| I²C Cmd | Name | Access | | | Protocol | Type | Min | Max | Default | Unit |
|---------|------|--------|--------|--------|----------|------|-----|-----|---------|------|
| | | SE | US | FA | | | | | | |
| 0x78/79 | *TurboSysR()* | R/W | R/W | R/W | Word | I2 | 0 | 32767 | | mΩ |

## 13.7  0x7A, 0x7B TurboEdv()

This command sets the minimal voltage at the system power converter input at which the system will still operate. This command writes to the data flash value **Min System Voltage**. It writes it once on the first use to adjust for possible changes in the system design from the time the battery pack was designed.

| I²C Cmd | Name | Access | | | Protocol | Type | Min | Max | Default | Unit |
|---------|------|--------|--------|--------|----------|------|-----|-----|---------|------|
| | | SE | US | FA | | | | | | |
| 0x7A/7B | *TurboEdv()* | R/W | R/W | R/W | Word | I2 | 0 | 32767 | | mV |

## 13.8  0x00, 0x01 ManufacturerAccess() and 0x3E, 0x3F AltManufacturerAccess()

*AltManufacturerAccess()* provides a method of reading and writing data in the Manufacturer Access System (MAC). The MAC command is sent via *AltManufacturerAccess()* by a block protocol. The result is returned on *AltManufacturerAccess()* via a block read.

Commands are sent by writing to registers 0x00/0x01 or 0x3E/0x3F. 0x3E and 0x3F work the same as 0x00 and 0x01, but are primarily intended for block writes and reads.

Example: Send a MAC *Gauging()* to enable IT via *AltManufacturerAccess().*

1.  With Impedance Track disabled, send *Gauging()* (0x0021) to *AltManufacturerAccess()*

    a.  Command Write, start address = 0x3E (or 0x00). Data = 21 00 (data must be sent in little endian).

2.  IT is enabled, *ManufacturingStatus()[GAUGE_EN]* = 1.

Example: Read *Chemical ID()* (0x0006) via *AltManufacturerAccess().*

1.  Send *Chemical ID()* to *AltManufacturerAccess().*

    a.  Command Write, start address = 0x3E (or 0x00). Data sent = 06 00 (data must be sent in little endian).

2.  Read the result from *AltManufacturerAccess()* and *MACData()*.

    a.  Command Read, start address = 0x3E length = 36 bytes. The first 4 bytes of the response will be 06 00 10 12.

    b.  The first two bytes "06 00" is the MAC command (for verification).

    c.  The second two bytes "10 12" are the Chem ID in little endian. That is, 0x1210 for ChemID 1210.

    d.  The last two bytes of the 36-byte block will be the checksum and length. The length in this case will be 6. The checksum is 0xFF – (sum of the first length – 2 bytes). The length and checksum are used to validate the block response.

It is recommended to send "command only" operations to 0x00 and 0x01, and to set the command for a read back in the same way. The reason for this is that it can always reset any legacy support options that may be in effect; whereas, some legacy support options use 0x3E and 0x3F for other purposes. However, 0x3E and 0x3F can always safely be used for block reads. For backward compatibility, a request of the device number or version reports a value for a read on 0x00/0x01. The response word for the MAC commands DEV and VERSION (0x0001 and 0x0002) should report 0xFFA5 as the legacy response. This is meant as a token to indicate to the host that the real response is on the extended block. "Command only" operations take place immediately after the word write.

### Table 13-2. AltManufacturerAccess() Command List

| Command | Function | Access | Format | Data Read on MACData() | Not Available in SEALED Mode | Type | Units |
|---|---|---|---|---|---|---|---|
| 0x0001 | DeviceType | R | Block | √ | — | hex | — |
| 0x0002 | FirmwareVersion | R | Block | √ | — | hex | — |
| 0x0003 | HardwareVersion | R | Block | √ | — | hex | — |
| 0x0004 | IFChecksum | R | Block | √ | — | hex | — |
| 0x0005 | StaticDFSignature | R | Block | √ | — | hex | — |
| 0x0006 | ChemID | R | Block | √ | — | hex | — |
| 0x0007 | Prev_MacWrite | R | Block | √ | — | hex | — |
| 0x0008 | StaticChemDFSignature | R | Block | √ | — | hex | — |
| 0x0009 | AllDFSignature | R | Block | √ | — | hex | — |
| 0x012 | Reset | W | — | — | — | hex | — |
| 0x0017 | QMax Day | R | Block | √ | — | U2 | days |
| 0x0021 | Gauging (IT Enable) | W | — | — | √ | hex | — |
| 0x0023 | LifetimeDataCollection | W | — | — | √ | hex | — |
| 0x0028 | LifetimeDataReset | W | — | — | √ | hex | — |
| 0x002D | CalibrationMode | W | — | — | √ | hex | — |
| 0x002E | LifetimeDataFlush | W | — | — | √ | hex | — |
| 0x002F | LifetimeDataSpeedUp Mode | W | — | — | √ | hex | — |
| 0x0030 | SealDevice | W | — | — | — | hex | — |
| 0x0035 | SecurityKeys | R/W | Block | √ | √ | hex | — |

**Table 13-2. AltManufacturerAccess() Command List (continued)**

| Command | Function | Access | Format | Data Read on MACData() | Not Available in SEALED Mode | Type | Units |
|---------|----------|--------|--------|------------------------|------------------------------|------|-------|
| 0x0041 | Reset | W | — | — | — | hex | — |
| 0x0044 | SetDeepSleep | W | — | — | — | hex | — |
| 0x0045 | ClearDeepSleep | W | — | — | — | hex | — |
| 0x0046 | PulseGPIO | W | — | — | — | hex | — |
| 0x0047 | TambientSync | W | — | — | — | hex | — |
| 0x0048 | ClearGPIO | W | — | — | — | hex | — |
| 0x0049 | SetGPIO | W | — | — | — | hex | — |
| 0x004A | Device Name | R/W | Block | √ | — | hex | — |
| 0x004B | Device Chem | R/W | Block | √ | — | hex | — |
| 0x004C | Manufacturer Name | R/W | Block | √ | — | hex | — |
| 0x004D | Manufacture Date | R/W | Block | √ | — | hex | — |
| 0x004E | Serial Number | R/W | — | √ | — | hex | — |
| 0x0054 | OperationStatus | R | Block | √ | — | hex | — |
| 0x0055 | ChargingStatus | R | Block | √ | — | hex | — |
| 0x0056 | GaugingStatus | R | Block | √ | — | hex | — |
| 0x0057 | ManufacturingStatus | R | Block | √ | — | hex | — |
| 0x0060 | LifetimeDataBlock1 | R | Block | √ | — | Mixed | Mixed |
| 0x0061 | LifetimeDataBlock2 | R | Block | √ | | Mixed | Mixed |
| 0x0062 | LifetimeDataBlock3 | R | Block | √ | — | U4 | s |
| 0x0063 | LifetimeDataBlock4 | R | Block | √ | | Mixed | Mixed |
| 0x0065 | LifetimeDataBlock6 | R | Block | √ | | Mixed | Mixed |
| 0x0065 | LifetimeDataBlock7 | R | Block | √ | | Mixed | Mixed |
| 0x0067 | LifetimeDataBlock8 | R | Block | √ | | Mixed | Mixed |
| 0x0068 | LifetimeDataBlock9 | R | Block | √ | | Mixed | Mixed |
| 0x0069 | LifetimeDataBlock10 | R | Block | √ | | Mixed | Mixed |
| 0x006A | LifetimeDataBlock11 | R | Block | √ | | Mixed | Mixed |
| 0x006B | LifetimeDataBlock12 | R | Block | √ | | Mixed | Mixed |
| 0x0070 | ManufacturerInfo | R/W | Block | √ | — | hex | — |
| 0x0071 | DAStatus1 | R | Block | √ | — | Mixed | Mixed |
| 0x0072 | DAStatus2 | R | Block | √ | — | Mixed | Mixed |
| 0x0073 | ITStatus1 | R | Block | √ | — | Mixed | Mixed |
| 0x0074 | ITStatus2 | R | Block | √ | — | Mixed | Mixed |
| 0x0075 | ITStatus3 | R | Block | √ | — | Mixed | Mixed |
| 0x0077 | FCC_SOH | R | Block | √ | — | hex | — |
| 0x0078 | Filtered Capacity | R | Block | √ | — | Hex | Mixed |
| 0x007A | ManufacturerInfoB | R/W | Block | √ | — | Hex | — |
| 0x007B | ManufacturerInfoC | R/W | Block | √ | — | Hex | — |
| 0x00B0 | ChargingVoltageOverride | R/W | Block | √ | √ | I2 | mV |
| 0x01yy | DFAccessRowAddress | R/W | Block | — | √ | hex | — |
| 0x0F00 | ROMMode | W | — | — | √ | hex | — |
| 0x7C40 | SwitchToHDQ | W | — | — | √ | hex | — |
| 0xF080 | ExitCalibrationOutput | R/W | Block | √ | √ | hex | — |
| 0xF081 | OutputCCandADCforCalibration | R/W | Block | √ | √ | hex | — |

## 13.8.1 AltManufacturerAccess() 0x0001 Device Type

The device can be checked for its part number.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x0001 to *AltManufacturerAccess()* | Returns the device part number on a subsequent read on *MACData()* in the following format: aaAA, where:<br>aaAA: Device type |

### 13.8.2 *AltManufacturerAccess() 0x0002 Firmware Version*

The device can be checked for its firmware version.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0002 to *AltManufacturerAccess()* | Returns the firmware revision on *MACData()* in the following format: ddDDvvVVbbBBTTzzZZRREE, where: ddDD: Device number vvVV: Version bbBB: build number TT: Firmware type zzZZ: Impedance Track version RR: Reserved EE: Reserved |

### 13.8.3 *AltManufacturerAccess() 0x0003 Hardware Version*

The device can be checked for its hardware version.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0003 to *AltManufacturerAccess()* | Returns the hardware version on a subsequent read on *MACData()* |

### 13.8.4 *AltManufacturerAccess() 0x0004 Instruction Flash Signature*

The device can return the instruction flash signature.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0004 to *AltManufacturerAccess()* | Returns the IF signature on a subsequent read on *MACData()* after a wait time of 250 ms |

### 13.8.5 *AltManufacturerAccess() 0x0005 Static DF Signature*

This command returns the signature of all static data flash parameters. The command does not update **Static DF Signature**.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0005 to *AltManufacturerAccess()* | Returns the signature of all static DF on a subsequent read on *MACData()* after a wait time of 250 ms. The MSB is set to 1 if the calculated signature does not match the signature stored in DF. |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| System Data | Integrity | Static DF Signature | H2 | 0 | 0x7FFF | 0x0000 | — | Static data flash signature. Use the MAC *StaticDFSignature()* (with MSB set to 0) to manually initialize this value. |

### 13.8.6 *AltManufacturerAccess() 0x0006 Chemical ID*

This command returns the chemical ID of the OCV tables used in the gauging algorithm.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0006 to *AltManufacturerAccess()* | Returns the chemical ID on subsequent read on *MACData()* |

### 13.8.7 *AltManufacturerAccess() 0x0007 Pre_MACWrite*

This command enables copying the last MAC into a 2-byte block.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0007 to *AltManufacturerAccess()* | Copies the last MAC information into a 2-byte block *MACData()* |

### 13.8.8 AltManufacturerAccess() 0x0008 Static Chem DF Signature

This command returns the signature of all static, chemistry-related data flash parameters. The command does not update **Static Chem DF Signature**.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0008 to *AltManufacturerAccess()* | Returns the signature of all static chemistry DF on a subsequent read on *MACData()* after a wait time of 250 ms. The MSB is set to 1 if the calculated signature does not match the signature stored in DF. |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| System Data | Integrity | Static Chem DF Signature | H2 | 0 | 0x7FFF | 0x3A4C | — | Static chemistry data signature. Use the MAC *StaticChemDFSignature()* (with MSB set to 0) to manually initialize this value. |

### 13.8.9 AltManufacturerAccess() 0x0009 All DF Signature

This command returns the signature of all data flash parameters. The command does not update **All DF Signature**.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0009 to *AltManufacturerAccess()* | Returns the signature of all DF parameters on a subsequent read on *MACData()* after a wait time of 250 ms. The MSB is set to 1 if the calculated signature does not match the signature stored in DF. It is normally expected that this signature will change due to updates of lifetime, gauging, and other information. |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| System Data | Integrity | All DF Signature | H2 | 0 | 0x7FFF | 0x0000 | — | All data flash signature. Use the MAC *AllDFSignature()* (with MSB set to 0) to manually initialize this value. |

### 13.8.10 AltManufacturerAccess() 0x0012 Device Reset

This command resets the device.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0012 to *AltManufacturerAccess()* | Resets the device |

---

**NOTE:** Command 0x0041 also resets the device.

---

### 13.8.11 AltManufacturerAccess() 0x0017 QMax Day

Instructs the fuel gauge to return the last day QMax was updated. The value is updated with the *Elapsed Time Count()* converted to days whenever QMax is updated.

### 13.8.12 AltManufacturerAccess() 0x0021 Gauging

This command enables or disables the gauging function to ease testing during manufacturing.

| Status | Condition | Action |
|---|---|---|
| Disable | *ManufacturingStatus()[GAUGE_EN]* = 1 AND 0x0021 to *AltManufacturerAccess()* | *ManufacturingStatus()[GAUGE_EN]* = 0 Disables the gauging feature |
| Enable | *ManufacturingStatus()[GAUGE_EN]* = 0 AND 0x0021 to *AltManufacturerAccess()* | *ManufacturingStatus()[GAUGE_EN]* = 1 Enables the gauging feature |

### 13.8.13  AltManufacturerAccess() 0x0023 Lifetime Data Collection

This command disables/enables lifetime data collection for manufacturing.

| Status | Condition | Action |
|---|---|---|
| Disable | *ManufacturingStatus()[LF_EN]* = 1 AND 0x0023 to *AltManufacturerAccess()* | *ManufacturingStatus()[LF_EN]* = 0 The lifetime data collection feature is disabled. |
| Enable | *ManufacturingStatus()[LF_EN]* = 0 AND 0x0023 to *AltManufacturerAccess()* | *ManufacturingStatus()[LF_EN]* = 1 The lifetime data collection feature is enabled. |

### 13.8.14  AltManufacturerAccess() 0x0028 Lifetime Data Reset

This command resets lifetime data in data flash for manufacturing.

| Status | Condition | Action |
|---|---|---|
| Reset | 0x0028 to *AltManufacturerAccess()* | Clears lifetime data in DF |

### 13.8.15  AltManufacturerAccess() 0x002D CALIBRATION Mode

This command disables/enables entry into CALIBRATION mode. The status is indicated by the *ManufacturingStatus()[CAL_EN]* flag. CALIBRATION mode is disabled upon a reset.

| Status | Condition | Action |
|---|---|---|
| Disable | *ManufacturingStatus()[CAL_EN]* = 1 AND 0x002D to *AltManufacturerAccess()* | *ManufacturingStatus()[CAL_EN]* = 0 Disables output of ADC and CC raw data on *MACData()* |
| Enable | *ManufacturingStatus()[CAL_EN]* = 0 AND 0x002D to *AltManufacturerAccess()* | *ManufacturingStatus()[CAL_EN]* = 1 Enables output of ADC and CC raw data on *MACData()*, controllable with 0xF081 on *AltManufacturerAccess()* |

### 13.8.16  ManufacturerAccess() 0x002E Lifetime Data Flush

This command flushes the RAM lifetime data to data flash to help streamline evaluation testing.

### 13.8.17  ManufacturerAccess() 0x002F Lifetime Data SPEED UP Mode

For ease of evaluation testing, this command enables a lifetime SPEED UP mode where every 1 s in real time counts as 1 hour in FW time. When the lifetime SPEED UP mode is enabled, the *ManufacturingStatus()[LT_TEST]* = 1.

The SPEED UP mode will be disabled if this command is sent again when *[LT_TEST]* = 1, the MAC *LifetimeDataReset()* command is sent, the MAC *SealDevice()* command is sent, or the device is reset.

### 13.8.18  AltManufacturerAccess() 0x0030 Seal Device

This command seals the device for the field, disabling certain commands and access to DF.

| Status | Condition | Action |
|---|---|---|
| Sealed | *OperationStatusA()[SEC1,SEC0]* = 0,1 or 1,0 AND 0x0030 to *AltManufacturerAccess()* | *OperationStatusA()[SEC1,SEC0]* = 1,1 Certain commands are not available. See Table 13-1 for details. |

### 13.8.19  AltManufacturerAccess() 0x0035 Security Keys

This is a read/write command that changes the Unseal, Full Access, and Lifetime reset keys. To read the keys, send the *SecurityKeys()* command to either the *AltManufacturerAccess()* 0x00 or 0x3E, followed by a read from *AltManufacturerAccess()*.

To change the keys, the write operations must be sent through *AltManufacturerAccess()* 0x3E with the *SecurityKeys()* followed by the keys. Each parameter entry must be sent in little endian.

> **NOTE:** Create each access key to be unique; otherwise, the copy key will not execute the intended function.

Example of changing the Unseal key to 0x0123, 0x4567; the Full Access key to 0x89AB, 0xCDEF; and the Lifetime reset key to 0x2244, 0x2131:

Byte 0: Unseal Key LSW
Byte 1: Unseal Key MSW
Byte 2: Full Access Key LSW
Byte 3: Full Access Key MSW
Byte 4: Lifetime Reset Key LSW
Byte 5: Lifetime Reset Key MSW

Write block: command through *AltManufacturerAccess()* starting address 0x3E:

data block (hex) = [35 00 23 01 67 45 AB 89 EF CD 44 22 31 21].

Starting address 0x60, data block(hex) = [0A 0C] (checksum followed by length)

Checksum = 0x52 = ~(0x35 + 0x00 + 0x23 + 0x01 + 0x67 + 0x45 + 0xAB + 0x89 + 0xEF + 0xCD + 0x44 + 0x22 + 0x31 + 0x21). The checksum is the 8-bit sum of the MSB and LSB of the command plus the (command length) bytes in the buffer. The final sum is the bitwise inversion of the result.

### 13.8.20  *AltManufacturerAccess() 0x0037 Authentication Key*

This command enters a new authentication key into the device.

| Status | Condition | Action |
|--------|-----------|--------|
| Initiate | *OperationStatusA()[SEC1,SEC0]* = 0,1 AND 0x0037 to *AltManufacturerAccess()* | *OperationStatusB()[AUTH]* = 1 160-bit random number available at *MACData()* |
| Enter Key | Correct 128-bit key written to *MACData()* in the format 0xAABBCCDDEEFFGGHHIIJJKKLLMMNNOOPP, where AA is LSB. In addition to this information, the checksum + length data block is required. | Wait time 250 ms *OperationStatusB()[AUTH]* = 0 The device returns a 160-bit HMAC digest at *MACData()* in the format: 0xAABBCCDDEEFFGGHHIIJJKKLLMMNNOOPPQQRR SSTTT, where AA is LSB. The HMAC digest was calculated using a challenge of all 0s + key. The result can be used to verify the key without allowing a plain text read back. |

### 13.8.21  *AltManufacturerAccess() 0x0041 Device Reset*

This command resets the device.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x0041 to *AltManufacturerAccess()* | Resets the device |

> **NOTE:** Command 0x0012 also resets the device.

### 13.8.22  *AltManufacturerAccess() 0x0044 Set Deep Sleep*

This command instructs the gauge to enter DEEP SLEEP mode when available.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x0044 to *AltManufacturerAccess()* | *OperationStatusB()[DPSLEEPM]* = 1 |

### 13.8.23  *AltManufacturerAccess() 0x0045 Clear Deep Sleep*

This command instructs the gauge to exit DEEP SLEEP mode.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x0045 to *AltManufacturerAccess()* | *OperationStatusB()[DPSLEEPM]* = 0 |

### 13.8.24  AltManufacturerAccess() 0x0046 Pulse GPIO

This command generates a single pulse or state change on the PULS pin.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x0046 to *AltManufacturerAccess()* | If **IO Config[GPIO_LEVEL_EN]** = 0, it generates a single pulse on the PULS pin with a polarity of **IO Config[GPIO_POL]** and a width of **GPIO Pulse Width**. If **IO Config[GPIO_LEVEL_EN]** = 1, it changes the state of the PULS pin. |

### 13.8.25  AltManufacturerAccess() 0x0047 Tambient Sync

This command synchronizes the ambient temperature in Impedance Track, T_ambient, to the presently measured cell temperature, *Temperature()*.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x0047 to *AltManufacturerAccess()* | Synchronize T_ambient to *Temperature()* and trigger an IT simulation if **I2C Gauging Configuration[TAMB_SYNC_SIM]** = 1 |

### 13.8.26  AltManufacturerAccess() 0x0047 Clear GPIO

This command clears the state on the PULS pin.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x0048 to *AltManufacturerAccess()* | If **IO Config[GPIO_LEVEL_EN]** = 1, it clears the state of the PULS pin and clears the *OperationStatus()[GPIO_LVL]* flag. |

### 13.8.27  AltManufacturerAccess() 0x0047 Set GPIO

This command sets state on the PULS pin.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x0049 to *AltManufacturerAccess()* | If **IO Config[GPIO_LEVEL_EN]** = 1, it sets the state of the PULS pin and sets the *OperationStatus()[GPIO_LVL]* flag. |

### 13.8.28  AltManufacturerAccess() 0x004A Device Name

This command returns the assigned pack name, as programmed in data flash.

| Status | Condition | Action |
|--------|-----------|--------|
| Enable | 0x004A to *AltManufacturerAccess()* | Returns the value stored in **Device Name** |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| I2C Configuration | Data | Device Name | S21 | — | — | BQ27Z561-R2 | ASCII | Assigned pack name |

### 13.8.29  AltManufacturerAccess() 0x004B Device Chem

This command returns the battery chemistry used in the pack, as programmed in data flash.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x004B to *AltManufacturerAccess()* | Returns the value stored in **Device Chemistry** |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| I2C Configuration | Data | Device Chemistry | S5 | — | — | LION | ASCII | The battery chemistry used in the pack. This parameter does not change with chemistry programming and must be updated manually. |

### 13.8.30   AltManufacturerAccess() 0x004C Manufacturer Name

This command returns the pack manufacturer name, as programmed in data flash.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x004C to *AltManufacturerAccess()* | Returns the value stored in **Manufacturer Name** |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| I2C Configuration | Data | Manufacturer Name | S21 | — | — | Texas Instruments | ASCII | Pack manufacturer name |

### 13.8.31   AltManufacturerAccess() 0x004D Manufacture Date

This command returns the manufacture date of the pack, as programmed in data flash.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x004D to *AltManufacturerAccess()* | Returns the value stored in **Manufacture Date** |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| I2C Configuration | Data | Manufacture Date | U2 | 0 | 65535 | 1980-1-1 | date | Manufacture date of the pack, which follows the format: Day + Month×32 + (Year−1980) × 512 Example: 10/27/2017 = 19291 (or 0x4B5B, binary split: 100101, 1010, 11011) |

### 13.8.32   AltManufacturerAccess() 0x004E Serial Number

This command returns the serial number of the pack, as programmed in data flash.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x004E to *AltManufacturerAccess()* | Returns the value stored in **Serial Number** |

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|---|---|---|---|---|---|---|---|---|
| I2C Configuration | Data | Serial Number | H2 | 0x0000 | 0xFFFF | 0x0001 | — | Serial number of the pack |

### 13.8.33   AltManufacturerAccess() 0x0054 OperationStatus

This command returns the *OperationStatusA()* and *OperationStatusB()* flags on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0054 to *AltManufacturerAccess()* | Outputs *OperationStatusA()* and *OperationStatusB()* flags on *MACData()* in the format AAaaBBbb where AAaa is **Operation Status A** and BBbb is **Operation Status B** |

### 13.8.33.1 Operation Status A

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SLEEP | RSVD | RSVD | RSVD | SS | RSVD | SEC1 | SEC0 | BTP_INT | DP_SLP | GPIO_LVL | RSVD | RSVD | RSVD | RSVD | RSVD |

**SLEEP (Bit 15)**: SLEEP mode conditions met

    1 = Active

    0 = Inactive

**RSVD (Bits 14–10)**: Reserved

**SEC1, SEC0 (Bits 9, 8)**: SECURITY Mode

    0, 0 = Reserved

    0, 1 = Full Access

    1, 0 = Unsealed

    1, 1 = Sealed

**BTP_INT (Bit 7)**: Battery Trip Point Interrupt. Setting and clearing this bit depends on various conditions. See Section 4.12 for details.

**DP_SLP (Bit 6)**: DEEP SLEEP mode status

    1 = Active

    0 = Inactive

**GPIO_LVL (Bit 5)**: GPIO pin level when used in level mode (*IO Config[GPIO_LEVEL_EN]* = 1)

    1 = High

    0 = Low

**RSVD (Bits 4–0)**: Reserved

### 13.8.33.2 Operation Status B

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | RSVD | RSVD | RSVD | RSVD | SLPAD | RSVD | INIT | DPSLEEPM | XL | RSVD | CAL | RSVD | AUTH | RSVD | RSVD |

**RSVD (Bits 15–11)**: Reserved

**SLPAD (Bit 10)**: SLPAD

    1 = Active

    0 = Inactive

**RSVD (Bit 9)**: Reserved

    1 = Active

    0 = Inactive

**INIT (Bit 8)**: Initialization after full reset

    1 = Active

    0 = Inactive

**DPSLEEPM (Bit 7)**: DEEP SLEEP command received

    1 = Active

    0 = Inactive

**XL (Bit 6)**: 400-kHz mode

1 = Active

0 = Inactive

**RSVD (Bit 5)**: Reserved

**CAL (Bit 4)**: Calibration output (raw ADC and CC data)

    1 = Active when either the MAC *OutputCCADCCal()* or *OutputShortedCCADCCal()* is sent and the raw CC and ADC data for calibration is available.

    0 = When the raw CC and ADC data for calibration is not available.

**RSVD (Bit 3)**: Reserved

**AUTH (Bit 2)**: Authentication in progress

    1 = Active

    0 = Inactive

**RSVD (Bits 1–0)**: Reserved

### 13.8.34  AltManufacturerAccess() 0x0055 ChargingStatus

This command returns the *Temp Range* and *Charging Status* flags on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0055 to *AltManufacturerAccess()* | Outputs three bytes of charging related data in the format AABBbb, where AA is *Temp Range* and BBbb is *Charging Status* flags on *MACData()* |

#### 13.8.34.1  Temp Range

*Temp Range* status bits indicate the current operating temperature range.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RSVD | OT | HT | STH | RT | STL | LT | UT |

**RSVD (Bit 7)**: Reserved

**OT (Bit 6)**: Overtemperature range

    1 = Temperature > T4

    0 = Inactive

**HT (Bit 5)**: High temperature range

    1 = T3 < Temperature < T4

    0 = Inactive

**STH (Bit 4)**: Standard temperature high range

    1 = T6 < Temperature < T3

    0 = Inactive

**RT (Bit 3)**: Recommended temperature range

    1 = T5 < Temperature < T6

    0 = Inactive

**STL (Bit 2)**: Standard temperature low range

    1 = T2 < Temperature < T5

    0 = Inactive

**LT (Bit 1)**: Low temperature range

    1 = T1 < Temperature < T2

0 =  Inactive

**UT (Bit 0)**: Undertemperature range

1 =  Temperature < T1

0 =  Inactive

### 13.8.34.2  Charging Status

Charging Status bits indicate various status bits related to charging.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|------|------|-----|------|------------|------------|-----|------|----|----|----|----|----|----|
| RSVD | RSVD | RSVD | RSVD | NCT | RSVD | CV_D GRD1 | CV_D GRD0 | VCT | MCHG | SU | IN | HV | MV | LV | PV |

**RSVD (Bits 15–12)**: Reserved

**NCT (Bit 11)**: NCT

1 =  Active

0 =  Inactive

**RSVD (Bit 10)**: Reserved

**CV_DGRD (Bits 9–8)**: Charge degradation

0,0 =  No degradation

0,1 =  Mode 1

1,0 =  Mode 2

1,1 =  Mode 3

**VCT (Bit 7)**: Charge termination

1 =  Active

0 =  Inactive

**MCHG (Bit 6)**: Maintenance charge

1 =  Active

0 =  Inactive

**SU (Bit 5)**: Charge suspend

1 =  Active

0 =  Inactive

**IN (Bit 4)**: Charge inhibit

1 =  Active

0 =  Inactive

**HV (Bit 3)**: High voltage region

1 =  Active

0 =  Inactive

**MV (Bit 2)**: Mid voltage region

1 =  Active

0 =  Inactive

**LV (Bit 1)**: Low voltage region

1 =  Active

> 0 = Inactive

**PV (Bit 0)**: Precharge voltage region

> 1 = Active
>
> 0 = Inactive

### 13.8.35  AltManufacturerAccess() 0x0056 GaugingStatus

This command returns the *GaugingStatus()* flags on *MACData()*.

> **NOTE:** The Battery Management Studio (BQSTUDIO) tool splits the returned value of *GaugingStatus()* into two separate registers: Gauging Status = lowest byte, and IT Status = 2nd and 3rd bytes, with the 4th byte reserved.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0056 to *AltManufacturerAccess()* | Outputs *GaugingStatus()* flags on *MACData()* |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | RSVD | RSVD | RSVD | RSVD | RSVD | RSVD | RSVD | RSVD | RSVD | QMA XDOD OK | OCV FR | LDMD | RX | QMax | VDQ |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NSFM | OCVP RED | SLP QMax | QEN | VOK | RDIS | RSVD | REST | RSVD | DSG | EDV | RSVD | TC | TD | FC | FD |

**RSVD (Bits 31–22)**: Reserved. Do not use.

**QMAXDODOK (Bit 21)**: Indicates whether the DOD is outside the cell chemistry's voltage flat zone. QMax updates do not happen if DOD is in the flat zone.

> 1 = Outside the flat zone
>
> 0 = In the flat zone

**OCVFR (Bit 20)**: Open circuit voltage in flat region (during RELAX)

> 1 = Detected
>
> 0 = Not detected

**LDMD (Bit 19)**: LOAD mode

> 1 = Constant power
>
> 0 = Constant current

**RX (Bit 18)**: Resistance update (toggles after every resistance update)

**QMax (Bit 17)**: QMax update (toggles after every QMax update)

**VDQ (Bit 16)**: Discharge qualified for learning (based on RU flag)

**NSFM (Bit 15)**: Negative scale factor mode

> 1 = Negative Ra scaling factor detected
>
> 0 = Negative Ra scaling factor not detected

**OCVPRED (Bit 14):** Open-circuit-voltage predicted

> 1 = Fast OCV prediction is performed in RELAX mode.
>
> 0 = Fast OCV prediction is not performed or not in RELAX mode.

**SLPQMax (Bit 13)**: QMax update during sleep

> 1 = Active
>
> 0 = Inactive

**QEN (Bit 12)**: Impedance Track gauging (Ra and QMax updates are enabled.)

    1 = Enabled

    0 = Disabled

**VOK (Bit 11)**: Voltage OK for QMax update

    1 = Detected

    0 = Not detected

**RDIS (Bit 10)**: Resistance updates

    1 = Disabled

    0 = Enabled

**RSVD (Bit 9)**: Reserved. Do not use.

**REST (Bit 8)**: Rest

    1 = The OCV reading is taken.

    0 = The OCV reading is not taken or is not in RELAX mode.

**RSVD (Bit 7)**: Reserved. Do not use.

**DSG (Bit 6)**: Discharge/relax

    1 = Charging not detected

    0 = Charging detected

**EDV (Bit 5)**: End-of-discharge termination voltage

    1 = Termination voltage reached during discharge

    0 = Termination voltage not reached or not in DISCHARGE mode

**RSVD (Bit 4)**: Reserved. Do not use.

**TC (Bit 3)**: Terminate charge

    1 = Detected

    0 = Not detected

**TD (Bit 2)**: Terminate discharge

    1 = Detected

    0 = Not detected

**FC (Bit 1)**: Fully charged

    1 = Detected

    0 = Not detected

**FD (Bit 0)**: Fully discharged

    1 = Detected

    0 = Not detected

### 13.8.36  *AltManufacturerAccess() 0x0057 ManufacturingStatus*

This command returns the *ManufacturingStatus()* flags on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0057 to *AltManufacturerAccess()* | Outputs *ManufacturingStatus()* flags on *MACData()* |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|----|----|----|----|----|----|----|----|
| CAL_EN | LT_TEST | RSVD | RSVD | RSVD | RSVD | RSVD | RSVD |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|----|----|
| RSVD | RSVD | LF_EN | RSVD | GAUGE_EN | RSVD | RSVD | RSVD |

**CAL_EN (Bit 15)**: CALIBRATION mode

  1 = Enabled

  0 = Disabled

**LT_TEST (Bit 14)**: LIFETIME SPEED UP mode

  1 = Enabled

  0 = Disabled

**RSVD (Bits 13–6)**: Reserved

**LF_EN (Bit 5)**: Lifetime data collection

  1 = Enabled

  0 = Disabled

**RSVD (Bit 4)**: Reserved

**GAUGE_EN (Bit 3)**: Gas gauging

  1 = Enabled

  0 = Disabled

**RSVD (Bits 2–0)**: Reserved

### 13.8.37 *AltManufacturerAccess() 0x0060 Lifetime Data Block 1*

This command returns the lifetime data on *MACData()* with the following format:
  aaAAbbBBccCCddDDeeEEffFFGGHHIIJJ.

| Value | Description |
|-------|-------------|
| AAaa | Max cell voltage |
| BBbb | Min cell voltage |
| CCcc | Max charge current |
| DDdd | Max discharge current |
| EEee | Max average discharge current |
| FFff | Max average discharge power |
| GG | Max cell temperature |
| HH | Min cell temperature |
| II | Max circuit temperature |
| JJ | Min circuit temperature |

### 13.8.38 *AltManufacturerAccess() 0x0062 Lifetime Data Block 3*

This command returns the lifetime data on *MACData()* with the following format:
  aaAAbbBB

| Value | Description |
|-------|-------------|
| AAaaBBbb | Total FW Runtime |

### 13.8.39  ManufacturerAccess() 0x0063 Lifetime Data Block 4

This command returns the lifetime data with the following format:

aaAAbbBBccCCddDDeeEEffFF.

| Value | Description |
|-------|-------------|
| AAaa | Count of valid charge terminations |
| BBbb | Cycle count at last valid charge termination |
| CCcc | Count of QMax updates |
| DDdd | Cycle count at last QMax update |
| EEee | Count or impedance table updates |
| FFff | Cycle count at last impedance update |

### 13.8.40  ManufacturerAccess() 0x0065-0x006B Lifetime SOC Range Table Data Blocks

These commands return the lifetime data on *MACData()*. See Table 5-1 for more information on the collected time based data.

Each block access returns data, as indicated below, where MAC 0x0065 returns SOC range one, MAC 0x0066 returns SOC range two, and so on:

aaAAbbBBCCccDDddEEeeFFffGGggHHhhIIiiJJjjKKkkLLllMMmmNNnnOOooPPpp.

| Value | Description |
|-------|-------------|
| AAaaBBbb | Seconds at temperature range 1 at SOC range = (MAC number – 0x0064) |
| CCccDDdd | Seconds at temperature range 2 at SOC range = (MAC number – 0x0064) |
| EEeeFFff | Seconds at temperature range 3 at SOC range = (MAC number – 0x0064) |
| GGggHHhh | Seconds at temperature range 4 at SOC range = (MAC number – 0x0064) |
| IIiiJJjj | Seconds at temperature range 5 at SOC range = (MAC number – 0x0064) |
| KKkkLLll | Seconds at temperature range 6 at SOC range = (MAC number – 0x0064) |
| MMmmNNnn | Seconds at temperature range 7 at SOC range = (MAC number – 0x0064) |
| OOooPPpp | Seconds at temperature range 8 at SOC range = (MAC number – 0x0064) |

### 13.8.41  AltManufacturerAccess() 0x0070 ManufacturerInfo

This command read or writes the data flash values of **Manufacturer Info Block A01** through **Manufacturer Info Block A32** on *MACData()*.

| Status | Condition | Action |
|--------|-----------|--------|
| Activate | 0x0070 to *AltManufacturerAccess()* | Outputs 32 bytes of Manufacturer Info on *MACData()* in the following format: AABBCCDDEEFFGGHHIIJJKKLLMMNN OOPPQQRRSSTTUUVVWWXXVVZZ112233445566 |

| Class | Subclass | Name | Type | Min | Max | Default | Unit |
|-------|----------|------|------|-----|-----|---------|------|
| System Data | Manufacturer Data | Manufacturer Info Block A01 | H1 | 0x00 | 0xFF | 0x61 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A02 | H1 | 0x00 | 0xFF | 0x62 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A03 | H1 | 0x00 | 0xFF | 0x63 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A04 | H1 | 0x00 | 0xFF | 0x64 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A05 | H1 | 0x00 | 0xFF | 0x65 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A06 | H1 | 0x00 | 0xFF | 0x66 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A07 | H1 | 0x00 | 0xFF | 0x67 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A08 | H1 | 0x00 | 0xFF | 0x68 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A09 | H1 | 0x00 | 0xFF | 0x69 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A10 | H1 | 0x00 | 0xFF | 0x6A | — |
| System Data | Manufacturer Data | Manufacturer Info Block A11 | H1 | 0x00 | 0xFF | 0x6B | — |

| Class | Subclass | Name | Type | Min | Max | Default | Unit |
|---|---|---|---|---|---|---|---|
| System Data | Manufacturer Data | Manufacturer Info Block A12 | H1 | 0x00 | 0xFF | 0x6C | — |
| System Data | Manufacturer Data | Manufacturer Info Block A13 | H1 | 0x00 | 0xFF | 0x6D | — |
| System Data | Manufacturer Data | Manufacturer Info Block A14 | H1 | 0x00 | 0xFF | 0x6E | — |
| System Data | Manufacturer Data | Manufacturer Info Block A15 | H1 | 0x00 | 0xFF | 0x6F | — |
| System Data | Manufacturer Data | Manufacturer Info Block A16 | H1 | 0x00 | 0xFF | 0x70 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A17 | H1 | 0x00 | 0xFF | 0x71 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A18 | H1 | 0x00 | 0xFF | 0x72 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A19 | H1 | 0x00 | 0xFF | 0x73 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A20 | H1 | 0x00 | 0xFF | 0x74 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A21 | H1 | 0x00 | 0xFF | 0x75 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A22 | H1 | 0x00 | 0xFF | 0x76 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A23 | H1 | 0x00 | 0xFF | 0x77 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A24 | H1 | 0x00 | 0xFF | 0x7A | — |
| System Data | Manufacturer Data | Manufacturer Info Block A25 | H1 | 0x00 | 0xFF | 0x78 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A26 | H1 | 0x00 | 0xFF | 0x79 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A27 | H1 | 0x00 | 0xFF | 0x30 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A28 | H1 | 0x00 | 0xFF | 0x31 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A29 | H1 | 0x00 | 0xFF | 0x32 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A30 | H1 | 0x00 | 0xFF | 0x33 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A31 | H1 | 0x00 | 0xFF | 0x34 | — |
| System Data | Manufacturer Data | Manufacturer Info Block A32 | H1 | 0x00 | 0xFF | 0x35 | — |

### 13.8.42  AltManufacturerAccess() 0x0071 DAStatus1

This command returns the cell voltage, cell current, cell power, power, and average power on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0071 to *AltManufacturerAccess()* | Outputs 32 bytes of data on *MACData()* in the following format: aaAAbbBBccCCddDDeeEEffFFggGGhhHHiiIIjjJJkkKKllLLmmM MnnNNooOOppPP where:<br>AAaa: Cell 1 voltage<br>BBbb:<br>CCcc:<br>DDdd:<br>EEee:<br>FFff:<br>GGgg: Cell 1 current. simultaneous current measured during Cell 1 Voltage measurement<br>HHhh:<br>IIii:<br>JJjj:<br>KKkk: Cell 1 Power. Calculated using Cell 1 Voltage and Cell 1 Current data<br>LLll:<br>MMmm:<br>NNnn:<br>OOoo: Power calculated by *Voltage()* × *Current()*<br>PPpp: Average power. Calculated by *Voltage()* × *AverageCurrent()* |

### 13.8.43  AltManufacturerAccess() 0x0072 DAStatus2

This command returns the internal temp sensor, TS1 *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0072 to *AltManufacturerAccess()* | Outputs 20 bytes of temperature data values on *MacData()* in the following format: aaAAbbBBccCCddDDeeEEffFFggGGhhHHiiIIjjJJ where:<br>AAaa: Int temperature<br>BBbb: TS1 temperature<br>CCcc:<br>DDdd:<br>EEee:<br>FFff: Cell temperature<br>GGgg:<br>HHhh: Cell 1 raw voltage<br>IIii: Cell 1 raw current<br>JJjj: Wake comparator current |

### 13.8.44 *AltManufacturerAccess()* 0x0073 ITStatus1

This command instructs the device to return Impedance Track related gauging information on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0073 to *AltManufacturerAccess()* | Outputs 20 bytes of IT data values on *MACData()* in the following format:<br>aaAAbbBBccCCddDDeeEEffFFggGGhhHHiiIIjjJJ where:<br>AAaa: True Rem Q. True remaining capacity in mAh from IT simulation before any filtering or smoothing function. This value can be negative or higher than FCC.<br>BBbb: True Rem E. True remaining energy in cWh from IT simulation before any filtering or smoothing function. This value can be negative or higher than FCC.<br>CCcc: Initial Q. Initial capacity calculated from IT simulation<br>DDdd: Initial E. Initial energy calculated from IT simulation<br>EEee: True Full Chg Q. True full charge capacity from IT simulation without the effects of any smoothing function<br>FFff: True Full Chg E. True full charge energy from IT simulation without the effects of any smoothing function<br>GGgg: T_sim. Temperature during the last simulation run<br>HHhh: T_ambient. Current assumed ambient temperature used by the IT algorithm for thermal modeling<br>IIii: Cell 1 RaScale. Ra table scaling factor of Cell 1<br>JJjj: Cell 1 CompRes. Last temperature compensated resistance of Cell 1 |

### 13.8.45 *AltManufacturerAccess()* 0x0074 ITStatus2

This command instructs the device to return Impedance Track related gauging information on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0074 to *AltManufacturerAccess()* | Outputs 20 bytes of IT data values on *MACData()* in the following format: AABBCCDDEEFFggGGhhHHiiIIjjJJkkKKllLLmmMM where:<br>AA:<br>BB: LStatus—Learned status of resistance table<br>Bit 3 \| Bit 2 \| Bit 1\| Bit 0<br>QMax \| ITEN \| CF1 \| CF0<br>CF1, CF0: QMax Status<br>0,0 = Battery OK<br>0,1 = QMax is first updated in the learning cycle.<br>1,0 = QMax and resistance table updated in the learning cycle<br>ITEN: IT enable<br>0 = IT disabled<br>1 = IT enabled<br>QMax: QMax field updates<br>0 = QMax is not updated in the field.<br>1= QMax is updated in the field.<br>CC: Cell Grid 0. Active grid point of Cell 1<br>DD:<br>EE:<br>FF:<br>GGggHHhh: State Time. Time passed since last state change (DISCHARGE, CHARGE, RELAX)<br>IIii: DOD0_0. Depth of discharge for Cell 1<br>JJjj: DOD0 Passed Q. Passed capacity since the last DOD0 update<br>KKkk: DOD0 Passed E. Passed energy since last DOD0 update<br>LLll: DOD0 Time. Time passed since the last DOD0 update<br>MMmm: Cell 1 DODEOC. Depth of discharge at valid charge termination for Cell 1 |

### 13.8.46  AltManufacturerAccess() 0x0075 ITStatus3

This command instructs the device to return Impedance Track related gauging information on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0075 to *AltManufacturerAccess()* | Outputs 14 bytes of IT data values on *MACData()* in the following format: aaAAbbBBccCCddDDeeEEffFFggGG where:<br>AAaa: QMax 0. QMax of Cell 1<br>BBbb: QMax DOD0 0. DOD0 at last QMax update of Cell 1<br>CCcc: QMax Passed Q. Pass capacity since last QMax update<br>DDdd: QMax Time. Time passed since last QMax update<br>EEee: Temp k. Thermal model temperature factor<br>FFff: Temp a. Thermal model temperature time constant<br>GGgg: Raw DOD0 1. Raw DOD0 of Cell 1 |

### 13.8.47  AltManufacturerAccess() 0x0077 FCC_SOH

This command returns the capacity and energy results of the IT simulation for state-of-health.

| Status | Condition | Action |
|---|---|---|
| Enable | 0x0077 to *AltManufacturerAccess()* | Returns the state-of-health capacities (4 bytes) on a subsequent read on *MACData()* in the following format: aaAAbbBB where:<br>AAaa: SOH FC@25C Q. State-of-health simulated at 25°C in mAh<br>BBbb: SOH FC@25C E. State-of-health simulated at 25°C in cWh |

### 13.8.48  AltManufacturerAccess() 0x0078 Filtered Capacity

This command instructs the device to return Impedance Track related gauging information on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x0078 to *AltManufacturerAccess()* | Outputs 8 bytes of IT data values on *MACData()* in the following format: AAaaBBbbCCccDDdd where:<br>AAaa: Flt Rem Q. Filtered remaining capacity (mAh)<br>BBbb: Flt Rem E. Filtered remaining capacity (cWh)<br>CCcc: Flt FCC Q. Filtered full charge capacity (mAh)<br>DDdd: Flt FCC E. Filtered full charge capacity (cWh) |

### 13.8.49  *AltManufacturerAccess() 0x007A ManufacturerInfoB*

This command reads or writes the data flash values of **Manufacturer Info Block B01** through **Manufacturer Info Block B32** on *MACData()*.

| Status | Condition | Action |
|---|---|---|
| Activate | 0x007A to *AltManufacturerAccess()* | Outputs 32 bytes of Manufacturer Info on *MACData()* in the following format: AABBCCDDEEFFGGHHIIJJKKLLMMNN OOPPQQRRSSTTUUVVWWXXVVZZ112233445566 |

| Class | Subclass | Name | Type | Min | Max | Default | Unit |
|---|---|---|---|---|---|---|---|
| System Data | Manufacturer Info B | Manufacturer Info Block B01 | H1 | 0x00 | 0xFF | 0x61 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B02 | H1 | 0x00 | 0xFF | 0x62 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B03 | H1 | 0x00 | 0xFF | 0x63 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B04 | H1 | 0x00 | 0xFF | 0x64 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B05 | H1 | 0x00 | 0xFF | 0x65 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B06 | H1 | 0x00 | 0xFF | 0x66 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B07 | H1 | 0x00 | 0xFF | 0x67 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B08 | H1 | 0x00 | 0xFF | 0x68 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B09 | H1 | 0x00 | 0xFF | 0x69 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B10 | H1 | 0x00 | 0xFF | 0x6A | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B11 | H1 | 0x00 | 0xFF | 0x6B | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B12 | H1 | 0x00 | 0xFF | 0x6C | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B13 | H1 | 0x00 | 0xFF | 0x6D | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B14 | H1 | 0x00 | 0xFF | 0x6E | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B15 | H1 | 0x00 | 0xFF | 0x6F | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B16 | H1 | 0x00 | 0xFF | 0x70 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B17 | H1 | 0x00 | 0xFF | 0x71 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B18 | H1 | 0x00 | 0xFF | 0x72 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B19 | H1 | 0x00 | 0xFF | 0x73 | — |

| Class | Subclass | Name | Type | Min | Max | Default | Unit |
|-------|----------|------|------|-----|-----|---------|------|
| System Data | Manufacturer Info B | Manufacturer Info Block B20 | H1 | 0x00 | 0xFF | 0x74 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B21 | H1 | 0x00 | 0xFF | 0x75 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B22 | H1 | 0x00 | 0xFF | 0x76 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B23 | H1 | 0x00 | 0xFF | 0x77 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B24 | H1 | 0x00 | 0xFF | 0x7A | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B25 | H1 | 0x00 | 0xFF | 0x78 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B26 | H1 | 0x00 | 0xFF | 0x79 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B27 | H1 | 0x00 | 0xFF | 0x30 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B28 | H1 | 0x00 | 0xFF | 0x31 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B29 | H1 | 0x00 | 0xFF | 0x32 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B30 | H1 | 0x00 | 0xFF | 0x33 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B31 | H1 | 0x00 | 0xFF | 0x34 | — |
| System Data | Manufacturer Info B | Manufacturer Info Block B32 | H1 | 0x00 | 0xFF | 0x35 | — |

### 13.8.50  AltManufacturerAccess() 0x007B ManufacturerInfoC

This command reads or writes the data flash values of **Manufacturer Info Block C01** through **Manufacturer Info Block C32** on *MACData()*. Data is organized with default values such as, for example, Info Block B in its table.

### 13.8.51  ManufacturerAccess() 0x00B0 ChargingVoltageOverride

This command enables writing the five advanced charge algorithm charging voltage values in SEALED mode to data flash in the following format: aaAAbbBBccCCddDDeeEE where:

| Value | Description | Unit |
|-------|-------------|------|
| AAaa | Low Temperature Charging Voltage | mV |
| BBbb | Standard Temperature Low Charging Voltage | mV |
| CCcc | Standard Temperature High Charging Voltage | mV |
| DDdd | High Temperature Charging Voltage | mV |
| EEee | Recommended Temperature Charging Voltage | mV |

### 13.8.52  AltManufacturerAccess() 0x0F00 ROM Mode

This command sends the device into ROM mode in preparation for reprogramming.

| Status | Condition | Action |
|--------|-----------|--------|
| ROM Mode | *OperationStatusA()[SEC1,SEC0]* = 0,1 AND 0x0F00 to *AltManufacturerAccess()* | The device goes into ROM mode to prepare for updates. The ROM command 0x08 returns it to FIRMWARE mode. (Note: ROM commands are sent to address 0x16 using the I²C protocol.) |

### 13.8.53  Data Flash Access() 0x4000–0x5FFF

Accessing data flash (DF) is only supported by the *AltManufacturerAccess()* by addressing the physical address.

To write to the DF, send the starting address, followed by the DF data block. The DF data block is the intended revised DF data to be updated. The size of the DF data block ranges from 1 byte to 32 bytes. All individual data must be sent in little endian.

Write to DF example:

Assuming: data1 locates at address 0x4000 and data2 locates at address 0x4002.
           Both data1 and data2 are U2 type.

To update data1 and data2, send a block write with command = 0x3E
    block =  starting address + DF data block
        =  0x00 + 0x40 + data1_LowByte + data1_HighByte + data2_LowByte + data2_HighByte

To read the DF, send a block write to the *AltManufacturerAccess()* followed by the starting address, then send a block read to the *AltManufacturerAccess()*. The return data contains the starting address followed by 32 bytes of DF data in little endian.

Read from DF example:
Taking the same assuming from the read DF example, to read DF:
a. Send the write block with command 0x3E, block = 0x00 + 0x40
b. Send the read block with command 0x3E
    The returned block  = a starting address + 32 bytes of DF data
        = 0x00 + 0x40 + data1_LowByte + data1_HighByte + data2_LowByte + data2_HighByte....
        data32_LowByte + data32_HighByte

The gauge supports an auto-increment on the address during a DF read. This greatly reduces the time required to read out the entire DF. To trigger auto-increment on the address during DF reads, follow the example above, but read from *AltManufacturerAccess()* through *MACDataLength()* for a total of 36 bytes. If another read block is performed starting from *AltManufacturerAccess()*, the gauge returns another 32 bytes of DF data, starting with address 0x4020.

### 13.8.54  AltManufacturerAccess() 0x7C40 Switch to HDQ

This command PERMANENTLY switches the device's communication protocol from I$^2$C to HDQ.

| Status | Condition | Action |
|---|---|---|
| Activate | *OperationStatusA()[SEC1,SEC0]* = 0,1 or 1,0 AND 0x7C40 to *AltManufacturerAccess()* | The device permanently switches the communication protocol to HDQ (includes ROM mode). |

> **NOTE:**  After switching to HDQ, **I2C Configuration[XL]** has no effect on gauge operation.

### 13.8.55  AltManufacturerAccess() 0xF080 Exit Calibration Output Mode

This command stops the output of calibration data to the *MACData()* command.

| Status | Condition | Action |
|---|---|---|
| Activate | *MACData()* = 1 AND 0xF080 to *AltManufacturerAccess()* | Stops output of ADC or CC data on *MACData()* |

### 13.8.56 *AltManufacturerAccess() 0xF081 Output CC and ADC for Calibration*

This command instructs the device to output the raw values for calibration purposes on *MACData()*. All values are updated every 1000 ms and the format of each value is 2's complement, MSB first.

| Status | Condition | Action |
|---|---|---|
| Disable | *OperationStatusB()[CAL]* = 1 AND 0xF080 to *AltManufacturerAccess()* | *OperationStatusB()[CAL]* = 0<br>Stops output of ADC and CC data on *MACData()* |
| Enable | 0xF081 to *AltManufacturerAccess()* | *OperationStatusB()[CAL]* = 1<br>Outputs the raw CC and AD values on *MACData()* in the format of ZZYYaaAAbbBBccCCddDDeeEEffFF ggGGhhHHiiIIjjJJkkKK:<br>ZZ: Rolling 8-bit counter, increments when values are refreshed.<br>YY: Status, 1 when *AltManufacturerAccess()* = 0xF081<br>AAaa: Current (coulomb counter)<br>BBaa: Cell 1 Voltage<br>CCaa:<br>DDaa:<br>EEee:<br>FFff:<br>GGgg:<br>HHhh: Cell 1 Current<br>IIii:<br>JJjj:<br>KKkk: |

# Data Flash Access and Format

## 14.1 Data Flash Access

### 14.1.1 Minimum Voltage

Data flash can only be updated when the cell voltage is above the **Valid Update Voltage**.

| Class | Subclass | Name | Type | Min | Max | Default | Unit | Description |
|-------|----------|------|------|-----|-----|---------|------|-------------|
| Power | Power | Valid Update Voltage | I2 | 0 | 32767 | 2800 | mV | Min cell voltage threshold for the flash update |

## 14.2 Data Formats

### 14.2.1 Unsigned Integer

Unsigned integers are stored without changes as 1-byte, 2-byte, or 4-byte values in little endian byte order.

```
     0
+----------+
| U1       |
| MSB      |
+----------+
```

```
     0          1
+----------+----------+
| U2       | U2       |
| LSB      | MSB      |
+----------+----------+
```

```
     0          1          2          3
+----------+----------+----------+----------+
| U4 L     | U4 L     | U4 H     | U4 H     |
| LSB      | MSB      | LSB      | MSB      |
+----------+----------+----------+----------+
```
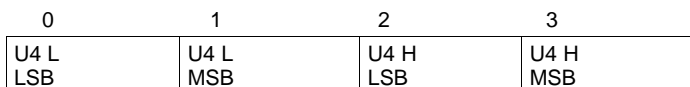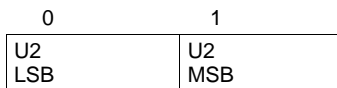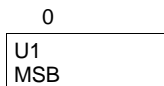
### 14.2.2 Integer

Integer values are stored in 2's-complement format in 1-byte, 2-byte, or 4-byte values in little endian byte order.

```
     0
+----------+
| I1       |
| MSB      |
+----------+
```

```
     0          1
+----------+----------+
| I2       | I2       |
| LSB      | MSB      |
+----------+----------+
```

```
     0          1          2          3
+----------+----------+----------+----------+
| I4 L     | I4 L     | I4 H     | I4 H     |
| LSB      | MSB      | LSB      | MSB      |
+----------+----------+----------+----------+
```
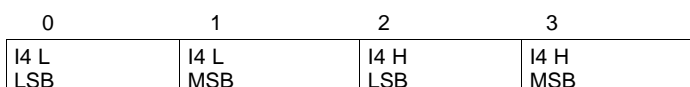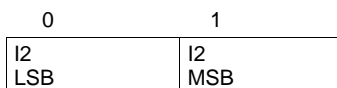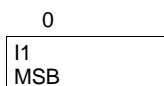
### 14.2.3  Floating Point

Floating point values are stored using the IEEE 754 single-precision, 4-byte format in little endian byte order.

| 0 | 1 | 2 | 3 |
|---|---|---|---|
| Fract [0–7] | Fract [8–15] | Exp[0] + Fract[16–22] | Sign + Exp[1–7] |

Where:

Exp: 8-bit exponent stored with an offset bias of 127. The values 00 and FF have special meaning.

Fract: 23-bit fraction. If the exponent is > 0, then the mantissa is 1.fract. If the exponent is 0, then the mantissa is 0.fract.

The floating point value depends on the special cases of the exponent:

- If the exponent is FF and the fraction is 0, this represents ± infinity.
- If the exponent is FF and the fraction is non–0, this represents "not a number" (NaN).
- If the exponent is 00, then the value is a subnormal number represented by $(-1)^{sign} \times 2^{-126} \times 0.\text{fraction}$.
- Otherwise, the value is a normalized number represented by $(-1)^{sign} \times 2^{(exponent - 127)} \times 1.\text{fraction}$.

### 14.2.4  Hex

Bit register definitions are stored in unsigned integer format.

### 14.2.5  String

String values are stored with the length byte first, followed by a number of data bytes defined with the length byte.

| 0 | 1 | . . . | N |
|---|---|---|---|
| Length | Data0 | . . . | DataN |

# *Data Flash Summary*

## 15.1 Data Flash Table

**Table 15-1. Data Flash Table**

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| Calibration | Voltage | 0x4000 | Cell Gain | I2 | −32767 | 32767 | 12101 | — |
| Calibration | Current | 0x4006 | CC Gain | F4 | 1.00E−01 | 4.00E+00 | 3.68 | — |
| Calibration | Current | 0x400A | Capacity Gain | F4 | 2.98262E+02 | 1.193046E+08 | 4390411.01 | — |
| Calibration | Temperature | 0x4014 | Internal Temp Offset | I1 | −128 | 127 | 0 | 0.1°C |
| Calibration | Temperature | 0x4015 | External1 Temp Offset | I1 | −128 | 127 | 0 | 0.1°C |
| Calibration | Internal Temp Model | 0x4440 | Int Gain | I2 | −32768 | 32767 | −13908 | — |
| Calibration | Internal Temp Model | 0x4442 | Int base offset | I2 | −32768 | 32767 | 6959 | — |
| Calibration | Internal Temp Model | 0x4444 | Int Minimum AD | I2 | −32768 | 32767 | 0 | — |
| Calibration | Internal Temp Model | 0x4446 | Int Maximum Temp | I2 | −32768 | 32767 | 6959 | 0.1°K |
| Calibration | Cell Temperature Model | 0x4448 | Coeff a1 | I2 | −32768 | 32767 | −17447 | — |
| Calibration | Cell Temperature Model | 0x444A | Coeff a2 | I2 | −32768 | 32767 | 29322 | — |
| Calibration | Cell Temperature Model | 0x444C | Coeff a3 | I2 | −32768 | 32767 | −25430 | — |
| Calibration | Cell Temperature Model | 0x444E | Coeff a4 | I2 | −32768 | 32767 | 29836 | — |
| Calibration | Cell Temperature Model | 0x4450 | Coeff a5 | I2 | −32768 | 32767 | 1200 | — |
| Calibration | Cell Temperature Model | 0x4452 | Coeff b1 | I2 | −32768 | 32767 | −293 | — |
| Calibration | Cell Temperature Model | 0x4454 | Coeff b2 | I2 | −32768 | 32767 | 552 | — |
| Calibration | Cell Temperature Model | 0x4456 | Coeff b3 | I2 | −32768 | 32767 | −2887 | — |
| Calibration | Cell Temperature Model | 0x4458 | Coeff b4 | I2 | −32768 | 32767 | 4591 | — |
| Calibration | Cell Temperature Model | 0x445A | Rc0 | I2 | −32768 | 32767 | 11703 | — |
| Calibration | Cell Temperature Model | 0x445C | Adc0 | I2 | −32768 | 32767 | 11703 | — |
| Calibration | Cell Temperature Model | 0x445E | Rpad | I2 | −32768 | 32767 | 0 | — |
| Calibration | Cell Temperature Model | 0x4460 | Rint | I2 | −32768 | 32767 | 0 | — |
| Calibration | Current Deadband | 0x4465 | Deadband | U1 | 0 | 255 | 3 | mA |
| Calibration | Current Deadband | 0x4466 | Coulomb Counter Deadband | U1 | 0 | 255 | 9 | 116 nV |
| Settings | Configuration | 0x4640 | I2C Gauging Configuration | H1 | 0x0 | 0x3F | 0x20 | Hex |
| Settings | Configuration | 0x4641 | I2C Configuration | H1 | 0x0 | 0xFF | 0x08 | Hex |
| Settings | Configuration | 0x4642 | Auth Config | H1 | 0x0 | 0x03 | 0x00 | Hex |

**Table 15-1. Data Flash Table (continued)**

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| Settings | Configuration | 0x4643 | Power Config | H1 | 0x0 | 0xFF | 0x10 | Hex |
| Settings | Configuration | 0x4644 | IO Config | H1 | 0x0 | 0xFF | 0x00 | Hex |
| Settings | Configuration | 0x464B | Init Voltage High Set | I2 | 0 | 5000 | 4500 | mV |
| Settings | Configuration | 0x464D | Init Voltage High Clear | I2 | 0 | 5000 | 4400 | mV |
| Settings | Configuration | 0x464F | Init Voltage Low Set | I2 | 0 | 5000 | 2500 | mV |
| Settings | Configuration | 0x4651 | Init Voltage Low Clear | I2 | 0 | 5000 | 2600 | mV |
| Settings | Configuration | 0x4653 | Init Temperature High Set | I1 | −128 | 127 | 60 | °C |
| Settings | Configuration | 0x4654 | Init Temperature High Clear | I1 | −128 | 127 | 55 | °C |
| Settings | Configuration | 0x4655 | Init Temperature Low Set | I1 | −128 | 127 | 0 | °C |
| Settings | Configuration | 0x4656 | Init Temperature Low Clear | I1 | −128 | 127 | 5 | °C |
| Settings | Configuration | 0x4657 | SOC Delta | U1 | 0 | 100 | 1 | % |
| Settings | Configuration | 0x4658 | GPIO Pulse Width | U1 | 0 | 255 | 7 | 25 ms |
| Settings | Configuration | 0x466E | SOC Flag Config A | H2 | 0x0 | 0x0FFF | 0x0C8C | Hex |
| Settings | Configuration | 0x4670 | SOC Flag Config B | H1 | 0x0 | 0xFF | 0x8C | Hex |
| Settings | Configuration | 0x4689 | IT Gauging Configuration | H2 | 0x0 | 0xFFFF | 0xF5FE | Hex |
| Settings | Configuration | 0x468D | IT Gauging Ext | H2 | 0x0000 | 0x003F | 0x007B | Hex |
| Settings | Configuration | 0x46A6 | Charging Configuration | H2 | 0x0 | 0xFFFF | 0x00 | Hex |
| Settings | Configuration | 0x471D | Temperature Enable | H1 | 0x0 | 0x03 | 0x02 | Hex |
| Settings | Configuration | 0x471E | DA Configuration | H2 | 0x0 | 0x10 | 0x0010 | Hex |
| Settings | BTP | 0x4645 | Init Discharge Set | I2 | 0 | 32767 | 150 | mAh |
| Settings | BTP | 0x4647 | Init Charge Set | I2 | 0 | 32767 | 175 | mAh |
| Settings | BTP | 0x4649 | Init Discharge Set RSOC | U1 | 0 | 100 | 10 | % |
| Settings | BTP | 0x464A | Init Charge Set RSOC | U1 | 0 | 100 | 15 | % |
| Settings | Lifetimes | 0x4729 | Lifetimes Configuration | H2 | 0x0 | 0x00FF | 0x0000 | Hex |
| Settings | Lifetimes | 0x472C | Time RSOC Threshold A | U1 | 0 | 100 | 95 | % |
| Settings | Lifetimes | 0x472D | Time RSOC Threshold B | U1 | 0 | 100 | 90 | % |
| Settings | Lifetimes | 0x472E | Time RSOC Threshold C | U1 | 0 | 100 | 80 | % |
| Settings | Lifetimes | 0x472F | Time RSOC Threshold D | U1 | 0 | 100 | 50 | % |
| Settings | Lifetimes | 0x4730 | Time RSOC Threshold E | U1 | 0 | 100 | 20 | % |
| Settings | Lifetimes | 0x4731 | Time RSOC Threshold F | U1 | 0 | 100 | 10 | % |
| Settings | Lifetimes | 0x4732 | Time RSOC Threshold G | U1 | 0 | 100 | 5 | % |
| Settings | Manufacturing | 0x440C | Mfg Status init | H2 | 0x0 | 0xFFFF | 0x0000 | Hex |
| Advanced Charge Algorithm | Temperature Ranges | 0x46A8 | T1 Temp | I1 | −128 | 127 | 0 | °C |
| Advanced Charge Algorithm | Temperature Ranges | 0x46A9 | T2 Temp | I1 | −128 | 127 | 10 | °C |
| Advanced Charge Algorithm | Temperature Ranges | 0x46AA | T5 Temp | I1 | −128 | 127 | 20 | °C |
| Advanced Charge Algorithm | Temperature Ranges | 0x46AB | T6 Temp | I1 | −128 | 127 | 25 | °C |
| Advanced Charge Algorithm | Temperature Ranges | 0x46AC | T3 Temp | I1 | −128 | 127 | 45 | °C |
| Advanced Charge Algorithm | Temperature Ranges | 0x46AD | T4 Temp | I1 | −128 | 127 | 55 | °C |
| Advanced Charge Algorithm | Temperature Ranges | 0x46AE | Hysteresis Temp | I1 | 0 | 127 | 1 | °C |
| Advanced Charge Algorithm | Low Temp Charging | 0x4382 | Voltage | I2 | 0 | 32767 | 4400 | mV |
| Advanced Charge Algorithm | Low Temp Charging | 0x46AF | Current Low | I2 | 0 | 32767 | 132 | mA |
| Advanced Charge Algorithm | Low Temp Charging | 0x46B1 | Current Med | I2 | 0 | 32767 | 352 | mA |
| Advanced Charge Algorithm | Low Temp Charging | 0x46B3 | Current High | I2 | 0 | 32767 | 264 | mA |

## Table 15-1. Data Flash Table (continued)

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| Advanced Charge Algorithm | Standard Temp Low Charging | 0x4384 | Voltage | I2 | 0 | 32767 | 4400 | mV |
| Advanced Charge Algorithm | Standard Temp Low Charging | 0x46B5 | Current Low | I2 | 0 | 32767 | 1980 | mA |
| Advanced Charge Algorithm | Standard Temp Low Charging | 0x46B7 | Current Med | I2 | 0 | 32767 | 4004 | mA |
| Advanced Charge Algorithm | Standard Temp Low Charging | 0x46B9 | Current High | I2 | 0 | 32767 | 2992 | mA |
| Advanced Charge Algorithm | Standard Temp High Charging | 0x4386 | Voltage | I2 | 0 | 32767 | 4400 | mV |
| Advanced Charge Algorithm | Standard Temp High Charging | 0x46BB | Current Low | I2 | 0 | 32767 | 1980 | mA |
| Advanced Charge Algorithm | Standard Temp High Charging | 0x46BD | Current Med | I2 | 0 | 32767 | 4004 | mA |
| Advanced Charge Algorithm | Standard Temp High Charging | 0x46BF | Current High | I2 | 0 | 32767 | 2992 | mA |
| Advanced Charge Algorithm | High Temp Charging | 0x4388 | Voltage | I2 | 0 | 32767 | 4100 | mV |
| Advanced Charge Algorithm | High Temp Charging | 0x46C1 | Current Low | I2 | 0 | 32767 | 1012 | mA |
| Advanced Charge Algorithm | High Temp Charging | 0x46C3 | Current Med | I2 | 0 | 32767 | 1980 | mA |
| Advanced Charge Algorithm | High Temp Charging | 0x46C5 | Current High | I2 | 0 | 32767 | 1496 | mA |
| Advanced Charge Algorithm | Rec Temp Charging | 0x438A | Voltage | I2 | 0 | 32767 | 4400 | mV |
| Advanced Charge Algorithm | Rec Temp Charging | 0x46C7 | Current Low | I2 | 0 | 32767 | 2508 | mA |
| Advanced Charge Algorithm | Rec Temp Charging | 0x46C9 | Current Med | I2 | 0 | 32767 | 4488 | mA |
| Advanced Charge Algorithm | Rec Temp Charging | 0x46CB | Current High | I2 | 0 | 32767 | 3520 | mA |
| Advanced Charge Algorithm | Pre-Charging | 0x46CD | Current | I2 | 0 | 32767 | 88 | mA |
| Advanced Charge Algorithm | Maintenance Charging | 0x46CF | Current | I2 | 0 | 32767 | 44 | mA |
| Advanced Charge Algorithm | Voltage Range | 0x46D1 | Precharge Start Voltage | I2 | 0 | 32767 | 2500 | mV |
| Advanced Charge Algorithm | Voltage Range | 0x46D3 | Charging Voltage Low | I2 | 0 | 32767 | 2900 | mV |
| Advanced Charge Algorithm | Voltage Range | 0x46D5 | Charging Voltage Med | I2 | 0 | 32767 | 3600 | mV |
| Advanced Charge Algorithm | Voltage Range | 0x46D7 | Charging Voltage High | I2 | 0 | 32767 | 4000 | mV |
| Advanced Charge Algorithm | Voltage Range | 0x46D9 | Charging Voltage Hysteresis | U1 | 0 | 255 | 0 | mV |
| Advanced Charge Algorithm | SoC Range | 0x46DA | Charging SoC Med | U1 | 0 | 100 | 50 | % |
| Advanced Charge Algorithm | SoC Range | 0x46DB | Charging SoC High | U1 | 0 | 100 | 75 | % |
| Advanced Charge Algorithm | SoC Range | 0x46DC | Charging SoC Hysteresis | U1 | 0 | 100 | 1 | % |
| Advanced Charge Algorithm | Degrade Mode 1 | 0x46DD | Cycle Threshold | U2 | 0 | 65535 | 50 | — |
| Advanced Charge Algorithm | Degrade Mode 1 | 0x46DF | SOH Threshold | U1 | 0 | 100 | 95 | % |
| Advanced Charge Algorithm | Degrade Mode 1 | 0x46E0 | Runtime Threshold | U2 | 0 | 65535 | 8760 | h |
| Advanced Charge Algorithm | Degrade Mode 1 | 0x46E2 | High SOC Threshold | U2 | 0 | 65535 | 8760 | h |

**Table 15-1. Data Flash Table (continued)**

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| Advanced Charge Algorithm | Degrade Mode 1 | 0x46E4 | High Temp Threshold | U2 | 0 | 65535 | 8760 | h |
| Advanced Charge Algorithm | Degrade Mode 1 | 0x46E6 | High Temp at SOC Threshold | U2 | 0 | 65535 | 8760 | h |
| Advanced Charge Algorithm | Degrade Mode 1 | 0x46E8 | Voltage Degradation | I2 | 0 | 32767 | 10 | mV |
| Advanced Charge Algorithm | Degrade Mode 1 | 0x46EA | Current Degradation | U1 | 0 | 100 | 10 | % |
| Advanced Charge Algorithm | Degrade Mode 2 | 0x46EB | Cycle Threshold | U2 | 0 | 65535 | 150 | — |
| Advanced Charge Algorithm | Degrade Mode 2 | 0x46ED | SOH Threshold | U1 | 0 | 100 | 80 | % |
| Advanced Charge Algorithm | Degrade Mode 2 | 0x46EE | Runtime Threshold | U2 | 0 | 65535 | 17520 | h |
| Advanced Charge Algorithm | Degrade Mode 2 | 0x46F0 | High SOC Threshold | U2 | 0 | 65535 | 17520 | h |
| Advanced Charge Algorithm | Degrade Mode 2 | 0x46F2 | High Temp Threshold | U2 | 0 | 65535 | 17520 | h |
| Advanced Charge Algorithm | Degrade Mode 2 | 0x46F4 | High Temp at SOC Threshold | U2 | 0 | 65535 | 17520 | h |
| Advanced Charge Algorithm | Degrade Mode 2 | 0x46F6 | Voltage Degradation | U2 | 0 | 65535 | 150 | mV |
| Advanced Charge Algorithm | Degrade Mode 2 | 0x46F8 | Current Degradation | U1 | 0 | 100 | 20 | % |
| Advanced Charge Algorithm | Degrade Mode 3 | 0x46F9 | Cycle Threshold | U2 | 0 | 65535 | 350 | — |
| Advanced Charge Algorithm | Degrade Mode 3 | 0x46FB | SOH Threshold | U1 | 0 | 100 | 60 | % |
| Advanced Charge Algorithm | Degrade Mode 3 | 0x46FC | Runtime Threshold | U2 | 0 | 65535 | 26280 | h |
| Advanced Charge Algorithm | Degrade Mode 3 | 0x46FE | High SOC Threshold | U2 | 0 | 65535 | 26280 | h |
| Advanced Charge Algorithm | Degrade Mode 3 | 0x4700 | High Temp Threshold | U2 | 0 | 65535 | 26280 | h |
| Advanced Charge Algorithm | Degrade Mode 3 | 0x4702 | High Temp at SOC Threshold | U2 | 0 | 65535 | 26280 | h |
| Advanced Charge Algorithm | Degrade Mode 3 | 0x4704 | Voltage Degradation | U2 | 0 | 65535 | 250 | mV |
| Advanced Charge Algorithm | Degrade Mode 3 | 0x4706 | Current Degradation | U1 | 0 | 100 | 40 | % |
| Advanced Charge Algorithm | Degrade Mode | 0x4200 | Runtime Degrade | U2 | 0 | 65535 | 0 | hours |
| Advanced Charge Algorithm | Degrade Mode | 0x4707 | Cycle Count Start Runtime | U1 | 0 | 255 | 1 | — |
| Advanced Charge Algorithm | Degrade Mode | 0x4708 | Runtime Update Interval | U1 | 0 | 18 | 10 | h |
| Advanced Charge Algorithm | CS Degrade | 0x4709 | Temperature Threshold | I2 | 0 | 32767 | 3232 | 0.1°K |
| Advanced Charge Algorithm | CS Degrade | 0x470B | Voltage Threshold | I2 | 0 | 32767 | 4200 | mV |
| Advanced Charge Algorithm | CS Degrade | 0x470D | Time Interval | U2 | 0 | 14400 | 300 | s |
| Advanced Charge Algorithm | CS Degrade | 0x470F | Delta Voltage | I2 | 0 | 32767 | 25 | mV |
| Advanced Charge Algorithm | CS Degrade | 0x4711 | Min CV | I2 | 0 | 32767 | 3000 | mV |
| Advanced Charge Algorithm | Termination Config | 0x4713 | Charge Term Taper Current | I2 | 0 | 32767 | 100 | mA |
| Advanced Charge Algorithm | Termination Config | 0x4717 | Charge Term Voltage Offset | I2 | 0 | 32767 | 100 | mV |

## Table 15-1. Data Flash Table (continued)

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| Advanced Charge Algorithm | Termination Config | 0x4719 | Charge Term Charging Voltage | I2 | 0 | 32767 | 4400 | mV |
| Advanced Charge Algorithm | IR Correction | 0x471C | Averaging Interval | U1 | 1 | 255 | 12 | s |
| Gas Gauging | Current Thresholds | 0x4721 | Dsg Current Threshold | I2 | −32768 | 32767 | 60 | mA |
| Gas Gauging | Current Thresholds | 0x4723 | Chg Current Threshold | I2 | −32768 | 32767 | 75 | mA |
| Gas Gauging | Current Thresholds | 0x4725 | Quit Current | I2 | 0 | 32767 | 40 | mA |
| Gas Gauging | Current Thresholds | 0x4727 | Dsg Relax Time | U1 | 0 | 255 | 60 | s |
| Gas Gauging | Current Thresholds | 0x4728 | Chg Relax Time | U1 | 0 | 255 | 60 | s |
| Gas Gauging | Max Load | 0x4663 | Max Load Current | I2 | −32768 | 0 | −500 | mA |
| Gas Gauging | Max Load | 0x4665 | Max Load Rsoc | U1 | 0 | 100 | 50 | % |
| Gas Gauging | Design | 0x4666 | Design Capacity mAh | I2 | 0 | 32767 | 5300 | mAh |
| Gas Gauging | Design | 0x4668 | Design Capacity cWh | I2 | 0 | 32767 | 2040 | cWh |
| Gas Gauging | Design | 0x466A | Design Voltage | I2 | 0 | 32767 | 3850 | mV |
| Gas Gauging | Cycle | 0x466C | Cycle Count Percentage | U1 | 0 | 100 | 90 | % |
| Gas Gauging | FD | 0x4671 | Set Voltage Threshold | I2 | 0 | 5000 | 3000 | mV |
| Gas Gauging | FD | 0x4673 | Clear Voltage Threshold | I2 | 0 | 5000 | 3100 | mV |
| Gas Gauging | FD | 0x4675 | Set % RSOC Threshold | U1 | 0 | 100 | 0 | % |
| Gas Gauging | FD | 0x4676 | Clear % RSOC Threshold | U1 | 0 | 100 | 5 | % |
| Gas Gauging | FC | 0x4677 | Set Voltage Threshold | I2 | 0 | 5000 | 4400 | mV |
| Gas Gauging | FC | 0x4679 | Clear Voltage Threshold | I2 | 0 | 5000 | 4300 | mV |
| Gas Gauging | FC | 0x467B | Set % RSOC Threshold | U1 | 0 | 100 | 100 | % |
| Gas Gauging | FC | 0x467C | Clear % RSOC Threshold | U1 | 0 | 100 | 95 | % |
| Gas Gauging | TD | 0x467D | Set Voltage Threshold | I2 | 0 | 5000 | 3200 | mV |
| Gas Gauging | TD | 0x467F | Clear Voltage Threshold | I2 | 0 | 5000 | 3300 | mV |
| Gas Gauging | TD | 0x4681 | Set % RSOC Threshold | U1 | 0 | 100 | 6 | % |
| Gas Gauging | TD | 0x4682 | Clear % RSOC Threshold | U1 | 0 | 100 | 8 | % |
| Gas Gauging | TC | 0x4683 | Set Voltage Threshold | I2 | 0 | 5000 | 4400 | mV |
| Gas Gauging | TC | 0x4685 | Clear Voltage Threshold | I2 | 0 | 5000 | 4300 | mV |
| Gas Gauging | TC | 0x4687 | Set % RSOC Threshold | U1 | 0 | 100 | 100 | % |
| Gas Gauging | TC | 0x4688 | Clear % RSOC Threshold | U1 | 0 | 100 | 95 | % |
| Gas Gauging | State | 0x4186 | Qmax Cell 1 | I2 | 0 | 32767 | 5359 | mAh |
| Gas Gauging | State | 0x4188 | Qmax Day | U2 | 0 | 65535 | 0 | — |
| Gas Gauging | State | 0x418A | Qmax Cycle Count | U2 | 0 | 65535 | 0 | — |
| Gas Gauging | State | 0x418C | Update Status | H1 | 0x0 | 0x0E | 0x0 | — |
| Gas Gauging | State | 0x418D | Cell 1 Chg Voltage at EoC | I2 | 0 | 32767 | 4375 | mV |
| Gas Gauging | State | 0x418F | Current at EoC | I2 | 0 | 32767 | 100 | mA |
| Gas Gauging | State | 0x4191 | Avg I Last Run | I2 | −32768 | 32767 | −500 | mA |
| Gas Gauging | State | 0x4193 | Avg P Last Run | I2 | −32768 | 32767 | −192 | cW |
| Gas Gauging | State | 0x4195 | Delta Voltage | I2 | −32768 | 32767 | 0 | mV |
| Gas Gauging | State | 0x4197 | Temp k | I2 | 0 | 32767 | 200 | 0.1°C/256 cW |
| Gas Gauging | State | 0x4199 | Temp a | I2 | 0 | 32767 | 1000 | s |
| Gas Gauging | State | 0x419B | Max Avg I Last Run | I2 | −32768 | 32767 | −500 | mA |
| Gas Gauging | State | 0x419D | Max Avg P Last Run | I2 | −32768 | 32767 | −192 | cW |
| Gas Gauging | State | 0x41C0 | Cycle Count | U2 | 0 | 65535 | 0 | — |
| Gas Gauging | State | 0x463A | SOH Temp k | I2 | 0 | 32767 | 100 | 0.1°C/256 cW |
| Gas Gauging | State | 0x463C | SOH Temp a | I2 | 0 | 32767 | 1000 | s |

## Table 15-1. Data Flash Table (continued)

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| Gas Gauging | IT Cfg | 0x4180 | Design Resistance | I2 | 1 | 32767 | 90 | mΩ |
| Gas Gauging | IT Cfg | 0x4182 | Pack Resistance | I2 | 0 | 32767 | 0 | mΩ |
| Gas Gauging | IT Cfg | 0x4184 | System Resistance | I2 | 0 | 32767 | 0 | mΩ |
| Gas Gauging | IT Cfg | 0x45CE | Ra Filter | U2 | 0 | 999 | 800 | 0.1% |
| Gas Gauging | IT Cfg | 0x45D1 | Ra Max Delta | U1 | 0 | 255 | 15 | % |
| Gas Gauging | IT Cfg | 0x45D4 | Resistance Parameter Filter | U2 | 1 | 65535 | 65142 | — |
| Gas Gauging | IT Cfg | 0x45D6 | Near EDV Ra Param Filter | U2 | 1 | 65535 | 62184 | — |
| Gas Gauging | IT Cfg | 0x45D8 | Cell 1 Interconnect Resistance | I2 | 0 | 32767 | 0 | $2^{-10}$ Ω |
| Gas Gauging | IT Cfg | 0x45DA | Max Current Change % | U1 | 0 | 100 | 10 | % |
| Gas Gauging | IT Cfg | 0x45DB | Resistance Update Voltage | I2 | 0 | 32767 | 50 | mV |
| Gas Gauging | IT Cfg | 0x4601 | Qmax Delta | U1 | 3 | 100 | 5 | % |
| Gas Gauging | IT Cfg | 0x4602 | Qmax Upper Bound | U1 | 100 | 255 | 130 | % |
| Gas Gauging | IT Cfg | 0x4603 | Cycle Adjust Threshold | U1 | 0 | 255 | 20 | — |
| Gas Gauging | IT Cfg | 0x4604 | Cycle Adj Incr | U1 | 0 | 255 | 5 | 0.01% |
| Gas Gauging | IT Cfg | 0x4605 | OCV Pred Active T Limit | U2 | 100 | 65535 | 200 | s |
| Gas Gauging | IT Cfg | 0x4607 | OCV Pred Transient T | U2 | 100 | 65535 | 300 | s |
| Gas Gauging | IT Cfg | 0x4609 | OCV Pred Measure Time | U2 | 0 | 65535 | 200 | s |
| Gas Gauging | IT Cfg | 0x460B | Term Voltage | I2 | 0 | 32767 | 3000 | mV |
| Gas Gauging | IT Cfg | 0x460D | Term V Hold Time | U1 | 0 | 255 | 2 | s |
| Gas Gauging | IT Cfg | 0x460E | Term Voltage Delta | I2 | 0 | 32767 | 200 | mV |
| Gas Gauging | IT Cfg | 0x4619 | Max Simulation Iterations | U1 | 20 | 50 | 50 | — |
| Gas Gauging | IT Cfg | 0x461A | Simulation Near Term Delta | I2 | 0 | 32767 | 250 | mV |
| Gas Gauging | IT Cfg | 0x462C | Fast Scale Start SOC | U1 | 0 | 100 | 10 | % |
| Gas Gauging | IT Cfg | 0x4635 | Min Delta Voltage | I2 | −32768 | 32767 | 0 | mV |
| Gas Gauging | IT Cfg | 0x468F | Load Select | U1 | 0 | 7 | 1 | — |
| Gas Gauging | IT Cfg | 0x4690 | Load Mode | U1 | 0 | 1 | 1 | — |
| Gas Gauging | IT Cfg | 0x4691 | User Rate-mA | I2 | −9000 | 0 | 0 | mA |
| Gas Gauging | IT Cfg | 0x4693 | User Rate-cW | I2 | −32768 | 0 | 0 | cW |
| Gas Gauging | IT Cfg | 0x4695 | Reserve Cap-mAh | I2 | 0 | 9000 | 0 | mAh |
| Gas Gauging | IT Cfg | 0x4697 | Reserve Cap-cWh | I2 | 0 | 32000 | 0 | cWh |
| Gas Gauging | IT Cfg | 0x46A4 | Predict Ambient Time | U2 | 0 | 65535 | 2000 | s |
| Gas Gauging | Smoothing | 0x4699 | Smooth Relax Time | U2 | 1 | 32767 | 1000 | s |
| Gas Gauging | Smoothing | 0x469B | Term Smooth Start Cell V Delta | I2 | 0 | 32767 | 150 | mV |
| Gas Gauging | Smoothing | 0x469D | Term Smooth Final Cell V Delta | I2 | 0 | 32767 | 100 | mV |
| Gas Gauging | Smoothing | 0x469F | Term Smooth Time | U1 | 1 | 32767 | 20 | s |
| Gas Gauging | SoH | 0x4639 | SoH Load Rate | U1 | 0 | 255 | 50 | 0.1 Hr rate |
| Power | Power | 0x4659 | Valid Update Voltage | I2 | 0 | 32767 | 2800 | mV |
| Power | Sleep | 0x465B | Sleep Current | I2 | 0 | 32767 | 15 | mA |
| Power | Sleep | 0x465D | Voltage Time | U1 | 1 | 20 | 5 | s |
| Power | Sleep | 0x465E | Deep Sleep Voltage Time | U1 | 1 | 20 | 20 | s |
| Power | Sleep | 0x465F | Wake Check Time | U1 | 0 | 20 | 1 | s |
| Power | Sleep | 0x4660 | Deep Sleep Current | I2 | 0 | 32767 | 10 | mA |
| Power | Hibernate | 0x4662 | Auto Time | U1 | 0 | 20 | 2 | s |
| System Data | Manufacturer Data | 0x4041 | Manufacturer Info Block A01 | H1 | 0x0 | 0xFF | 0x61 | Hex |
| System Data | Manufacturer Data | 0x4042 | Manufacturer Info Block A02 | H1 | 0x0 | 0xFF | 0x62 | Hex |
| System Data | Manufacturer Data | 0x4043 | Manufacturer Info Block A03 | H1 | 0x0 | 0xFF | 0x63 | Hex |

## Table 15-1. Data Flash Table (continued)

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| System Data | Manufacturer Data | 0x4044 | Manufacturer Info Block A04 | H1 | 0x0 | 0xFF | 0x64 | Hex |
| System Data | Manufacturer Data | 0x4045 | Manufacturer Info Block A05 | H1 | 0x0 | 0xFF | 0x65 | Hex |
| System Data | Manufacturer Data | 0x4046 | Manufacturer Info Block A06 | H1 | 0x0 | 0xFF | 0x66 | Hex |
| System Data | Manufacturer Data | 0x4047 | Manufacturer Info Block A07 | H1 | 0x0 | 0xFF | 0x67 | Hex |
| System Data | Manufacturer Data | 0x4048 | Manufacturer Info Block A08 | H1 | 0x0 | 0xFF | 0x68 | Hex |
| System Data | Manufacturer Data | 0x4049 | Manufacturer Info Block A09 | H1 | 0x0 | 0xFF | 0x69 | Hex |
| System Data | Manufacturer Data | 0x404A | Manufacturer Info Block A10 | H1 | 0x0 | 0xFF | 0x6A | Hex |
| System Data | Manufacturer Data | 0x404B | Manufacturer Info Block A11 | H1 | 0x0 | 0xFF | 0x6B | Hex |
| System Data | Manufacturer Data | 0x404C | Manufacturer Info Block A12 | H1 | 0x0 | 0xFF | 0x6C | Hex |
| System Data | Manufacturer Data | 0x404D | Manufacturer Info Block A13 | H1 | 0x0 | 0xFF | 0x6D | Hex |
| System Data | Manufacturer Data | 0x404E | Manufacturer Info Block A14 | H1 | 0x0 | 0xFF | 0x6E | Hex |
| System Data | Manufacturer Data | 0x404F | Manufacturer Info Block A15 | H1 | 0x0 | 0xFF | 0x6F | Hex |
| System Data | Manufacturer Data | 0x4050 | Manufacturer Info Block A16 | H1 | 0x0 | 0xFF | 0x70 | Hex |
| System Data | Manufacturer Data | 0x4051 | Manufacturer Info Block A17 | H1 | 0x0 | 0xFF | 0x71 | Hex |
| System Data | Manufacturer Data | 0x4052 | Manufacturer Info Block A18 | H1 | 0x0 | 0xFF | 0x72 | Hex |
| System Data | Manufacturer Data | 0x4053 | Manufacturer Info Block A19 | H1 | 0x0 | 0xFF | 0x73 | Hex |
| System Data | Manufacturer Data | 0x4054 | Manufacturer Info Block A20 | H1 | 0x0 | 0xFF | 0x74 | Hex |
| System Data | Manufacturer Data | 0x4055 | Manufacturer Info Block A21 | H1 | 0x0 | 0xFF | 0x75 | Hex |
| System Data | Manufacturer Data | 0x4056 | Manufacturer Info Block A22 | H1 | 0x0 | 0xFF | 0x76 | Hex |
| System Data | Manufacturer Data | 0x4057 | Manufacturer Info Block A23 | H1 | 0x0 | 0xFF | 0x77 | Hex |
| System Data | Manufacturer Data | 0x4058 | Manufacturer Info Block A24 | H1 | 0x0 | 0xFF | 0x7A | Hex |
| System Data | Manufacturer Data | 0x4059 | Manufacturer Info Block A25 | H1 | 0x0 | 0xFF | 0x78 | Hex |
| System Data | Manufacturer Data | 0x405A | Manufacturer Info Block A26 | H1 | 0x0 | 0xFF | 0x79 | Hex |
| System Data | Manufacturer Data | 0x405B | Manufacturer Info Block A27 | H1 | 0x0 | 0xFF | 0x30 | Hex |
| System Data | Manufacturer Data | 0x405C | Manufacturer Info Block A28 | H1 | 0x0 | 0xFF | 0x31 | Hex |
| System Data | Manufacturer Data | 0x405D | Manufacturer Info Block A29 | H1 | 0x0 | 0xFF | 0x32 | Hex |
| System Data | Manufacturer Data | 0x405E | Manufacturer Info Block A30 | H1 | 0x0 | 0xFF | 0x33 | Hex |
| System Data | Manufacturer Data | 0x405F | Manufacturer Info Block A31 | H1 | 0x0 | 0xFF | 0x34 | Hex |
| System Data | Manufacturer Data | 0x4060 | Manufacturer Info Block A32 | H1 | 0x0 | 0xFF | 0x35 | Hex |
| System Data | Manufacturer Info B | 0x4062 | Manufacturer Info Block B01 | H1 | 0x0 | 0xFF | 0x61 | Hex |

**Table 15-1. Data Flash Table (continued)**

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| System Data | Manufacturer Info B | 0x4063 | Manufacturer Info Block B02 | H1 | 0x0 | 0xFF | 0x62 | Hex |
| System Data | Manufacturer Info B | 0x4064 | Manufacturer Info Block B03 | H1 | 0x0 | 0xFF | 0x63 | Hex |
| System Data | Manufacturer Info B | 0x4065 | Manufacturer Info Block B04 | H1 | 0x0 | 0xFF | 0x64 | Hex |
| System Data | Manufacturer Info B | 0x4066 | Manufacturer Info Block B05 | H1 | 0x0 | 0xFF | 0x65 | Hex |
| System Data | Manufacturer Info B | 0x4067 | Manufacturer Info Block B06 | H1 | 0x0 | 0xFF | 0x66 | Hex |
| System Data | Manufacturer Info B | 0x4068 | Manufacturer Info Block B07 | H1 | 0x0 | 0xFF | 0x67 | Hex |
| System Data | Manufacturer Info B | 0x4069 | Manufacturer Info Block B08 | H1 | 0x0 | 0xFF | 0x68 | Hex |
| System Data | Manufacturer Info B | 0x406A | Manufacturer Info Block B09 | H1 | 0x0 | 0xFF | 0x69 | Hex |
| System Data | Manufacturer Info B | 0x406B | Manufacturer Info Block B10 | H1 | 0x0 | 0xFF | 0x6A | Hex |
| System Data | Manufacturer Info B | 0x406C | Manufacturer Info Block B11 | H1 | 0x0 | 0xFF | 0x6B | Hex |
| System Data | Manufacturer Info B | 0x406D | Manufacturer Info Block B12 | H1 | 0x0 | 0xFF | 0x6C | Hex |
| System Data | Manufacturer Info B | 0x406E | Manufacturer Info Block B13 | H1 | 0x0 | 0xFF | 0x6D | Hex |
| System Data | Manufacturer Info B | 0x406F | Manufacturer Info Block B14 | H1 | 0x0 | 0xFF | 0x6E | Hex |
| System Data | Manufacturer Info B | 0x4070 | Manufacturer Info Block B15 | H1 | 0x0 | 0xFF | 0x6F | Hex |
| System Data | Manufacturer Info B | 0x4071 | Manufacturer Info Block B16 | H1 | 0x0 | 0xFF | 0x70 | Hex |
| System Data | Manufacturer Info B | 0x4072 | Manufacturer Info Block B17 | H1 | 0x0 | 0xFF | 0x71 | Hex |
| System Data | Manufacturer Info B | 0x4073 | Manufacturer Info Block B18 | H1 | 0x0 | 0xFF | 0x72 | Hex |
| System Data | Manufacturer Info B | 0x4074 | Manufacturer Info Block B19 | H1 | 0x0 | 0xFF | 0x73 | Hex |
| System Data | Manufacturer Info B | 0x4075 | Manufacturer Info Block B20 | H1 | 0x0 | 0xFF | 0x74 | Hex |
| System Data | Manufacturer Info B | 0x4076 | Manufacturer Info Block B21 | H1 | 0x0 | 0xFF | 0x75 | Hex |
| System Data | Manufacturer Info B | 0x4077 | Manufacturer Info Block B22 | H1 | 0x0 | 0xFF | 0x76 | Hex |
| System Data | Manufacturer Info B | 0x4078 | Manufacturer Info Block B23 | H1 | 0x0 | 0xFF | 0x77 | Hex |
| System Data | Manufacturer Info B | 0x4079 | Manufacturer Info Block B24 | H1 | 0x0 | 0xFF | 0x7A | Hex |
| System Data | Manufacturer Info B | 0x407A | Manufacturer Info Block B25 | H1 | 0x0 | 0xFF | 0x78 | Hex |
| System Data | Manufacturer Info B | 0x407B | Manufacturer Info Block B26 | H1 | 0x0 | 0xFF | 0x79 | Hex |
| System Data | Manufacturer Info B | 0x407C | Manufacturer Info Block B27 | H1 | 0x0 | 0xFF | 0x30 | Hex |
| System Data | Manufacturer Info B | 0x407D | Manufacturer Info Block B28 | H1 | 0x0 | 0xFF | 0x31 | Hex |
| System Data | Manufacturer Info B | 0x407E | Manufacturer Info Block B29 | H1 | 0x0 | 0xFF | 0x32 | Hex |
| System Data | Manufacturer Info B | 0x407F | Manufacturer Info Block B30 | H1 | 0x0 | 0xFF | 0x33 | Hex |
| System Data | Manufacturer Info B | 0x4080 | Manufacturer Info Block B31 | H1 | 0x0 | 0xFF | 0x34 | Hex |

## Table 15-1. Data Flash Table (continued)

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|-------|----------|---------|------|------|-----------|-----------|---------|-------|
| System Data | Manufacturer Info B | 0x4081 | Manufacturer Info Block B32 | H1 | 0x0 | 0xFF | 0x35 | Hex |
| System Data | Manufacturer Info C | 0x4083 | Manufacturer Info Block C01 | H1 | 0x0 | 0xFF | 0x61 | Hex |
| System Data | Manufacturer Info C | 0x4084 | Manufacturer Info Block C02 | H1 | 0x0 | 0xFF | 0x62 | Hex |
| System Data | Manufacturer Info C | 0x4085 | Manufacturer Info Block C03 | H1 | 0x0 | 0xFF | 0x63 | Hex |
| System Data | Manufacturer Info C | 0x4086 | Manufacturer Info Block C04 | H1 | 0x0 | 0xFF | 0x64 | Hex |
| System Data | Manufacturer Info C | 0x4087 | Manufacturer Info Block C05 | H1 | 0x0 | 0xFF | 0x65 | Hex |
| System Data | Manufacturer Info C | 0x4088 | Manufacturer Info Block C06 | H1 | 0x0 | 0xFF | 0x66 | Hex |
| System Data | Manufacturer Info C | 0x4089 | Manufacturer Info Block C07 | H1 | 0x0 | 0xFF | 0x67 | Hex |
| System Data | Manufacturer Info C | 0x408A | Manufacturer Info Block C08 | H1 | 0x0 | 0xFF | 0x68 | Hex |
| System Data | Manufacturer Info C | 0x408B | Manufacturer Info Block C09 | H1 | 0x0 | 0xFF | 0x69 | Hex |
| System Data | Manufacturer Info C | 0x408C | Manufacturer Info Block C10 | H1 | 0x0 | 0xFF | 0x6A | Hex |
| System Data | Manufacturer Info C | 0x408D | Manufacturer Info Block C11 | H1 | 0x0 | 0xFF | 0x6B | Hex |
| System Data | Manufacturer Info C | 0x408E | Manufacturer Info Block C12 | H1 | 0x0 | 0xFF | 0x6C | Hex |
| System Data | Manufacturer Info C | 0x408F | Manufacturer Info Block C13 | H1 | 0x0 | 0xFF | 0x6D | Hex |
| System Data | Manufacturer Info C | 0x4090 | Manufacturer Info Block C14 | H1 | 0x0 | 0xFF | 0x6E | Hex |
| System Data | Manufacturer Info C | 0x4091 | Manufacturer Info Block C15 | H1 | 0x0 | 0xFF | 0x6F | Hex |
| System Data | Manufacturer Info C | 0x4092 | Manufacturer Info Block C16 | H1 | 0x0 | 0xFF | 0x70 | Hex |
| System Data | Manufacturer Info C | 0x4093 | Manufacturer Info Block C17 | H1 | 0x0 | 0xFF | 0x71 | Hex |
| System Data | Manufacturer Info C | 0x4094 | Manufacturer Info Block C18 | H1 | 0x0 | 0xFF | 0x72 | Hex |
| System Data | Manufacturer Info C | 0x4095 | Manufacturer Info Block C19 | H1 | 0x0 | 0xFF | 0x73 | Hex |
| System Data | Manufacturer Info C | 0x4096 | Manufacturer Info Block C20 | H1 | 0x0 | 0xFF | 0x74 | Hex |
| System Data | Manufacturer Info C | 0x4097 | Manufacturer Info Block C21 | H1 | 0x0 | 0xFF | 0x75 | Hex |
| System Data | Manufacturer Info C | 0x4098 | Manufacturer Info Block C22 | H1 | 0x0 | 0xFF | 0x76 | Hex |
| System Data | Manufacturer Info C | 0x4099 | Manufacturer Info Block C23 | H1 | 0x0 | 0xFF | 0x77 | Hex |
| System Data | Manufacturer Info C | 0x409A | Manufacturer Info Block C24 | H1 | 0x0 | 0xFF | 0x7A | Hex |
| System Data | Manufacturer Info C | 0x409B | Manufacturer Info Block C25 | H1 | 0x0 | 0xFF | 0x78 | Hex |
| System Data | Manufacturer Info C | 0x409C | Manufacturer Info Block C26 | H1 | 0x0 | 0xFF | 0x79 | Hex |
| System Data | Manufacturer Info C | 0x409D | Manufacturer Info Block C27 | H1 | 0x0 | 0xFF | 0x30 | Hex |
| System Data | Manufacturer Info C | 0x409E | Manufacturer Info Block C28 | H1 | 0x0 | 0xFF | 0x31 | Hex |
| System Data | Manufacturer Info C | 0x409F | Manufacturer Info Block C29 | H1 | 0x0 | 0xFF | 0x32 | Hex |

**Table 15-1. Data Flash Table (continued)**

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| System Data | Manufacturer Info C | 0x40A0 | Manufacturer Info Clock C30 | H1 | 0x0 | 0xFF | 0x33 | Hex |
| System Data | Manufacturer Info C | 0x40A1 | Manufacturer Info Block C31 | H1 | 0x0 | 0xFF | 0x34 | Hex |
| System Data | Manufacturer Info C | 0x40A2 | Manufacturer Info Block C32 | H1 | 0x0 | 0xFF | 0x35 | Hex |
| System Data | Integrity | 0x40A3 | Static DF Signature | H2 | 0x0 | 0x7FFF | 0x0 | Hex |
| System Data | Integrity | 0x40A5 | Static Chem DF Signature | H2 | 0x0 | 0x7FFF | 0x3A4C | Hex |
| System Data | Integrity | 0x40A7 | All DF Signature | H2 | 0x0 | 0x7FFF | 0x0 | Hex |
| I2C Configuration | Data | 0x40A9 | Manufacture Date | U2 | 0 | 65535 | 0 | date |
| I2C Configuration | Data | 0x40AB | Serial Number | H2 | 0x0 | 0xFFFF | 0x0001 | Hex |
| I2C Configuration | Data | 0x40AD | Manufacturer Name | S21 | x | x | Texas Instruments | — |
| I2C Configuration | Data | 0x40C2 | Device Name | S21 | x | x | BQ27Z561 | — |
| I2C Configuration | Data | 0x40D7 | Device Chemistry | S5 | x | x | LION | — |
| Lifetimes | Voltage | 0x4202 | Cell 1 Max Voltage | I2 | 0 | 32767 | 0 | mV |
| Lifetimes | Voltage | 0x4204 | Cell 1 Min Voltage | I2 | 0 | 32767 | 32767 | mV |
| Lifetimes | Current | 0x4206 | Max Charge Current | I2 | 0 | 32767 | 0 | mA |
| Lifetimes | Current | 0x4208 | Max Discharge Current | I2 | −32768 | 0 | 0 | mA |
| Lifetimes | Current | 0x420A | Max Avg Dsg Current | I2 | −32768 | 0 | 0 | mA |
| Lifetimes | Current | 0x420C | Max Avg Dsg Power | I2 | −32768 | 0 | 0 | cW |
| Lifetimes | Temperature | 0x420E | Max Temp Cell | I1 | −128 | 127 | −128 | °C |
| Lifetimes | Temperature | 0x420F | Min Temp Cell | I1 | −128 | 127 | 127 | °C |
| Lifetimes | Temperature | 0x4210 | Max Temp Int Sensor | I1 | −128 | 127 | −128 | °C |
| Lifetimes | Temperature | 0x4211 | Min Temp Int Sensor | I1 | −128 | 127 | 127 | °C |
| Lifetimes | Charging Events | 0x4212 | No Valid Charge Term | U2 | 0 | 32767 | 0 | events |
| Lifetimes | Charging Events | 0x4214 | Last Valid Charge Term | U2 | 0 | 32767 | 0 | cycles |
| Lifetimes | Gauging Events | 0x4216 | No Of Qmax Updates | U2 | 0 | 32767 | 0 | events |
| Lifetimes | Gauging Events | 0x4218 | Last Qmax Update | U2 | 0 | 32767 | 0 | cycles |
| Lifetimes | Gauging Events | 0x421A | No Of Ra Updates | U2 | 0 | 32767 | 0 | events |
| Lifetimes | Gauging Events | 0x421C | Last Ra Update | U2 | 0 | 32767 | 0 | cycles |
| Lifetimes | Time | 0x4221 | Total Fw Runtime | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4225 | Time Spent In UT RSOC A | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4229 | Time Spent In UT RSOC B | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x422D | Time Spent In UT RSOC C | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4231 | Time Spent In UT RSOC D | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4235 | Time Spent In UT RSOC E | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4239 | Time Spent In UT RSOC F | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x423D | Time Spent In UT RSOC G | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4241 | Time Spent In UT RSOC H | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4245 | Time Spent In LT RSOC A | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4249 | Time Spent In LT RSOC B | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x424D | Time Spent In LT RSOC C | U4 | 0 | 4.294967295e+009 | 0 | s |

## Table 15-1. Data Flash Table (continued)

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|---|---|---|---|---|---|---|---|---|
| Lifetimes | Time | 0x4251 | Time Spent In LT RSOC D | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4255 | Time Spent In LT RSOC E | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4259 | Time Spent In LT RSOC F | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x425D | Time Spent In LT RSOC G | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4261 | Time Spent In LT RSOC H | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4265 | Time Spent In STL RSOC A | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4269 | Time Spent In STL RSOC B | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x426D | Time Spent In STL RSOC C | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4271 | Time Spent In STL RSOC D | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4275 | Time Spent In STL RSOC E | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4279 | Time Spent In STL RSOC F | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x427D | Time Spent In STL RSOC G | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4281 | Time Spent In STL RSOC H | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4285 | Time Spent In RT RSOC A | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4289 | Time Spent In RT RSOC B | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x428D | Time Spent In RT RSOC C | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4291 | Time Spent In RT RSOC D | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4295 | Time Spent In RT RSOC E | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x4299 | Time Spent In RT RSOC F | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x429D | Time Spent In RT RSOC G | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42A1 | Time Spent In RT RSOC H | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42A5 | Time Spent In STH RSOC A | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42A9 | Time Spent In STH RSOC B | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42AD | Time Spent In STH RSOC C | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42B1 | Time Spent In STH RSOC D | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42B5 | Time Spent In STH RSOC E | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42B9 | Time Spent In STH RSOC F | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42BD | Time Spent In STH RSOC G | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42C1 | Time Spent In STH RSOC H | U4 | 0 | 4.294967 295e+009 | 0 | s |
| Lifetimes | Time | 0x42C5 | Time Spent In HT RSOC A | U4 | 0 | 4.294967 295e+009 | 0 | s |

**Table 15-1. Data Flash Table (continued)**

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|-------|----------|---------|------|------|-----------|-----------|---------|-------|
| Lifetimes | Time | 0x42C9 | Time Spent In HT RSOC B | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42CD | Time Spent In HT RSOC C | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42D1 | Time Spent In HT RSOC D | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42D5 | Time Spent In HT RSOC E | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42D9 | Time Spent In HT RSOC F | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42DD | Time Spent In HT RSOC G | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42E1 | Time Spent In HT RSOC H | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42E5 | Time Spent In OT RSOC A | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42E9 | Time Spent In OT RSOC B | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42ED | Time Spent In OT RSOC C | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42F1 | Time Spent In OT RSOC D | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42F5 | Time Spent In OT RSOC E | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42F9 | Time Spent In OT RSOC F | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x42FD | Time Spent In OT RSOC G | U4 | 0 | 4.294967295e+009 | 0 | s |
| Lifetimes | Time | 0x4301 | Time Spent In OT RSOC H | U4 | 0 | 4.294967295e+009 | 0 | s |
| Ra Table | R_a0 | 0x4100 | Cell0 R_a flag | H2 | 0x0 | 0xFFFF | 0xFF55 | — |
| Ra Table | R_a0 | 0x4102 | Cell0 R_a 0 | I2 | 0 | 32767 | 159 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x4104 | Cell0 R_a 1 | I2 | 0 | 32767 | 58 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x4106 | Cell0 R_a 2 | I2 | 0 | 32767 | 65 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x4108 | Cell0 R_a 3 | I2 | 0 | 32767 | 79 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x410A | Cell0 R_a 4 | I2 | 0 | 32767 | 90 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x410C | Cell0 R_a 5 | I2 | 0 | 32767 | 63 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x410E | Cell0 R_a 6 | I2 | 0 | 32767 | 76 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x4110 | Cell0 R_a 7 | I2 | 0 | 32767 | 82 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x4112 | Cell0 R_a 8 | I2 | 0 | 32767 | 82 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x4114 | Cell0 R_a 9 | I2 | 0 | 32767 | 87 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x4116 | Cell0 R_a 10 | I2 | 0 | 32767 | 87 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x4118 | Cell0 R_a 11 | I2 | 0 | 32767 | 101 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x411A | Cell0 R_a 12 | I2 | 0 | 32767 | 118 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x411C | Cell0 R_a 13 | I2 | 0 | 32767 | 191 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0 | 0x411E | Cell0 R_a 14 | I2 | 0 | 32767 | 282 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x4140 | xCell0 R_a flag | H2 | 0x0 | 0xFFFF | 0xFFFF | — |
| Ra Table | R_a0x | 0x4142 | xCell0 R_a 0 | I2 | 0 | 32767 | 159 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x4144 | xCell0 R_a 1 | I2 | 0 | 32767 | 58 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x4146 | xCell0 R_a 2 | I2 | 0 | 32767 | 65 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x4148 | xCell0 R_a 3 | I2 | 0 | 32767 | 79 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x414A | xCell0 R_a 4 | I2 | 0 | 32767 | 90 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x414C | xCell0 R_a 5 | I2 | 0 | 32767 | 63 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x414E | xCell0 R_a 6 | I2 | 0 | 32767 | 76 | $2^{-10}\ \Omega$ |

**Table 15-1. Data Flash Table (continued)**

| Class | Subclass | Address | Name | Type | Min Value | Max Value | Default | Units |
|-------|----------|---------|------|------|-----------|-----------|---------|-------|
| Ra Table | R_a0x | 0x4150 | xCell0 R_a 7 | I2 | 0 | 32767 | 82 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x4152 | xCell0 R_a 8 | I2 | 0 | 32767 | 82 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x4154 | xCell0 R_a 9 | I2 | 0 | 32767 | 87 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x4156 | xCell0 R_a 10 | I2 | 0 | 32767 | 87 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x4158 | xCell0 R_a 11 | I2 | 0 | 32767 | 101 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x415A | xCell0 R_a 12 | I2 | 0 | 32767 | 118 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x415C | xCell0 R_a 13 | I2 | 0 | 32767 | 191 | $2^{-10}\ \Omega$ |
| Ra Table | R_a0x | 0x415E | xCell0 R_a 14 | I2 | 0 | 32767 | 282 | $2^{-10}\ \Omega$ |

# Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

**Changes from A Revision (February 2020) to B Revision** **Page**

• Changed *Authentication Flow* ................................................................................................................. 41

# IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.