bq27621-G1

Technical Reference

TEXAS INSTRUMENTS

Literature Number: SLUUAD4C
December 2013–Revised August 2014

Contents

Preface

This document is a detailed Technical Reference Manual (TRM) for using and configuring the bq27621-G1 battery fuel gauge. This TRM document is intended to complement but not supersede any information contained in the separate bq27621-G1 datasheet. Refer to the *bq27621-G1 DataSheet (SLUSBB3)*.

Another useful reference document is the *bq27621-G1 Quick Start Guide (SLUUAP5)*.

Formatting conventions used in this document:

Information Type	Formatting Convention	Example
Commands	*Italics* with *parentheses* and no breaking spaces	*RemainingCapacity( )* command
Data Memory	*Italics*, **bold**, and breaking spaces	***Design Capacity*** data
Register bits and flags	Brackets and *Italics*	*[SOC1]* bit
Data Memory bits	Brackets, *italics*, and **bold**	***[TEMPS]*** bit
Modes and states	ALL CAPITALS	UNSEALED mode

Revision History

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

General Description

The bq27621-G1 battery fuel gauge accurately predicts the battery capacity and other operational characteristics of a single Li-based rechargeable cell. It can be interrogated by a system processor to provide cell information, such as state-of-charge (SOC). The device is orderable in one predefined standard configuration but can be configured to support other chemistries using I^2C commands:

- The bq27621-G1 fuel gauge is predefined for $LiCoO_2$-based batteries which have 4.2-V, 4.3-V, and 4.35-V maximum charging voltages. One orderable part number contains three different battery profiles which can be selected using I^2C commands.

Unlike some other fuel gauges, the bq27621-G1 fuel gauge cannot be programmed with specific battery chemistry profiles. For many battery types and applications, the predefined standard chemistry profiles available in the bq27621-G1 fuel gauge are sufficient matches from a gauging perspective.

Information is accessed through a series of commands, called *Standard Commands*. Further capabilities are provided by the additional *Extended Commands* set. Both sets of commands, indicated by the general format *Command( )*, are used to read and write information contained within the control and status registers, as well as its data locations. Commands are sent from system to gauge using the I^2C serial communications engine, and can be executed during application development, system manufacture, or end-equipment operation.

The key to the fuel gauging prediction of the bq27621-G1 fuel gauge is Texas Instruments proprietary Dynamic Voltage Correlation algorithm. This algorithm eliminates the need for a sense resistor when calculating remaining battery capacity (mAh) and SOC (%). This algorithm uses cell voltage measurements and cell characteristics to create SOC predictions that can achieve high accuracy across a wide variety of operating conditions.

The fuel gauge estimates charge and discharge activity by monitoring the cell voltage. Cell impedance is computed based on estimated current, open-circuit voltage (OCV), and cell voltage under loaded conditions.

The fuel gauge uses an integrated temperature sensor for estimating cell temperature. Alternatively, the system processor can provide temperature data for the fuel gauge.

To minimize power consumption, the fuel gauge has several power modes: INITIALIZATION, NORMAL, SLEEP, HIBERNATE, and SHUTDOWN. The fuel gauge passes automatically between these modes, depending upon the occurrence of specific events, though a system processor can initiate some of these modes directly.

Functional Description

2.1 Fuel Gauging

The bq27621-G1 battery fuel gauge measures the cell voltage and temperature to determine battery SOC. The fuel gauge estimates charge and discharge current by monitoring the voltage across the BAT and V_{SS} terminals.

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 application load is applied, the impedance of the cell is measured by comparing the OCV obtained from a predefined function for present SOC with the measured voltage under load. Measurements of OCV and charge integration determine chemical SOC and chemical capacity (Qmax). The initial value for Qmax is defined by ***Design Capacity*** and should match the cell manufacturers' data sheet. An impedance profile, along with SOC and the Qmax value, are used to determine *FullChargeCapacity( )* and *StateOfCharge( )*, specifically for the present load and temperature. *FullChargeCapacity( )* is reported as capacity available from a fully-charged battery under the present load and temperature until *Voltage( )* reaches the ***Terminate Voltage***. *NominalAvailableCapacity( )* and *FullAvailableCapacity( )* are the uncompensated (no or light load) versions of *RemainingCapacity( )* and *FullChargeCapacity( )*, respectively.

The bq27621-G1 has two flags, *[SOC1]* (state-of-charge initial) and *[SOCF]* (state-of-charge final), accessed by the *Flags( )* command that warns when the battery SOC has fallen to critical levels. When *StateOfCharge( )* falls below the first capacity threshold, specified in ***SOC1 Set Threshold***, the *[SOC1]* flag is set. The flag is cleared once *StateofCharge( )* rises above ***SOC1 Set Threshold***.

When *StateofCharge( )* falls below the second capacity threshold, ***SOCF Set Threshold***, the *[SOCF]* flag is set, serving as a final discharge warning. If ***SOCF Set Threshold*** = –1, the flag is inoperative during discharge. Similarly, when *StateofCharge( )* rises above ***SOCF Clear Threshold*** and the *[SOCF]* flag has already been set, the *[SOCF]* flag is cleared.

2.2 Temperature Measurement

Accurate fuel gauging requires accurate knowledge of the battery temperature. If the internal temperature sensor of the fuel gauge is used (***Op Config [TEMPS]*** = 0), the gauge should be placed as close to the battery as possible. If the system processor is used to update the fuel gauge *Temperature( )* register (***Op Config [TEMPS]*** = 1), then it should be using an accurate measurement for this purpose.

Regardless of which sensor is used for measurement, the system processor can request the current battery temperature being used by the algorithm by calling the *Temperature( )* function.

2.3 Current Measurement

The fuel gauge estimates current using the battery voltage, temperature, and battery model parameters. The estimated current is available through the *EffectiveCurrent( )* command.

2.4 Operating Modes

The fuel gauge has different operating modes: POR, INITIALIZATION, NORMAL, CONFIG UPDATE, SLEEP, and HIBERNATE. Upon power up from OFF or SHUTDOWN, a power-on reset (POR) occurs and the fuel gauge begins INITIALIZATION. In NORMAL mode, the fuel gauge is fully powered and can execute any allowable task. Configuration data in RAM can be updated by the host using the CONFIG UPDATE mode. In SLEEP mode the fuel gauge turns off the high-frequency oscillator clock to enter a reduced-power state, periodically taking measurements and performing calculations. In HIBERNATE mode the fuel gauge is in a very-low-power state, but can be woken up by communication or certain I/O activity.

In SHUTDOWN mode, the LDO is disabled so internal power and all RAM-based volatile data is lost. Because no gauging occurs in SHUTDOWN mode, additional gauging error can be introduced if the system has significant battery charge or discharge activity prior to re-initialization.

2.4.1 POR and INITIALIZATION Modes

Upon power-on reset (POR), the fuel gauge copies ROM-based configuration defaults to RAM and begins INITIALIZATION mode where essential data is initialized. It remains in INITIALIZATION mode as a halted-CPU state when an adapter or other power source is present to power the fuel gauge (and system), yet no battery has been detected. The occurrence of POR or a *RESET* subcommand sets the *Flags( ) [ITPOR]* status bit to indicate that RAM has returned to ROM default data. When battery insertion is detected, a series of initialization activities begin including an OCV measurement.

Some commands, issued by a system processor, can be processed while the fuel gauge is halted in this mode. The gauge wakes up to process the command, and then returns to the halted state awaiting battery insertion.

2.4.2 CONFIG UPDATE Mode

If the application requires different configuration data for the fuel gauge, the system processor can update RAM-based Data Memory parameters using the *SET_CFGUPDATE* subcommand to enter CONFIG UPDATE mode. Operation in this mode is indicated by the *Flags( ) [CFGUPD]* status bit. In this mode, fuel gauging is suspended while the host uses the *Extended Commands* to modify the configuration data blocks. To resume fuel gauging, the host sends a *SOFT_RESET* subcommand to exit CONFIG UPDATE mode which clears both *Flags( ) [ITPOR]* and *[CFGUPD]* bits. After a timeout of approximately 240 seconds (4 minutes), the gauge automatically exits the CONFIG UPDATE mode if it has not received a *SOFT_RESET* subcommand from the host.

2.4.3 NORMAL Mode

The fuel gauge is in NORMAL mode when not in any other power mode. During this mode, *EffectiveCurrent( )*, *Voltage( ),* and *Temperature( )* measurements are updated every three seconds. Decisions to change states are also made. This mode is exited by activating a different power mode.

Because the gauge consumes the most power in NORMAL mode, the algorithm minimizes the time the fuel gauge remains in this mode.

2.4.4 SLEEP Mode

SLEEP mode is entered automatically if the feature is enabled (***Sleep Time*** > 0) and when *EffectiveCurrent( )* is below the programmable level ***Sleep Current*** (default = 10 mA). Once entry into SLEEP mode has been qualified, but prior to entering it, the fuel gauge performs an ADC autocalibration to minimize offset.

During SLEEP mode, the fuel gauge remains in a very-low-power idle state and automatically wakes up periodically. The number of seconds that the gauge will stay in SLEEP before waking to take periodic measurements is set by the parameter ***Sleep Time*** (default = 20 seconds). If ***Sleep Time*** is set to 0 then the gauge will never go do SLEEP.

After making the measurements on the interval defined by ***Sleep Time***, the fuel gauge will exit SLEEP mode when *EffectiveCurrent( )* rises above ***Sleep Current*** (default = 10 mA).

2.4.5 HIBERNATE Mode

HIBERNATE mode could be used when the system equipment needs to enter a very-low-power state, and minimal gauge power consumption is required. This mode is ideal when a system equipment is set to its own HIBERNATE, SHUTDOWN, or OFF modes.

Before the fuel gauge can enter HIBERNATE mode, the system must use the *SET_HIBERNATE* subcommand to set the *[HIBERNATE]* bit of the *CONTROL_STATUS* register. The gauge will automatically enter hibernate mode when the magnitude of the average cell current has fallen below ***Hibernate Current***. The gauge will remain in HIBERNATE mode until the system issues a direct I^2C command to the gauge. I^2C communication that is not directed to the gauge will only briefly wake it up and the gauge immediately returns to HIBERNATE mode.

It is the responsibility of the system to wake the fuel gauge after it has gone into HIBERNATE mode and to prevent a charger from charging the battery before the *Flags( ) [OCVTAKEN]* bit is set which signals an OCV reading is taken. After waking, the gauge can proceed with the initialization of the battery information. During HIBERNATE mode, RAM-based data values are maintained, but gauging status is lost. Upon exit from HIBERNATE mode, the gauge will immediately re-acquire measurements and re-initialize all gauging predictions. The gauge will automatically enter HIBERNATE mode, if the average current goes below the hibernate current threshold or if the cell voltage goes below the hibernate voltage threshold.

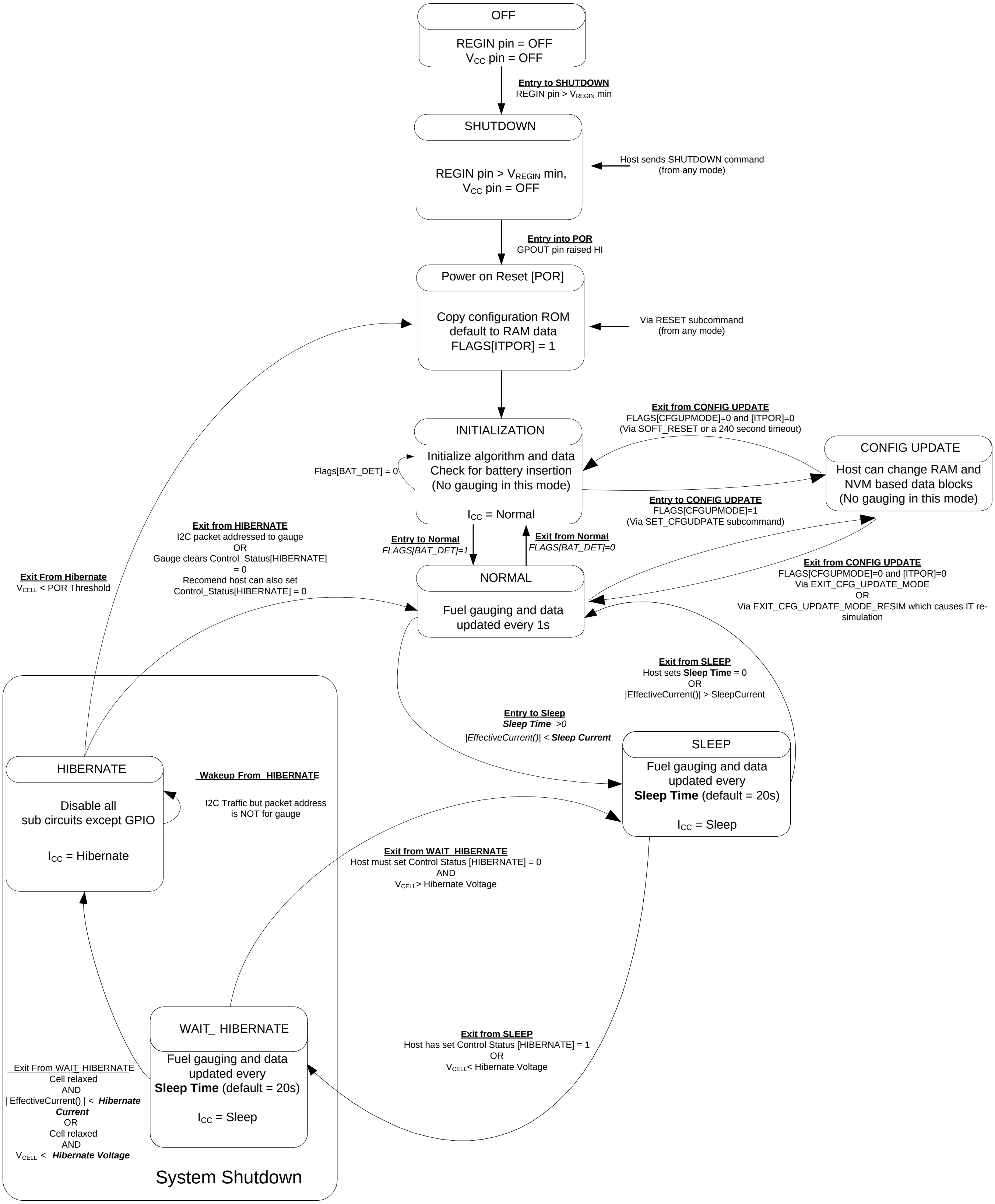

Figure 2-1. Power Mode Diagram

2.5 Pin Descriptions

2.5.1 GPOUT Pin

The GPOUT pin is a multiplexed pin and the polarity of the pin output can be selected via the ***[GPIOPOL]*** bit of ***Op Config***. The function is defined by ***Op Config [BATLOWEN]***. If ***[BATLOWEN]*** is set, the battery low indicator (BAT_LOW) function for GPOUT pin is selected. If it is cleared, the SOC interrupt (SOC_INT) function is selected for GPOUT.

When the BAT_LOW function is activated, the signaling on the multiplexed pin follows the status of the *[SOC1]* bit in the *Flags( )* register. The fuel gauge has two flags accessed by the *Flags( )* function that warns when the battery SOC has fallen to critical levels. When *StateOfCharge( )* falls below the first capacity threshold, specified in ***SOC1 Set Threshold***, the *[SOC1]* flag is set. The flag is cleared once *StateOfCharge( )* rises above ***SOC1 Set Threshold***. The GPOUT pin automatically reflects the status of the *[SOC1]* flag when ***[BATLOWEN]*** = 1 and ***[GPIOPOL]*** = 0. The polarity can be flipped by setting ***[GPIOPOL]*** = 1.

When *StateOfCharge( )* falls below the second capacity threshold, ***SOCF Set Threshold***, the *[SOCF]* flag is set, serving as a final discharge warning. Similarly, when *StateOfCharge( )* rises above ***SOCF Clear Threshold*** and the *[SOCF]* flag has already been set, the *[SOCF]* flag is cleared.

When the SOC_INT function is activated, the GPOUT pin generates 1-ms pulse width under various conditions as described in Table 2-1.

Table 2-1. SOC_INT Function Definition

	Enable Condition	Pulse Width	Description
Change in SOC	(***SOCI Delta***) ≠ 0	1 ms	During charge, when the SOC is greater than (>) the points, 100% – n × (***SOCI Delta***) and 100%; During discharge, when the SOC reaches (≤) the points, 100% – n × (***SOCI Delta***) and 0%; where n is an integer starting from 0 to the number generating SOC no less than 0% Examples: For ***SOCI Delta*** = 1% (default), the SOC_INT intervals are 0%, 1%, 2%, …, 99%, and 100%. For ***SOCI Delta*** = 10%, the SOC_INT intervals are 0%, 10%, 20%, …, 90%, and 100%.
State Change	(***SOCI Delta***) ≠ 0	1 ms	Upon detection of entry to a charge or a discharge state. Relaxation is not included.
Battery Removal	***Op Config [BIE]*** is set	1 ms	When battery removal is detected by BIN pin.

2.5.2 Battery Detection (BIN)

The function of the ***Op Config [BIE]*** bit is described in Table 2-2. When battery insertion is detected and INITIALIZATION mode is completed, the fuel gauge transitions to NORMAL mode to start fuel gauging. When battery insertion is not detected, the fuel gauge remains in INITIALIZATION mode. Do not leave the BIN pin floating; external termination through a resistor is required.

Table 2-2. Battery Detection

Op Config [BIE]	Battery Insertion Requirement	Battery Removal Requirement
1	(1) Host drives BIN pin from logic high to low to signal battery insertion. or (2) A weak pullup resistor can be used (between BIN and V_{CC} pins). When a battery pack with a pulldown resistor is connected, it can generate a logic low to signal battery insertion.	(1) Host drives BIN pin from logic low to high to signal battery removal. or (2) When a battery pack with a pulldown resistor is removed, the weak pullup resistor can generate a logic high to signal battery removal.
0	Host sends *BAT_INSERT* subcommand to signal battery insertion. The BIN pin must be connected to V_{SS} through a weak pulldown resistor; 4.7 MΩ is a typical value.	Host sends *BAT_REMOVE* subcommand to signal battery removal.

Application Examples

3.1 Data Memory Parameter Update Example

This following example shows the command sequence needed to modify a Data Memory parameter. For this example, the default ***Design Capacity*** is updated from 2425 mAh to 1200 mAh. All device writes (wr) and reads (rd) are implied to I²C 8-bit addresses 0xAA and 0xAB, respectively.

Step	Description	Pseudo Code

1	If the device has been previously SEALED, UNSEAL it by sending the appropriate keys to *Control( )* (0x00 and 0x01). Write the first 2 bytes of the UNSEAL key using the *Control(0x8000)* command. Without writing any other bytes to the device, write the second (identical) 2 bytes of the UNSEAL key using the *Control(0x8000)* command.
	Note: Remaining steps below will use this single packet method when writing multiple bytes.
		`//Two-byte incremental Method`
		`wr 0x00 0x00 0x80;`
		`wr 0x00 0x00 0x80;`
		`//Alternative single byte method`
		`wr 0x00 0x00;`
		`wr 0x01 0x80;`
		`wr 0x00 0x00;`
		`wr 0x01 0x80;`

2	Send *SET_CFGUPDATE* subcommand, *Control(0x0013)*	`wr 0x00 0x13 0x00;`
3	Confirm CONFIG UPDATE mode by polling *Flags( )* register until bit 4 is set. May take up to 1 second.	`rd 0x06 Flags_register;`
4	Write 0x00 using *BlockDataControl( )* command (0x61) to enable block data memory control.	`wr 0x61 0x00;`
5	Write 0x52 using the *DataClass( )* command (0x3E) to access the State subclass (82 decimal, 0x52 hex) containing the ***Design Capacity*** parameter.	`wr 0x3E 0x52;`
6	Write the block offset location using *DataBlock( )* command (0x3F). **Note**: To access data located at offset 0 to 31 use offset = 0x00. To access data located at offset 32 to 41 use offset = 0x01.	`wr 0x3F 0x00;`
7	Read the 1-byte checksum using the *BlockDataChecksum( )* command (0x60). Expect 0x8A for this example.	`rd 0x60 OLD_Csum;`
8	Read both ***Design Capacity*** bytes starting at 0x43 (offset = 3). Block data starts at 0x40, so to read the data of a specific offset, use address 0x40 + mod (offset, 32). Expect 0x09 0x79 for a 2425-mAh default value.	`rd 0x43 OLD_DesCap_MSB;` `rd 0x44 OLD_DesCap_LSB;`
9	Write both ***Design Capacity*** bytes starting at 0x43. (offset = 3) For this example, the new value is 1200 mAh. (0x04B0 hex)	`wr 0x43 0x04;` `wr 0x44 0xB0;`

10	Compute the new block checksum. The checksum is (255 – x) where x is the 8-bit summation of the *BlockData( )* (0x40 to 0x5F) on a byte-by-byte basis. A quick way to calculate the new checksum uses a data replacement method with the old and new data summation bytes. Refer to the code for the indicated method.	`temp = mod(255 - OLD_Csum`
		`- OLD_DesCap_MSB`
		`- OLD_DesCap_LSB, 256);`
		`NEW_Csum = 255 - mod(temp +`
		`+ 0x04 + 0xB0, 256);`

11	Write new checksum. The data is actually transferred to the Data Memory when the correct checksum for the whole block (0x40 to 0x5F) is written to *BlockDataChecksum( )* (0x60). For this example New_Csum is 0x1F.	`wr 0x60 New_Csum;` `//Example: wr 0x60 0x58`
12	Exit CONFIG UPDATE mode by sending *SOFT_RESET* subcommand, *Control(0x0042)*	`wr 0x00 0x42 0x00;`
13	Confirm CONFIG UPDATE has been exited by polling *Flags( )* register until bit 4 is cleared. May take up to 1 second.	`rd 0x06 Flags_register;`
14	If the device was previously SEALED, return to SEALED mode sending by the *Control(0x0020)* command.	`wr 0x00 0x20 0x00;`

3.2 Procedure to Select Alternate Chemistry

This device has three unique battery profiles stored in memory. Upon initialization or reset, the gauge will automatically use the default profile, *CHEM_ID* = 0x1202. This is appropriate for most standard $LiCoO_2$ batteries which have a maximum charging voltage of 4.2 V. The two alternative battery profiles can be selected by the system while updating the other configuration parameters in CONFIG UPDATE mode. These alternative battery profiles are *CHEM_ID* = 0x1210 (appropriate for most chemistries with a maximum charging voltage of 4.3 V) and *CHEM_ID* = 0x354 (appropriate for most chemistries with a maximum charging voltage of 4.35 V). The procedure that modifies which battery profile is to be used by the fuel gauging algorithm is:

1. Enter CONFIG UPDATE mode. This is done by sending the *SET_CFGUPDATE* subcommand (0x0013) to the *Control( )* register. Verify that we have entered CONFIG UPDATE mode. This is done by reading the *Flags( )* register and verifying that *[CFGUPD]* (bit 11) is set.
2. Choose alternate chemistry. Send *ALT_CHEM1* subcommand (0x0031) to choose alternate chemistry 1 (*CHEM_ID* = 0x1210) or *ALT_CHEM2* subcommand (0x0032) to choose alternate chemistry 2 (*CHEM_ID* = 0x354). Verify that the command has been executed successfully. This is done by reading the *CONTROL_STATUS* register and verifying that *[CHEMCHNG]* (Bit 0) is clear.
3. Exit CONFIG UPDATE mode. This is done by sending the *EXIT_CFGUPDATE* subcommand (0x0043) to the *Control( )* register.
4. Verify if new chemistry has been selected. This is done by sending the *CHEM_ID* subcommand (0x0008) and reading the result from the *Control( )* register. The device should report 0x1210 for chemistry 1 and 0x324 for chemistry 2. It will report 0x1202 if the default chemistry profile is in use.

The gauge must be in the UNSEALED state for the *SET_CFGUPDATE* subcommand to be successful.

Standard Commands

The bq27621-G1 fuel gauge uses a series of 2-byte standard commands to enable system reading and writing of battery information. Each standard command has an associated command-code pair, as indicated in Table 4-1. Because each command consists of two bytes of data, two consecutive I^2C transmissions must be executed both to initiate the command function, and to read or write the corresponding two bytes of data.

Table 4-1. Standard Commands

NAME		COMMAND CODE	UNIT	SEALED ACCESS
Control( )	CNTL	0x00 and 0x01	NA	RW
Temperature( )	TEMP	0x02 and 0x03	0.1°K	RW
Voltage( )	VOLT	0x04 and 0x05	mV	R
Flags( )	FLAGS	0x06 and 0x07	NA	R
NominalAvailableCapacity( )		0x08 and 0x09	mAh	R
FullAvailableCapacity( )		0x0A and 0x0B	mAh	R
RemainingCapacity( )	RM	0x0C and 0x0D	mAh	R
FullChargeCapacity( )	FCC	0x0E and 0x0F	mAh	R
EffectiveCurrent( )		0x10 and 0x11	mA	R
AveragePower( )		0x18 and 0x19	mW	R
StateOfCharge( )	SOC	0x1C and 0x1D	%	R
InternalTemperature( )		0x1E and 0x1F	0.1°K	R
RemainingCapacityUnfiltered( )		0x28 and 0x29	mAh	R
RemainingCapacityFiltered( )		0x2A and 0x2B	mAh	R
FullChargeCapacityUnfiltered( )		0x2C and 0x2D	mAh	R
FullChargeCapacityFiltered( )		0x2E and 0x2F	mAh	R
StateOfChargeUnfiltered( )		0x30 and 0x31	%	R
OperationConfiguration( )		0x3A and 0x3B	NA	R

4.1 Control( ): 0x00 and 0x01

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

Table 4-2. *Control( )* Subcommands

CNTL FUNCTION	CNTL DATA	SEALED ACCESS	DESCRIPTION
CONTROL_STATUS	0x0000	Yes	Reports the status of device.
DEVICE_TYPE	0x0001	Yes	Reports the device type (0x0621).
FW_VERSION	0x0002	Yes	Reports the firmware version of the device.
PREV_MACWRITE	0x0007	Yes	Returns previous MAC command code.
CHEM_ID	0x0008	Yes	Reports the chemical identifier of the battery profile currently used by the fuel gauging algorithm.
BAT_INSERT	0x000C	Yes	Forces the *Flags( ) [BAT_DET]* bit set when the ***Op Config [BIE]*** bit is 0.
BAT_REMOVE	0x000D	Yes	Forces the *Flags( ) [BAT_DET]* bit clear when the ***Op Config [BIE]*** bit is 0.
TOGGLE_POWERMIN	0x0010	Yes	Sets *CONTROL_STATUS [POWERMIN]* to 1.
SET_HIBERNATE	0x0011	Yes	Forces *CONTROL_STATUS [HIBERNATE]* to 1.
CLEAR_HIBERNATE	0x0012	Yes	Forces *CONTROL_STATUS [HIBERNATE]* to 0.
SET_CFGUPDATE	0x0013	No	Forces *Flags( ) [CFGUPD]* to 1 and gauge enters CONFIG UPDATE mode.
SHUTDOWN_ENABLE	0x001B	No	Enables device SHUTDOWN mode.
SHUTDOWN	0x001C	No	Commands the device to enter SHUTDOWN mode.
SEALED	0x0020	No	Places the device in SEALED access mode.
TOGGLE_GPOUT	0x0023	Yes	Tests the GPIO pin by sending a pulse signal
ALT_CHEM1	0x0031	No	Selects alternate chemistry 1 (0x1210)
ALT_CHEM2	0x0032	No	Selects alternate chemistry 2 (0x354)
RESET	0x0041	No	Performs a full device reset.
SOFT_RESET	0x0042	No	Gauge exits CONFIG UPDATE mode.
EXIT_CFGUPDATE	0x0043	No	Exits CONFIG UPDATE mode without an OCV measurement and without resimulating to update *StateOfCharge( ).*
EXIT_RESIM	0x0044	No	Exits CONFIG UPDATE mode without an OCV measurement and resimulates with the updated configuration data to update *StateOfCharge( ).*

4.1.1 CONTROL_STATUS: 0x0000

Instructs the fuel gauge to return status information to *Control( )* addresses 0x00 and 0x01. The read-only status word contains status bits that are set or cleared either automatically as conditions warrant or through using specified subcommands.

Table 4-3. CONTROL_STATUS Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	SHUTDOWNEN	WDRESET	SS	CALMODE	RSVD	RSVD	OCVCMDCOMP	OCVFAIL
Low Byte	INITCOMP	HIBERNATE	POWERMIN	SLEEP	LDMD	RSVD	RSVD	CHEMCHNG

High Byte

SHUTDOWNEN = Indicates the fuel gauge has received the *SHUTDOWN_ENABLE* subcommand and is enabled for SHUTDOWN. Active when set.

WDRESET = Indicates the fuel gauge has performed a Watchdog Reset. Active when set.

SS = Indicates the fuel gauge is in the SEALED state. Active when set.

CALMODE = Indicates the fuel gauge is in calibration mode. Active when set.

RSVD = Bits 3:2 are reserved.

OCVCMDCOMP = Indicates that the OCV measurement is complete. After a battery insert this bit will be clear. If the host is able to stay at a reduced power to provide a better SOC estimate, then after this bit sets it is OK for the host system to go to normal power because the measurements needed of OCV and thus SOC have been completed. It can take up to another second for the host to see *[INITCOMP]* while the gauge runs simulations for remaining and full capacity.

OCVFAIL = if set indicates that the gauge measured a large jump of current while trying to determine SOC, therefore SOC may not be reliable.

Low Byte

INITCOMP = Initialization completion bit indicating the initialization completed. True when set.

HIBERNATE = Indicates a request for entry into HIBERNATE from SLEEP mode has been issued. True when set.

POWERMIN = Indicates the fuel gauge is in minimum power mode. True when set. The state can be detected by monitoring the power used by the fuel gauge.

SLEEP = Indicates the fuel gauge is in SLEEP mode. True when set.

LDMD = Indicates the algorithm is using constant-power model. True when set. Default is 0.

RSVD = Bits 2:1 are reserved.

CHEMCHNG = Bit is set by a successful MAC command for chemistry change and gets cleared when all of the RAM has been updated. The purpose is that after a CHEMCHNG, the host will want to exit CONFIG UPDATE mode, typically with a *SOFT_RESET* or *EXIT_RESIM* subcommand. But the host should wait until this bit (CHEMCHNG) is cleared so that SOC and capacities will be determined correctly.

4.1.2 DEVICE_TYPE: 0x0001

Instructs the fuel gauge to return the device type to addresses 0x00 and 0x01. The value returned is 0x0621.

4.1.3 FW_VERSION: 0x0002

Instructs the fuel gauge to return the firmware version to addresses 0x00 and 0x01.

4.1.4 PREV_MACWRITE: 0x0007

Instructs the fuel gauge to return the previous command written to addresses 0x00 and 0x01. The value returned is limited to less than 0x0015.

4.1.5 CHEM_ID: 0x0008

Instructs the fuel gauge to return the chemical identifier of the battery profile currently being used for fuel gauging to addresses 0x00 and 0x01. The default value for *CHEM_ID* is 0x1202. Alternative values are 0x1210 and 0x354.

4.1.6 BAT_INSERT: 0x000C

This subcommand forces the *Flags( ) [BAT_DET]* bit to set when the battery insertion detection is disabled via ***Op Config [BIE]*** = 0. In this case, the gauge does not detect battery insertion from the logic state of the BIN pin, but relies on the *BAT_INSERT* subcommand to indicate battery presence in the system. This subcommand also starts gauging.

4.1.7 BAT_REMOVE: 0x000D

This subcommand forces the *Flags( ) [BAT_DET]* bit to clear when the battery insertion detection is disabled via ***Op Config [BIE]*** = 0. In this case, the gauge does not detect battery removal from the logic state of the BIN pin, but relies on the *BAT_REMOVE* subcommand to indicate battery removal from the system.

4.1.8 TOGGLE_POWERMIN: 0x0010

Instructs the fuel gauge to enable or disable low-power operation. If the *CONTROL_STATUS [POWERMIN]* bit is set to 0, this subcommand forces it to 1 and vice versa. The POWERMIN mode disables one of the oscillators whenever possible to conserve power. This can result in slightly longer stretching of the I^2C clock signal when it needs to wake up. If other devices on the I^2C bus do not fully conform to the I^2C specification and have troubles with the extra clock stretching, then the POWERMIN mode can be disabled.

4.1.9 SET_HIBERNATE: 0x0011

Instructs the fuel gauge to force the *CONTROL_STATUS [HIBERNATE]* bit to 1. If the necessary conditions are met, this enables the gauge to enter the HIBERNATE power mode after the transition to SLEEP power state is detected. The *[HIBERNATE]* bit is automatically cleared upon exiting from HIBERNATE mode.

4.1.10 CLEAR_HIBERNATE: 0x0012

Instructs the fuel gauge to force the *CONTROL_STATUS [HIBERNATE]* bit to 0. This prevents the gauge from entering the HIBERNATE power mode after the transition to SLEEP power state is detected. It can also be used to force the gauge out of HIBERNATE mode.

4.1.11 SET_CFGUPDATE: 0x0013

Instructs the fuel gauge to set the *Flags( ) [CFGUPD]* bit to 1 and enter CONFIG UPDATE mode. This subcommand is only available when the fuel gauge is UNSEALED.

NOTE: A *SOFT_RESET* subcommand (0x0042) is typically used to exit CONFIG UPDATE mode to resume normal gauging.

4.1.12 SHUTDOWN_ENABLE: 0x001B

Instructs the fuel gauge to enable SHUTDOWN mode and sets the *CONTROL_STATUS [SHUTDOWNEN]* status bit.

4.1.13 SHUTDOWN: 0x001C

Instructs the fuel gauge to immediately enter SHUTDOWN mode after receiving this subcommand. The SHUTDOWN mode is effectively a powerdown mode with only a small circuit biased by the BAT pin used for wake up detection. To enter SHUTDOWN mode, the *SHUTDOWN_ENABLE* subcommand must have been previously received. The fuel gauge wakes up from SHUTDOWN mode upon detection of a low-to-high transition on the open-drain GPOUT pin. All gauging stops while the gauge is in SHUTDOWN mode. An exit from SHUTDOWN mode reinitializes the gauge to the default setting and it must be reconfigured by the host.

4.1.14 SEALED: 0x0020

Instructs the fuel gauge to transition from UNSEALED state to SEALED state. The fuel gauge should always be set to SEALED state for use in end equipment. The SEALED state blocks accidental writes of specific subcommands and most Standard and Extended Commands. See Table 4-1, Table 4-2, and Table 5-1.

4.1.15 TOGGLE_GPOUT: 0x0023

This subcommand can be useful for system level debug or test purposes. It instructs the fuel gauge to pulse the GPOUT pin for approximately 1 ms within 1 second of receiving the subcommand.

NOTE: The GPOUT pin must be configured for the SOC_INT output function with ***Op Config [BATLOWEN]*** cleared.

4.1.16 ALT_CHEM1: 0x0031

This subcommand instructs the fuel gauge select alternate chemistry 1 (*CHEM_ID* = 0x1210). This subcommand is only available when the fuel gauge is UNSEALED and in the CONFIG UPDATE mode. After exiting from CONFIG UPDATE mode, wait for the *CONTROL_STATUS [CHEMCHNG]* bit to clear before reading gauging results. This subcommand should only be sent when the gauge is in CONFIG UPDATE mode (*Flags( ) [CFGUPD]* bit is set).

4.1.17 ALT_CHEM2: 0x0032

This subcommand instructs the fuel gauge select alternate chemistry 2 (*CHEM_ID* = 0x354). This subcommand is only available when the fuel gauge is UNSEALED and in the CONFIG UPDATE mode. After exiting from CONFIG UPDATE mode, wait for the *CONTROL_STATUS [CHEMCHNG]* bit to clear before reading gauging results. This subcommand should only be sent when the gauge is in CONFIG UPDATE mode (*Flags( ) [CFGUPD]* bit is set).

4.1.18 RESET: 0x0041

This subcommand instructs the fuel gauge to perform a full device reset and reinitialize RAM data to the default values from ROM and is therefore not typically used in field operation. The gauge sets the *Flags( ) [ITPOR]* bit and enters the INITIALIZE mode. See Figure 2-1. This subcommand is only available when the fuel gauge is UNSEALED.

4.1.19 SOFT_RESET: 0x0042

This subcommand instructs the fuel gauge to perform a partial (soft) reset from any mode with an OCV measurement. The *Flags( ) [ITPOR]* and *[CFGUPD]* bits are cleared and a resimulation occurs to update *StateOfCharge( )*. See Figure 2-1. This command is only available when the fuel gauge is UNSEALED.

4.1.20 EXIT_CFGUPDATE: 0x0043

This subcommand exits CONFIG UPDATE mode without an OCV measurement and without resimulating to update *StateOfCharge( )*. The *Flags( ) [ITPOR]* and *[CFGUPD]* bits are cleared. This command is only available when the fuel gauge is UNSEALED.

4.1.21 EXIT_RESIM: 0x0044

This subcommand exits CONFIG UPDATE mode without an OCV measurement. The *Flags( ) [ITPOR]* and *[CFGUPD]* bits are cleared and a resimulation occurs to update *StateOfCharge( )*. This command is only available when the fuel gauge is UNSEALED.

4.2 Temperature( ): 0x02 and 0x03

This read- or write-word function returns an unsigned integer value of the temperature in units of 0.1°K measured by the fuel gauge. If ***Op Config [TEMPS]*** bit = 0 (default), a read command will return the internal temperature sensor value and a write command will be ignored. If ***Op Config [TEMPS]*** bit = 1, a write command sets the temperature to be used for gauging calculations while a read command returns to temperature previously written.

4.3 Voltage( ): 0x04 and 0x05

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

4.4 Flags( ): 0x06 and 0x07

This read-word function returns the contents of the fuel gauging status register which depicts the current operating status.

Table 4-4. Flags Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	OT	UT	RSVD	RSVD	RSVD	RSVD	FC	CHG
Low Byte	OCVTAKEN	RSVD	ITPOR	CFGUPD	BAT_DET	SOC1	SOCF	DSG

High Byte

OT = Over-Temperature condition is detected. *[OT]* is set when *Temperature( )* ≥ ***Over Temp*** (default = 55°C). *[OT]* is cleared when *Temperature( )* < ***Over Temp*** – ***Temp Hys***.

UT = Under-Temperature condition is detected. *[UT]* is set when *Temperature( )* ≤ ***Under Temp*** (default = 0°C). *[UT]* is cleared when *Temperature( )* > ***Under Temp*** + ***Temp Hys***.

RSVD = Bits 5:2 are reserved

FC = Full-charge is detected. If the ***FC Set %*** is a positive threshold, *[FC]* is set when SOC ≥ ***FC Set %*** and is cleared when SOC ≤ ***FC Clear %*** (default = 98%). By default, ***FC Set %*** = –1, therefore, *[FC]* is set when the fuel gauge has detected charge termination.

CHG = Fast charging allowed. If SOC changes from 98% to 99% during charging, the *[CHG]* bit is cleared. The *[CHG]* bit will become set again when SOC ≤ 95%.

Low Byte

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

RSVD = Bit 6 is reserved.

ITPOR = Indicates a power-on reset or *RESET* subcommand has occurred. If set, this bit generally indicates that the RAM configuration registers have been reset to default values and the host should reload the configuration parameters using the CONFIG UPDATE mode. This bit is cleared after the *SOFT_RESET* subcommand is received.

CFGUPD = Fuel gauge is in CONFIG UPDATE mode. True when set. Default is 0. See Section 2.4.2, *CONFIG UPDATE Mode*, for details.

BAT_DET = Battery insertion detected. True when set. When ***Op Config [BIE]*** is set, *[BAT_DET]* is set by detecting a logic high-to-low transition at the BIN pin. When ***Op Config [BIE]*** is low, *[BAT_DET]* is set when host issues *BAT_INSERT* subcommand and is clear when host issues *BAT_REMOVE* subcommand. Gauge predictions are no valid unless *[BAT_DET]* is set.

SOC1 = If set, *StateOfCharge( )* ≤ ***SOC1 Set Threshold***. The *[SOC1]* bit will remain set until *StateOfCharge( )* ≥ ***SOC1 Clear Threshold***.

SOCF = If set, *StateOfCharge( )* ≤ ***SOCF Set Threshold***. The *[SOCF]* bit will remain set until *StateOfCharge( )* ≥ ***SOCF Clear Threshold***.

DSG = Discharging detected. True when set.

4.5 NominalAvailableCapacity( ): 0x08 and 0x09

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

4.6 FullAvailableCapacity( ): 0x0A and 0x0B

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

4.7 RemainingCapacity( ): 0x0C and 0x0D

This read-only command pair returns the remaining battery capacity compensated for load and temperature. If ***OpConfigB [SMOOTHEN]*** = 1, this register is equal to *RemainingCapacityFiltered( )*; otherwise, it is equal to *RemainingCapacityUnfiltered( )*. Units are mAh.

4.8 FullChargeCapacity( ): 0x0E and 0x0F

This read-only command pair returns the compensated capacity of the battery when fully charged. *FullChargeCapacity( )* is updated at regular intervals and is compensated for load and temperature. If ***OpConfigB [SMOOTHEN]*** = 1, this register is equal to *FullChargeCapacityFiltered( )*; otherwise, it is equal to *FullChargeCapacityUnfiltered( )*. Units are mAh.

4.9 EffectiveCurrent( ): 0x10 and 0x11

This read-only command pair returns a signed integer value that is the average estimated current flow. In NORMAL mode, it is updated once per second. Large current spikes of short duration will be averaged out in this measurement. Instantaneous accuracy of this reported current estimate may fluctuate, but the long term average is sufficient for gauging. Units are mA.

4.10 AveragePower( ): 0x18 and 0x19

This read-only function returns an signed integer value of the 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.

4.11 StateOfCharge( ): 0x1C and 0x1D

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

4.12 InternalTemperature( ): 0x1E and 0x1F

This read-only function returns an unsigned integer value of the internal temperature sensor in units of 0.1°K measured by the fuel gauge. If ***Op Config [TEMPS]*** = 0, this command will return the same value as *Temperature( )*.

4.13 RemainingCapacityUnfiltered( ): 0x28 and 0x29

This read-only command pair returns the true battery capacity remaining. This value can jump as the gauge updates its predictions dynamically. Units are mAh.

4.14 RemainingCapacityFiltered( ): 0x2A and 0x2B

This read-only command pair returns the filtered battery capacity remaining. This value is not allowed to jump unless *RemainingCapacityUnfiltered( )* reaches empty or full before *RemainingCapacityFiltered( )* does. Units are mAh.

4.15 FullChargeCapacityUnfiltered( ): 0x2C and 0x2D

This read-only command pair returns the compensated capacity of the battery when fully charged. Units are mAh. *FullChargeCapacityUnfiltered( )* is updated at regular intervals.

4.16 FullChargeCapacityFiltered( ): 0x2E and 0x2F

This read-only command pair returns the filtered compensated capacity of the battery when fully charged. Units are mAh. *FullChargeCapacityFiltered( )* is updated at regular intervals. It has no physical meaning and is manipulated to ensure the *StateOfCharge( )* register is smoothed if ***OpConfigB [SMOOTHEN]*** = 1.

4.17 StateOfChargeUnfiltered( ): 0x30 and 0x31

This read-only command pair returns the true state-of-charge. Units are %. *StateOfChargeUnfiltered( )* is updated at regular intervals, and may jump as the gauge updates its predictions dynamically. *StateOfChargeUnfiltered( )* is always calculated as *RemainingCapacityUnfiltered( )* / *FullChargeCapacityUnfiltered( )*, rounded up to the nearest whole percentage point.

Extended Commands

Extended commands offer additional functionality beyond the standard set of commands. They are used in the same manner; however, unlike standard commands, extended commands are not limited to 2-byte words. The number of command bytes for a given extended command ranges in size from single to multiple bytes, as specified in Table 5-1.

Table 5-1. Extended Commands

Name	Command Code	Unit	SEALED Access[1] [2]	UNSEALED Access[1] [2]
OpConfig( )	0x3A and 0x3B	NA	R	R
DesignCapacity( )	0x3C and 0x3D	mAh	R	R
DataClass( ) [2]	0x3E	NA	NA	RW
DataBlock( ) [2]	0x3F	NA	NA	RW
BlockData( )	0x40 through 0x5F	NA	NA	RW
BlockDataCheckSum( )	0x60	NA	NA	RW
BlockDataControl( )	0x61	NA	NA	RW
Reserved	0x62 through 0x7F	NA	R	R

(1) SEALED and UNSEALED states are entered via commands to *Control( )* 0x00 and 0x01
(2) In SEALED mode, data cannot be accessed through commands 0x3E and 0x3F.

5.1 OpConfig( ): 0x3A and 0x3B

SEALED and UNSEALED Access: This command returns the ***Op Config*** Data Memory register setting which is most useful for system-level debug to quickly determine device configuration.

5.2 DesignCapacity( ): 0x3C and 0x3D

SEALED and UNSEALED Access: This command returns the ***Design Capacity*** Data Memory value and is most useful for system-level debug to quickly determine device configuration.

5.3 DataClass( ): 0x3E

UNSEALED Access: This command sets the data class to be accessed. The class to be accessed should be entered in hexadecimal.

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

5.4 DataBlock( ): 0x3F

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

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

5.5 BlockData( ): 0x40 through 0x5F

UNSEALED Access: This data block is the remainder of the 32-byte data block when accessing general block data.

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

5.6 BlockDataChecksum( ): 0x60

UNSEALED Access: This byte contains the checksum on the 32-bytes of block data read or written. The least-significant byte of the sum of the data bytes written must be complemented ([255 – x] , for x the least-significant byte) before being written to 0x60. For a block write, the correct complemented checksum must be written before the *BlockData( )* is transferred to RAM.

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

5.7 BlockDataControl( ): 0x61

UNSEALED Access: This command is used to control the data access mode. Writing 0x00 to this command enables *BlockData( )* to access to RAM.

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

5.8 Reserved – 0x62 through 0x7F

Data Memory

6.1 Data Memory Interface

6.1.1 Accessing the Data Memory

The fuel gauge's Data Memory contains initialization, default, cell status, calibration, configuration, and user information. Most Data Memory parameters reside in volatile RAM that are initialized by associated parameters from ROM. However, some Data Memory parameters are directly accessed from ROM and do not have an associated RAM copy. The Data Memory can be accessed in several different ways, depending on what mode the fuel gauge is operating in and what data is being accessed.

Commonly accessed Data Memory locations, frequently read by a system, are conveniently accessed through specific instructions, already described in Chapter 5, *Extended Commands*. These commands are available when the fuel gauge is either in UNSEALED or SEALED modes.

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

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

Occasionally, a Data Memory class is larger than the 32-byte block size. In this case, the *DataBlock( )* command designates in which 32-byte block the desired locations reside. The correct command address is then given by 0x40 + offset modulo 32. For an example of this type of Data Memory access, see Section 3.1.

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

None of the data written to memory are bounded by the fuel gauge — the values are not rejected by the fuel gauge. Writing an incorrect value may result in hardware failure due to firmware program interpretation of the invalid data.

6.1.2 Access Modes

The fuel gauge provides two access modes, UNSEALED and SEALED, that control the Data Memory access permissions. The default access mode of the fuel gauge is UNSEALED, so the system processor must send a SEALED subcommand after a gauge reset in order to utilize the data protection feature.

6.1.3 SEALING and UNSEALING Data Memory Access

The fuel gauge implements a key-access scheme to transition from SEALED to UNSEALED mode. Once SEALED through the associated subcommand, a unique set of two keys must be sent to the fuel gauge via the *Control( )* command to return to UNSEALED mode. The keys must be sent consecutively, with no other data being written to the *Control( )* register in between.

When in the SEALED mode, the *CONTROL_STATUS [SS]* bit is set; but when the ***Sealed to Unsealed*** keys are correctly received by the fuel gauge, the *[SS]* bit is cleared. The ***Sealed to Unsealed*** key has two identical words stored in ROM with a value of 0x80008000. Then, *Control( )* should supply 0x8000 and 0x8000 (again) to unseal the part.

6.2 Data Types Summary

Table 6-1. Data Type Decoder

Data Type	Length	Min Value	Max Value
B	1	0x00	0xFF
	2	0x00	0xFFFF
	4	0x00	0xFFFF FFFF
I	1	–128	127
	2	–32768	32767
	4	−2,147,483,648	2,147,483,647
Sx		1-byte string	X-byte string
U	1	0	255
	2	0	65535
	4	0	4,294,967,295

6.3 Data Memory Summary Tables

Table 6-2. Data Memory Summary—Configuration Class

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
2	Safety	0	OT	I2	0	1200	550	0.1 °C
		2	UT	I2	–200	100	0	0.1 °C
		4	T Hysteresis	I1	0	127	50	0.1 °C
36	Charge Termination	0	Min Taper Capacity	I2	0	1000	25	mAh
		2	Current Taper Window	U1	0	60	40	s
		5	FC Set %	I1	–1	100	–1	%
		6	FC Clear %	I1	0	100	98	%
		7	DODatEOC Delta T	I2	0	1000	50	0.1 °C
48	Data	0	Design Voltage	I2	2000	5000	3600	mV
49	Discharge	0	SOC1 Set Threshold	U1	0	100	10	%
		1	SOC1 Clear Threshold	U1	0	100	15	%
		2	SOCF Set Threshold	U1	0	100	2	%
		3	SOCF Clear Threshold	U1	0	100	5	%
64	Registers	0	Op Config	H2	0x0	0xFFFF	0xB4D8	Flag
		2	OpConfig B	H1	0x0	0xFF	0x0C	Flag
		3	DF Version	U1	0	255	0	Num
68	Power	0	Sleep Time	U1	0	100	20	s
		1	Hibernate I	U2	0	8000	3	mA
		3	Hibernate V	U2	0	5000	2200	mV

Table 6-3. Data Memory Summary—Gas (Fuel) Gauging Class

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	21	Ra Filter	U2	0	1000	1000	Num
		30	ResRelax Time	U2	0	65535	500	s
		32	User Rate-mA	I2	–32768	0	0	mA
		34	User Rate-mW	I2	–32768	0	0	mW
		36	Reserve Cap-mAh	I2	0	9000	0	mAh
		38	Reserve Cap-mWh	I2	0	32767	0	mWh
		47	Max Res Scale	U2	0	32767	3000	Num
		49	Min Res Scale	U2	0	32767	300	Num
		51	Fast Scale Start SOC	U1	0	100	10	%
		54	Min Delta Voltage	I2	0	32767	0	mV
		56	Max Delta Voltage	I2	0	32767	200	mV
		58	Delta V Max Delta	I2	0	100	100	mV
		60	Term V Valid Time	U1	0	100	2	s
		64	DVC DeadBand	U1	0	100	15	mA
		70	RaScl OCV Rst Temp Thresh	U1	0	127	15	°C

Table 6-3. Data Memory Summary—Gas (Fuel) Gauging Class (continued)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	0	Dsg I Rate Threshold	I2	0	2000	167	0.1 h
		2	Chg I Rate Threshold	I2	0	2000	100	0.1 h
		4	Quit I Rate	I2	0	1000	250	0.1 h
		6	Dsg Relax Time	U2	0	65535	60	s
		8	Chg Relax Time	U1	0	255	60	s
		9	Quit Relax Time	U1	0	255	1	s
		10	Transient Factor Charge	U1	0	255	179	Num
		11	Transient Factor Discharge	U1	0	255	179	Num
		12	Max IR Correct	U2	0	1000	400	mV
82	State	0	Qmax Cell 0	I2	0	32767	17203	Num
		2	Load Select/Mode	H1	0x0	0xFF	0x00	Hex
		3	Design Capacity	I2	0	8000	2425	mAh
		5	Design Energy	I2	0	32767	7275	mWh
		7	Default Design Cap	I2	0	32767	2425	mAh
		9	Terminate Voltage	I2	2500	3700	3200	mV
		19	SOCI Delta	U1	0	100	1	%
		20	Taper Rate	I2	0	2000	200	0.1 Hr rate
		22	Taper Voltage	I2	0	5000	4100	mV
		24	Sleep Current	I2	0	1000	10	mA
		26	V at Chg Term	I2	0	5000	4190	mV
		28	Avg I Last Run	I2	–32768	–1	–50	0.1 Hr rate
		30	Avg P Last Run	I2	–32768	–1	–50	0.1 Hr rate
		32	Delta Voltage	I2	0	1000	1	mV
		36	Chem ID	H2	0x0	0xFFFF	0x1202	Flag

Table 6-4. Data Memory Summary—Ra Tables Class

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
85	Ra	0	R_a0 0	I2	0	32767	29	Num
		2	R_a0 1	I2	0	32767	30	Num
		4	R_a0 2	I2	0	32767	34	Num
		6	R_a0 3	I2	0	32767	46	Num
		8	R_a0 4	I2	0	32767	38	Num
		10	R_a0 5	I2	0	32767	32	Num
		12	R_a0 6	I2	0	32767	37	Num
		14	R_a0 7	I2	0	32767	31	Num
		16	R_a0 8	I2	0	32767	32	Num
		18	R_a0 9	I2	0	32767	35	Num
		20	R_a0 10	I2	0	32767	39	Num
		22	R_a0 11	I2	0	32767	39	Num
		24	R_a0 12	I2	0	32767	61	Num
		26	R_a0 13	I2	0	32767	115	Num
		28	R_a0 14	I2	0	32767	200	Num

Table 6-5. Data Memory Summary—Security Class

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
112	Codes	0	Sealed to Unsealed	H4	0x0001 0001	0xFFFF FFFF	0x8000 8000	Hex

6.4 Data Memory Parameter Descriptions

6.4.1 Configuration Class

6.4.1.1 Safety Subclass

6.4.1.1.1 Over-Temperature, Under-Temperature, Temperature Hysteresis

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
2	Safety	0	OT	I2	0	1200	550	0.1 °C
		2	UT	I2	–200	100	0	0.1 °C
		4	T Hysteresis	I1	0	127	50	0.1 °C

An Over-Temperature condition is detected if *Temperature( )* ≥ ***Over Temp*** (default = 55°C) and indicated by setting the *Flags( ) [OT]* bit. The *[OT]* bit is cleared when *Temperature( )* < ***Over Temp*** – ***Temp Hys*** (default = 50°C).

An Under-Temperature condition is detected if *Temperature( )* ≤ ***Under Temp*** (default = 0°C) and indicated by setting the *Flags( ) [UT]* bit. The *[UT]* bit is cleared when *Temperature( )* > ***Under Temp*** + ***Temp Hys*** (default = 5°C).

6.4.1.2 Charge Termination Subclass

6.4.1.2.1 Minimum Taper Capacity, Current Taper Window

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
36	Charge Termination	0	Min Taper Capacity	I2	0	1000	25	mAh
		2	Current Taper Window	U1	0	60	40	s

See Section 6.4.2.3.6 for details on ***Min Taper Capacity*** and ***Current Taper Window***.

6.4.1.2.2 Full Charge Set % (FC Set %), Full Charge Clear % (FC Clear %)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
36	Charge Termination	5	FC Set %	I1	–1	100	–1	%
		6	FC Clear %	I1	0	100	98	%

The *Flags( ) [FC]* bit is set when SOC reaches ***FC Set %*** and is cleared when it drops below ***FC Clear %***.

The *Flags( ) [FC]* bit is set when Primary Charge Termination conditions are met and ***FC Set*** is set to –1%.

See Section 6.4.2.3.6 for details about the Primary Charge Termination Algorithm.

6.4.1.2.3 DOD at EOC Delta Temperature

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
36	Charge Termination	7	DODatEOC Delta T	I2	0	1000	50	0.1 °C

During relaxation and charge start, *RemainingCapacity( )* = *FullChargeCapacity( )* – Qstart. But with temperature decreases, Qstart can become much smaller than old *FullChargeCapacity( )* resulting in an over-estimation of *RemainingCapacity( )*. To improve accuracy, gauging predictions are updated when the temperature has changed by at least ***DODatEOC Delta T*** since the last simulation.

6.4.1.3 Data Subclass

6.4.1.3.1 Design Voltage

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
48	Data	0	Design Voltage	I2	2000	5000	3600	mV

The ***Design Voltage*** is the nominal battery capacity, usually printed on the battery label. Typical values are 3600 mV or 3700 mV. It is typically multiplied by ***Design Capacity*** to calculate ***Design Energy***.

6.4.1.4 Discharge Subclass

6.4.1.4.1 State-of-Charge 1 Set or Clear Threshold (SOC1 Set Threshold, SOC1 Clear Threshold)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
49	Discharge	0	SOC1 Set Threshold	U1	0	100	10	%
		1	SOC1 Clear Threshold	U1	0	100	15	%

When *StateOfCharge( )* falls to or below the first capacity threshold as specified in ***SOC1 Set Threshold***, the *Flags( ) [SOC1]* bit is set. This bit is cleared once *StateOfCharge( )* rises to or above ***SOC1 Clear Threshold***.

These values are up to the user's preference.

6.4.1.4.2 State-of-Charge Final Set or Clear Threshold (SOCF Set Threshold, SOCF Clear Threshold)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
49	Discharge	2	SOCF Set Threshold	U1	0	100	2	%
		3	SOCF Clear Threshold	U1	0	100	5	%

When *StateOfCharge( )* falls to or below the final capacity threshold as specified in ***SOCF Set Threshold***, the *Flags( ) [SOCF]* bit is set. This bit is cleared once *StateOfCharge( )* rises to or above ***SOCF Clear Threshold***. The *[SOCF]* bit serves as the final discharge warning.

These values are up to the user's preference.

6.4.1.5 Registers Subclass

6.4.1.5.1 Operation Configuration

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
64	Registers	0	Op Config	H2	0x0	0xFFFF	0xB4D8	Flag

Table 6-6. Op Config Register Definition

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	RSVD1	ADOLP	BIE	RSVD1	GPIOPOL	ADMLP	RSVD	RSVD
Default =	1	0	1	1	0	1	0	0
	0xB4							
Low Byte	VCONSEN	RSVD1	RSVD	RMFCC	FCONVEN	BATLOWEN	NODVAVG	TEMPS
Default =	1	1	0	1	1	0	0	0
	0xD8							

High Byte

RSVD1 = This bit is reserved. The default setting of 1 should be used.

ADOLP = If cleared (default), the ADC conversion time is largest to achieve maximum accuracy. If set, power consumption will be slightly reduced but the shorter ADC conversion time could result in reduced accuracy of measurements.

BIE = Battery Insertion Enable. If set, the battery insertion is detection via the BIN pin input. If cleared, the detection relies on the host to issue *BAT_INSERT* subcommand to indicate battery presence in the system.

RSVD1 = This bit is reserved. The default setting of 1 should be used.

GPIOPOL = GPOUT pin is active-high if set or active-low if cleared.

ADMLP = I set (default), the ADC is in low-power conversion mode.

RSVD = Bits 1 and 0 are reserved.

Low Byte

VCONSEN = Enables voltage consistency check. True when set. Default is recommended.

RSVD1 = This bit is reserved. The default setting of 1 should be used.

RSVD = Bit 5 is reserved

RMFCC = RM is updated with the value from FCC on valid charge termination. True when set.

FCONVEN = Enables Fast SOC Convergence. True when set. Default is recommended.

BATLOWEN = If set, the BAT_LOW function for GPOUT pin is selected. If cleared, the SOC_INT function is selected for GPOUT.

NODVAVG = True when set. Default is recommended.

TEMPS = Selects the temperature source. Enables the host to write *Temperature( )* if set. If cleared, the internal temperature sensor is used for *Temperature( )*.

6.4.1.5.2 Operation Configuration B

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
64	Registers	2	OpConfig B	H1	0x0	0xFF	0x0C	Flag

Table 6-7. Op ConfigB Register Definition

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RSVD	RSVD	DEF_SEAL	RSVD	RCJUMPOK	SMTHEN	RSVD	RSVD
Default =	0	0	0	0	1	1	0	0
	0x0C							

RSVD = Bits 7:6 are reserved.
DEF_SEAL= Seals the device. True when set.
RSVD = Bit 4 is reserved.
RCJUMPOK = Allows SOC to jump to true if the gauge is in relaxation and a resimulation due to temperature causes the jump.
SMTHEN = Enables the SOC smoothing feature. True when set.
RSVD = Bits 1 and 0 are reserved.

6.4.1.5.3 DF Version

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
64	Registers	3	DF Version	U1	0	255	0	Num

This register is not used by the fuel gauge. It is recommended to be used for storing a version number of the gauge configuration parameter set. This is useful for version control of the golden file.

6.4.1.6 Power Subclass

6.4.1.6.1 Sleep Time

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
68	Power	0	Sleep Time	U1	0	100	20	s

It is not recommended to modify this parameter. It represents the number of seconds that the gauge will stay in SLEEP mode before periodically waking to update its measurements and register set. To disable SLEEP, set this value to 0.

6.4.1.6.2 Hibernate Current

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
68	Power	1	Hibernate I	U2	0	8000	3	mA

When |*EffectiveCurrent( )*| < ***Hibernate I***, the gauge enters HIBERNATE mode if *CONTROL_STATUS [HIBERNATE]* bit is set and cell is relaxed. This setting should be below any normal application currents. Gauging stops while in this mode.

6.4.1.6.3 Hibernate Voltage

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
68	Power	3	Hibernate V	U2	0	5000	2200	mV

When |Voltage| < ***Hibernate V***, the gauge enters HIBERNATE mode. The *CONTROL_STATUS [HIBERNATE]* bit is ignored if the voltage is below this threshold. This setting should be below any normal application voltage.

6.4.2 Gas (Fuel) Gauging Class

6.4.2.1 IT Cfg Subclass

6.4.2.1.1 Ra Filter

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	21	Ra Filter	U2	0	1000	1000	Num

This value should not be changed.

6.4.2.1.2 Simulation Res Relax Time (ResRelax Time)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	30	ResRelax Time	U2	0	65535	500	s

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

6.4.2.1.3 User Rate-mA

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	32	User Rate-mA	I2	–32768	0	0	mA

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

6.4.2.1.4 User Rate-mW

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	34	User Rate-mW	I2	–32768	0	0	mW

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

6.4.2.1.5 Reserve Capacity (Reserve Cap-mAh)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	36	Reserve Cap-mAh	I2	0	9000	0	mAh

Reserve Cap-mAh determines how much actual remaining capacity exists after reaching 0 *RemainingCapacity( )* before ***Terminate Voltage*** is reached. This register is only used if ***Load Mode*** = 0 (constant current). A no-load rate of compensation is applied to this reserve capacity. This is a specialized function to allow time for a controlled shutdown after 0 *RemainingCapacity( )* is reached.

6.4.2.1.6 Reserve Energy (Reserve Cap-mWh)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	38	Reserve Cap-mWh	I2	0	32767	0	mWh

Reserve Cap-mWh determines how much actual remaining capacity exists after reaching 0 *RemainingCapacity( )* before ***Terminate Voltage*** is reached. This register is only used if ***Load Mode*** = 1 (constant-power). A no-load rate of compensation is applied to this reserve capacity. This is a specialized function to allow time for a controlled shutdown after 0 *RemainingCapacity( )* is reached.

6.4.2.1.7 Maximum Resistance Scale, Minimum Resistance Scale

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	47	Max Res Scale	U2	0	32767	3000	Num
		49	Min Res Scale	U2	0	32767	300	Num

Min Res Scale and ***Max Res Scale*** specify the allowed change in simulated resistance during Fast Ra Scaling algorithm. A value of 1000 corresponds to 1× and a value of 200 corresponds to 0.2×.

6.4.2.1.8 Fast Scale Start SOC

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	51	Fast Scale Start SOC	U1	0	100	10	%

Fast Scale Start SOC specifies SOC thresholds for Fast Ra Scaling activation. Fast Ra Scaling is activated when either of the following conditions is true:

- SOC < ***Fast Scale Start SOC***
- Voltage < (***Terminate Voltage*** + 200 mV)

6.4.2.1.9 Minimum Delta Voltage, Maximum Delta Voltage

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	54	Min Delta Voltage	I2	0	32767	0	mV
		56	Max Delta Voltage	I2	0	32767	200	mV

These parameters are the lower and upper bounds on the value that ***Delta Voltage*** is allowed to learn.

6.4.2.1.10 Delta Voltage Maximum Delta

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	58	Delta V Max Delta	I2	0	100	100	mV

This parameter limits the amount of change allowed for each update of ***Delta Voltage***. ***Delta Voltage*** will only be updated in Data Memory after a discharge of at least 500 seconds has occurred and stopped.

6.4.2.1.11 Terminate Voltage Valid Time

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	60	Term V Valid Time	U1	0	100	2	s

The voltage must dip below ***Terminate Voltage*** for at least this many seconds before *RemainingCapacity( )* and *StateOfCharge( )* will be forced to zero.

6.4.2.1.12 DVC Dead Band

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	64	DVC DeadBand	U1	0	100	15	mA

The recommended value for ***DVC DeadBand*** is C / 100, or ***Design Capacity*** / 100. Any estimated current below ***DVC DeadBand*** will result in *EffectiveCurrent()* = 0 mA.

6.4.2.1.13 RaScl OCV Reset Temperature Threshold

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	70	RaScl OCV Rst Temp Thresh	U1	0	127	15	°C

The default value is sufficient for most applications.

6.4.2.2 Current Thresholds Subclass

6.4.2.2.1 Discharge and Charge Detection Rate Threshold, Quit Current Rate and Relax Time, Discharge and Charge Relax Time

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	0	Dsg I Rate Threshold	I2	0	2000	167	0.1 h
		2	Chg I Rate Threshold	I2	0	2000	100	0.1 h
		4	Quit I Rate	I2	0	1000	250	0.1 h
		6	Dsg Relax Time	U2	0	65535	60	s
		8	Chg Relax Time	U1	0	255	60	s
		9	Quit Relax Time	U1	0	255	1	s

The gauging algorithm transitions between three states: discharge, charge, and relaxation modes of operation. During charge mode, the *[DSG]* bit of the *Flags( )* register is cleared, and during discharge and relaxation mode it is set. Entry and exit for each mode is controlled by six parameters in the Current Thresholds Subclass.

The discharge current threshold can be calculated as ***Design Capacity*** / (***Dsg I Rate Threshold*** × 0.1). The default is effectively C / 16.7.

The charge current threshold can be calculated as ***Design Capacity*** / (***Chg I Rate Threshold*** × 0.1). The default is effectively C / 10.

The quit current threshold can be calculated as ***Design Capacity*** / (***Quit I Rate*** × 0.1). The default is effectively C / 25.

Charge mode is exited and relaxation mode is entered when *EffectiveCurrent( )* goes below the quit current threshold for the number of seconds specified in ***Charge Relax Time*** (default 60 s). Discharge mode is entered when *EffectiveCurrent( )* goes below the discharge current threshold for ***Quit Relax Time*** (default 1 s). Discharge mode is exited and relaxation mode is entered when *EffectiveCurrent( )* goes above negative quit current threshold for ***Dsg Relax Time*** (default 60 s). Charge mode is entered when *EffectiveCurrent( )* goes above the charge current threshold for ***Charge Relax Time*** (default 60 s).

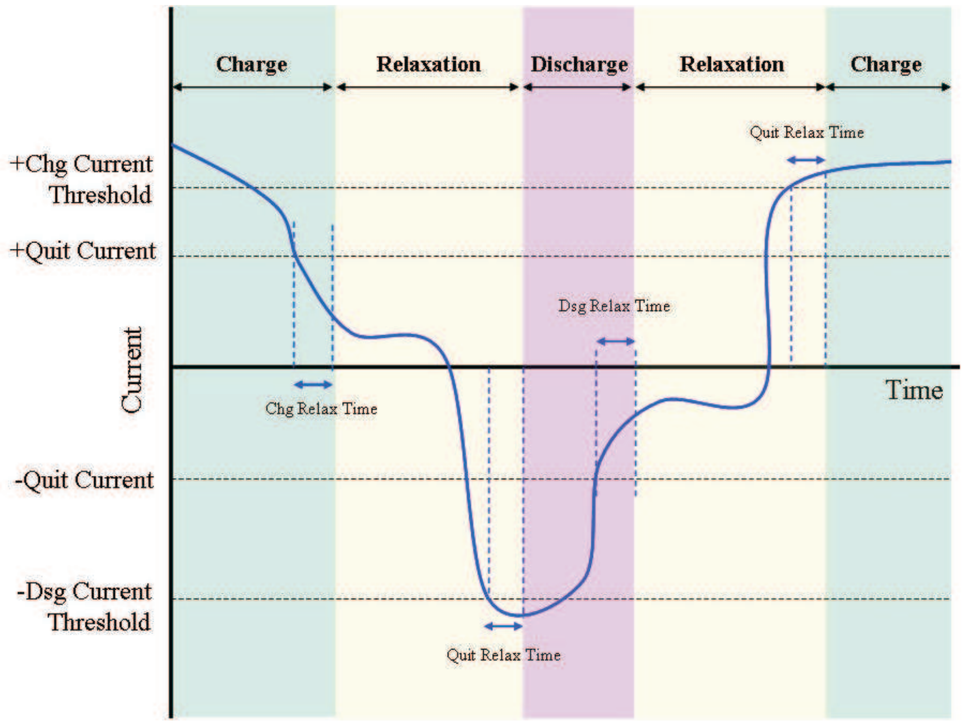

Figure 6-1. Example of Algorithm Operation Mode Changes With Varying *DataRAM.EffectiveCurrent( )*

6.4.2.2.2 Transient Factor Charge, Transient Factor Discharge

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	10	Transient Factor Charge	U1	0	255	179	Num
		11	Transient Factor Discharge	U1	0	255	179	Num

The ***Transient Factor Charge*** and ***Transient Factor Discharge*** parameters provide an adjustment of the computed resistance due to transient voltage readings upon pack insertion for either charge or discharge conditions. The values range from 0 to 255 (default 179) and used as a scaling factor / 256 to adjust resistance such that R_adj = R × Transient_Factor / 256.

6.4.2.2.3 Max IR Correct

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
81	Current Thresholds	12	Max IR Correct	U2	0	1000	400	mV

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

6.4.2.3 State Subclass

6.4.2.3.1 Qmax Cell 0

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	0	Qmax Cell 0	I2	0	32767	17203	Num

Qmax describes the maximum chemical capacity of the cell, and is determined by comparing states of charge before and after applying the load with the amount of charge passed. It corresponds to capacity at a low rate (~ C/20) of discharge. To translate the Qmax register to mAh units, use this formula:

Qmax (mAh) = ***Qmax Cell 0*** × ***Design Capacity*** / 2^{14}

6.4.2.3.2 Load Select and Mode

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	2	Load Select/Mode	H1	0x0	0xFF	0x00	Hex

Load Mode configures the fuel gauge to use either a constant-current or constant-power model for the gauging algorithm. When ***Load Mode*** is 0, the Constant Current Model is used. When 1 (default), the Constant Power Model is used. The *CONTROL_STATUS [LDMD]* bit reflects the status of ***Load Mode***.

Load Select is used in conjunction with ***Load Mode*** to define the type of load model that computes the load-compensated capacity in the gauging algorithm.

If ***Load Mode*** = 0 (constant current), then the following options are available:

Table 6-8. Load Select/Mode Parameter Encoding

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Load Mode	RSVD	RSVD	RSVD	RSVD	Load Select[2:0]		
Default =	0	0	0	0	0	0	0	0
	0x00							

Load Mode = Bit 7 contains the value for ***Load Mode***. See Table 6-9 and Table 6-10 for operational details.

RSVD = Bits 6:3 are reserved. Set to 0 for proper operation.

Load Select[2:0] = Bits 2:0 contain the value for ***Load Select***. See Table 6-9 and Table 6-10 for operational details. Default is 1.

Table 6-9. Current Model Used When Load Mode = 0 (Default)

Load Select Value	Current Model Used
0 (default)	Average discharge current from previous cycle: There is an internal register that records the average discharge current through each entire discharge cycle. The previous average is stored in ***Avg I Last Run***.
1	Present average discharge current: This is the average discharge current from the beginning of this discharge cycle until present time.
2	Average current: based off the *EffectiveCurrent( )*
3	Current: based off of a low-pass-filtered version of *EffectiveCurrent( )* (τ = 14 s)
4	Design capacity / 5: C Rate based off of ***Design Capacity*** / 5 or a C / 5 rate in mA.
All others	Reserved

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

Table 6-10. Current Model Used When Load Mode = 1

Load Select Value	Power Model Used
0 (default)	Average discharge power from previous cycle: There is an internal register that records the average discharge power through each entire discharge cycle. The previous average is stored in ***Avg P Last Run***.
1	Present average discharge power: This is the average discharge power from the beginning of this discharge cycle until present time.
2	Average current × voltage: based off the *EffectiveCurrent( )* and *Voltage( )*.
3	Current × voltage: based off of a low-pass-filtered version of *EffectiveCurrent( )* (τ = 14 s) and *Voltage( )*.
4	Design energy / 5: C Rate based off of ***Design Energy*** / 5 or a C / 5 rate in mA .
5	Reserved
6	User_Rate-mW: Use the value in ***User Rate-mW***. This provides a user-configurable load model.
All others	Reserved

6.4.2.3.3 Design Capacity, Design Energy, Default Design Capacity

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	3	Design Capacity	I2	0	8000	2425	mAh
		5	Design Energy	I2	0	32767	7275	mWh
		7	Default Design Cap	I2	0	32767	2425	mAh

Design Capacity is used for compensated battery capacity remaining and capacity when fully charged calculations are done by the gauge. It is also used for constant-current model for gauging algorithm when ***Load Mode*** is 0 (constant-current) and ***Load Select*** is 4 (***Design Capacity*** / 5 for constant discharge). The *CONTROL_STATUS [LDMD]* bit indicates the gauging algorithm is assuming the constant-current model when cleared.

Design Energy is used for compensated battery capacity remaining and capacity when fully charged calculations are done by the gauge. It is also used for constant-power model for the gauging algorithm when ***Load Mode*** is 1 (constant-power) and ***Load Select*** is 4 (***Design Energy*** / 5 for constant discharge). The *CONTROL_STATUS [LDMD]* bit indicates the gauging algorithm is assuming the constant-power model when set.

These values should be set based on the battery specification. See the data sheet from the battery manufacturer.

Default Design Cap contains the capacity of the pack originally used to generate the *CHEM_ID* data and is used along with ***Design Capacity*** or ***Design Energy*** to scale data for the gauging algorithm. ***Default Design Cap*** should never be modified by the user. It will automatically update if the ALT_CHEM1 or ALT_CHEM2 subcommands are used to change the battery profile in use.

6.4.2.3.4 Terminate Voltage

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	9	Terminate Voltage	I2	2500	3700	3200	mV

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

This register is application dependent. It should be set based on battery cell specification to prevent damage to the cells or the absolute minimum system input voltage, taking into account impedance drop from the PCB traces, FETs, and wires.

Terminate Voltage should typically be set to the lowest possible value at which the system will operate, in order to maximize run-time and capacity extracted from the battery. The gauge will automatically learn the load spikes characteristic of the system during operation and store it in ***Delta Voltage***, thereby adding margin to capacity predictions when necessary. The effect is that StateOfCharge( ) will reach 0% at ***Terminate Voltage*** + ***Delta Voltage*** and *RemainingCapacity( )* will represent the amount of charge available from the present depth of discharge until that voltage is reached.

6.4.2.3.5 SOCI Delta

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	19	SOCI Delta	U1	0	100	1	%

The ***SOCI Delta*** parameter is active when the SOC_INT function is activated when ***Op Config [BATLOWEN]*** is cleared. In this case, the GPOUT pin generates interrupts with an approximate 1-ms pulse width under various conditions as described in Table 2-1.

6.4.2.3.6 Taper Rate, Taper Voltage

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	20	Taper Rate	I2	0	2000	200	0.1 Hr rate
		22	Taper Voltage	I2	0	5000	4100	mV

Taper Rate is used in the Primary Charge Termination Algorithm. *EffectiveCurrent( )* is integrated over each of the two 40-second periods separately and averaged separately to determine two averages (IRateAvg1, IRateAvg2).

The ***Taper Voltage*** threshold defines the minimum voltage necessary for as a qualifier for detection of charge termination.

Three requirements must be met to qualify for Primary Charge Termination:

- During two consecutive periods of 40 seconds:
  IRateAvg1 < ***Taper Rate*** and IRateAvg2 < ***Taper Rate***
- During the same periods: Accumulated change in capacity > ***Min Taper Capacity*** per 40 seconds
- *Voltage( )* > ***Taper Voltage***

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

6.4.2.3.7 Sleep Current

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	24	Sleep Current	I2	0	1000	10	mA

When *EffectiveCurrent( )* is less than ***Sleep Current*** or greater than (–)***Sleep Current***, the gauge enters SLEEP mode if the feature is enable by setting ***Sleep Time*** > 0. The default setting of ***Sleep Time*** is 20 seconds.

This setting should be below any normal application currents.

6.4.2.3.8 Voltage at Charge Termination (V at Chg Term)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	26	V at Chg Term	I2	0	5000	4190	mV

V at Chg Term should be initialized to the typical charging voltage of the system. Typically, if using the default battery profile (CHEM_ID = 0x1202), the charging voltage will be 4200 mV and the default value of ***V at Chg Term*** can be used. If using ALT_CHEM1 (CHEM_ID = 0x1210) then ***V at Chg Term*** could be initialized to 4300 mV. If using ALT_CHEM2 (CHEM_ID = 0x354), ***V at Chg Term*** could be initialized to 4350 mV.

V at Chg Term will automatically be updated and learned by the gauge during system operation whenever charge termination is detected. It represents the full charge point for a given system which can vary from charger to charger and also depends on temperature and other factors.

6.4.2.3.9 Average Current Last Run (Avg I Last Run)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	28	Avg I Last Run	I2	–32768	–1	–50	0.1 Hr rate

The gauge logs the current averaged from the beginning to the end of each discharge period. It stores this average current from the previous discharge in this register. This register can be initialized to a typical system current load. It is updated by the gauge after a discharge lasts for at least 500 seconds and stops. The default represents a C/5 load. It should always be a negative value.

6.4.2.3.10 Average Power Last Run (Avg P Last Run)

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	30	Avg P Last Run	I2	–32768	–1	–50	0.1 Hr rate

The gauge logs the power averaged from the beginning to the end of each discharge period. It stores this average power from the previous discharge in this register. To get a correct average power reading the gauge continuously multiplies current times voltage to get power. It then logs this data to derive the average power. This register can be initialized to a typical system power load. It is updated by the gauge after a discharge lasts for at least 500 seconds and stops. The default represents a C/5 load. It should always be a negative value.

6.4.2.3.11 Delta Voltage

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	32	Delta Voltage	I2	0	1000	1	mV

The gauge stores the maximum difference of Voltage during short load spikes and normal load, so the gauging algorithm can calculate *RemainingCapacity( )* for pulsed loads. It is added to ***Terminate Voltage*** for simulations.

6.4.2.3.12 Chemistry ID

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
82	State	36	Chem ID	H2	0x0	0xFFFF	0x1202	Flag

The fuel gauge updates information on chemical depth of discharge (DOD 0) based on open-circuit voltage (OCV) readings when in a relaxed state. DOD is found by correlating DOD with OCV using a predefined table DOD (OCV,T) stored as reserved Data Memory parameters. The table is specific for a particular chemistry and can be identified by reading the chemistry ID through sending *CHEM_ID* subcommand 0x0008, then reading ***Chem ID***. This Data Memory parameter should not be changed directly. It will automatically update if the ALT_CHEM1 or ALT_CHEM2 subcommand is used.

6.4.3 Ra Tables Class

6.4.3.1 R_a0 Data Table

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
85	Ra	0	R_a0 0	I2	0	32767	29	Num
		2	R_a0 1	I2	0	32767	30	Num
		4	R_a0 2	I2	0	32767	34	Num
		6	R_a0 3	I2	0	32767	46	Num
		8	R_a0 4	I2	0	32767	38	Num
		10	R_a0 5	I2	0	32767	32	Num
		12	R_a0 6	I2	0	32767	37	Num
		14	R_a0 7	I2	0	32767	31	Num
		16	R_a0 8	I2	0	32767	32	Num
		18	R_a0 9	I2	0	32767	35	Num
		20	R_a0 10	I2	0	32767	39	Num
		22	R_a0 11	I2	0	32767	39	Num
		24	R_a0 12	I2	0	32767	61	Num
		26	R_a0 13	I2	0	32767	115	Num
		28	R_a0 14	I2	0	32767	200	Num

These resistance profiles are used by the gauge for the gauging algorithm. The only reason this data is displayed and accessible is to give the user the ability to update the resistance data on golden image files. This resistance profile description is for information purposes only. It is not intended to give a detailed functional description for the gauge resistance algorithms.

6.4.4 *Security Class*

6.4.4.1 Codes Subclass

6.4.4.1.1 Sealed to Unsealed

Subclass ID	Subclass	Offset	Name	Type	Value			Unit
					Min	Max	Default	
112	Codes	0	Sealed to Unsealed	H4	0x0001 0001	0xFFFF FFFF	0x8000 8000	Hex

The fuel gauge implements a key-access scheme to transition from SEALED to UNSEALED mode. Once SEALED via the associated subcommand, a unique set of two keys must be sent to the fuel gauge via the *Control( )* command to return to UNSEALED mode. The keys must be sent consecutively, with no other data being written to the *Control( )* register in between.

When in the SEALED mode, the *CONTROL_STATUS [SS]* bit is set; but when the ***Sealed to Unsealed*** keys are correctly received by the fuel gauge, the *[SS]* bit is cleared. The ***Sealed to Unsealed*** key has two identical words stored in ROM with a value of 0x8000 8000. Then, *Control( )* should supply 0x8000 and 0x8000 (again) to unseal the part.

Glossary

ACK	Acknowledge character
ADC	Analog-to-digital converter
BI	Battery insert
CE	Chip enable
Charge Mode	Refers to a mode to where the gauge read *EffectiveCurrent( )* > ***Chg I Rate Threshold*** for at least 1 second.
Clear	Refers to a bit in a register becoming a logic LOW or 0. The bqStudio software represents a clear bit with the color green.
Discharge Mode	Refers to a mode where the gauge read *EffectiveCurrent( )* < (–)***Dsg I Rate Threshold*** for at least 1 second.
DOD	Depth of discharge in percent as related to Qmax. 100% corresponds to empty battery.
DOD0	Depth of discharge that was looked up in the DOD (OCV) table based on OCV measurement in relaxed state.
EOC	End of charge
FC	Fully charged
FCC	Full charge capacity. Total capacity of the battery compensated for present load current, temperature, and aging effects (reduction in chemical capacity and increase in internal impedance).
Flag	This word usually represents a read-only status bit that indicates some action has occurred or is occurring. This bit typically cannot be modified. The flags are set and cleared automatically by the fuel gauge.
GPIO	General-purpose input output
IC	Integrated circuit
IO	Input or output
I^2C	Inter-integrated circuit two-wire serial communication protocol
LDO	Low dropout voltage regulator
LSB	Least significant bit
MAC	Manufacturer access command or control command
mAh	Milliamp-hour
MSB	Most significant bit
mWh	Milliwatt-hour
NACK	Negative acknowledge character
OCV	Open-circuit voltage. Voltage measured on fully-relaxed battery with no load applied.
OTC	Overtemperature in charge
OTD	Overtemperature in discharge
POR	Power-on reset
Qmax	Maximum chemical capacity
Relaxation Mode	Refers to a mode to where the gauge read *EffectiveCurrent( )* < ***Quit I Rate*** for at least 60 seconds.
RM	Remaining capacity
RW	Read or write
SCL	Serial clock: programmable serial clock used in the I^2C interface
SDA	Serial data: serial data bus in the I^2C interface
SE	Shutdown enable
Set	Refers to a bit in a register becoming a logic high or 1. The bqStudio software represents a set bit with the color red.
SOC	State-of-charge in percent related to FCC
SOC1	State-of-charge initial
SOCF	State-of-charge final
System	The word system is sometimes used in this document. When used, it always means a host system that is consuming current from the battery pack.

IMPORTANT NOTICE

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265