BQ27427

Technical Reference Manual

Literature Number: SLUUCD5
JANUARY 2023

Table of Contents

About This Manual

This is a detailed technical reference manual (TRM) for using and configuring the BQ27427 battery fuel gauge. This TRM should complement but not supersede information in the *BQ27427 System-Side Impedance Track™ Fuel Gauge with an Integrated Sense Resistor Data Sheet*.

1.1 Notational 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

1.2 Trademarks

Impedance Track™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

1.3 Glossary

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

This page intentionally left blank.

Chapter 1
General Description

The BQ27427 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).

Unlike some other Impedance Track™ fuel gauges, the BQ27427 cannot be programmed with specific battery chemistry profiles. Three pre-defined chemistry profiles (shown below) are available in the device memory. For many battery types and applications, these profiles are sufficient matches from a gauging perspective.

Chem ID	Voltage
3230	4.35 V (default)
1202	4.2 V
3142	4.4 V

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, as 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 the system to the 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 high-accuracy, fuel gauging prediction is Texas Instruments proprietary Impedance Track algorithm. This algorithm uses cell measurements, characteristics, and properties to create SOC predictions that can achieve high accuracy across a wide variety of operating conditions and over the lifetime of the battery.

The fuel gauge measures the charging and discharging of the battery by monitoring the voltage across a small-value, integrated sense resistor. Cell impedance is computed based on current, open-circuit voltage (OCV), and cell voltage under loading 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, 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.

This page intentionally left blank.

Chapter 2

Functional Description

2.1 Fuel Gauging

The BQ27427 battery fuel gauge measures the cell voltage, temperature, and current to determine battery SOC. The fuel gauge monitors the charging and discharging of the battery by sensing the voltage across an integrated sense resistor (7 mΩ, typical) between the BAT and SRX pins. By integrating the charge passing through the battery, the battery SOC is adjusted during the charging or discharging of the battery.

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 the 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. The fuel gauge acquires and updates the battery-impedance profile during normal battery usage. The 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 fuel gauge has two flags, *[SOC1]* and *[SOCF]*, accessed by the *Flags()* command that warn when the battery SOC has fallen to critical levels. When *StateOfCharge()* falls below the first capacity threshold, as specified in ***SOC1 Set Threshold***, the *[SOC1]* (state-of-charge initial) flag is set. The flag is cleared once *StateOfCharge()* rises above ***SOC1 Set Threshold***. All units are in %.

When *StateOfCharge()* falls below the second capacity threshold, ***SOCF Set Threshold***, the *[SOCF]* (state-of-charge final) 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. All units are in %.

2.2 Temperature Measurement

Temperature information for the fuel gauge can be configured from three separate sources:

1. Internal Temperature Sensor
2. External Temperature Sensor (Thermistor)
3. Host Commanded Temperature

The ***OpConfig [TEMPS]*** bits can be used to select the source, as shown below:

OpConfig [TEMPS]	Temperature Source
00	Internal Temperature Sensor is used as Temperature Source.
01	External Thermistor is used as Temperature Source.
10	Host written Temperature is used as Temperature Source.

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 measures current by sensing the voltage across an integrated sense resistor (7 mΩ, typical). Internally, voltage passes through a gain stage before conversion by the coulomb counter. The current measurement data is available through the *AverageCurrent()* command.

2.4 Operating Modes

The fuel gauge has different operating modes: POR, INITIALIZATION, NORMAL, CONFIG UPDATE, and SLEEP. Upon powering 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.

2.4.1 SHUTDOWN Mode

In SHUTDOWN mode, the LDO output is disabled so internal power and all RAM-based volatile data are lost. The host can command the gauge to immediately enter SHUTDOWN mode by first unsealing the gauge and then enabling the mode with a *SHUTDOWN_ENABLE* subcommand (Section 5.1.11) followed by the *SHUTDOWN* subcommand (Section 5.1.12). To exit SHUTDOWN mode, the GPOUT pin must be raised from logic low to logic high for at least 200 µs.

2.4.2 POR and INITIALIZATION Modes

Upon a POR, the fuel gauge copies ROM-based configuration defaults to RAM and begins INITIALIZATION mode where essential data is initialized. The occurrence of a POR or a *Control() RESET* subcommand will set 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. In addition, *CONTROL_STATUS [QMAX_UP]* and *[RES_UP]* bits are cleared to allow unfiltered learning of Qmax and impedance. Completion of INITIALIZATION mode is indicated by the *CONTROL_STATUS [INITCOMP]* bit.

2.4.3 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 *Control() SET_CFGUPDATE* subcommand to enter the CONFIG UPDATE mode. Operation in this mode is indicated by the *Flags() [CFGUPMODE]* status bit. In this mode, fuel gauging is suspended while the host uses the extended data commands to modify the configuration data blocks. To resume fuel gauging, the host must send a *Control() SOFT_RESET* subcommand to exit the CONFIG UPDATE mode which clears both *Flags() [ITPOR]* and *[CFGUPMODE]* bits. After a timeout of approximately 240 seconds (4 minutes), the gauge will automatically exit the CONFIG UPDATE mode if it has not received a *SOFT_RESET* subcommand from the host.

2.4.4 NORMAL Mode

The fuel gauge is in NORMAL mode when not in any other power mode. During this mode, *AverageCurrent()*, *Voltage()*, and *Temperature()* measurements are taken once per second, and the interface data set is updated. 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 Impedance Track algorithm minimizes the time the fuel gauge remains in this mode.

2.4.5 SLEEP Mode

SLEEP mode is entered automatically if the feature is enabled (***OpConfig [SLEEP]*** = 1) and *AverageCurrent()* 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 may perform an ADC autocalibration to minimize the offset.

During SLEEP mode, the fuel gauge remains in a very-low-power idle state and automatically wakes up briefly every 48seconds to take data measurements.

After taking the measurements on the 20-second interval, the fuel gauge will exit SLEEP mode when *AverageCurrent()* rises above ***Sleep Current*** (default = 10 mA). Alternatively, an early wake-up before the 20-second internal is possible if the instantaneous current detected by an internal hardware comparator is above an approximate threshold of ±30 mA.

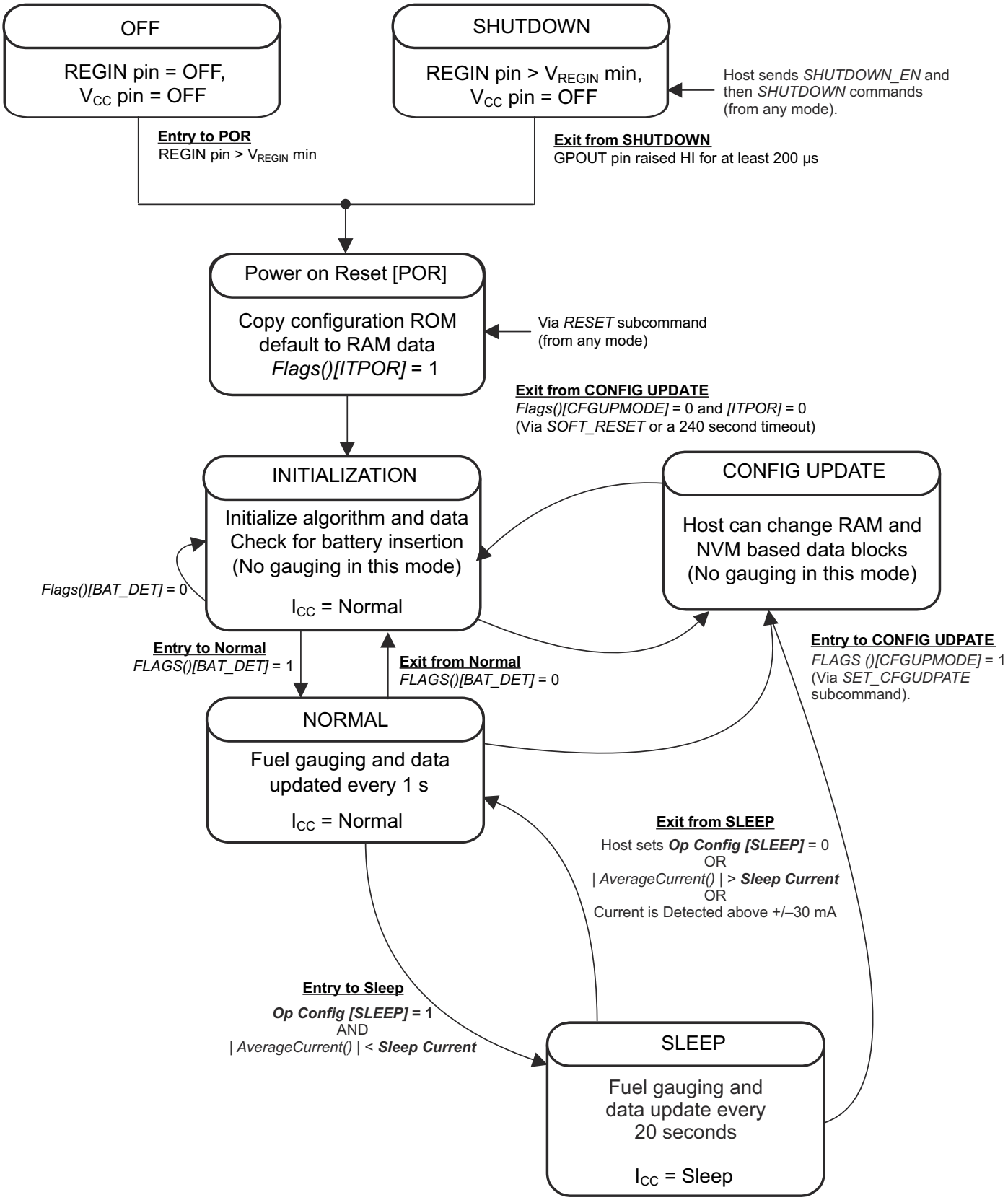


Figure 2-1. Power Mode Diagram

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 ***OpConfig [GPIO_POL]*** bit. The function is defined by the ***OpConfig [BATLOWEN]*** bit. If the bit is set, the Battery Low Indicator (BAT_LOW) function is selected for the GPOUT pin. If it is cleared, the SOC interrupt (SOC_INT) function is selected for the GPOUT pin.

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]*** = 1. The polarity can be flipped by setting ***[GPIOPOL]*** = 0.

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 a 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	***OpConfig [BIE]*** bit is set.	1 ms	When battery removal is detected by the BIN pin.
Initialization Complete	Always	1 ms	After initial gauge predictions are updated upon exit from POR, the *ControlStatus() [INITCOMP]* bit is set.
PULSE_SOC_INT command	Subcommand	1 ms	Instructs the fuel gauge to pulse the GPOUT pin for approximately 1 ms with 1 second of receiving the command. Mostly used for debug purposes.

2.5.2 Battery Detection (BIN)

The function of ***OpConfig [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 Impedance Track fuel gauging. When battery insertion is not detected, the fuel gauge remains in INITIALIZATION mode.

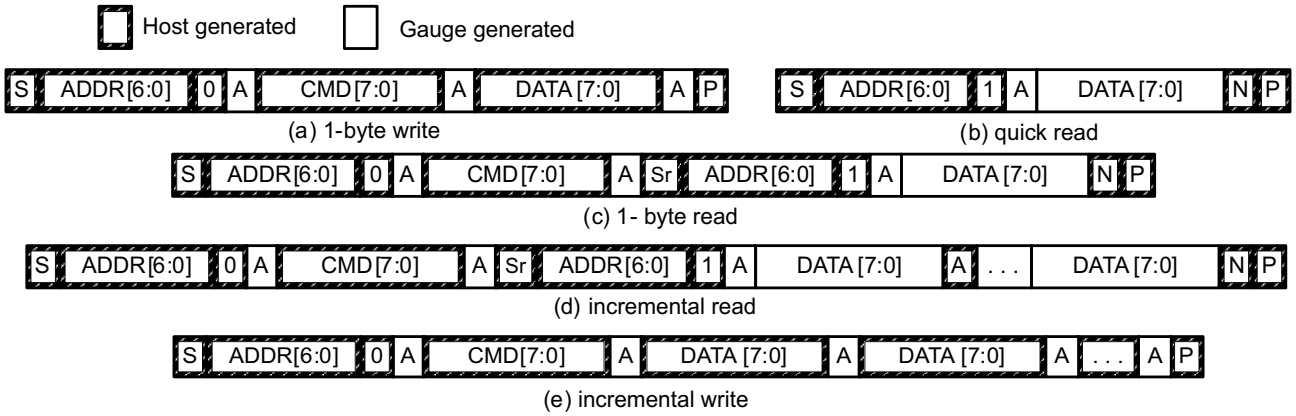
Table 2-2. Battery Detection

OpConfig [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 the 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.	Host sends *BAT_REMOVE* subcommand to signal battery removal.

The BQ27427 can communicate with a host with an I²C interface with both 100-KHz and 400-KHz modes.

3.1 I²C Interface

The fuel gauge supports the standard I²C read, incremental read, quick read, one-byte write, and incremental write functions. The 7-bit device address (ADDR) is the most significant 7 bits of the hex address and is fixed as 1010101. The first 8 bits of the I²C protocol are, therefore, 0xAA or 0xAB for write or read, respectively.

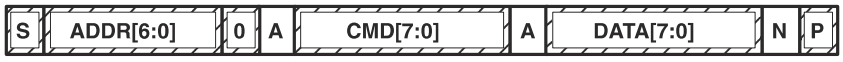


(S = Start, Sr = Repeated Start, A = Acknowledge, N = No Acknowledge , and P = Stop).

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 whenever data is acknowledged by the fuel gauge or the I²C primary. “Quick writes” function in the same manner and are a convenient means of sending multiple bytes to consecutive command locations (such as two-byte commands that require two bytes of data).

The following command sequences are not supported:

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



Attempt to read an address above 0x6B (NACK command):

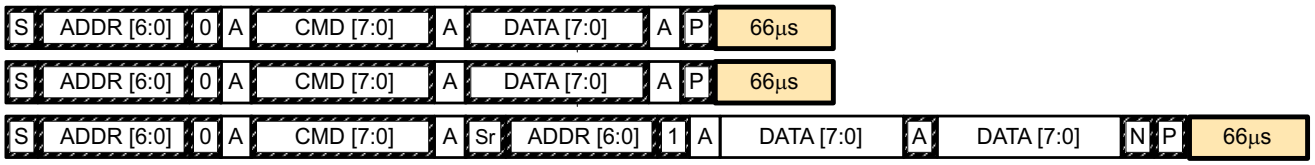
3.2 I²C Time Out

The I²C engine releases both SDA and SCL if the I²C bus is held low for 2 seconds. If the fuel gauge is holding the lines, releasing them frees them for the primary to drive the lines. If an external condition is holding either of the lines low, the I²C engine enters the low-power SLEEP mode.

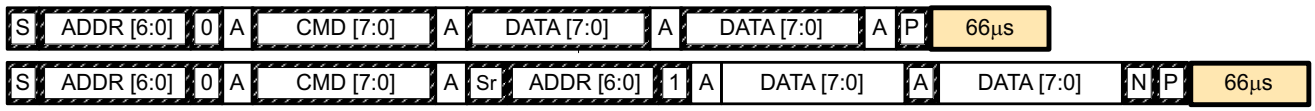
3.3 I²C Command Waiting Time

To ensure proper operation at 400 kHz, a $t_{(BUF)}$ ≥ 66 µs bus-free waiting time must be inserted between all packets addressed to the fuel gauge. In addition, if the SCL clock frequency (f_{SCL}) is > 100 kHz, use

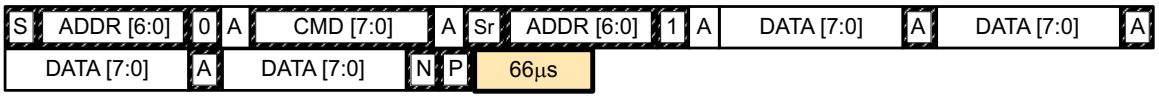
individual 1-byte write commands for proper data flow control. The following diagram shows the standard waiting time required between issuing the control subcommand the reading the status result. For read-write standard command, a minimum of 2 seconds is required to get the result updated. For read-only standard commands, there is no waiting time required, but the host must not issue any standard command more than two times per second. Otherwise, the gauge could result in a reset issue due to the expiration of the watchdog timer.



Waiting time inserted between two 1-byte write packets for a subcommand and reading results
(required for 100 kHz < f_{SCL} ≤ 400 kHz)



Waiting time inserted between incremental 2-byte write packet for a subcommand and reading results
(acceptable for f_{SCL} ≤ 100 kHz)



Waiting time inserted after incremental read

3.4 I²C Clock Stretching

A clock stretch can occur during all modes of fuel gauge operation. In SLEEP mode, a short ≤ 100-µs clock stretch occurs on all I²C traffic as the device must wake-up to process the packet. In the other modes (INITIALIZATION, NORMAL), a ≤ 4-ms clock stretching period may occur within packets addressed for the fuel gauge as the I²C interface performs normal data flow control.

Chapter 4

Application Examples

4.1 Data Memory Parameter Update Example

The following example shows the command sequence needed to modify a Data Memory parameter. For this example, the default ***Design Capacity*** is updated from 1000 mAh to 1200 mAh. All device writes (wr) and reads (rd) refer to the I^2C 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: The remaining steps in this table 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 CFGUPDATE 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 *DataBlockClass()* 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).	`rd 0x60 OLD_Csum;`
8	Read both ***Design Capacity*** bytes starting at 0x46 (offset = 6). Block data starts at 0x40, so to read the data of a specific offset, use address 0x40 + mod(offset, 32). **Note**: LSB byte is coincidentally the same value as the checksum.	`rd 0x46 OLD_DesCap_MSB;` `rd 0x47 OLD_DesCap_LSB;`
9	Write both ***Design Capacity*** bytes starting at 0x46 (offset = 6). For this example, the new value is 1200 mAh. (0x04B0 hex)	`wr 0x46 0x04;` `wr 0x47 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 0x1F`
12	Exit CFGUPDATE mode by sending *SOFT_RESET* subcommand, *Control(0x0042)*.	`wr 0x00 0x42 0x00;`

Step	Description	Pseudo Code
13	Confirm CFGUPDATE 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 by sending the *Control(0x0020)* subcommand.	`wr 0x00 0x20 0x00;`

4.2 Chemistry Profile Change Example

The following example illustrates how to change the chemistry profile from the default profile that is provided. In this example, the chemistry profile is changed from the default 4.35 V (3230) to 4.4 V (3142).

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: The remaining steps in this table 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	Read Current Chem ID.	`rd 0x00 0x08 0x00;`
3	Send SET_CFGUPDATE subcommand, *Control(0x0013).*	`wr 0x00 0x13 0x00;`
4	Confirm CFGUPDATE mode by polling *Flags()* register until bit 4 is set. The host should wait for 1 second to ensure IT processing has been stopped.	`rd 0x06 Flags_register;`
5	Change Chemistry to CHEM_C (chem ID = 3142, 4.4 V).	`wr 0x00 0x32 0x00;`
6	Exit CFGUPDATE mode by sending the SOFT_RESET subcommand.	`wr 0x00 0x42 0x00;`
7	Confirm CFGUPDATE has been exited by polling *Flags()* register until bit 4 is cleared. (This may take up to 1 second.) SOC will be valid after 2 seconds	`rd 0x06 Flags_register;`
8	Read Current Chem ID to see if it is updated.	`rd 0x00 0x08 0x00;`
9	If the device was previously SEALED, return to SEALED mode by sending the *Control(0x0020)* subcommand.	`wr 0x00 0x20 0x00;`

Chapter 5

Standard Commands

The 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 Standard Commands. 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 5-1. 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
AverageCurrent()		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	mAh	R

5.1 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 Control() Subcommands.

Table 5-2. 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.
FW_VERSION	0x0002	Yes	Reports the firmware version of the device.
DM_CODE	0x0004	Yes	Reports the Data Memory Code number stored in NVM.
PREV_MACWRITE	0x0007	Yes	Returns previous MAC command code.
CHEM_ID	0x0008	Yes	Reports the chemical identifier of the battery profile used by the fuel gauge.
BAT_INSERT	0x000C	Yes	Forces the *Flags() [BAT_DET]* bit set when the ***OpConfig [BIE]*** bit is 0.
BAT_REMOVE	0x000D	Yes	Forces the *Flags() [BAT_DET]* bit clear when the ***OpConfig [BIE]*** bit is 0.
SET_CFGUPDATE	0x0013	No	Forces the *CONTROL_STATUS [CFGUPMODE]* bit to 1 and the gauge enters CONFIG UPDATE mode.
SMOOTH_SYNC	0x0019	Yes	Synchronizes *RemCapSmooth()* and *FCCSmooth()* with *RemCapTrue()* and *FCCTrue()*
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.
PULSE_SOC_INT	0x0023	Yes	Commands the device to toggle the GPOUT pin for 1 ms
CHEM_A	0x0030	No	Dynamically changes existing Chem ID to Chem ID - 3230
CHEM_B	0x0031	No	Dynamically changes existing Chem ID to Chem ID - 1202
CHEM_C	0x0032	No	Dynamically changes existing Chem ID to Chem ID - 3142
RESET	0x0041	No	Performs a full device reset.
SOFT_RESET	0x0042	No	Gauge exits CONFIG UPDATE mode

5.1.1 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 5-3. CONTROL_STATUS Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	SHUTDOWNEN	WDRESET	SS	CALMODE	CCA	BCA	QMAX_UP	RES_UP
Low Byte	INITCOMP	RSVD	RSVD	SLEEP	LDMD	RUP_DIS	VOK	CHEM CHANGE

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.

CCA = Indicates the Coulomb Counter Auto-Calibration routine is active. The CCA routine will take place approximately 3 minutes and 45 seconds after the initialization as well as periodically as conditions permit. Active when set.

BCA = Indicates the fuel gauge board calibration routine is active. Active when set.

QMAX_UP = Indicates Qmax has updated. True when set. This bit is cleared after a POR or when the *Flags() [BAT_DET]* bit is set. When this bit is cleared, it enables fast learning of battery Qmax.

RES_UP = Indicates that resistance has been updated. True when set. This bit is cleared after a POR or when the *Flags() [BAT_DET]* bit is set. Also, this bit can only be set after Qmax is updated (*[QMAX_UP]* bit is set). When this bit is cleared, it enables fast learning of battery impedance.

Low Byte

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

RSVD = Reserved

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 1.

RUP_DIS = Indicates the Ra table updates are disabled. Updates are disabled when set.

VOK = Indicates cell voltages are ok for Qmax updates. True when set.

CHEMCHANGE = Indicates that the Device Chemistry table has been dynamically changed. True when set. The bit is cleared when the active chemistry table has been updated.

5.1.2 DEVICE_TYPE: 0x0001

Instructs the fuel gauge to return the device type to addresses 0x00 and 0x01. The value returned is 0x0427.

5.1.3 FW_VERSION: 0x0002

Instructs the fuel gauge to return the firmware version to addresses 0x00 and 0x01. The return value is 0x0202.

5.1.4 DM_CODE: 0x0004

Instructs the fuel gauge to return the 8-bit ***DM Code*** as the least significant byte of the 16-bit return value at addresses 0x00 and 0x01. The *DM_CODE* subcommand provides a simple method to determine the configuration code stored in Data Memory.

5.1.5 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.

5.1.6 CHEM_ID: 0x0008

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

5.1.7 BAT_INSERT: 0x000C

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

5.1.8 BAT_REMOVE: 0x000D

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

5.1.9 SET_CFGUPDATE: 0x0013

Instructs the fuel gauge to set the *Flags() [CFGUPMODE]* bit to 1 and enter CONFIG UPDATE mode. This command is only available when the fuel gauge is UNSEALED. After the command is sent, the host must wait a minimum of 1100ms to start modifying any parameters.

Note

A *SOFT_RESET* subcommand is used to exit CONFIG UPDATE mode to resume normal gauging.

5.1.10 SMOOTH_SYNC: 0x0019

This synchronizes *RemCapSmooth()* and *FCCSmooth()* with *RemCapTrue()* and *FCCTrue()*.

5.1.11 SHUTDOWN_ENABLE: 0x001B

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

5.1.12 SHUTDOWN: 0x001C

Instructs the fuel gauge to immediately enter SHUTDOWN mode after receiving this subcommand. The SHUTDOWN mode is effectively a power-down mode with only a small circuit biased by the BAT pin which is used for wake-up detection. To enter SHUTDOWN mode, the *SHUTDOWN_ENABLE* subcommand must have been previously received. To exit SHUTDOWN mode, the GPOUT pin must be raised from logic low to logic high for at least 200 µs.

5.1.13 SEALED: 0x0020

Instructs the fuel gauge to transition from UNSEALED state to SEALED state and will set bit 7 (0x80) in the ***Update Status*** register to 1. The fuel gauge should always be set to the SEALED state for use in end equipment. The SEALED state blocks accidental writes of specific subcommands and most Standard and Extended Commands. See Table 5-1, Table 5-2, and Table 6-1.

5.1.14 PULSE_SOC_INT: 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 command.

Note

The GPOUT pin must be configured for the SOC_INT output function with the ***OpConfig [BATLOWEN]*** bit cleared.

5.1.15 CHEM_A:/B/C 0x0030/0x0031/0x0032

This subcommand instructs the fuel gauge to dynamically change the active chemistry after reset. A SOFT_RESET should be performed after running the CHEM_A/B/C commands. This subcommand is only available when the fuel gauge is UNSEALED. After this command is executed, a control status word ***CHEMCHANGE*** will be set to indicate that the chemistry has been changed.

5.1.16 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. Refer to Figure 2-1. This subcommand is only available when the fuel gauge is UNSEALED. It is recommended to only issue the *RESET* in CONFIG UPDATE mode.

5.1.17 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 *[CFGUPMODE]* bits are cleared and a re-simulation occurs to update both *StateOfCharge()* and *StateOfChargeUnfiltered()*. Refer to Figure 2-1. Upon exit from CONFIG UPDATE mode, the fuel gauge will check bit 7 (0x80) in the ***Update Status*** register. If bit 7 (0x80) in the Update Status register is set the fuel gauge will be placed into the SEALED state. This subcommand is only available when the fuel gauge is UNSEALED.

5.2 Temperature(): 0x02 and 0x03

This read- and write-word function returns an unsigned integer value of the temperature in units of 0.1 K measured by the fuel gauge. If ***OpConfig [TEMPS]*** bits = 00 (default), a read command will return the internal temperature sensor value and a write command will be ignored. If ***OpConfig [TEMPS]*** bits = 01, a read command will return the temperature measured through an external thermistor. If ***OpConfig [TEMPS]*** bits = 10,

a write command sets the temperature to be used for gauging calculations, while a read command returns to the temperature previously written.

5.3 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.

5.4 Flags(): 0x06 and 0x07

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

Table 5-4. Flags Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	OT	UT	RSVD	RSVD	RSVD	RSVD	FC	CHG
Low Byte	OCVTAKEN	DOD CORRECT	ITPOR	CFGUPMODE	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 RELAXATION mode and set to 1 when OCV measurement is performed in RELAXATION mode.

DOD Correct = This indicates that DOD correction is being applied.

ITPOR = Indicates a POR 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.

CFGUPMODE = Fuel gauge is in CONFIG UPDATE mode. True when set. Default is 0. Refer to Section 2.4.3 for details.

BAT_DET = Battery insertion detected. True when set. When ***OpConfig [BIE]*** is set, *[BAT_DET]* is set by detecting a logic high-to-low transition at the BIN pin. When ***OpConfig [BIE]*** is low, *[BAT_DET]* is set when host issues the *BAT_INSERT* subcommand and is cleared when host issues the *BAT_REMOVE* subcommand. Gauge predictions are not 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.

5.5 NominalAvailableCapacity(): 0x08 and 0x09

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

5.6 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.

5.7 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.

5.8 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.

5.9 AverageCurrent(): 0x10 and 0x11

This read-only command pair returns a signed integer value that is the average current flow through the sense resistor. In NORMAL mode, it is updated once per second and is calculated by dividing the 1-second change in coulomb counter data by 1 second. Large current spikes of short duration will be averaged out in this measurement. Units are mA.

5.10 AveragePower(): 0x18 and 0x19

This read-only function returns a signed integer value of the average power during the charging or discharging of the battery. 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.

5.11 StateOfCharge(): 0x1C and 0x1D

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

5.12 InternalTemperature(): 0x1E and 0x1F

This read-only function returns an unsigned integer value of the internal temperature sensor in units of 0.1 K as measured by the fuel gauge. If ***OpConfig [TEMPS]*** = 00, this command will return the same value as *Temperature()*.

5.13 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.

5.14 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.

5.15 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.

5.16 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.

5.17 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.

This page intentionally left blank.

Chapter 6
Extended Data Commands

Extended data 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 6-1.

Table 6-1. Extended Commands

Name	Command Code	Unit	SEALED Access(1) (2)	UNSEALED Access(1) (2)
DataClass() (2)	0x3E	NA	NA	RW
DataBlock() (2)	0x3F	NA	RW	RW
BlockData()	0x40 through 0x5F	NA	R	RW
BlockDataCheckSum()	0x60	NA	RW	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.

6.1 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.

6.2 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.

6.3 BlockData(): 0x40 Through 0x5F

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

6.4 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()* will be transferred to RAM.

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

6.5 BlockDataControl(): 0x61

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

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

6.6 Reserved—0x62 Through 0x7F

Chapter 7
Data Memory

7.1 Data Memory Interface

7.1.1 Accessing the Data Memory

The Data Memory contains initialization, default, cell status, calibration, configuration, and user information. Most Data Memory parameters reside in volatile RAM 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 in which mode the fuel gauge is operating 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 6, *Extended Data Commands*. These commands are available when the fuel gauge is either in UNSEALED or SEALED mode.

Most Data Memory locations, however, are only accessible in the UNSEALED mode by use of the evaluation software or by Data Memory block transfers. These locations should be optimized and/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 4.1.

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 incorrect operation due to firmware program interpretation of the invalid data. The written data is not persistent, so a POR does resolve the fault.

7.1.2 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 to utilize the data protection feature. It is highly recommended to keep the gauge in SEALED mode unless to access specific information that can be accessed only during UNSEALED mode.

7.1.3 SEALING and UNSEALING Data Memory Access

The fuel gauge implements a key-access scheme to transition from SEALED to UNSEALED mode. Once SEALED 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; therefore, *Control()* should supply 0x8000 and 0x8000 (again) to unseal the part.

7.2 Data Types Summary

Table 7-1. Data Type Decoder

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

7.3 Data Flash Summary

Table 7-2. Data Flash Summary

Class	Subclass	Subclass ID	Offset	Type	Name	Min	Max	Default	Units
Configuration	Safety	2	0	I2	Over Temp	–1200	1200	550	0.1°C
Configuration	Safety	2	2	I2	Under Temp	–1200	1200	0	0.1°C
Configuration	Safety	2	4	U1	Temp Hys	0	255	50	0.1°C
Configuration	Charge Termination	36	3	I1	TCA Set %	–1	100	99	%
Configuration	Charge Termination	36	4	I1	TCA Clear %	–1	100	95	%
Configuration	Charge Termination	36	5	I1	FC Set %	–1	100	–1	%
Configuration	Charge Termination	36	6	I1	FC Clear %	0	100	98	%
Configuration	Charge Termination	36	7	I2	DODatEOC Delta T	0	1000	50	0.1°C
Configuration	Discharge	49	0	U1	SOC1 Set Threshold	0	100	10	%
Configuration	Discharge	49	1	U1	SOC1 Clear Threshold	0	100	15	%
Configuration	Discharge	49	2	U1	SOCF Set Threshold	0	100	2	%
Configuration	Discharge	49	3	U1	SOCF Clear Threshold	0	100	5	%
Configuration	Registers	64	0	H2	OpConfig	0	ffff	6478	Flag
Configuration	Registers	64	2	H1	OpConfigB	0	ff	0f	Flag
Configuration	Registers	64	3	H1	OpConfigC	0	ff	9f	Flag
Configuration	Registers	64	4	H1	OpConfigD	0	ff	23	Flag
Gas Gauging	IT Cfg	80	7	U2	OCV Wait Time	0	65535	60	s
Gas Gauging	IT Cfg	80	18	U2	Ra Filter	0	1000	800	Num
Gas Gauging	IT Cfg	80	20	I2	Res V Drop	0	32767	32767	mV
Gas Gauging	IT Cfg	80	22	U2	Samples to Wake	0	65535	240	s

Table 7-2. Data Flash Summary (continued)

Class	Subclass	Subclass ID	Offset	Type	Name	Min	Max	Default	Units
Gas Gauging	IT Cfg	80	24	U2	Qmax Max Time	0	65535	18000	s
Gas Gauging	IT Cfg	80	31	U1	DOD Valid Time	0	255	25	s
Gas Gauging	IT Cfg	80	33	U1	Fast Qmax Start DOD %	0	100	92	%
Gas Gauging	IT Cfg	80	34	U1	Fast Qmax End DOD %	0	100	96	%
Gas Gauging	IT Cfg	80	35	I2	Fast Qmax Start Volt Delta	0	4200	125	mV
Gas Gauging	IT Cfg	80	37	U2	Fast Qmax Current Threshold	0	1000	4	Hr rate
Gas Gauging	IT Cfg	80	39	U1	Fast Qmax Min Points	0	255	3	Num
Gas Gauging	IT Cfg	80	43	U1	Max Qmax Change	0	255	20	%
Gas Gauging	IT Cfg	80	44	U1	Qmax Max Delta %	0	255	10	%DCap
Gas Gauging	IT Cfg	80	45	U1	Max % Default Qmax	0	255	120	%DCap
Gas Gauging	IT Cfg	80	46	U1	Qmax Filter	0	255	96	Num
Gas Gauging	IT Cfg	80	48	U2	ResRelax Time	0	65535	500	s
Gas Gauging	IT Cfg	80	50	I2	User Rate-mA	–32768	0	0	mA
Gas Gauging	IT Cfg	80	52	I2	User Rate-mW	–32768	0	0	mW
Gas Gauging	IT Cfg	80	57	U1	Max Sim Rate	0	255	1	Hr rate
Gas Gauging	IT Cfg	80	58	U1	Min Sim Rate	0	255	20	Hr rate
Gas Gauging	IT Cfg	80	59	U2	Ra Max Delta	0	32767	8	4 mΩ
Gas Gauging	IT Cfg	80	68	I2	Min Delta Voltage	0	32767	0	mV
Gas Gauging	IT Cfg	80	70	I2	Max Delta Voltage	0	32767	200	mV
Gas Gauging	IT Cfg	80	72	I2	DeltaV Max dV	0	32767	100	mV
Gas Gauging	IT Cfg	80	74	U1	TermV Valid t	0	255	2	s
Gas Gauging	IT Cfg	80	75	I2	Trace Resistance	0	32767	0	mΩ
Gas Gauging	IT Cfg	80	77	I2	Downstream Resistance	0	32767	0	mΩ
Gas Gauging	IT Cfg	80	79	U2	Predict Ambient Time	0	65535	2000	s
Gas Gauging	IT Cfg	80	81	U1	Design Energy Scale	1	10	1	Num
Gas Gauging	IT Cfg	80	82	U1	Fast Scale Load Select	0	6	3	Num
Gas Gauging	IT Cfg	80	83	U1	Chg DOD Correction Start SOC	0	101	90	Num
Gas Gauging	IT Cfg	80	84	U1	Chg DOD Correction Taper Ratio	0	20	20	Num
Gas Gauging	Current Thresholds	81	0	I2	Dsg Current Threshold	0	2000	167	.1 Hr rate
Gas Gauging	Current Thresholds	81	2	I2	Chg Current Threshold	0	2000	100	.1 Hr rate
Gas Gauging	Current Thresholds	81	4	I2	Quit Current	0	2000	250	.1 Hr rate
Gas Gauging	Current Thresholds	81	6	U2	Dsg Relax Time	0	65535	60	s
Gas Gauging	Current Thresholds	81	8	U1	Chg Relax Time	0	255	60	s
Gas Gauging	Current Thresholds	81	9	U1	Quit Relax Time	0	255	1	s
Gas Gauging	Current Thresholds	81	12	U2	Max IR Correct	0	1000	400	mV
Gas Gauging	State	82	0	I2	Qmax Cell 0	0	32767	16384	Num
Gas Gauging	State	82	2	H1	Update Status	0	ff	0	Hex
Gas Gauging	State	82	3	I2	Reserve Cap-mAh	0	9000	0	mAh
Gas Gauging	State	82	5	H1	Load Select/Mode	0	ff	81	Hex
Gas Gauging	State	82	6	I2	Design Capacity	0	8000	1340	mAh
Gas Gauging	State	82	8	I2	Design Energy	0	32767	4960	mWh
Gas Gauging	State	82	10	I2	Terminate Voltage	2500	3700	3200	mV
Gas Gauging	State	82	16	I2	T Rise	0	32767	20	Num

Table 7-2. Data Flash Summary (continued)

Class	Subclass	Subclass ID	Offset	Type	Name	Min	Max	Default	Units
Gas Gauging	State	82	18	I2	T Time Constant	0	32767	1000	s
Gas Gauging	State	82	20	U1	SOCI Delta	0	100	1	%
Gas Gauging	State	82	21	I2	Taper Rate	0	2000	100	.1 Hr rate
Gas Gauging	State	82	23	I2	Sleep Current	0	1000	10	mA
Gas Gauging	State	82	25	I2	Avg I Last Run	–32768	–1	–50	.1 Hr rate
Gas Gauging	State	82	27	I2	Avg P Last Run	–32768	–1	–50	.1 Hr rate
Gas Gauging	State	82	29	I2	Delta Voltage	0	1000	1	mV
Ra Tables	Ra0 RAM	89	0	I2	Ra 0	0	32767	78	Num
Ra Tables	Ra0 RAM	89	2	I2	Ra 1	0	32767	35	Num
Ra Tables	Ra0 RAM	89	4	I2	Ra 2	0	32767	39	Num
Ra Tables	Ra0 RAM	89	6	I2	Ra 3	0	32767	45	Num
Ra Tables	Ra0 RAM	89	8	I2	Ra 4	0	32767	42	Num
Ra Tables	Ra0 RAM	89	10	I2	Ra 5	0	32767	36	Num
Ra Tables	Ra0 RAM	89	12	I2	Ra 6	0	32767	39	Num
Ra Tables	Ra0 RAM	89	14	I2	Ra 7	0	32767	36	Num
Ra Tables	Ra0 RAM	89	16	I2	Ra 8	0	32767	35	Num
Ra Tables	Ra0 RAM	89	18	I2	Ra 9	0	32767	37	Num
Ra Tables	Ra0 RAM	89	20	I2	Ra 10	0	32767	38	Num
Ra Tables	Ra0 RAM	89	22	I2	Ra 11	0	32767	40	Num
Ra Tables	Ra0 RAM	89	24	I2	Ra 12	0	32767	46	Num
Ra Tables	Ra0 RAM	89	26	I2	Ra 13	0	32767	54	Num
Ra Tables	Ra0 RAM	89	28	I2	Ra 14	0	32767	46	Num
Chemistry Info	Chem Data	109	2	I2	Q Invalid MaxV	0	32767	3811	mV
Chemistry Info	Chem Data	109	4	I2	Q Invalid MinV	0	32767	3750	mV
Chemistry Info	Chem Data	109	6	I2	V at Chg Term	0	5000	4340	mV
Chemistry Info	Chem Data	109	8	I2	Taper Voltage	0	5000	4250	mV
Calibration	Data	104	0	I1	Board Offset	–128	127	0	Counts
Calibration	Data	104	1	I1	Int Temp Offset	–128	127	0	0.1°C
Calibration	Data	104	2	I1	Ext Temp Offset	–128	127	0	0.1°C
Calibration	Data	104	3	I1	Pack V Offset	–128	127	0	mV
Calibration	Data	104	4	I2	Ext a Coef 1	–32768	32767	–11130	Num
Calibration	Data	104	6	I2	Ext a Coef 2	–32768	32767	19142	Num
Calibration	Data	104	8	I2	Ext a Coef 3	–32768	32767	–19262	Num
Calibration	Data	104	10	I2	Ext a Coef 4	–32768	32767	28203	Num
Calibration	Data	104	12	I2	Ext a Coef 5	–32768	32767	892	Num
Calibration	Data	104	14	I2	Ext b Coef 1	–32768	32767	328	Num
Calibration	Data	104	16	I2	Ext b Coef 2	–32768	32767	–605	Num
Calibration	Data	104	18	I2	Ext b Coef 3	–32768	32767	–2443	Num
Calibration	Data	104	20	I2	Ext b Coef 4	–32768	32767	4696	Num
Calibration	CC Cal	105	2	I2	CC Cal Temp	0	32767	2982	K
Calibration	Current	107	1	U1	Deadband	0	255	5	mA
Security	Codes	112	0	H4	Sealed to Unsealed	10001	ffffffff	80008000	Hex

7.4 Data Memory Parameter Descriptions

7.4.1 Configuration Class

7.4.1.1 Safety Subclass

7.4.1.1.1 Over-Temperature Threshold, Under-Temperature Threshold, Temperature Hysteresis

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Safety	2	0	I2	Over Temp	–1200	1200	550	0.1°C
		2	I2	Under Temp	–1200	1200	0	0.1°C
		4	U1	Temp Hys	0	255	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).

7.4.1.2 Charge Termination Subclass

7.4.1.2.1 Terminate Charge Alarm Set %, Terminate Charge Alarm Clear %

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Charge Termination	36	3	I1	TCA Set %	–1	100	99	%
		4	I1	TCA Clear %	–1	100	95	%

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

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

See Section 7.4.2.3.10 for details about the Primary Charge Termination Algorithm.

7.4.1.2.2 Full Charge Set %, Full Charge Clear %

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Charge Termination	36	5	I1	FC Set %	–1	100	–1	%
		6	I1	FC Clear %	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 7.4.2.3.10 for details about the Primary Charge Termination Algorithm.

7.4.1.2.3 DOD at EOC Delta Temperature

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

This represents the temperature change threshold to update Qstart and *RemainingCapacity()* due to temperature changes. During relaxation and at the start of charging, the remaining capacity is calculated as *RemainingCapacity()* = *FullChargeCapacity()* – Qstart. As temperature decreases, Qstart can become

much smaller than the old *FullChargeCapacity()* value, resulting in an overestimation of *RemainingCapacity()*. To improve accuracy, *FullChargeCapacity()* is updated when the temperature change since the last *FullChargeCapacity()* update is greater than ***DODatEOC Delta T*** × 0.1°C. The default value is 50.

Note

The units are a tenth of a °C, which means a value of 50 corresponds to 5°C.

7.4.1.3 Discharge Subclass

7.4.1.3.1 State-of-Charge 1 Set Threshold, State-of-Charge 1 Clear Threshold

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

7.4.1.3.2 State-of-Charge Final Set Threshold, State-of-Charge Final Clear Threshold

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

7.4.1.4 Registers

7.4.1.4.1 Operation Configuration (OpConfig) Register

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Registers	64	0	H2	OpConfig	0	FFFF	6478	Flag

Table 7-3. OpConfig Register Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	RSVD0	RSVD1	BIE	RSVD0	GPIOPOL	RSVD1	RSVD0	RSVD0
Default =	0	1	1	0	0	1	0	0
	0x64							
Low Byte	RSVD0	ResFactStep	SLEEP	RMFCC	FastConv En	BATLOWEN	Temp Source[1]	Temp Source[0]
Default =	0	1	1	1	1	0	0	0
	0x78							

High Byte

RSVD0 = Reserved. Default is 0. (Set to 0 for proper operation.)

RSVD1 = Reserved. Default is 1. (Set to 1 for proper operation.)

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

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

Low Byte

ResFactStep = Enables Ra step up/down to Max/Min Res factor before disabling Ra updates

SLEEP = The fuel gauge can enter sleep, if operating conditions allow. True when set.

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

FastConvEn = Enables Fast SOC Convergence. True when set.

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

TempSource[1:0] = Selects the temperature source. Enables the host to write *Temperature()* if set. If cleared, the internal temperature sensor is used for *Temperature()*.

00 = Internal Temperature Sensor is used as Temperature Source.

01 = External Thermistor is used as Temperature Source.

10 = Host written Temperature is used as Temperature Source.

7.4.1.4.2 Operation Configuration B (OpConfigB) Register

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Registers	64	2	H1	OpConfigB	0	FF	0F	Flag

Table 7-4. OpConfigB Register Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	RSVD0	RSVD0	RSVD0	RSVD0	RSVD1	SMOOTHEN	RSVD1	RSVD1
Default =	0	0	0	0	1	1	1	1
	0x0F							

SMOOTHEN = Enables the SOC smoothing feature. True when set.

RSVD0 = Reserved. Default is 0. (Set to 0 for proper operation.)

RSVD1 = Reserved. Default is 1. (Set to 1 for proper operation.)

7.4.1.4.3 Operation Configuration C (OpConfigC) Register

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Registers	64	2	H1	OpConfigC	0	FF	9F	Flag

Table 7-5. OpConfigC Register Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	RSVD1	Non Removable	RSVD0	SOCHold99	SOCHold 1	SOCHold OvrChg	SOCHold OvrDsg	Predict Ambient
Default =	1	0	0	1	1	1	1	1
	0x9F							

RSVD1 = Reserved. Default is 1. (Set to 1 for proper operation.)

Non Removable = If Set, Fuel gauge assumes battery is present and ignores BIE functionality or the Battery insert commands. If cleared, fuel gauge relies on BIE functionality or battery insertion commands to indicate battery presence

RSVD0= Reserved: Default is 0.

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

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

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

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

PredictAmbient =

0 = Disables ambient temperature adaptability for gauging

1 = Enables ambient temperature adaptability for gauging

7.4.1.4.4 Operation Configuration D (OpConfigD) Register

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Registers	64	2	H1	OpConfigD	0	FF	23	Flag

Table 7-6. OpConfigD Register Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	FilterAlways	NoDeltaV Avg	Postpone LgDrops0	ChemID[1]	ChemID[0]	DODCorChgA lways	DODCorChg	DODCor DSG
Default =	0	0	1	0	0	0	1	1
	0x23							

Filter Always = If set, Qmax/Ra filters always applied. If cleared, the first simulation will be unfiltered.

NoDeltaVAvg = If set, the last instantaneous change in *Voltage()* from steady state determines end of discharge voltage. If cleared, average variance from steady state voltage used to determine end of discharge voltage.

PostponeLgDrops0 = If set, when a simulation results in RemCap = 0 before Ra scaling begins, then the reporting of RemCap = 0 is delayed till fast scaling starts. If cleared, if simulation results in RemCap = 0, then RemCap is reported as 0 instantly.

ChemID[1:0] =

00 = Chem ID 3230 is used.

01 = Chem ID 1202 is used.

10 = Chem ID 3142 is used.

11 = RSVD

DODCorChgAlways = If set, enable present DoD recalculation during every charge cycle.

DODCorChg = Enable present DoD recalculation during charging only. True when set. Default setting is recommended.

DODCorDSG= Enable present DoD recalculation during discharging only. True when set. Default setting is recommended.

7.4.2 Gas (Fuel) Gauging Class

7.4.2.1 IT Cfg Subclass

7.4.2.1.1 OCV Wait Time

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	7	U2	OCV Wait Time	0	65535	60	s

7.4.2.1.2 Ra Filter

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	18	U2	Ra Filter	0	1000	800	Num

Ra table updates are filtered. This is a weighting factor that takes a certain percentage of the previous Ra table value, and the remaining percentage comes from the newest calculated Ra value. This prevents resistances in the Ra table from changing quickly.

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

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

7.4.2.1.3 Resistance Update Voltage Drop

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	20	I2	Res V Drop	0	32767	32767	mV

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

7.4.2.1.4 Samples to Wake

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	22	U2	Samples to Wake	0	65535	240	s

7.4.2.1.5 Qmax Max Time

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	24	U2	Qmax Max Time	0	65535	18000	s

7.4.2.1.6 Fast Qmax Start DOD%, Fast Qmax Start Voltage Delta, Fast Qmax Current Threshold

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	33	U1	Fast Qmax Start DOD %	0	100	92	%
		35	I2	Fast Qmax Start Volt Delta	0	4200	125	mV
		37	U2	Fast Qmax Current Threshold	0	1000	4	h rate

Fast Qmax measurement starts when the following conditions are met,

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

7.4.2.1.7 Fast Qmax End DOD%, Fast Qmax Minimum Data Points

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	34	U1	Fast Qmax End DOD %	0	100	96	%
		39	U1	Fast Qmax Min Points	0	255	3	Num

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

- Number of Fast Qmax measurements > ***Fast Qmax Min Points***
- DOD > ***Fast Qmax End DOD%***

7.4.2.1.8 Maximum Qmax Change

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	45	U1	Max Qmax Change	0	255	20	%

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

7.4.2.1.9 Qmax Maximum Delta %

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	46	U1	Qmax Max Delta %	0	255	10	%

This is the percentage of *DesignCapacity()* that limits how much Qmax may grow or shrink during any one Qmax update.

The default is 10%.

7.4.2.1.10 Maximum % Default Qmax

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	47	U1	Max % Default Qmax	0	255	120	%

Provides an upper limit to the value to which Qmax can be learned. The default value is sufficient for most applications.

7.4.2.1.11 Qmax Filter

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	48	U1	Qmax Filter	0	255	96	Num

Qmax updates are filtered to prevent corrupt values. It is not recommended to change this value.

7.4.2.1.12 Simulation ResRelax Time (ResRelax Time)

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	50	U2	ResRelax Time	0	65535	500	s

ResRelax Time, or resistance relaxation time, is used for transient modeling. It represents the time it takes for the internal resistance to be fully saturated. This way the gauge will not simulate immediate large IR drops when it calculates the instantaneous voltage from the battery under load. The default value is 500 seconds, which is sufficient for most applications.

This value is used for Impedance Track transient modeling of effective resistance. The resistance increases from 0 to a final value determined by the Ra table, as defined by the exponent with time constant ***ResRelax Time*** during discharge simulation. The default value is optimized for typical cell behavior.

7.4.2.1.13 User-Defined Rate-Current

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	52	I2	User Rate-mA	–32768	0	0	mA

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

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

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

7.4.2.1.14 User-Defined Rate-Power

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	54	I2	User Rate-mW	–32768	0	0	mW

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

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

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

7.4.2.1.15 Maximum Simulation Rate, Minimum Simulation Rate

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	61	U1	Max Sim Rate	0	255	1	h rate
		62	U1	Min Sim Rate	0	255	20	h rate

Max Sim Rate:

Maximum IT simulation rate (inversed). 2 implies C/2. This is the maximum load used in IT simulations in terms of C-rate. This register defaults to 1.

Min Sim Rate:

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

7.4.2.1.16 Ra Max Delta

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	63	U2	Ra Max Delta	0	32767	8	4 mΩ

During the update of Ra values, a filtering process is performed to eliminate unexpected fluctuations in the updated Ra values. ***Ra Max Delta*** limits the change in Ra values to an absolute magnitude per Ra update.

7.4.2.1.17 Minimum Delta Voltage, Maximum Delta Voltage

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	72	I2	Min Delta Voltage	0	32767	0	mV
		74	I2	Max Delta Voltage	0	32767	200	mV

These parameters are the lower and upper bounds on the value that ***Delta Voltage*** is allowed to learn, and are saved during discharge cycles.

7.4.2.1.18 DeltaV Maximum Delta Voltage

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	76	I2	DeltaV Max dV	0	32767	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.

7.4.2.1.19 Terminate Voltage Valid Time

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
IT Cfg	80	78	U1	TermV Valid t	0	255	2	s

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

7.4.2.1.20 Trace Resistance

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

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

7.4.2.1.21 Downstream Resistance

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

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

7.4.2.1.22 Predict Ambient Time

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	79	Predict Ambient Time	U2	0	65535	2000	s

Predict Ambient Time determines the wait time before the algorithm starts to predict the ambient temperature during charge/discharge.

7.4.2.1.23 Design Energy Scale

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	81	Design Energy Scale	U1	1	10	1	Num

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

7.4.2.1.24 Fast Scale Load Select

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	82	Fast Scale Load Select	U1	0	6	3	Num

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

7.4.2.1.25 Chg DOD Correction Start SOC

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	83	Chg DOD Correction Start SOC	U1	0	101	90	Num

DOD correction during charge will start when SOC is above ***Chg DOD Correction Start SOC***.

7.4.2.1.26 Chg DOD Correction Taper Ratio

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	84	Chg DOD Correction Taper Ratio	U1	0	20	20	Num

DOD correction during charge will be applied when current is below ***Chg DOD Correction Taper Ratio***.

7.4.2.2 Current Thresholds Subclass

7.4.2.2.1 Discharge and Charge Detection Threshold, Quit Current and Relax Time, Discharge and Charge Relax Time

Subclass ID	Subclass	Offset	Type	Name	Value			Unit
					Min	Max	Default	
81	Current Thresholds	0	I2	Dsg Current Threshold	0	2000	167	0.1 h
		2	I2	Chg Current Threshold	0	2000	100	0.1 h
		4	I2	Quit Current	0	1000	250	0.1 h
		6	U2	Dsg Relax Time	0	65535	60	s
		8	U1	Chg Relax Time	0	255	60	s
		9	U1	Quit Relax Time	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 Current Threshold*** × 0.1). The default is effectively C/16.7.

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

The quit current threshold can be calculated as ***Design Capacity***/(***Quit Current*** × 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).

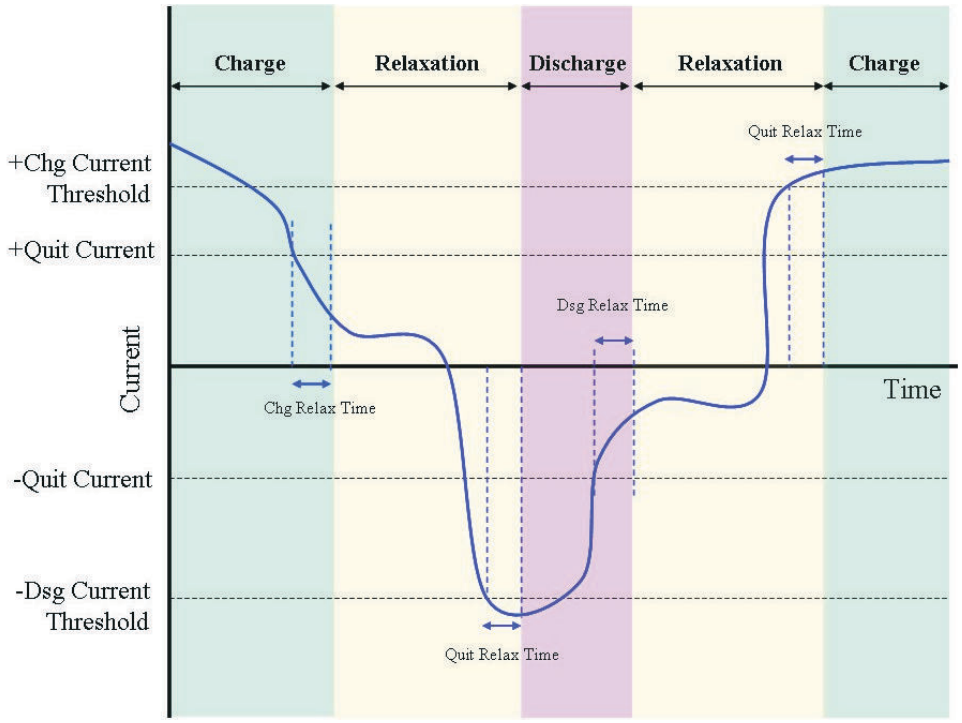


Figure 7-1. Example of Algorithm Operation Mode Changes with Varying *DataRAM.Average Current()*

7.4.2.2.2 Max IR Correct

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

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

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

This is specific to handheld applications. The default is 400 mV.

7.4.2.3 State Subclass

7.4.2.3.1 Qmax Cell 0

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	0	I2	Qmax Cell 0	0	32767	16384	Num

Qmax contains 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 low rate (~C/20) of discharge. For high accuracy, this value is periodically updated by the gauge during operation. The Impedance Track algorithm updates this value and maintains it.

To translate the Qmax register to mAh, use this formula:

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

7.4.2.3.2 Update Status

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	2	H1	Update Status	0	FF	0	Hex

Bit 0 (0x01) and bit 1 (0x02) of the ***Update Status*** register indicate whether or not the fuel gauge will apply limits to changes in Qmax updates and Ra table updates. When bit 0 (0x01) and bit 1 (0x02) of the ***Update Status*** register are cleared, the gauge will apply limits to changes in Qmax and the Ra table. Bit 0 (0x01) and bit 1 (0x02) are cleared by default and should remain cleared during operation. Only if a learning cycle is to be completed during initial configuration of the gauge's golden file should bit 0 (0x01) and bit 1 (0x02) be set.

Bit 7 (0x80) of the ***Update Status*** register indicates the default SEALED state of the fuel gauge. This bit is checked after POR and after exit of CONFIG UPDATE mode to see if the gauge should be placed into the SEALED or UNSEALED state. If bit 7 (0x80) is set then the gauge will be placed into the SEALED state.

7.4.2.3.3 Reserve Capacity (mAh)

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	3	I2	Reserve Cap-mAh	0	9000	0	mAh

Reserve Cap-mAh determines how much actual remaining capacity exists after reaching zero *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 zero *RemainingCapacity()* is reached.

7.4.2.3.4 Load Select, Load Mode

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	5	H1	Load Select/Mode	0	FF	81	Hex

Load Mode configures the fuel gauge to use either a constant-current or constant-power model for the Impedance Track algorithm. When ***Load Mode*** is 0, the Constant Current Model is used. This provides a better estimation of remaining run time, especially close to the end of discharge where current increases to compensate for decreasing battery voltage. When ***Load Mode*** is 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 Impedance Track algorithm.

Load Select defines the type of power or current model to be used to compute load-compensated capacity in the Impedance Track algorithm. By default, ***Load Select*** is set to 1, which means the IT algorithm will use a running average of the current discharge period. Once the discharge stops, the algorithm stores the average in data flash as the ***Avg I Last Run*** and ***Avg P Last Run*** variables. For simulations during RELAXATION, CHARGE, and at the start of DISCHARGE (since a new average has not been gathered), it would use data flash values for simulations. Once the discharge has lasted 500 seconds, the gauge re-simulates using the new running average. Thereafter, during discharge, it uses the continuous running average for any subsequent simulations.

Table 7-7. Load Select/Mode Parameter Encoding

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Load Mode	RSVD	RSVD	RSVD	RSVD	Load Select[2:0]		
Default =	1	0	0	0	0	0	0	1
	0x81							

Load Mode = Bit 7 contains the value for ***Load Mode***. Refer to Table 7-8 and Table 7-9 for operational details.

RSVD = Reserved. Set to 0 for proper operation.

Load Select[2:0] = Bits 2:0 contain the value for ***Load Select***. Refer to Table 7-8 and Table 7-9 for operational details. Default is 1.

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

Table 7-8. Current Model Used When Load Mode = 0

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

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

Table 7-9. Power Model Used When Load Mode = 1

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

7.4.2.3.5 Design Capacity, Design Energy, Default Design Capacity

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	6	I2	Design Capacity	0	8000	1340	mAh
		8	I2	Design Energy	0	32767	4960	mWh

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 Impedance Track 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 Impedance Track algorithm is assuming 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 Impedance Track 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 Impedance Track algorithm is using constant-power model when set.

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

7.4.2.3.6 Terminate Voltage

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	10	I2	Terminate Voltage	2500	3700	3200	mV

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

This register is application dependent. It should be set based on the battery cell specification to prevent damage to the cells or the absolute minimum system input voltage, taking into account the 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 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.

7.4.2.3.7 Thermal Rise Factor (T Rise)

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	16	I2	T Rise	0	32767	20	Num

The ***T Rise*** Factor reflects the level of system heating due to self-heating of the cell during discharge. This number can be measured empirically.

This is the thermal rise factor that is used in the single time constant heating-cooling thermal modeling. If set to 0, this feature is disabled and simulations in the IT algorithm will not account for self-heating of the battery cell. Larger values of ***T Rise*** lead to higher temperature rise estimates for the IT simulation. ***T Rise*** defaults to 20.

7.4.2.3.8 Thermal Time Constant (T Time Constant)

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	18	I2	T Time Constant	0	32767	1000	s

T Time Constant reflects the time constant of system heating due to self-heating of the cell during discharge. This number can be measured empirically.

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

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

7.4.2.3.9 SOC Interrupt Delta

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	20	U1	SOCI Delta	0	100	1	%

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

7.4.2.3.10 Taper Rate, Taper Voltage

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	21	I2	Taper Rate	0	2000	100	0.1 h rate
		29	I2	Taper Voltage	0	5000	4100	mV

Taper Rate is used in the Primary Charge Termination Algorithm. *AverageCurrent()* 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 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 per 40-second period
- *Voltage()* > ***Taper Voltage***

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

7.4.2.3.11 Sleep Current

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	23	I2	Sleep Current	0	1000	10	mA

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

This setting should be below any normal application currents.

7.4.2.3.12 Voltage at Charge Termination

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	33	I2	V at Chg Term	0	5000	4190	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.

7.4.2.3.13 Average Current Last Run

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	25	I2	Avg I Last Run	–32768	–1	–50	0.1 h 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. This register should never be modified; it is only updated by the fuel gauge when the gauge exits DISCHARGE mode.

7.4.2.3.14 Average Power Last Run

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	27	I2	Avg P Last Run	–32768	–1	–50	0.1 h 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. This register should never be modified; it is only updated by the fuel gauge when the gauge exits DISCHARGE mode.

7.4.2.3.15 Delta Voltage

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	29	I2	Delta Voltage	0	1000	1	mV

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

This value will never update to a value less than ***Min Delta Voltage*** or greater than ***Max Delta Voltage***. If ***Min Delta Voltage*** is set to a value above zero, then ***Delta Voltage*** should also be initialized to at least the same value as ***Min Delta Voltage***.

7.4.3 Ra Table Class

7.4.3.1 R_a RAM Subclass

Subclass	Subclass ID	Offset	Name	Type	Value			Unit
					Min	Max	Default	
Ra0 RAM	89	0	Ra 0	I2	0	32767	78	Num
		2	Ra 1	I2	0	32767	35	Num
		4	Ra 2	I2	0	32767	39	Num
		6	Ra 3	I2	0	32767	45	Num
		8	Ra 4	I2	0	32767	42	Num
		10	Ra 5	I2	0	32767	36	Num
		12	Ra 6	I2	0	32767	39	Num
		14	Ra 7	I2	0	32767	36	Num
		16	Ra 8	I2	0	32767	35	Num
		18	Ra 9	I2	0	32767	37	Num
		20	Ra 10	I2	0	32767	38	Num
		22	Ra 11	I2	0	32767	40	Num
		24	Ra 12	I2	0	32767	46	Num
		26	Ra 13	I2	0	32767	54	Num
		28	Ra 14	I2	0	32767	46	Num

The Ra Table class has 15 values. The R_a RAM is initialized from ROM upon gauge reset. Each of these values represents a resistance value normalized at 25°C for the associated Qmax Cell 0-based SOC grid point as found by the following rules:

For Cell0 Ra M where:

- If 0 ≤ M ≤ 7: The data is the resistance normalized at 25° and scaled by ***Design Capacity*** for:
  SOC = 100% – (M × 11.1%)
- If 8 ≤ M ≤ 14: The data is the resistance normalized at 25° and scaled by ***Design Capacity*** for:
  SOC = 100% – [77.7% + (M – 7) × 3.3%]

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

Normal Setting:

These resistance profiles are used by the gauge for the Impedance Track 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. It is important to note that this data is in mΩ and is normalized to 25°C and scaled by ***Design Capacity***. The following are useful observations to note with this data throughout the application development cycle:

- Watch for negative values in the Ra Table class. Negative numbers in profiles should never be anywhere in this class.
- Watch for smooth consistent transitions from one profile grid point value to the next throughout each profile. As the gauge does resistance profile updates, these values should be roughly consistent from one learned update to another without huge jumps in consecutive grid points.

7.4.4 Chemistry Class

7.4.4.1 Chem Data Subclass

7.4.4.1.1 Q Invalid Max V and Q Invalid Min V

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	2	I2	Q Invalid MaxV	0	32767	3811	mV
		4	I2	Q Invalid MinV	0	32767	3750	mV

Q Invalid Max V and ***Q Invalid Min V*** specify the Qmax disqualification voltage region generally known as the flat region of the OCV versus DOD curve. OCV measurement for Qmax calculation is disallowed in this region.

7.4.4.1.2 V at Chg Term

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	6	I2	V at Chg Term	0	5000	4340	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.

7.4.4.1.3 Taper Voltage

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
State	82	8	I2	Taper Voltage	0	5000	4250	mV

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

7.4.5 Calibration Class

7.4.5.1 Data Subclass

7.4.5.1.1 Board Offset

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Data	104	0	I1	Board Offset	–128	127	0	Counts

Board Offset is the second offset register. It calibrates all that the internal ***CC Offset*** does not calibrate out. This includes board layout, sense resistor, and copper trace, and other potential offsets that are external to the fuel gauge. The simplified ground circuit design in the fuel gauge requires a separate board offset for each tested device.

7.4.5.1.2 Internal Temperature Offset and External Temperature Offset

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Data	104	1	I1	Int Temp Offset	–128	127	0	0.1°C
Data	104	2	I1	Ext Temp Offset	–128	127	0	0.1°C

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

Ext Temp Offset is for calibrating the offset of the thermistor connected to the BIN pin of the fuel gauge.

7.4.5.1.3 Pack Voltage Offset

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Data	104	3	I1	Pack V Offset	–128	127	0	mV

This is the offset to calibrate the gauge analog-to-digital converter for cell voltage measurement.

Pack V Offset should not require modification by the user. It is modified by the Voltage Calibration function from CALIBRATION mode.

7.4.5.1.4 Ext a Coef and Ext b Coef

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Data	104	4	I2	Ext a Coef 1	–32768	32767	–11130	Num
Data	104	6	I2	Ext a Coef 2	–32768	32767	19142	Num
Data	104	8	I2	Ext a Coef 3	–32768	32767	–19262	Num
Data	104	10	I2	Ext a Coef 4	–32768	32767	28203	Num
Data	104	12	I2	Ext a Coef 5	–32768	32767	892	Num
Data	104	14	I2	Ext b Coef 1	–32768	32767	328	Num
Data	104	16	I2	Ext b Coef 2	–32768	32767	–605	Num
Data	104	18	I2	Ext b Coef 3	–32768	32767	–2443	Num
Data	104	20	I2	Ext b Coef 4	–32768	32767	4696	Num

Ext a Coef and ***Ext b Coef*** are the thermistor temperature linearization polynomial coefficients. The default values have been computed with a Semitec 103AT thermistor. If a different type of thermistor is used, then the coefficients will need to be changed. Contact TI to generate coefficients for a different thermistor.

7.4.5.2 CC Cal Subclass

7.4.5.2.1 CC Cal Temp

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
CC Cal	105	2	I2	CC Cal Temp	0	32767	2982	0.1 K

CC Cal Temp is the temperature at the time of current calibration. It is also used for RDL temperature compensation.

7.4.5.3 Current Subclass

7.4.5.3.1 Deadband

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

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

Only a few reasons may require changing the default value:

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

7.4.6 Security Class

7.4.6.1 Codes Subclass

7.4.6.1.1 Sealed to Unsealed

Subclass	Subclass ID	Offset	Type	Name	Value			Unit
					Min	Max	Default	
Codes	112	0	H4	Sealed to Unsealed	10001	FFFF FFFF	8000 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 after 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.

After the fuel gauge exits CONFIG UPDATE mode, the fuel gauge will check bit 7 (0x80) in the ***Update Status*** register. If bit 7 (0x80) is set, the fuel gauge will be placed into the SEALED state. If the fuel gauge is placed into SEALED mode on the exit of CONFIG UPDATE mode, the fuel gauge will not be allowed to go to the UNSEALED state for 4 seconds upon exiting CONFIG UPDATE mode. Any subcommand greater than 0x001A will restart the 4-second timer.

This page intentionally left blank.

Chapter 8

Updating BQ27427 Configuration Parameters

The BQ27427 gas gauge has RAM-based data memory parameters that can be updated if needed by the host processor. It may be useful that the application system main processor perform the actual firmware updating, rather than having an external tool do it. This appendix provides all the information necessary to implement a system firmware that sends the required I^2C commands to update the RAM-based parameters in the BQ27427 device.

8.1 Gauge Mode FlashStream (gm.fs) Files

With the Battery Management Studio (bqStudio) software, users can generate specific instruction files called gm.fs files, which contain the necessary I^2C commands that a host can send to the BQ27427 device to program the RAM-based data memory parameters.

The gm.fs file is an ASCII text file that contains commands and data. Each line of the file represents one command and potentially 96 bytes of data, as described in the following text. No row contains more than 96 data bytes. The first two characters of each row represent the command, followed by a ":

"W:" — Indicates that the row is a command to write one or more bytes of data.

"C:" — Indicates that the row is a command to read and compare one or more bytes of data.

"X:" — Indicates that the row is a command to wait a given number of milliseconds before proceeding.

White space is used to separate fields within the gm.fs files. Each row contains *only one* of the four commands. The commands discussed in this section can be implemented by a system that can perform multi-byte or single-byte operations for I^2C.

Figure 8-1 shows a typical gm.fs file snippet generated from the BQStudio software.

```
;-----------------------------------------------------
;Verify Existing Firmware Version
;-----------------------------------------------------
W: AA 00 01 00
C: AA 00 21 06
W: AA 00 02 00
C: AA 00 05 01
;-----------------------------------------------------
;SET_CFGUPDATE
;-----------------------------------------------------
W: AA 00 13 00
X: 1500
W: AA 3E 02 00
W: AA 40 02 26 00 00 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
W: AA 60 A5
X: 200
W: AA 3E 02 00
C: AA 60 A5
W: AA 3E 24 00
W: AA 40 00 19 28 63 5F FF 62 00 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
W: AA 60 69
X: 200
W: AA 3E 24 00
C: AA 60 69
W: AA 3E 30 00
W: AA 40 0E 10 FD FF 38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
W: AA 60 AD
X: 200
W: AA 3E 30 00
C: AA 60 AD
W: AA 3E 31 00
W: AA 40 0A 0F 02 05 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
W: AA 60 AD
```

Normal text file | length : 5015 | lines : 181 | Ln : 112 | Col : 6 | Sel : 0 | 0 | UNIX | ANSI as UTF-8 | INS

Figure 8-1. Typical gm.fs File Snippet

8.2 Write Command

The write command "W:" instructs the I^{2}C primary to write one or more bytes to a given I^{2}C address and given register address. The I^{2}C address format used throughout this document is based on an 8-bit representation of the address. The format of this sequence is:

```
"W: I2CAddr RegAddr Byte0 Byte1 Byte2…"
```

For example, the following:

```
W: AA 55 AB CD EF 00
```

Indicates that the I^{2}C primary writes the byte sequence 0xAB 0xCD 0xEF 0x00 to register 0x55 of the device addressed at 0xAA.

More precisely, it indicates to write the following data to the device address 0xAA:

0xAB to register 0x55

0xCD to register 0x56

0xEF to register 0x57

0x00 to register 0x58

8.3 Read and Compare Command

The read and compare command is formatted identically to the write command. The data presented with this command matches the data read exactly, or the operation should cease with an error indication to the user. The gm.fs file contains no information about program flow or decision making. If a read and compare command

results in data which does not match the expected values then the interpreting program needs to handle the next step itself. It should not continue with further commands, but would typically go back to the beginning of the gm.fs file and try again several times before giving up.

The format of this sequence is:

```
"C: i2cAddr RegAddr Byte0 Byte1 Byte2"
```

An example of this command is as follows:

```
C: AA 55 AB CD EF 00
```

This example expects the primary to read back 4 bytes from the register address 0x55 of the device addressed at 0xAA and then compare the data to the values given on the line command in this same order as 0xAB, 0xCD, 0xEF, and 0x00.

8.4 Wait Command

The wait command indicates that the host waits a minimum of the given number of milliseconds before continuing to the next row of the FlashStream file. A wait command is typically used to allow the fuel gauge processor to complete a process before proceeding to the next command in the file.

For example, the following:

```
X: 200
```

Indicates that the I^2C primary must wait at least 200 ms before continuing.

8.5 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 *Control()SET_CFGUPDATE* subcommand to enter the CONFIG UPDATE mode.

Note

To ensure that the gas gauge has entered CONFIG UPDATE mode correctly, there needs to be at least an 1100-ms delay after sending the *SET_CFGUPDATE*. Operation in this mode is indicated by the *Flags()[CFGUPMODE]* status bit.

In this mode, fuel gauging is suspended while the host uses the extended data commands to modify the configuration data blocks. To resume fuel gauging, the host must send a *Control()SOFT_RESET* subcommand to exit the CONFIG UPDATE mode, which clears both *Flags()[ITPOR]* and *[CFGUPMODE]* 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.

The memory of the BQ27427 device is separated into memory subclasses defined in this document. The memory cannot be directly addressed, but is updated through a sequence of extended commands that can access each block of memory indirectly. The gm.fs file updates these blocks to write the proper configuration so the BQ27427 device can have proper gauging performance and match the system characteristics. These updates are stored in RAM and need to be re-programmed any time the device loses power. (The *[ITPOR]* bit in the *Flags()* register indicates that the RAM configuration has been reset to the defaults, and is in need of updating using the gm.fs file.)

Table 8-1 shows the “Ra Tables” subclass in the BQ27427 gauge (see also Table 7-2).

Table 8-1. Ra Tables Subclass

Class	Subclass	Subclass ID	Offset	Type	Name	Min	Max	Default	Units
Ra Tables	Ra0 RAM	89	0	I2	Ra 0	0	32767	78	Num
Ra Tables	Ra0 RAM	89	2	I2	Ra 1	0	32767	35	Num

Table 8-1. Ra Tables Subclass (continued)

Class	Subclass	Subclass ID	Offset	Type	Name	Min	Max	Default	Units
Ra Tables	Ra0 RAM	89	4	I2	Ra 2	0	32767	39	Num
Ra Tables	Ra0 RAM	89	6	I2	Ra 3	0	32767	45	Num
Ra Tables	Ra0 RAM	89	8	I2	Ra 4	0	32767	42	Num
Ra Tables	Ra0 RAM	89	10	I2	Ra 5	0	32767	36	Num
Ra Tables	Ra0 RAM	89	12	I2	Ra 6	0	32767	39	Num
Ra Tables	Ra0 RAM	89	14	I2	Ra 7	0	32767	36	Num
Ra Tables	Ra0 RAM	89	16	I2	Ra 8	0	32767	35	Num
Ra Tables	Ra0 RAM	89	18	I2	Ra 9	0	32767	37	Num
Ra Tables	Ra0 RAM	89	20	I2	Ra 10	0	32767	38	Num
Ra Tables	Ra0 RAM	89	22	I2	Ra 11	0	32767	40	Num
Ra Tables	Ra0 RAM	89	24	I2	Ra 12	0	32767	46	Num
Ra Tables	Ra0 RAM	89	26	I2	Ra 13	0	32767	54	Num
Ra Tables	Ra0 RAM	89	28	I2	Ra 14	0	32767	46	Num

The subclass ID is defined by the decimal number 89, which is hexadecimal 0x59. To program this subclass ID, send the following sequence to the BQ27427 *after* entering CONFIG UPDATE mode.

	Commands	Description
1	W: AA 3E 59 00	Write to I^2C address AA, register 3E, the bytes 0x59 and 0x00. That is, 0x59 is written to 0x3E and 0x00 is written to address 0x3F. This is to setup the subclass ID for writes.
2	X: Delay (5 ms)	After setting up the subclass ID, delay for the BQ27427 to set up the block data to write all the 32 bytes. Wait a minimum of 5 ms after sending the command in (1).
3	W: AA 40 00 0B 00 0B 00 0D 00 11 00 0E 00 0C 00 0E 00 0C 00 0C 00 0D 00 0F 00 0F 00 17 00 2B 00 4B 00 00	Send 32 bytes (1 block) of data to address 0x40. This is the actual data for the gas gauge data memory area.
4	W: AA 60 “checksum”	Write the checksum into register 0x60
5	X: Delay (5 ms)	Need a delay after checksum operation of 5 ms
6	W: AA 3E 59 00	Write the block address for checksum computation
7	X: Delay (5 ms)	Need a delay for checksum operation of 5 ms
8	C: AA 60 D3	The checksum is computed and read from location 0x60 to compare the checksum to the checksum the gas gauge computed.

Revision History

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

DATE	REVISION	NOTES
January 2023	*	Initial Release

This page intentionally left blank.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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, regulatory 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 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.

TI objects to and rejects any additional or different terms you may have proposed.

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