

bq27520-G4 System-Side Impedance Track™ Fuel Gauge With Integrated LDO

Technical Reference Manual



Literature Number: SLUUA35

August 2013

Preface	5
1 General Description	6
2 Standard Data Commands	7
2.1 Control() : 0x00 and 0x01	8
2.1.1 CONTROL_STATUS: 0x0000	9
2.1.2 DEVICE_TYPE: 0x0001	9
2.1.3 FW_VERSION: 0x0002	9
2.1.4 PREV_MACWRITE: 0x0007	9
2.1.5 CHEM_ID: 0x0008	9
2.1.6 OCV_CMD: 0x000C	10
2.1.7 BAT_INSERT: 0x000D	10
2.1.8 BAT_REMOVE: 0x000E	10
2.1.9 SET_HIBERNATE: 0x0011	10
2.1.10 CLEAR_HIBERNATE: 0x0012	10
2.1.11 SET_SNOOZE: 0x0013	10
2.1.12 CLEAR_SNOOZE: 0x0014	10
2.1.13 DF_VERSION: 0x001F	10
2.1.14 SEALED: 0x0020	10
2.1.15 IT_ENABLE: 0x0021	11
2.1.16 RESET: 0x0041	11
2.2 AtRate() : 0x02 and 0x03	11
2.3 AtRateTimeToEmpty() : 0x04 and 0x05	11
2.4 Temperature() : 0x06 and 0x07	11
2.5 Voltage() : 0x08 and 0x09	11
2.6 Flags() : 0x0A and 0x0B	12
2.7 NominalAvailableCapacity() : 0x0C and 0x0D	12
2.8 FullAvailableCapacity() : 0x0E and 0x0F	12
2.9 RemainingCapacity() : 0x10 and 0x11	12
2.10 FullChargeCapacity() : 0x12 and 0x13	13
2.11 AverageCurrent() : 0x14 and 0x15	13
2.12 TimeToEmpty() : 0x16 and 0x17	13
2.13 StandbyCurrent() : 0x18 and 0x19	13
2.14 StandbyTimeToEmpty() : 0x1A and 0x1B	13
2.15 StateofHealth() : 0x1C and 0x1D	13
2.16 CycleCount() : 0x1E and 0x1F	14
2.17 StateOfCharge() : 0x20 and 0x21	14
2.18 InstantaneousCurrent() : 0x22 and 0x23	14
2.19 InternalTemperature() : 0x28 and 0x29	14
2.20 ResistanceScale() : 0x2A and 0x2B	14
2.21 OperationConfiguration() : 0x2C and 0x2D	14
2.22 DesignCapacity() : 0x2E and 0x2F	14
2.23 UnfilteredRM() : 0x6C and 0x6D	14
2.24 FilteredRM() : 0x6E and 0x6F	14
2.25 UnfilteredFCC() : 0x70 and 0x71	14
2.26 FilteredFCC() : 0x72 and 0x73	15

2.27	TrueSOC() : 0x74 and 0x75	15
3	Extended Data Commands	16
3.1	DataFlashClass() : 0x3E	16
3.2	DataFlashBlock() : 0x3F	16
3.3	BlockData() : 0x40 to 0x5F	16
3.4	BlockDataChecksum() : 0x60	17
3.5	BlockDataControl() : 0x61	17
3.6	ApplicationStatus() : 0x6A	17
4	Data Flash Interface	18
4.1	Accessing The Data Flash	18
4.2	Manufacturer Information Block	19
4.3	Device Access Modes	19
4.4	Sealing and Unsealing Data Flash	19
4.5	Data Flash Summary	20
4.6	Data Flash Parameter Update Example	27
4.6.1	Modify WRTEMP of OpConfig B Register	28
5	Functional Description	30
5.1	Impedance Track™ Variables	30
5.1.1	Load Mode	30
5.1.2	Load Select	30
5.1.3	Reserve Cap-mAh, Reserve Cap-mWh/cWh	31
5.1.4	Design Energy Scale	31
5.1.5	Dsg Current Threshold	31
5.1.6	Chg Current Threshold	31
5.1.7	Quit Current, Dsg Relax Time, Chg Relax Time, and Quit Relax Time	31
5.1.8	Qmax Cell 0 and Qmax Cell 1	32
5.1.9	Update Status 0 and Update Status 1	32
5.1.10	Avg I Last Run	32
5.1.11	Avg P Last Run	32
5.1.12	Delta Voltage	32
5.1.13	Default Ra and Ra Tables	32
5.1.14	Fast Resistance Scaling	33
5.1.15	Fast Qmax Update	33
5.1.16	SOC Smoothing	34
5.1.17	Flash Updates	34
5.2	Device Configuration	35
5.2.1	Operation Configuration (Op Config) Register	35
5.2.2	Operation Configuration B (OpConfig B) Register	36
5.2.3	Operation Configuration C (OpConfig C) Register	36
5.2.4	Operation Configuration D (OpConfig D) Register	37
5.2.5	Operation Configuration E (OpConfig E) Register	37
5.3	External Pin Functions	38
5.3.1	Pin Function Code (PFC) Descriptions	38
5.3.2	BAT_LOW Pin	38
5.3.3	Battery Presence Detection Using the BI/TOUT Pin	38
5.3.4	SOC_INT Pin Behavior	39
5.3.5	Power Path Control With the BAT_GD Pin	39
5.3.6	Wake-Up Comparator	40
5.3.7	Autocalibration	40
5.4	Temperature Measurement	41
5.4.1	Overtemperature Indication	41
5.5	Charging and Charge—Termination Indication	42
5.5.1	Detecting Charge Termination	42

5.5.2	Charge Inhibit and Suspend	42
5.6	Power Modes	43
5.6.1	BAT INSERT CHECK Mode	46
5.6.2	NORMAL Mode	46
5.6.3	SLEEP Mode	46
5.6.4	SNOOZE Mode	46
5.6.5	HIBERNATE Mode	47
5.7	Application-Specific Information	47
5.7.1	Battery Profile Storage and Selection	47
5.7.2	First OCV and Impedance Measurement	48
5.8	Additional Data Flash Parameter Descriptions	48
5.8.1	TCA Set %	48
5.8.2	TCA Clear %	48
5.8.3	FC Set %	48
5.8.4	FC Clear %	48
5.8.5	DOD at EOC Delta Temperature	49
5.8.6	Default Temperature	49
5.8.7	Device Name	49
5.8.8	Data Flash Version	49
5.8.9	SOC1 Set Threshold	49
5.8.10	SOC1 Clear Threshold	49
5.8.11	Final Voltage and Final Volt Time	49
5.8.12	Def Avg I Last Run and Def Avg P Last Run	49
5.8.13	Max Res Factor	49
5.8.14	Min Res Factor	50
5.8.15	Ra Filter	50
5.8.16	ResRelax Time	50
5.8.17	Max Sim Rate, Min Sim Rate	50
5.8.18	Transient Factor Charge and Discharge	50
5.8.19	Max IR Correct	50
5.8.20	Thermal Modeling	50
5.8.21	Cell 0 and 1 V at Chg Term	51
5.8.22	Calibration, Data, ID = 104	51
6	Communications	53
6.1	I ² C Interface	53
6.2	I ² C Time Out	53
6.3	I ² C Command Waiting Time	54
6.4	I ² C Clock Stretching	54
7	Reference Schematic	55
A	Open-Circuit Voltage Measurement Background	57
A.1	Background	57
A.1.1	OCV Qualification and Calculation	57
A.1.2	OCV Calculation Assumption	57
A.1.3	OCV Timing	57
A.2	OCV Timing and OCV_CMD Use Recommendations	59
A.2.1	ACTIVE Mode (Fuel Gauge is not in SLEEP Mode)	59
A.2.2	SLEEP Mode	59
A.2.3	Initial OCV – POR	59
B	Glossary	61

Preface

This document is a detailed Technical Reference Manual (TRM) for using and configuring the bq27520-G4 battery fuel gauge. This TRM document is intended to complement but not supersede any information contained in the separate bq27520-G4 datasheet.

Refer to the [bq27520-G4 Datasheet \(SLUSB20\)](#).

Formatting conventions used in this document:

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

Related Documentation from Texas Instruments

To obtain a copy of any of the following TI documents, call the Texas Instruments Literature Response Center at (800) 477-8924 or the Product Information Center (PIC) at (972) 644-5580. When ordering, identify this document by its title and literature number. Updated documents also can be obtained through the TI Web site at www.ti.com.

1. *bq27520-G4, System-Side Impedance Track™ Fuel Gauge With Integrated LDO Data Sheet (SLUSB20)*
2. *Going to Production with the bq275xx Application Report (SLUA449)*
3. *Theory and Implementation of Impedance Track™ Battery Fuel-Gauging Algorithm in bq2750x Family Application Report (SLUA450)*
4. *Host System Calibration Method Application Report (SLUA640)*

Revision History

Version	Change Date	Description
—	August 2013	Initial Release

General Description

The bq27520-G4 fuel gauge accurately predicts the battery capacity and other operational characteristics of a single series, Li-based rechargeable cell. It can be interrogated by a system processor to provide cell information, such as time-to-empty (TTE), state-of-charge (SOC), and the SOC interrupt signal to the host.

Information is accessed through a series of commands, called *Standard Commands*. Further capabilities are provided by the additional *Extended Commands* set. Both sets of commands, indicated by the general format *Command()*, read and write information contained within the device control and status registers, as well as its data flash locations. Commands are sent from system to gauge using the I²C™ serial communications engine, and can be executed during application development, system manufacture, or end-equipment operation.

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

The key to the high-accuracy gas gauging prediction is Texas Instruments proprietary Impedance Track™ algorithm. This algorithm uses cell measurements, characteristics, and properties to create state-of-charge predictions that can achieve less than 1% error across a wide variety of operating conditions and over the lifetime of the battery. See application report [SLUA450](#), *Theory and Implementation of Impedance Track™ Battery Fuel-Gauging Algorithm in bq2750x Family*.

The fuel gauge measures charge and discharge activity by monitoring the voltage across a small-value series sense resistor (5 mΩ to 20 mΩ, typical) located between the system V_{SS} and the battery PACK-terminal. When a cell is attached to the device, cell impedance is learned, based on cell current, cell open-circuit voltage (OCV), and cell voltage under loading conditions.

The external temperature sensing is optimized with the use of a high-accuracy negative temperature coefficient (NTC) thermistor with R₂₅ = 10.0 kΩ ±1%. B_{25/85} = 3435 kΩ ± 1% (such as Semitec NTC 103AT). Alternatively, the fuel gauge can also be configured to use its internal temperature sensor or receive temperature data from the host processor. When an external thermistor is used, a 18.2-kΩ pull-up resistor between BI/TOUT and TS pins is also required. The fuel gauge uses temperature to monitor the battery-pack environment, which is used for fuel gauging and cell protection functionality.

To minimize power consumption, the device has different power modes: NORMAL, SNOOZE, SLEEP, HIBERNATE, and BAT INSERT CHECK. 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. More details can be found in [Section 5.6](#), *Power Modes*.

Standard Data Commands

The bq27520-G4 fuel gauge uses a series of 2-byte standard commands to enable system reading and writing of battery information. Each standard command has an associated command-code pair, as indicated in [Table 2-1](#). Because each command consists of two bytes of data, two consecutive I²C transmissions must be executed both to initiate the command function and to read or write the corresponding two bytes of data. Additional options for transferring data are described in [Chapter 3, Extended Data Commands](#). Read and write permissions depend on the active access mode, SEALED or UNSEALED. For details, see [Section 4.3, Device Access Modes](#). See [Chapter 6, Communications](#), for I²C details.

Table 2-1. Standard Commands

NAME		COMMAND CODE	UNIT	SEALED ACCESS
<i>Control()</i>	CNTL	0x00 and 0x01	NA	RW
<i>AtRate()</i>	AR	0x02 and 0x03	mA	RW
<i>AtRateTimeToEmpty()</i>	ARTTE	0x04 and 0x05	Minutes	R
<i>Temperature()</i>	TEMP	0x06 and 0x07	0.1°K	RW
<i>Voltage()</i>	VOLT	0x08 and 0x09	mV	R
<i>Flags()</i>	FLAGS	0x0A and 0x0B	NA	R
<i>NominalAvailableCapacity()</i>	NAC	0x0C and 0x0D	mAh	R
<i>FullAvailableCapacity()</i>	FAC	0x0E and 0x0F	mAh	R
<i>RemainingCapacity()</i>	RM	0x10 and 0x11	mAh	R
<i>FullChargeCapacity()</i>	FCC	0x12 and 0x13	mAh	R
<i>AverageCurrent()</i>	AI	0x14 and 0x15	mA	R
<i>TimeToEmpty()</i>	TTE	0x16 and 0x17	Minutes	R
<i>StandbyCurrent()</i>	SI	0x18 and 0x19	mA	R
<i>StandbyTimeToEmpty()</i>	STTE	0x1A and 0x1B	Minutes	R
<i>StateOfHealth()</i>	SOH	0x1C and 0x1D	% / num	R
<i>CycleCount()</i>	CC	0x1E and 0x1F	num	R
<i>StateOfCharge()</i>	SOC	0x20 and 0x21	%	R
<i>InstantaneousCurrent()</i>		0x22 and 0x23	mA	R
<i>InternalTemperature()</i>	INTTEMP	0x28 and 0x29	0.1°K	R
<i>ResistanceScale()</i>		0x2A and 0x2B		R
<i>OperationConfiguration()</i>	Op Config	0x2C and 0x2D	NA	R
<i>DesignCapacity()</i>		0x2E and 0x2F	mAh	R
<i>UnfilteredRM()</i>	UFRM	0x6C and 0x6D	mAh	R
<i>FilteredRM()</i>	FRM	0x6E and 0x6F	mAh	R
<i>UnfilteredFCC()</i>	UFFCC	0x70 and 0x71	mAh	R
<i>FilteredFCC()</i>	FFCC	0x72 and 0x73	mAh	R
<i>TrueSOC()</i>	UFSOC	0x74 and 0x75	%	R

2.1 Control(): 0x00 and 0x01

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

Table 2-2. Control() Subcommands

CNTL FUNCTION	CNTL DATA	SEALED ACCESS	DESCRIPTION
CONTROL_STATUS	0x0000	Yes	Reports the status of DF checksum, hibernate, Impedance Track™, etc.
DEVICE_TYPE	0x0001	Yes	Reports the device type (for example: 0x0520)
FW_VERSION	0x0002	Yes	Reports the firmware version on the device type
PREV_MACWRITE	0x0007	Yes	Returns previous <i>Control()</i> subcommand code
CHEM_ID	0x0008	Yes	Reports the chemical identifier of the Impedance Track™ configuration
OCV_CMD	0x000C	Yes	Requests the fuel gauge to take an OCV measurement
BAT_INSERT	0x000D	Yes	Forces <i>Flags()</i> [BAT_DET] bit set when OpConfig B [BIE] bit = 0
BAT_REMOVE	0x000E	Yes	Forces <i>Flags()</i> [BAT_DET] bit clear when OpConfig B [BIE] bit = 0
SET_HIBERNATE	0x0011	Yes	Forces <i>CONTROL_STATUS</i> [HIBERNATE] bit to 1
CLEAR_HIBERNATE	0x0012	Yes	Forces <i>CONTROL_STATUS</i> [HIBERNATE] bit to 0
SET_SNOOZE	0x0013	Yes	Forces <i>CONTROL_STATUS</i> [SNOOZE] bit to 1
CLEAR_SNOOZE	0x0014	Yes	Forces <i>CONTROL_STATUS</i> [SNOOZE] bit to 0
DF_VERSION	0x001F	Yes	Returns the Data Flash Version code
SEALED	0x0020	No	Places the fuel gauge in SEALED access mode
IT_ENABLE	0x0021	No	Enables the Impedance Track™ (IT) algorithm
RESET	0x0041	No	Forces a full reset of the fuel gauge

Example using *DEVICE_TYPE* subcommand:

- To device address 0xAA, starting at command 0x00, write two bytes of data: 0x01 and 0x00.
- Then read the response using an incremental read. To device address 0xAB, starting at command 0x00, read two bytes.

2.1.1 CONTROL_STATUS: 0x0000

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

Table 2-3. CONTROL_STATUS Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	–	FAS	SS	–	CCA	BCA	OCVCMDCOMP	OCVFAIL
Low Byte	INITCOMP	HIBERNATE	SNOOZE	SLEEP	LDMD	RUP_DIS	VOK	QEN

High Byte

FAS = Status bit indicating the fuel gauge is in FULL ACCESS SEALED state. Active when set.

SS = Status bit indicating the fuel gauge is in SEALED state. Active when set.

CCA = Status bit indicating the fuel gauge Coulomb Counter Calibration routine is active. The CCA routine takes place approximately 1 minute after the initialization and periodically as gauging conditions change. Active when set. (See [Section 5.3.7, Autocalibration](#))

BCA = Status bit indicating the fuel gauge board calibration routine is active. Active when set.

OCVCMDCOMP = Status bit indicating the fuel gauge has executed the OCV command. This bit can only be set with the presence of a battery. True when set.

OCVFAIL = Status bit indicating an OCV reading failed due to the current. This bit can only be set with the presence of a battery. True when set.

Low Byte

INITCOMP = Initialization completion bit indicating the initialization completed. This bit can only be set with the presence of a battery and can be monitored to determine when fuel gauge values are valid. It is recommended to poll this bit at initialization or startup. True when set.

HIBERNATE = Status bit indicating a request for entry into the HIBERNATE mode from the SLEEP mode. True when set. Default is 0.

SNOOZE = Status bit indicating the SNOOZE mode is enabled. True when set.

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

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

RUP_DIS = When set, this status bit indicates resistance table updates are disabled. This bit is set on initialization or initial battery insertion, or if resistance updates exceed allowable values. This bit automatically clears after sufficient battery relaxation.

VOK = Status bit indicating *Voltage()* is okay for Qmax updates and calculations. True when set.

QEN = Status bit indicating Qmax updates are enabled during end-equipment operation as long as Impedance Track™ is enabled. True when set.

2.1.2 DEVICE_TYPE: 0x0001

Instructs the fuel gauge to return the device type to addresses 0x00 and 0x01. The bq27520-G4 device type returned is 0x0520.

2.1.3 FW_VERSION: 0x0002

Instructs the fuel gauge to return the firmware version (0x0329) to addresses 0x00 and 0x01.

2.1.4 PREV_MACWRITE: 0x0007

Instructs the fuel gauge to return the previous subcommand written to addresses 0x00 and 0x01.

NOTE: This subcommand is only supported for previous subcommand codes 0x0000 through 0x0014. For subcommand codes greater than 0x0009, a value of 0x0007 is returned.

2.1.5 CHEM_ID: 0x0008

Instructs the fuel gauge to return the chemical identifier value stored in data flash (see [Table 4-6](#)) for the Impedance Track™ configuration to addresses 0x00 and 0x01.

2.1.6 OCV_CMD: 0x000C

Requests the fuel gauge to take an open-circuit voltage (OCV) reading. This command can only be issued after the *CONTROL_STATUS [INITCOMP]* bit is set, indicating the initialization has been completed. The OCV measurement takes place at the beginning of the next repeated 1-second firmware synchronization clock. If the **OpConfig D [SOC_OCV]** bit is set, the SOC_INT pin pulses for approximately 165 ms to indicate the measurement window. (See also [Table 5-10](#).) See [Appendix A, Open-Circuit Voltage Measurement Background](#), for more details on OCV measurements and recommended usage of this command.

NOTE: The *CONTROL_STATUS [OCVFAIL]* bit is set if the *OCV_CMD* subcommand is received when the *Flags() [CHG_INH]* bit is set.

2.1.7 BAT_INSERT: 0x000D

Instructs the fuel gauge to force the *Flags() [BAT_DET]* bit to be set and informs the gauge of the presence of a battery when the insertion detection feature is disabled (**OpConfig B [BIE]** bit = 0). Alternatively, battery presence detection can be enabled (**OpConfig B [BIE]** bit = 1) to monitor the external thermistor network. (See [Section 5.3.3, Battery Presence Detection Using the BI/TOUT Pin](#).)

2.1.8 BAT_REMOVE: 0x000E

Instructs the fuel gauge to force the *Flags() [BAT_DET]* bit to clear when the battery insertion detection is disabled. (**OpConfig B [BIE]** bit = 0). Alternatively, battery presence detection can be enabled (**OpConfig B [BIE]** bit = 1) to monitor the external thermistor network. (See [Section 5.3.3, Battery Presence Detection Using the BI/TOUT Pin](#).)

2.1.9 SET_HIBERNATE: 0x0011

Instructs the fuel gauge to force the *CONTROL_STATUS [HIBERNATE]* bit to 1. This allows the gauge to enter the HIBERNATE power mode after the transition to the SLEEP power mode is detected and the required conditions are met. The *[HIBERNATE]* bit is automatically cleared upon exiting the HIBERNATE mode.

2.1.10 CLEAR_HIBERNATE: 0x0012

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

2.1.11 SET_SNOOZE: 0x0013

Instructs the fuel gauge to set the *CONTROL_STATUS [SNOOZE]* bit to 1. This enables the SNOOZE power mode. The gauge enters the SNOOZE power mode after the transition conditions are met.

2.1.12 CLEAR_SNOOZE: 0x0014

Instructs the fuel gauge to set the *CONTROL_STATUS [SNOOZE]* bit to 0. This disables the SNOOZE power mode. The gauge exits from the SNOOZE power mode after the *SNOOZE* bit is cleared.

2.1.13 DF_VERSION: 0x001F

Instructs the fuel gauge to return the 16-bit data flash revision code to addresses 0x00 and 0x01. The code is stored in **Data Flash Version** and provides a simple method for the customer to control data flash revisions. The default *DF_VERSION* is 0x0000 as configured in data flash.

2.1.14 SEALED: 0x0020

Instructs the fuel gauge to transition from the UNSEALED state to the SEALED state. The fuel gauge must always be set to the SEALED state for use in end-equipment.

2.1.15 IT_ENABLE: 0x0021

Forces the fuel gauge to begin the Impedance Track™ algorithm, and sets **IT Enable** = 0x01 and both **CONTROL_STATUS [VOK, QEN]** bits = 1. The [VOK] bit is cleared if **Voltage()** is not suitable for a Qmax update. This subcommand is only available when the fuel gauge is UNSEALED and is typically enabled at the last step of production after system test is completed. If it is not enabled, then Qmax and Ra cannot be learned.

2.1.16 RESET: 0x0041

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

2.2 AtRate(): 0x02 and 0x03

The **AtRate()** read- and write-word function is the first half of a two-function command set that sets the **AtRate** value used in calculations made by the **AtRateTimeToEmpty()** function. The **AtRate()** units are in mA.

The **AtRate()** value is a signed integer, with negative values interpreted as a discharge current value. The **AtRateTimeToEmpty()** function returns the predicted operating time at the **AtRate** value of discharge. The default value for **AtRate()** is 0 and forces **AtRateTimeToEmpty()** to return 65,535. Both the **AtRate()** and **AtRateTimeToEmpty()** commands must only be used in the NORMAL mode.

2.3 AtRateTimeToEmpty(): 0x04 and 0x05

This read-word function returns an unsigned integer value of the predicted remaining operating time if the battery is discharged at the **AtRate()** value in minutes with a range of 0 to 65,534. A value of 65,535 indicates **AtRate()** = 0. The fuel gauge updates **AtRateTimeToEmpty()** within 1 second after the system sets the **AtRate()** value. The fuel gauge automatically updates **AtRateTimeToEmpty()** based on the **AtRate()** value every second. Both the **AtRate()** and **AtRateTimeToEmpty()** commands must only be used in the NORMAL mode.

2.4 Temperature(): 0x06 and 0x07

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. See [Table 2-4, Temperature Measurement Options](#), and [Section 5.4, Temperature Measurement](#).

Table 2-4. Temperature Measurement Options

OpConfig B [WRTEMP]	Op Config [TEMPS]	Temperature() Read Command	Temperature() Write Command
0	0	Returns internal temperature as read from an internal sensor. This data is also available using the <i>InternalTemperature()</i> function.	The data is ignored.
0	1	Returns external temperature read from an external thermistor.	
1	X	Returns the <i>Temperature()</i> value previously written.	Sets the <i>Temperature()</i> to be used for gauging calculations.

2.5 Voltage(): 0x08 and 0x09

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

2.6 **Flags():** 0x0A and 0x0B

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

Table 2-5. Flags Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	OTC	OTD	–	CALMODE	CHG_INH	XCHG	FC	CHG
Low Byte	–	–	OCV_GD	WAIT_ID	BAT_DET	SOC1	SYSDOWN	DSG

High Byte

- OTC = Overtemperature in charge condition is detected. True when set. See [Table 4-3](#), Safety Subclass parameters for threshold settings. If the **OpConfig D [SOC_OT]** bit = 1, SOC_INT pin toggles once [OTC] bit is set.
- OTD = Overtemperature in discharge condition is detected. True when set. If the **OpConfig D [SOC_OT]** bit = 1, SOC_INT pin toggles once [OTD] bit is set.
- CALMODE = Status bit indicating the calibration function is active. True when set. This bit must be cleared when the device is in NORMAL mode.
- CHG_INH = Charge inhibit: If set, indicates that charging should not begin because *Temperature()* is outside the range [**Charge Inhibit Temp Low, Charge Inhibit Temp High**]. True when set.
- XCHG = Charge suspend alert (temperature outside the range [**Suspend Temperature Low, Suspend Temperature High**]). True when set.
- FC = Full-charged is detected. If **FC Set%** = –1, the [FC] bit is set when the fuel gauge has detected charge termination. Alternatively, if **FC Set%** is configured with a positive % threshold, the [FC] bit is set when the *StateOfCharge()* is larger than the **FC Set%** threshold and cleared when the *StateOfCharge()* is lower than the **FC Clear%** threshold. (See [Section 5.5](#), *Charging and Charge Termination Indication*)
- CHG = Indicates OK to charge. This bit is set if *StateOfCharge()* is below **TCA Set %** and *Temperature()* is within the ranges set by **Chg Inhibit Temp Low/High** and **Suspend Low/High Temp**. This bit is cleared when *StateOfCharge()* rises above **TCA Clear %**, unless **TCA Clear %** is set to –1. If **TCA Clear %** = –1, then this bit is cleared when primary charge termination is detected. If **FC Set %** = –1, then the CHG bit clears at the same moment as the FC bit is set when primary charge termination is detected.

Low Byte

- OCV_GD = Good OCV measurement taken. True when set.
- WAIT_ID = Waiting to identify inserted battery. True when set.
- BAT_DET = Battery detected. True when set.
- SOC1 = State-of-charge threshold 1 (**SOC1 Set**) reached. SOC_INT pin toggles once when this bit is set or cleared if **OpConfig B [BL_INT]** bit = 1. True when set.
- SYSDOWN = System down bit indicating the system should shut down. See [Table 4-3](#), Discharge Subclass, **SysDown** parameters for threshold settings. True when set. SOC_INT pin toggles once if set.
- DSG = Discharging detected. True when set.

2.7 **NominalAvailableCapacity():** 0x0C and 0x0D

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

2.8 **FullAvailableCapacity():** 0x0E and 0x0F

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

2.9 **RemainingCapacity():** 0x10 and 0x11

This read-only command pair returns the compensated battery capacity remaining (*UnfilteredRM()*) when the **OpConfig D [SMTHEN]** bit is cleared or filtered compensated battery capacity remaining (*FilteredRM()*) when the **[SMTHEN]** bit is set. Units are mAh.

2.10 FullChargeCapacity(): 0x12 and 0x13

This read-only command pair returns the compensated capacity of fully charged battery (*UnfilteredFCC()*) when the **OpConfig D [SMTHEN]** bit is cleared or filtered compensated capacity of fully charged battery (*FilteredFCC()*) when the **[SMTHEN]** bit is set. Units are mAh. *FullChargeCapacity()* is updated at regular intervals, as specified by the Impedance Track™ algorithm.

2.11 AverageCurrent(): 0x14 and 0x15

This read-only command pair returns a signed integer value that is the average current flow through the sense resistor. 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 are averaged out in this measurement. Units are mA.

2.12 TimeToEmpty(): 0x16 and 0x17

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

2.13 StandbyCurrent(): 0x18 and 0x19

This read-only function returns a signed integer value of the measured standby current through the sense resistor. The *StandbyCurrent()* is an adaptive measurement. Initially it reports the standby current programmed in **Initial Standby**, and after spending several seconds in standby, reports the measured standby current.

The register value is updated every second when the measured current is above the **Deadband** and is less than or equal to $2 \times$ **Initial Standby**. The first and last values that meet this criteria are not included, because they may not be stable values. To approximate a 1-minute time constant, each new *StandbyCurrent()* value is computed by taking approximately 93% weight of the last standby current and approximately 7% of the current measured average current.

2.14 StandbyTimeToEmpty(): 0x1A and 0x1B

This read-only function returns an unsigned integer value of the predicted remaining battery life at the standby rate of discharge in minutes. The computation uses *NominalAvailableCapacity()* (NAC), the uncompensated remaining capacity, for this computation. A value of 65,535 indicates battery is not being discharged.

2.15 StateofHealth(): 0x1C and 0x1D

0x28 SOH percentage: this read-only function returns an unsigned integer value, expressed as a percentage of the ratio of predicted *FCC(25°C, SOH LoadI)* over the *DesignCapacity()*. The *FCC(25°C, SOH LoadI)* is the calculated full charge capacity at 25°C and the *SOH LoadI* which is specified in the data flash. The range of the returned SOH percentage is 0x00 to 0x64, indicating 0 to 100%, correspondingly.

0x29 SOH status: this read-only function returns an unsigned integer value, indicating the status of the SOH percentage. The meanings of the returned value are:

- 0x00: SOH not valid before initialization
- 0x01: Instant SOH value ready
- 0x02: Initial SOH value ready. The calculation is based on unlearned Qmax and is updated at the first grid point during discharge after cell insertion.
- 0x03: SOH value ready. The calculation is based on an updated learned Qmax value. The updated Qmax value is measured after charge/relaxation or Fast Qmax conditions are met.
- 0x04 to 0xFF: Reserved

2.16 **CycleCount(): 0x1E and 0x1F**

This read-only function returns an unsigned integer value of the number of cycles that the active cell has experienced with a range of 0 to 65535. One cycle occurs when accumulated discharge \geq **CC Threshold**. The gauge maintains a separate cycle counter for both cell profiles and resets to 0 if the insertion of a new pack has been detected.

2.17 **StateOfCharge(): 0x20 and 0x21**

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%. $StateOfCharge() = RemainingCapacity() \div FullChargeCapacity()$ rounded up to the nearest whole percentage point.

2.18 **InstantaneousCurrent(): 0x22 and 0x23**

This read-only function returns a signed integer value that is the instantaneous current flow through the sense resistor. The conversion time is 125 ms. It is updated every second. Units are mA.

2.19 **InternalTemperature(): 0x28 and 0x29**

This read-only function returns an unsigned integer value of the internal temperature sensor in units of 0.1°K measured by the fuel gauge. This function can be useful as an additional system-level temperature monitor if the main *Temperature()* function is configured for external or host-reported temperature.

2.20 **ResistanceScale(): 0x2A and 0x2B**

This read-only function returns the resistance scale value when the Fast Resistance Scaling feature is enabled via the **OpConfig B [FCE]** bit. (See [Section 5.1.14, Fast Resistance Scaling](#).)

2.21 **OperationConfiguration(): 0x2C and 0x2D**

This read-only function returns the contents of the data flash **Op Config** register and is most useful for system level debug to quickly determine device configuration.

2.22 **DesignCapacity(): 0x2E and 0x2F**

This read-only function returns the value stored in **Design Capacity** and is expressed in mAh. This is intended to be the theoretical or nominal capacity of a new pack, and is used for the calculation of *StateOfHealth()*.

2.23 **UnfilteredRM(): 0x6C and 0x6D**

This read-only command pair returns the compensated battery capacity remaining. When the **OpConfig D [SMTHEN]** bit is cleared, this value is reported in the *RemainingCapacity()* register. Units are mAh.

2.24 **FilteredRM(): 0x6E and 0x6F**

This read-only command pair returns the filtered compensated battery capacity remaining. When the **OpConfig D [SMTHEN]** bit is set, this value is reported in the *RemainingCapacity()* register. Units are mAh.

2.25 **UnfilteredFCC(): 0x70 and 0x71**

This read-only command pair returns the compensated capacity of the battery when fully charged. When the **OpConfig D [SMTHEN]** bit is cleared, this value is reported in the *RemainingCapacity()* register. Units are mAh. *UnFilteredFCC()* is updated at regular intervals, as specified by the Impedance Track™ algorithm.

2.26 *FilteredFCC()*: 0x72 and 0x73

This read-only command pair returns the filtered compensated capacity of the battery when fully charged. When the **OpConfig D [SMTHEN]** bit is set, this value is reported in the *RemainingCapacity()* register. Units are mAh. *FilteredFCC()* is updated at regular intervals, as specified by the Impedance Track™ algorithm.

2.27 *TrueSOC()*: 0x74 and 0x75

This read-only function returns an unsigned integer value of the predicted remaining battery capacity expressed as a percentage of *UnfilteredFCC()*, with a range of 0 to 100%. When the **OpConfig D [SMTHEN]** bit is cleared, this value is reported in the *StateOfCharge()* register.

Extended Data Commands

Extended commands offer additional functionality beyond the standard set of commands. They are used in the same manner; however, unlike standard commands, extended commands are not limited to 2-byte words. The number of command bytes for a given extended command range in size from single to multiple bytes is specified in [Table 3-1](#). See [Section 4.1](#) for details on accessing the data flash and [Section 4.6](#) for an example to update a data flash parameter.

Table 3-1. Extended Data Commands

Name	Command Code	Unit	SEALED Access ^{(1) (2)}	UNSEALED Access ^{(1) (2)}
Reserved	0x34 to 0x3D	NA	R	R
<i>DataFlashClass()</i> ⁽²⁾	0x3E	NA	NA	RW
<i>DataFlashBlock()</i> ⁽²⁾	0x3F	NA	RW	RW
<i>BlockData()</i>	0x40 to 0x5F	NA	R	RW
<i>BlockDataChecksum()</i>	0x60	NA	RW	RW
<i>BlockDataControl()</i>	0x61	NA	NA	RW
<i>ApplicationStatus()</i>	0x6A	NA	R	R
Reserved	0x6B to 0x7F	NA	R	R

⁽¹⁾ SEALED and UNSEALED states are entered via commands to *Control()* 0x00 and 0x01.

⁽²⁾ In sealed mode, data flash cannot be accessed through commands 0x3E and 0x3F.

3.1 *DataFlashClass()*: 0x3E

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

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

3.2 *DataFlashBlock()*: 0x3F

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

SEALED Access: This command directs which data flash block is accessed by the *BlockData()* command. Writing a 0x01 or 0x02 instructs the *BlockData()* command to transfer the **Manufacturer Info Block**. All other *DataFlashBlock()* values are reserved.

3.3 *BlockData()*: 0x40 to 0x5F

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

SEALED Access: This data block is the remainder of the 32-byte data block when accessing **Manufacturer Info Block**.

3.4 **BlockDataChecksum(): 0x60**

UNSEALED Access: This byte contains the checksum on the 32 bytes of block data read from or written to data flash. The least-significant byte of the sum of the data bytes written must be complemented ($[255 - x]$, x is the least-significant byte) before being written to 0x60.

SEALED Access: This byte contains the checksum for the 32 bytes of block data written to **the Manufacturer Info Block**. The least-significant byte of the sum of the data bytes written must be complemented ($[255 - x]$, x is the least-significant byte) before being written to 0x60.

3.5 **BlockDataControl(): 0x61**

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

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

3.6 **ApplicationStatus(): 0x6A**

This byte function allows the system to read the **Application Status** data flash location. See [Table 5-12, ApplicationStatus\(\) Bit Definitions](#), for specific bit definitions.

Data Flash Interface

4.1 Accessing The Data Flash

The bq27520-G4 data flash is a non-volatile memory that contains initialization, default, cell status, calibration, configuration, and user information. The data flash can be accessed in several different ways, depending in what mode the fuel gauge is operating and what data is being accessed.

Commonly accessed data flash memory locations, frequently read by a system, are conveniently accessed through specific instructions, already described in [Chapter 3, Extended Data Commands](#). These commands are available when the fuel gauge is either in UNSEALED mode or SEALED mode.

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

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

Occasionally, a data flash class is larger than the 32-byte block size. In this case, the *DataFlashBlock()* command designates in which 32-byte block the desired locations reside. The correct command address is then given by $0x40 + \text{offset} \textit{ modulo } 32$. For example, to access **Terminate Voltage** in the *Gas Gauging* class, *DataFlashClass()* is issued 80 (0x50) to set the class. Because the offset is 55, it must reside in the second 32-byte block. Hence, *DataFlashBlock()* is issued 0x01 to set the block offset, and the offset used to index into the *BlockData()* memory area is $0x40 + 55 \textit{ modulo } 32 = 0x40 + 23 = 0x40 + 0x17 = 0x57$.

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

None of the data written to memory are bounded by the bq27520-G4 fuel gauge – the values are not rejected by the fuel gauge. Writing an incorrect value may result in hardware failure due to firmware program interpretation of the invalid data. The written data is persistent, so a power-on reset does not resolve the fault.

4.2 Manufacturer Information Block

The fuel gauge contains 32 bytes of user programmable data flash storage called the **Manufacturer Info Block**. The method for accessing these memory locations is slightly different, depending on whether the device is in UNSEALED or SEALED mode.

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

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

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

4.3 Device Access Modes

The fuel gauge provides three security modes (FULL ACCESS, UNSEALED, and SEALED) that control data flash access permissions, according to [Table 4-1](#).

Table 4-1. Data Flash Access

Security Mode	Data Flash	Manufacturer Info Block
FULL ACCESS	RW	RW
UNSEALED	RW	RW
SEALED	None	R

Although FULL ACCESS and UNSEALED modes appear identical, only FULL ACCESS allows the fuel gauge to read or write access-mode transition keys.

4.4 Sealing and Unsealing Data Flash

The fuel gauge implements a key-access scheme to transition between SEALED, UNSEALED, and FULL ACCESS modes. Each transition requires that a unique set of two keys be sent to the fuel gauge via the *Control()* control command. The keys must be sent consecutively, with no other data being written to the *Control()* register in between. Do not set the two keys to identical values.

NOTE: To avoid conflict, the keys must be different from the codes presented in the *CNTL DATA* column of [Table 2-2, Control\(\) Subcommands](#).

When in the SEALED mode, the *CONTROL_STATUS [SS]* bit is set, but when the UNSEAL keys are correctly received by the fuel gauge, the *[SS]* bit is cleared. When the FULL ACCESS keys are correctly received, then the *CONTROL_STATUS [FAS]* bit is cleared.

Both sets of keys for each level are 2 bytes each in length and are stored in data flash. The UNSEAL key (stored at **Unseal Key 0** and **Unseal Key 1**) and the FULL ACCESS keys (stored at **Full-Access Key 0** and **Full-Access Key 1**) can only be updated when in FULL ACCESS mode. The order of the keys is **Key 1** followed by **Key 0**. The order of the bytes entered through the *Control()* command is the reverse of what is read from the part. For example, if the **Key 1** and **Key 0** of the **UNSEAL keys** returns 0x1234 and 0x5678, then the *Control()* should supply 0x3412 and 0x7856 to unseal the part.

4.5 Data Flash Summary

Table 4-3 through Table 4-10 summarize the data flash locations available to the user, including their default, minimum, and maximum values.

Table 4-2. Data Type Decoder

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

Table 4-3. Data Flash Summary—Configuration Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
2	Safety	0	OT Chg	I2	0	1200	550	0.1°C
		2	OT Chg Time	U1	0	60	2	s
		3	OT Chg Recovery	I2	0	1200	500	0.1°C
		5	OT Dsg	I2	0	1200	600	0.1°C
		7	OT Dsg Time	U1	0	60	2	s
		8	OT Dsg Recovery	I2	0	1200	550	0.1°C
32	Charge Inhibit Cfg	0	Chg Inhibit Temp Low	I2	-400	1200	0	0.1°C
		2	Chg Inhibit Temp High	I2	-400	1200	450	0.1°C
		4	Temp Hys	I2	0	100	50	0.1°C
34	Charge	2	Charging Voltage	I2	0	4600	4200	mV
		4	Delta Temp	I2	0	500	50	0.1°C
		6	Suspend Low Temp	I2	-400	1200	-50	0.1°C
		8	Suspend High Temp	I2	-400	1200	550	0.1°C
36	Charge Termination	2	Taper Current	I2	0	1000	100	mA
		6	Taper Voltage	I2	0	1000	100	mV
		9	TCA Set %	I1	-1	100	99	%
		10	TCA Clear %	I1	-1	100	95	%
		11	FC Set %	I1	-1	100	-1	%
		12	FC Clear %	I1	-1	100	98	%
		13	DODatEOC Delta T	I2	0	1000	50	0.1°C

Table 4-3. Data Flash Summary—Configuration Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
48	Data	4	Initial Standby	I1	-256	0	-10	mA
		7	CC Threshold	I2	100	32767	900	mAh
		10	Design Capacity	I2	0	65535	1000	mAh
		12	Des Energy Scale	U1	0	65535	1	num
		13	SOH LoadI	I2	-32767	0	-400	mA
		15	Default Temperature	I2	2732	3732	2982	0.1°C
		17	Device Name	S8	x	x	bq27520	
		25	Data Flash Version	H2	0x0	0xFFFF	0x0	num
49	Discharge	0	SOC1 Set Threshold	U2	0	5000	150	mAh
		2	SOC1 Clear Threshold	U2	0	5000	175	mAh
		9	SysDown Set Volt Threshold	I2	0	4200	3150	mV
		11	SysDown Set Volt Time	U1	0	60	2	s
		12	SysDown Clear Volt	I2	0	4200	3400	mV
		14	Final Voltage	U2	0	4200	3100	mV
		16	Final Volt Time	U1	0	60	2	s
		21	Def Avg I Last Run	I2	-32768	32767	-299	mA
		23	Def Avg P Last Run	I2	-32768	32767	-1131	mW
64	Registers	0	Op Config	H2	0x0000	0xFFFF	0x0973	flags
		7	SOC Delta	U1	0	25	1	%
		8	i2c Timeout	U1	0	7	4	%
		9	DF Wr Ind Wait	U2	0	65535	0	%
		11	OpConfig B	H1	0x00	0xFF	0x4A	flags
		12	OpConfig C	H1	0x00	0xFF	0x2C	flags
		13	OpConfig D	H1	0x00	0xFF	0x5E	flags
		14	OpConfig E	H1	0x00	0xFF	0x00	flags
68	Power	0	Flash Update OK Voltage	I2	0	4200	2800	mV
		4	Sleep Current	I2	0	100	10	mA
		13	Hibernate I	U2	0	700	8	mA
		15	Hibernate V	U2	2400	3000	2550	mV

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

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
57	Manufacturer Info	0 through 31	Block 0 through 31	H1	0x00	0xFF	0x00	

Table 4-5. Data Flash Summary—Gas Gauging Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	0	Load Select	U1	0	255	1	Number
		1	Load Mode	U1	0	255	1	Number
		21	Max Res Factor	U1	0	255	15	num
		22	Min Res Factor	U1	0	255	7	num
		24	Ra Filter	U2	0	1000	800	num
		41	Fast Qmax Start DOD %	U1	0	255	92	%
		42	Fast Qmax End DOD %	U1	0	255	96	%
		43	Fast Qm Start V Delta	I2	0	1000	125	mV
		45	Fast Qmax Current Threshold	I2	0	1000	4	C/r
		47	Fast Qmax Min Points	U1	0	65535	3	num
		49	Min % Passed Chg for Qm	U1	1	100	37	%
		53	Qmax Filter	U1	0	255	96	mAh
		54	Max % Default Qmax	U1	0	255	110	%
		55	Terminate Voltage	I2	-32768	32767	3200	mV
		57	Term V Delta	I2	0	4200	200	mV
		60	ResRelax Time	U2	0	65534	500	s
		64	User Rate-mA	I2	-32000	-100	0	mA
		66	User Rate-m/cW	I2	-32000	-350	0	mW or cW
		68	Reserve Cap-mAh	I2	0	32000	0	mAh
		70	Reserve Cap-m/cWh	I2	0	32000	0	mWh or cWh
		75	Min Delta Voltage	I2	-32000	32000	0	num
		77	Max Sim Rate	U1	0	255	1	C/rate
		78	Min Sim Rate	U1	0	255	20	C/rate
		79	Ra Max Delta	U2	0	65535	44	mΩ
		81	Qmax Max Delta %	U1	0	65535	5	%
		82	DeltaV Max dV	U2	0	65535	10	mV
		84	Max Res Scale	U2	0	32767	5000	Num
		86	Min Res Scale	U2	0	32767	200	Num
88	Fast Scale Start SOC	U1	0	100	10	%		
81	Current Thresholds	0	Dsg Current Threshold	I2	0	2000	60	mA
		2	Chg Current Threshold	I2	0	2000	75	mA
		4	Quit Current	I2	0	1000	40	mA
		6	Dsg Relax Time	U2	0	8191	60	s
		8	Chg Relax Time	U1	0	255	60	s
		9	Quit Relax Time	U1	0	63	1	s
		10	Transient Factor Charge	U1	0	255	128	num
		11	Transient Factor Discharge	U1	0	255	128	num
		12	Max IR Correct	U2	0	1000	400	mV

Table 4-5. Data Flash Summary—Gas Gauging Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	0	IT Enable	H1	0x0	0x3	0x0	num
		1	App Status	H1	0x0	0xFFFF	0x0	flags
		2	Qmax Cell 0	I2	0	32767	1000	mAh
		4	Cycle Count 0	U2	0	65535	0	Count
		6	Update Status 0	H1	0x0	0x3	0x0	num
		7	Qmax Cell 1	I2	0	32767	1000	mAh
		9	Cycle Count 1	U2	0	65535	0	Count
		11	Update Status 1	H1	0x0	0x3	0x0	num
		12	Avg I Last Run	I2	-32768	32767	-299	mA
		14	Avg P Last Run	I2	-32768	32767	-1131	cW or mW
		16	Delta Voltage	I2	-32768	32767	2	mV
		20	T Rise	U2	0	65535	20	Num
		22	T Time Constant	U2	0	65535	1000	Num
		24	Cell 0 V at Chg Term	I2	3800	4500	4200	mV
26	Cell 1 V at Chg Term	I2	3800	4500	4200	mV		

Table 4-6. Data Flash Summary—OCV Table Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
83	OCVa0 Table	0	Chem ID	H2	0x0	0xFFFF	0x100	flags
		2	Qmax Cell 0	I2	0	32767	1000	mAh
		4	Update Status	H1	0x0	0x3	0x0	num
84	OCVa1 Table	0	Chem ID	H2	0x0	0xFFFF	0x100	flags
		2	Qmax Cell 1	I2	0	32767	1000	mAh
		4	Update Status	H1	0x0	0x3	0x0	num

Table 4-7. Data Flash Summary—Default Ra Tables Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
87	Def0 Ra	0	Def0 Ra status	H1	0x0	0x0	0xFF	
		1	Def0 Ra flag	H1	0x0	0x0	0x55	
		2	Def0 Ra Base R	I2	-200	200	41	
		4	Def0 Ra Gain	H1	0x0	0x0	0x0	
		5	Def0 Ra 1	I1	-128	127	2	$2^{-10} \Omega$
		6	Def0 Ra 2	I1	-128	127	-4	$2^{-10} \Omega$
		7	Def0 Ra 3	I1	-128	127	0	$2^{-10} \Omega$
		8	Def0 Ra 4	I1	-128	127	-2	$2^{-10} \Omega$
		9	Def0 Ra 5	I1	-128	127	2	$2^{-10} \Omega$
		10	Def0 Ra 6	I1	-128	127	6	$2^{-10} \Omega$
		11	Def0 Ra 7	I1	-128	127	7	$2^{-10} \Omega$
		12	Def0 Ra 8	I1	-128	127	5	$2^{-10} \Omega$
		13	Def0 Ra 9	I1	-128	127	8	$2^{-10} \Omega$
		14	Def0 Ra 10	I1	-128	127	15	$2^{-10} \Omega$
		15	Def0 Ra 11	I1	-128	127	30	$2^{-10} \Omega$
		16	Def0 Ra 12	I1	-128	127	54	$2^{-10} \Omega$
		17	Def0 Ra 13	I1	-128	127	87	$2^{-10} \Omega$
		18	Def0 Ra 14	I1	-128	127	115	$2^{-10} \Omega$
88	Def1 Ra	0	Def1 Ra status	H1	0x0	0x0	0xFF	
		1	Def1 Ra flag	H1	0x0	0x0	0x55	
		2	Def1 Ra Base R	I2	-200	200	41	
		4	Def1 Ra Gain	H1	0x0	0x0	0x0	
		5	Def1 Ra 1	I1	-128	127	2	$2^{-10} \Omega$
		6	Def1 Ra 2	I1	-128	127	-4	$2^{-10} \Omega$
		7	Def1 Ra 3	I1	-128	127	0	$2^{-10} \Omega$
		8	Def1 Ra 4	I1	-128	127	-2	$2^{-10} \Omega$
		9	Def1 Ra 5	I1	-128	127	2	$2^{-10} \Omega$
		10	Def1 Ra 6	I1	-128	127	6	$2^{-10} \Omega$
		11	Def1 Ra 7	I1	-128	127	7	$2^{-10} \Omega$
		12	Def1 Ra 8	I1	-128	127	5	$2^{-10} \Omega$
		13	Def1 Ra 9	I1	-128	127	8	$2^{-10} \Omega$
		14	Def1 Ra 10	I1	-128	127	15	$2^{-10} \Omega$
		15	Def1 Ra 11	I1	-128	127	30	$2^{-10} \Omega$
		16	Def1 Ra 12	I1	-128	127	54	$2^{-10} \Omega$
		17	Def1 Ra 13	I1	-128	127	87	$2^{-10} \Omega$
		18	Def1 Ra 14	I1	-128	127	115	$2^{-10} \Omega$

Table 4-8. Data Flash Summary—Ra Tables Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
91	Pack0 Ra	0	Pack0 Ra status	H1	0x0	0x0	0xFF	
		1	Pack0 Ra flag	H1	0x0	0x0	0x55	
		2	Pack0 Ra Base R	I2	-200	200	41	
		4	Pack0 Ra Gain	H1	0x0	0x0	0x0	
		5	Pack0 Ra 1	I1	-128	127	2	2 ⁻¹⁰ Ω
		6	Pack0 Ra 2	I1	-128	127	-4	2 ⁻¹⁰ Ω
		7	Pack0 Ra 3	I1	-128	127	0	2 ⁻¹⁰ Ω
		8	Pack0 Ra 4	I1	-128	127	-2	2 ⁻¹⁰ Ω
		9	Pack0 Ra 5	I1	-128	127	2	2 ⁻¹⁰ Ω
		10	Pack0 Ra 6	I1	-128	127	6	2 ⁻¹⁰ Ω
		11	Pack0 Ra 7	I1	-128	127	7	2 ⁻¹⁰ Ω
		12	Pack0 Ra 8	I1	-128	127	5	2 ⁻¹⁰ Ω
		13	Pack0 Ra 9	I1	-128	127	8	2 ⁻¹⁰ Ω
		14	Pack0 Ra 10	I1	-128	127	15	2 ⁻¹⁰ Ω
		15	Pack0 Ra 11	I1	-128	127	30	2 ⁻¹⁰ Ω
		16	Pack0 Ra 12	I1	-128	127	54	2 ⁻¹⁰ Ω
		17	Pack0 Ra 13	I1	-128	127	87	2 ⁻¹⁰ Ω
		18	Pack0 Ra 14	I1	-128	127	115	2 ⁻¹⁰ Ω
92	Pack1 Ra	0	Pack1 Ra status	H1	0x0	0x0	0xFF	
		1	Pack1 Ra flag	H1	0x0	0x0	0x55	
		2	Pack1 Ra Base R	I2	-200	200	41	
		4	Pack1 Ra Gain	H1	0x0	0x0	0x0	
		5	Pack1 Ra 1	I1	-128	127	2	2 ⁻¹⁰ Ω
		6	Pack1 Ra 2	I1	-128	127	-4	2 ⁻¹⁰ Ω
		7	Pack1 Ra 3	I1	-128	127	0	2 ⁻¹⁰ Ω
		8	Pack1 Ra 4	I1	-128	127	-2	2 ⁻¹⁰ Ω
		9	Pack1 Ra 5	I1	-128	127	2	2 ⁻¹⁰ Ω
		10	Pack1 Ra 6	I1	-128	127	6	2 ⁻¹⁰ Ω
		11	Pack1 Ra 7	I1	-128	127	7	2 ⁻¹⁰ Ω
		12	Pack1 Ra 8	I1	-128	127	5	2 ⁻¹⁰ Ω
		13	Pack1 Ra 9	I1	-128	127	8	2 ⁻¹⁰ Ω
		14	Pack1 Ra 10	I1	-128	127	15	2 ⁻¹⁰ Ω
		15	Pack1 Ra 11	I1	-128	127	30	2 ⁻¹⁰ Ω
		16	Pack1 Ra 12	I1	-128	127	54	2 ⁻¹⁰ Ω
		17	Pack1 Ra 13	I1	-128	127	87	2 ⁻¹⁰ Ω
		18	Pack1 Ra 14	I1	-128	127	115	2 ⁻¹⁰ Ω

Table 4-8. Data Flash Summary—Ra Tables Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
93	Pack0 Rax	0	Pack0 Rax status	H1	0x0	0x0	0xFF	
		1	Pack0 Rax flag	H1	0x0	0x0	0xFF	
		2	Pack0 Rax Base R	I2	-200	200	41	
		4	Pack0 Rax Gain	H1	0x0	0x0	0x0	
		5	Pack0 Rax 1	I1	-128	127	2	$2^{-10} \Omega$
		6	Pack0 Rax 2	I1	-128	127	-4	$2^{-10} \Omega$
		7	Pack0 Rax 3	I1	-128	127	0	$2^{-10} \Omega$
		8	Pack0 Rax 4	I1	-128	127	-2	$2^{-10} \Omega$
		9	Pack0 Rax 5	I1	-128	127	2	$2^{-10} \Omega$
		10	Pack0 Rax 6	I1	-128	127	6	$2^{-10} \Omega$
		11	Pack0 Rax 7	I1	-128	127	7	$2^{-10} \Omega$
		12	Pack0 Rax 8	I1	-128	127	5	$2^{-10} \Omega$
		13	Pack0 Rax 9	I1	-128	127	8	$2^{-10} \Omega$
		14	Pack0 Rax 10	I1	-128	127	15	$2^{-10} \Omega$
		15	Pack0 Rax11	I1	-128	127	30	$2^{-10} \Omega$
		16	Pack0 Rax 12	I1	-128	127	54	$2^{-10} \Omega$
		17	Pack0 Rax 13	I1	-128	127	87	$2^{-10} \Omega$
		18	Pack0 Rax 14	I1	-128	127	115	$2^{-10} \Omega$
94	Pack1 Rax	0	Pack1 Rax status	H1	0x0	0x0	0xFF	
		1	Pack1 Rax flag	H1	0x0	0x0	0xFF	
		2	Pack1 Rax Base R	I2	-200	200	41	
		4	Pack1 Rax Gain	H1	0x0	0x0	0x0	
		5	Pack1 Rax 1	I1	-128	127	2	$2^{-10} \Omega$
		6	Pack1 Rax 2	I1	-128	127	-4	$2^{-10} \Omega$
		7	Pack1 Rax 3	I1	-128	127	0	$2^{-10} \Omega$
		8	Pack1 Rax 4	I1	-128	127	-2	$2^{-10} \Omega$
		9	Pack1 Rax 5	I1	-128	127	2	$2^{-10} \Omega$
		10	Pack1 Rax 6	I1	-128	127	6	$2^{-10} \Omega$
		11	Pack1 Rax 7	I1	-128	127	7	$2^{-10} \Omega$
		12	Pack1 Rax 8	I1	-128	127	5	$2^{-10} \Omega$
		13	Pack1 Rax 9	I1	-128	127	8	$2^{-10} \Omega$
		14	Pack1 Rax 10	I1	-128	127	15	$2^{-10} \Omega$
		15	Pack1 Rax 11	I1	-128	127	30	$2^{-10} \Omega$
		16	Pack1 Rax 12	I1	-128	127	54	$2^{-10} \Omega$
		17	Pack1 Rax 13	I1	-128	127	87	$2^{-10} \Omega$
		18	Pack1 Rax 14	I1	-128	127	115	$2^{-10} \Omega$

Table 4-9. Data Flash Summary—Calibration Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	0	CC Gain	F4 ⁽¹⁾	0.119	47.68	10.0 ⁽²⁾	Number
		4	CC Delta	F4 ⁽¹⁾	4.76	190	10.0 ⁽²⁾	Number
		8	CC Offset	I2	-157	157	5.76 ⁽²⁾	mA
		12	Board Offset	I1	-0.96	0.96	0	µA
		13	Int Temp Offset	I1	-128	127	0	Number
		14	Ext Temp Offset	I1	-128	127	0	Number
		15	Pack V Offset	I1	-128	127	0	Number
106	Temp Model	0	Ext a Coef 1	I2	-32768	32767	-11130	num
		2	Ext a Coef 2	I2	-32768	32767	19142	num
		4	Ext a Coef 3	I2	-32768	32767	-19262	num
		6	Ext a Coef 4	I2	-32768	32767	28203	num
		8	Ext a Coef 5	I2	-32768	32767	892	num
		10	Ext b Coef 1	I2	-32768	32767	328	num
		12	Ext b Coef 2	I2	-32768	32767	-605	num
		14	Ext b Coef 3	I2	-32768	32767	-2443	num
		16	Ext b Coef 4	I2	-32768	32767	4696	num
107	Current	1	Deadband	U1	0	255	5	mA

⁽¹⁾ Not IEEE floating point.

⁽²⁾ Displayed as the value EVSW displayed. Data Flash value is different. For CC calibration values, please follow the *Host System Calibration Method* Application Report ([SLUA640](#)).

Table 4-10. Data Flash Summary—Security Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
112	Codes	0	Sealed to Unsealed	H4	0x0	0xFFFF FFFF	0x0	
		4	Unsealed to Full	H4	0x0	0xFFFF FFFF	0x0	

Table 4-11. Data Flash to EVSW or GaugeStudio Conversion

Class	Subclass ID	Subclass	Offset	Name	Data Type	Data Flash Default	Data Flash Unit	EVSW Default	EVSW Unit	Data Flash to EVSW Conversion
Configuration	48	Data	15	Default Temperature	I2	2982	°K	250	0.1°C	DF - 2732
	64	Registers	9	DF Wr Ind Wait	U2	0	0.2 µs	0	µs	DF × 5
Calibration	104	Data	0	CC Gain	F4	0.4768	number	10	mΩ	4.768 ÷ DF
			4	CC Delta	F4	567744.6	number	10	mΩ	5677445 ÷ DF
			8	CC Offset	I2	-1200	number	-5.76	mV	DF × 0.0048
			12	Board Offset	I1	0	number	0	µV	DF × 0.0075

4.6 Data Flash Parameter Update Example

This section shows an example of the command sequence that modifies a data flash parameter while device firmware is still running. It can update one or more parameters without going to ROM mode and loading a new data flash image (.dfi, .dmi, or .dffs file).

For this example, the **OpConfig B [WRTEMP]** bit of the fuel gauge is changed from 0 to 1.

Some bq27520-G4 pins are configured via the **OpConfig B** register. This register is programmed and read via the methods described in [Section 4.1, Accessing The Data Flash](#). See [Section 4.5, Data Flash Summary](#), for the location (subclass and offset) of these configuration registers.

Table 4-12. OpConfig B Register Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	WRTEMP	BIE	BL_INT	GNDSEL	FCE	DFWrIndBL	RFACTSTEP	RSVD
Default	0	1	0	0	1	0	1	0
0x4A								

WRTEMP = Enables the temperature write. The temperature is expected to be written by the host and is used for gauging. Neither the external thermistor or internal temperature sensor is used. True when set.

Note that subclass ID and offset values in [Table 4-13](#) are in decimal format. The example below has converted these to hexadecimal. For example, the **OpConfig B** subclass is d64 = 0x40.

Table 4-13. Data Flash Summary—Configuration

Subclass ID	Subclass	Offset	Name	Data Type	Min Value	Max Value	Default Value	Unit
64	Registers	11	OpConfig B	H1	0x00	0xFF	0x4A	hex

4.6.1 Modify WRTEMP of OpConfig B Register

- Unseal the device by using the *Control()* (0x00 and 0x01) command if the device is SEALED.
 - Write the first 2 bytes of the UNSEAL key using the *Control(0x0414)* command.
(wr 0x00 0x14 0x04)
 - Write the second 2 bytes of the UNSEAL key using the *Control(0x3672)* command.
(wr 0x00 0x72 0x36)
- Write 0x00 using *BlockDataControl()* command (0x61) to enable block data flash control.
(wr 0x61 0x00)
- Write 0x40 (**OpConfig B** Subclass) using the *DataFlashClass()* command (0x3E) to access the registers subclass.
(wr 0x3E 0x40)
- Write the block offset location using *DataFlashClass()* command (0x3F). To access data located at offset 0 to 31 use offset = 0x00. To access data located at offset 32 to 41 use offset = 0x01. For example, **OpConfig B** (offset = 11) is in the first block so use:
(wr 0x3F 0x00)
- To read the data of a specific offset use address 0x40 + mod(offset, 32). For example **OpConfig B** (offset = 11) is located at 0x4B, read 1 byte starting at 0x4B address.
(rd 0x4B old_OP_CONF_B_BYTE)
In our example, assume WRTEMP(MSB) is cleared.
- To read the 1-byte checksum use the *BlockDataChecksum()* command (0x60).
(rd 0x60 OLD_checksum)
- In this example, set WRTEMP by setting the most-significant bit of OP_CONF_B_BYTE.
- The new value for OP_CONF_B_BYTE can be written by writing to the specific offset location. For example to write 1-byte OP_CONF_B_BYTE new value with MSB set to **OpConfig B** (offset = 11) located at 0x4B, use command: (wr 0x4B new_OP_CONF_B_BYTE)
- The data is actually transferred to the data flash when the correct checksum for the whole block (0x40 to 0x5F) is written to *BlockDataChecksum()* (0x60).
(wr 0x60 NEW_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 is to make use of the old checksum:
 - temp = mod(255 – OLD_checksum – old_OP_CONF_B_BYTE, 256)
 - NEW_checksum = 255 – mod(temp + new_OP_CONF_B_BYTE, 256)

10. RESET the gauge to ensure the new data flash parameter goes into effect by using *Control(0x0041)*.
(wr 0x00 0x41 0x00)
If previously sealed, then the gauge automatically becomes sealed again after RESET.
11. If not previously sealed, then SEAL the gauge by using *Control(0x0020)*.
(wr 0x00 0x20 0x00)

Functional Description

5.1 Impedance Track™ Variables

The bq27520-G4 fuel gauge has a number of data flash variables that permit the user to customize the Impedance Track™ algorithm for optimal performance. These variables are dependent upon the power characteristics of the application as well as the cell itself.

5.1.1 Load Mode

Load Mode selects either the constant-current or constant-power model for the Impedance Track™ algorithm as used in **Load Select** (see [Section 5.1.2](#)). When **Load Mode** is 0, the constant-current model is used (default). When **Load Mode** is 1, the constant-power model is used. The **CONTROL_STATUS [LDMD]** bit shows the status of **Load Mode**.

5.1.2 Load Select

Load Select defines the type of power or current model that computes the load-compensated capacity in the Impedance Track™ algorithm.

If **Load Mode** = 0 (constant-current model), then the options presented in [Table 5-1](#) are available.

Table 5-1. Constant-Current Model Used When Load Mode = 0

Load Select Value	Current Model Used
0	The average discharge current from previous cycle, Avg I Last Run . (See Section 5.1.10)
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 on <i>AverageCurrent()</i>
3	Current: based off of a low-pass-filtered version of <i>AverageCurrent()</i> ($\tau = 14$ s)
4	Design capacity / 5: C Rate based off of Design Capacity /5 or a C/5 rate in mA.
5	AtRate (mA): Use whatever current is in <i>AtRate()</i>
6	User_Rate-mA: Use the value in <i>User_Rate-mA</i> . This mode provides a completely user-configurable method.

If **Load Mode** = 1 (constant-power model), then the options shown in [Table 5-2](#) are available.

Table 5-2. Constant-Power Model Used When Load Mode = 1

Load Select Value	Power Model Used
0	The average discharge power from previous cycle, Avg P Last Run . (See Section 5.1.11)
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 x voltage: based off the <i>AverageCurrent()</i> and <i>Voltage()</i>
3	Current x voltage: based off of a low-pass-filtered version of <i>AverageCurrent()</i> ($\tau = 14$ s) and <i>Voltage()</i>
4	Design energy / 5: C Rate based off of Design Energy /5 or a C/5 rate in mA.
5	AtRate (10 mW): Use the value is in <i>AtRate()</i> .
6	User_Rate–10mW: Use the value in <i>User_Rate–10mW</i> . This mode provides a completely user-configurable method.

5.1.3 Reserve Cap-mAh, Reserve Cap-mWh/cWh

Reserve Cap-mAh (*Load Mode* = 0) or **Reserve Cap-mWh/cWh** (*Load Mode* = 1) determines how much actual remaining capacity exists when the fuel gauge reports zero for *RemainingCapacity()* before reaching the **Terminate Voltage**. This accommodates a controlled shutdown scheme based on battery capacity rather than a specific voltage. A loaded rate or no-load rate of compensation can be selected for **Reserve Cap-mAh** via the **Op Config [RESCAP]** bit.

5.1.4 Design Energy Scale

Design energy scaling accommodates large capacity battery packs greater than approximately 6000 mAh. **Des Energy Scale** selects the scale and unit of a set of data flash parameters. The value of **Des Energy Scale** can be either 1 or 10, only. For batteries less than 6000 mAh, a setting of 1 is recommended. For batteries greater than 6000 mAh, a setting of 10 is recommended.

Table 5-3. Data Flash Parameter Scale and Unit Based on Design Energy Scale

Data Flash	Design Energy Scale = 1 (default)	Design Energy Scale = 10
<i>Design Energy</i>	mWh	cWh
<i>Reserve Capacity-mWh/cWh</i>	mWh	cWh
<i>Avg Power Last Run</i>	mW	cW
<i>User Rate-mW/cW</i>	mWh	cWh
<i>T Rise</i>	No Scale (example: default is 20)	Scaled by x10 (example: default is 200)

5.1.5 Dsg Current Threshold

This register is used as a threshold by many functions in the fuel gauge to determine if significant discharge current is flowing into or out of the cell. The default for this register is in [Table 4-5](#), Current Thresholds Subclass, which should be sufficient for most applications. This threshold should be set low enough to be below any normal application load current but high enough to prevent noise or drift from affecting the measurement.

5.1.6 Chg Current Threshold

This register is used as a threshold by many functions in the fuel gauge to determine if significant charge current is flowing into or out of the cell. The default for this register is in [Table 4-5](#), Current Thresholds Subclass, which should be sufficient for most applications. This threshold should be set low enough to be below any normal charge current but high enough to prevent noise or drift from affecting the measurement.

5.1.7 Quit Current, Dsg Relax Time, Chg Relax Time, and Quit Relax Time

The **Quit Current** is used as part of the Impedance Track™ algorithm to determine when the fuel gauge enters the relaxation mode from a current-flowing mode in either the charge direction or the discharge direction. The value of **Quit Current** is set to a default value in [Table 4-5](#), Current Thresholds Subclass, and should be above the standby current of the system.

Either of the following criteria must be met to enter the relaxation mode:

- $|AverageCurrent()| < |Quit Current|$ for **Dsg Relax Time**
- $|AverageCurrent()| < |Quit Current|$ for **Chg Relax Time**

After about 5 minutes in relaxation mode, the fuel gauge attempts to take accurate OCV readings. An additional requirement of $dV/dt < 1 \mu V/s$ is required for the fuel gauge to perform Qmax updates. These updates are used in the Impedance Track™ algorithms. It is critical that the battery voltage be relaxed during OCV readings and that the current is not higher than C/20 when attempting to go into relaxation mode.

Quit Relax Time specifies the minimum time required for *AverageCurrent()* to remain above the **Dsg Current Threshold** or **Chg Current Threshold** before exiting the relaxation mode. See Application Report [SLUUA450](#), *Theory and Implementation of Impedance Track Battery Fuel-Gauging Algorithm*, for more information.

5.1.8 Qmax Cell 0 and Qmax Cell 1

Generically called **Qmax**, these dynamic variables contain the respective maximum chemical capacity of the active cell profiles, and are determined by comparing states of charge before and after applying the load with the amount of charge passed. They also correspond to capacity at a low rate of discharge, such as the C/20 rate. For high accuracy, this value is periodically updated by the fuel gauge during operation. Based on the battery cell capacity information, the initial value of chemical capacity should be entered in the **Qmax n** field for each default cell profile. The Impedance Track™ algorithm updates these values and maintains them in the associated cell profiles.

Qmax Max Delta % is the percent of *DesignCapacity()* to limit how much Qmax may grow or shrink during any one Qmax update. The default is 5%.

Min % Passed Chg for Qm represents the approximate change in SOC that is required as part of the qualification for Qmax updates. It is not recommended to change this value.

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

Although there is variation between batteries, in general Qmax should only decrease over time and use. **Max % Default Qmax** prevents erroneous updates from being recorded if they are too high.

5.1.9 Update Status 0 and Update Status 1

The **Update Status n** registers are modified automatically by the fuel gauge and the TI evaluation software during the process of creating a golden Data Flash file. A golden file with optimized Qmax and resistance values should have the **Update Status = 2**.

5.1.10 Avg I Last Run

The fuel gauge computes average current from the beginning to the end of each discharge cycle and stores the average current from the previous discharge period in this register if the duration is > 500 seconds. This register should not be modified by the host as it is automatically updated by the fuel gauge when required. For a golden file, this register should be initially set to a typical system discharge current level.

5.1.11 Avg P Last Run

The fuel gauge computes average power from the beginning to the end of each discharge cycle and stores this average power from the previous discharge period in this register if the duration is > 500 seconds. Average Power is computed by continuously averaging the product of *InstantaneousCurrent()* and *Voltage()*. This register should not be modified by the host as it is automatically updated by the fuel gauge when required. For a golden file, this register should be initially set to a typical system discharge power level.

5.1.12 Delta Voltage

The fuel gauge stores the maximum difference of *Voltage()* during short load spikes and normal load, so the Impedance Track™ algorithm can calculate remaining capacity for pulsed loads. It is not recommended to change this value.

Min DeltaV is the minimum **Delta Voltage** that is saved during discharge cycles. The default is 0 mV.

DeltaV Max dV limits on how far **Delta Voltage** grows or shrinks on one grid update (in mV). This register defaults to 10.

5.1.13 Default Ra and Ra Tables

These tables contain encoded data and, with the exception of the **Default Ra Tables**, are automatically updated during device operation. No user changes should be made except for reading or writing the values from a pre-learned pack (part of the process for creating golden image files).

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. This value should be set to 15% of the Ra[4] value. Value needs to be manually adjusted after chemistry change.

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

5.1.14 Fast Resistance Scaling

Fast resistance scaling improves convergency of remaining capacity and terminates the voltage at end of discharge. The feature is enabled via the **OpConfig B [FCE]** bit and operates when cell voltage is below (**Terminate Voltage + Term V Delta**) or **StateofCharge()** is less than **Fast Scale Start SOC**. For most applications, the default values of **Term V Delta** and **Fast Scale Start SOC** are recommended. It is also recommended to keep (**Terminate Voltage + Term V Delta**) below 3.6 V for most battery applications.

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

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

5.1.15 Fast Qmax Update

Fast Qmax provides a method to compute Qmax based on full charge and end-of-discharge conditions without requiring battery relaxation. The feature is enabled via the **OpConfig E [DSGFASTQM, CHGFASTQM]** bits. Several data flash parameters (**Fast Qmax Start DOD%**, **Fast Qmax End DOD%**, **Fast Qm Start V Delta**, **Fast Qmax Current Threshold**, **Fast Qmax Min Points**, and **Term V Delta**) configure the algorithm; default settings are recommended.

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

For traditional Qmax learning, two DOD points must be captured by the gauge during cell relaxation. These DOD points must be separated by at least 37% DOD, and neither can be taken in the flat voltage region or at extreme temperatures. By using the Fast Qmax feature, either or both relaxed DOD points can be replaced by a Fast Qmax DOD point. Although Qmax learning does not need to occur frequently, the Fast Qmax Update is useful for systems where a full relaxation of the battery is rare.

If the CHGFASTQM is enabled, a DOD point is captured in RAM at the end of a full charge termination (when the FC bit is set). This DOD point can be qualified for a Qmax update when the next discharge begins, if a traditional relaxed DOD update did not occur.

If the DSGFASTQM is enabled, a DOD point can be captured near the end of discharge to empty. There are more qualification requirements for this DOD point. As the discharge approaches empty, the algorithm will start to try qualifying Fast Qmax DOD samples. It will begin looking for samples every 30 seconds when the following conditions are met:

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

When the discharge stops, the Fast Qmax DOD point will be qualified if the following conditions are met:

- Number of Fast Qmax measurements > **Fast Qmax Min Points**
- DOD > **Fast Qmax End DOD%**, or
Voltage < (**Terminate Voltage + Fast Qmax Volt Buffer**)

If the discharge is deep enough, and the previous requirements are met, a DOD point that can be used for a Qmax update is qualified.

5.1.16 SOC Smoothing

Rapid changes in operating conditions, such as temperature or discharge current, can lead to sudden changes in the algorithm's immediate calculation of *RemainingCapacity()*, *FullChargeCapacity()*, and *StateOfCharge()*. SOC smoothing provides filtered data to the host resulting in more gradual changes to SOC-related data when conditions vary and can provide a better end-user experience. The feature is enabled via the **OpConfig D [SMTHEN]** bit and has one configuration option available via the **OpConfig D [RCJUMPOK]** bit.

Both smoothed and unsmoothed registers are available at the higher register addresses, but the **OpConfig D [SMTHEN]** bit determines which values get reported in the *RemainingCapacity()*, *FullChargeCapacity()*, and *StateOfCharge()* registers.

5.1.17 Flash Updates

Data flash can only be updated if $Voltage() \geq \text{Flash Update OK Voltage}$. Flash programming current can cause an increase in LDO dropout. The value of **Flash Update OK Voltage** must be selected such that the V_{CC} voltage does not fall below its minimum of 2.4 V during flash write operations. Data flash updates can occur at any time during gauge operation. During data flash updates, the gauge may stretch the I²C clock significantly. See [Section 6.4, I²C Clock Stretching](#), for more information.

The SOC_INT pin can be configured to generate a pulse before and during data flash updates if desired. This is disabled by default. See [Section 5.3.4, SOC_INT Pin Behavior](#), for details.

5.2 Device Configuration

The configuration options are configured via the following **Operation Configuration** data flash registers. These registers are programmed and read via the methods described in [Section 4.1, Accessing the Data Flash](#). See [Table 4-3, Registers Subclass](#), for the location (subclass and offset) of these configuration registers. A faster way to read the current value of the **Operation Configuration** register is to use the `OperationConfiguration()` function.

5.2.1 Operation Configuration (Op Config) Register

Table 5-4. Op Config Register Bit Definition

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	RESCAP	BATG_OVR	INT_BREM	PFC_CFG1	PFC_CFG0	IWAKE	RSNS1	RSNS0
Default	0	0	0	0	1	0	0	1
0x09								
Low Byte	INT_FOCV	IDSELEN	SLEEP	RMFCC	SOCI_POL	BATG_POL	BATL_POL	TEMPS
Default	0	1	1	1	0	0	1	1
0x73								

High Byte (0x09)

- RESCAP = Selects the rate of compensation method for the reserve capacity calculation. If clear (default), a loaded rate is used. If set, a no-load rate is used. (See [Section 5.1.3, Reserve Cap-mAh, Reserve Cap-mWh/cWh](#))
- BATG_OVR = BAT_GD override bit. If the gauge enters HIBERNATE only due to the cell voltage, the BAT_GD pin does not negate. This option may be useful if the BAT_GD pin is interfaced with a charger IC. True when set. (See [Section 5.3.5, Power Path Control With the BAT_GD Pin](#))
- INT_BERM = Battery removal interrupt bit. The SOC_INT pin pulses 1 ms when the battery removal interrupt is enabled. True when set. (See [Table 5-10, SOC_INT Pulse Conditions and Widths](#))
- PFC_CFG1, PFC_CFG0 = Pin function code (PFC) mode selection: PFC 0, 1, 2, or 3 selected by 00, 01, 10, or 11, respectively (see [Section 5.3.1, Pin Function Code \(PFC\) Descriptions](#)).
- IWAKE, RSNS1, RSNS0 = These bits configure the current wake function (see [Section 5.3.6, Wake-Up Comparator](#)).

Low Byte (0x73)

- INT_FOCV = Indication of the measurement of the OCV during the initialization. The SOC_INT pin pulses during the first measurement if this bit is set. True when set. This option may be useful if the SOC_INT pulse can trigger the system to enter a low-power state for the best possible OCV measurement. (See [Table 5-10, SOC_INT Pulse Conditions and Widths](#))
- IDSELEN = Enables cell profile identification feature. True when set. (See [Section 5.7.1, Battery Profile Storage and Selection](#))
- 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. (See [Section 5.5.1, Detecting Charge Termination](#))
- SOCI_POL = SOC_INT pin polarity control. Active-low is 0. Active-high is 1.
- BATG_POL = BAT_GD pin polarity control. Active-low is 0. Active-high is 1.
- BATL_POL = BAT_LOW pin polarity control. Active-low is 0. Active-high is 1.
- TEMPS = Selects external thermistor for `Temperature()` measurements. True when set. (See [Section 5.3.1, Pin Function Code \(PFC\) Descriptions](#), and [Section 2.4, Temperature\(\): 0x06 and 0x07](#))

5.2.2 Operation Configuration B (OpConfig B) Register

Table 5-5. OpConfig B Register Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	WRTEMP	BIE	BL_INT	GNDSEL	FCE	DFWrlndBL	RFACTSTEP	RSVD
Default	0	1	0	0	1	0	1	0
0x4A								

WRTEMP = Enables the temperature write. The temperature is expected to be written by the host and is used for gauging. Neither the external thermistor or internal temperature sensor is used. True when set. (See [Section 5.3.1, Pin Function Code \(PFC\) Descriptions](#), and [Section 2.4, Temperature\(\)](#): 0x06 and 0x07)

BIE = Battery Insertion Detection feature enable. When enabled, the gauge detects battery insertion using the TS pin. If disabled, the gauge relies on the host to set and clear the *Flags()* [BAT_DET] bit using BAT_INSERT or BAT_REMOVE subcommands. True when set. (See [Section 5.3.3, Battery Presence Detection Using the BI/TOUT Pin](#), and [Section 5.6.1, BAT INSERT CHECK Mode](#))

BL_INT = Enables toggle of SOC_INT pin upon the state change of *Flags()* [SOC1] in addition to the BAT_LOW pin's discrete output of the battery low condition. True when set. (See [Table 5-10, SOC_INT Pulse Conditions and Widths](#))

GNDSEL = The ADC ground select control. The V_{SS} (pin D1) is selected as ground reference when the bit is clear. Pin A1 is selected when the bit is set. The default value is recommend for typical applications.

FCE = Fast Convergence Enable for Resistance Scaling. Configures algorithm to use fast convergence method. The default value is recommend for typical applications. (See [Section 5.1.14, Fast Resistance Scaling](#))

DFWrlndBL = Data Flash Write Indication. SOC_INT pin is used for indication if the bit is clear. BAT_LOW is used for indication if the bit is set.

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

RSVD = Bit 0 is reserved.

5.2.3 Operation Configuration C (OpConfig C) Register

Table 5-6. OpConfig C Register Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	BATGSPUEN	RSVD	BATLSPUEN	RSVD	VconsEn	SlpWkChg	DeltaVOpt1	DeltaVOpt0
Default	0	0	1	0	1	1	0	0
0x2C								

BATGSPUEN = Enables internal pull-up resistor to V_{CC} (2.5 V) on BAT_GD pin. True when set.

RSVD = Bit 6 is reserved.

BATLSPUEN = Enables internal pull-up resistor to V_{CC} (2.5 V) on BAT_LOW pin. True when set.

RSVD = Bit 4 is reserved.

VconsEn = Enables Voltage Consistency checking function. Use default value for proper operation.

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

DeltaVOpt[1:0] = Configures options for determination of **Delta Voltage** which is defined as the maximum difference in *Voltage()* during normal load and short load spikes. **Delta Voltage** is used as a compensation factor for calculating *RemainingCapacity()* under pulsed loads.

00 = Standard DeltaV. Average variance from steady state voltage used to determine end-of-discharge voltage. (Default)

01 = No Averaging. The last instantaneous change in *Voltage()* from steady state determines the end-of-discharge voltage.

10 = Use the value in **Min Delta Voltage**.

11 = Not used.

5.2.4 Operation Configuration D (OpConfig D) Register

Table 5-7. OpConfig D Register Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	RCJUMPOK	SMTHEN	SOC_STATE	SOC_OCV	SOC_DFW	SOC_OT	CHGDODEOC	RSVD
Default	0	1	0	1	1	1	1	0
0x5E								

- RCJUMPOK = Allows SOC to change due to a temperature change during relaxation when the SOC smoothing algorithm is enabled. True when set. (See [Section 5.1.16, SOC Smoothing](#))
- SMTHEN = Enables SOC smoothing function. (See [Section 5.1.16, SOC Smoothing](#))
- SOC_STATE = Enables SOC_INT pin function to generate a pulse due to an Impedance Track™ algorithm state change. (See [Table 5-10, SOC_INT Pulse Conditions and Widths](#))
- SOC_OCV = Enables SOC_INT pin function to generate a pulse due to OCV command. (See [Table 5-10, SOC_INT Pulse Conditions and Widths](#))
- SOC_DFW = Enables SOC_INT pin function to generate a pulse due to data flash write. (See [Table 5-10, SOC_INT Pulse Conditions and Widths](#))
- SOC_OT = Enables SOC_INT pin function to generate a pulse due to overtemperature conditions in conjunction with the assertion of *Flags() [OTC or OTD]*. (See [Table 5-10, SOC_INT Pulse Conditions and Widths](#))
- CHGDODEOC = Enables DoD at End-of-Charge recalculation during charging only. True when set. The default setting is recommended.
- RSVD = Bit 0 is reserved.

5.2.5 Operation Configuration E (OpConfig E) Register

Table 5-8. OpConfig E Register Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	RSVD	RSVD	RSVD	RSVD	DSGFASTQM	CHGFASTQM	RSVD	RSVD
Default	0	0	0	0	0	0	0	0
0x00								

- RSVD = Bits 7, 6, 5, and 4 are reserved.
- DSGFASTQM / CHGFASTQM = Enables end of discharge (DSG) / end of charge (CHG) related Fast Qmax function. See [Section 5.1.15, Fast Qmax Update](#), for additional details. Use the defaults for most applications.
- RSVD = Bits 1 and 0 are reserved.

5.3 External Pin Functions

5.3.1 Pin Function Code (PFC) Descriptions

This fuel gauge has several pin-function configurations available for the end application. Each configuration is assigned a pin function code, or PFC, specified by the **Op Config [PFC_CFG1, PFC_CFG0]** bits (see [Table 5-9](#)). If the fuel gauge is configured to measure external temperature via the **Op Config [TEMPS]** bit, a voltage bias of approximately 125 ms is applied periodically to the external thermistor network in order to make a temperature measurement.

Table 5-9. Pin Function Code Summary

PFC	PFC_CFG [1:0]	External Thermistor Bias Rate ([TEMPS] = 1 only)			BAT_GD Pin Usage for PFC	Pin Function Description
		Discharge	Charge	Sleep		
0	00	1 / s	1 / s	1 / 20 s	NA	A dedicated external thermistor is used for the fuel gauge to monitor battery temperature in all conditions. The BAT_GD pin is not used to interface with a charger IC.
1	01				Temperature-based Charge Inhibit	A dedicated external thermistor is used for the fuel gauge to monitor battery temperature in all conditions. If battery charging temperature falls outside of the preset range defined in data flash, a charger can be disabled via the BAT_GD pin until cell temperature recovers. See Section 5.5.2, Charge Inhibit and Suspend , for additional details.
2	10		None		NA	A shared external thermistor is supported between the fuel gauge and a charger IC; however, the BAT_GD pin is not used to interface with the charger IC. The fuel gauge biases the thermistor for battery temperature measurement and BAT_INSERT_CHECK mode (if OpConfig B [BIE] bit = 1) under discharge and relaxation conditions only so the charger IC can separately bias the thermistor during charge mode. Bias networks required by the fuel gauge and the charger for the thermistor must be identical.
3	11		1 / s		Follows <i>Flags() [FC]</i> flags bit	Disables a battery charger IC when fuel gauge has determined the battery is fully charged. The BAT_GD pin reflects the logical status of the <i>Flags() [FC]</i> bit and is typically connected directly to the charger Charge Enable/Disable (CE/CD) pin or via a network to drive the charger Temperature Sense (TS) pin.

5.3.2 BAT_LOW Pin

The BAT_LOW pin provides a system processor with an external indicator of battery status. The signaling on the BAT_LOW pin follows the status of the *Flags() [SOC1]* bit. The BAT_LOW pin polarity is configured via the **Op Config [BATL_POL]** bit. The internal pull-up to V_{CC} (2.5 V) is enabled via the **OpConfig C [BATGSPUEN]** bit.

5.3.3 Battery Presence Detection Using the BI/TOUT Pin

During power-up or hibernate activities, or any other activity where the fuel gauge needs to determine whether or not a battery is connected, the fuel gauge applies a test for battery presence when the **OpConfig B [BIE]** bit is set. First, the BI/TOUT pin is put into high-Z status. The weak 1.8-MΩ pull-up resistor keeps the pin high while no battery is present. When a battery is inserted (or is already inserted) into the system device, the BI/TOUT pin is pulled low. This state is detected by the fuel gauge, which polls this pin every second when the gauge has power. A battery-disconnected status is assumed when the fuel gauge reads a thermistor voltage that is near 2.5 V.

When a thermistor is not used by the system for the gauge to detect battery insertion, there are two options. First, the BI/TOUT pin can be tied to V_{SS} with a resistor so the gauge always considers a battery to be present if it has power. Second, the **OpConfig B [BIE]** bit can be cleared so host can inform the gauge of the battery status via the *BAT_INSERT* and *BAT_REMOVE* subcommands.

5.3.4 SOC_INT Pin Behavior

The SOC_INT pin generates a pulse of different pulse widths under various conditions as indicated by [Table 5-10](#). After initialization, only one SOC_INT pulse is generated within any given one-second time slot and, therefore, may indicate multiple event conditions.

Table 5-10. SOC_INT Pulse Conditions and Widths

Pulse Condition	Enable Condition	Pulse Width	Comment
Change of <i>StateOfCharge()</i>	(<i>SOC Delta</i>) ≠ 0	1 ms	During charge, when the SOC is greater than (>) the points: 100% – n × (<i>SOC Delta</i>) and 100%; During discharge, when the SOC reaches (≤) the points: 100% – n × (<i>SOC Delta</i>) and 0%; where n is an integer starting from 0 to the number generating SOC no less than 0% Examples: For <i>SOC Delta</i> = 1% (default), the SOC_INT intervals are 0%, 1%, 2%, ..., 99%, and 100%. For <i>SOC Delta</i> = 10%, the SOC_INT intervals are 0%, 10%, 20%, ..., 90%, and 100%.
Change of <i>Flags() [SOC1]</i> state (Set or Clear)	<i>OpConfig B [BL_INT]</i> = 1	1 ms	When SOC reached the <i>SOC1 Set</i> or <i>Clear Threshold</i> set in the Data Flash.
Change of <i>Flags() [SYSDOWN]</i> state (Set or Clear)	Always	1 ms	When the <i>Voltage()</i> has reached <i>SysDown Set Volt Threshold</i> or <i>SysDown Clear Volt</i> threshold.
Battery State Change	(<i>SOC Delta</i>) ≠ 0 and <i>OpConfig D [SOC_STATE]</i> = 1	1 ms	Upon detection of a state change in battery charging and discharging. Relaxation is not included.
Battery Removal	<i>OpConfig B [BIE]</i> = 1 and <i>Op Config [INT_BREM]</i> = 1	1 ms	
OCV measurement after initialization	<i>Op Config [INT_FOCV]</i> = 1	Approximately 380 ms	Within 1.5 seconds after a POR event or the receipt of either <i>BAT_INSERT</i> or <i>RESET</i> subcommand, SOC_INT begins a pulse for the duration of the OCV measurement and initialization time period.
OCV measurement from <i>OCV_CMD</i> subcommand	<i>OpConfig D [SOC_OCV]</i> = 1	Approximately 260 ms	Within 1 second after receipt of <i>OCV_CMD</i> subcommand, SOC_INT begins a pulse for the duration of the OCV measurement execution time period.
After initialization and <i>DF Wr Ind Wait</i> ≠ 0	<i>OpConfig D [SOC_DFW]</i> = 1	Programmable (see comment)	SOC_INT pin indicates the data flash update. The gauge waits <i>DF Wr Ind Wait</i> × 5 μs after the SOC_INT signal to start the data flash update. This function is disabled if <i>DF Wr Ind Wait</i> = 0.
<i>Flags() [OTC]</i> or <i>[OTD]</i>	<i>OpConfig D [SOC_OT]</i> = 1	1 ms	Upon first assertion of <i>Flags() [OTC]</i> or <i>[OTD]</i> overtemperature conditions.

5.3.5 Power Path Control With the BAT_GD Pin

The fuel gauge must operate in conjunction with other electronics in a system appliance, such as chargers or other ICs and application circuits that draw appreciable power. After a battery is inserted into the system, it is preferable that no charging current or discharging current higher than C/20 is present, so that an accurate OCV can be read. The OCV helps determine which battery profile to use, as it constitutes part of the battery impedance measurement and determines initial SOC. To disable these functions, the BAT_GD pin can be connected to the Charger Enable/Disable (CE/CD) pin to disable the charging function. Once an OCV reading has been made, the BAT_GD pin is asserted, thereby enabling battery charging and regular discharge of the battery. The *Op Config [BATG_POL]* bit can change the polarity of the BAT_GD pin in case the default configuration needs to be changed for the system application.

[Figure 5-1](#) and [Figure 5-2](#) detail how the BAT_GD pin functions in the context of battery insertion and removal, as well as NORMAL versus SLEEP modes.

In PFC 1, the BAT_GD pin also disables battery charging when the fuel gauge reads battery temperatures outside the range defined by [*Charge Inhibit Temp Low, Charge Inhibit Temp High*]. The BAT_GD line is asserted once temperature falls within the range [*Charge Inhibit Temp Low + Temp Hys, Charge Inhibit Temp High – Temp Hys*].

5.3.6 Wake-Up Comparator

The wake-up comparator indicates a change in cell current while the fuel gauge is in SLEEP mode. The **Op Config [RSNS1:RSNS0]** bits select the appropriate comparator threshold for the sense resistor value used. The **Op Config [IWAKE]** bit selects one of two possible voltage threshold ranges for the given sense resistor selection. An internal interrupt is generated when the threshold is reached in either the charge or discharge direction. Setting both **[RSNS1]** and **[RSNS0]** bits to 0 disables this feature.

Table 5-11. I_{WAKE} Threshold Settings⁽¹⁾

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

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

5.3.7 Autocalibration

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

Autocalibration of the coulomb counter begins on entry to SLEEP mode, except if *Temperature()* is $\leq 5^{\circ}\text{C}$ or *Temperature()* $\geq 45^{\circ}\text{C}$.

The fuel gauge also performs a single offset when:

- The condition of *AverageCurrent()* ≤ 100 mA
- {voltage change since last offset calibration ≥ 256 mV} or {temperature change since last offset calibration is greater than 8°C for ≥ 60 s}.

Capacity and current measurements continue at the last measured rate during the offset calibration when these measurements cannot be performed. If the battery voltage drops more than 32 mV during the offset calibration, the load current has likely increased; hence, the offset calibration is aborted. The *CONTROL_STATUS [CCA]* bit is set during coulomb counter autocalibration.

5.4 Temperature Measurement

The fuel gauge typically measures battery temperature via its TS input to supply battery temperature status information to the fuel gauging algorithm and charger-control sections of the gauge. Alternatively, it can be configured to use an internal on-chip temperature sensor or receive temperature data from the host processor. See [Section 2.4, *Temperature\(\)*: 0x06 and 0x07](#), for specific information on configuration options. Regardless of which temperature configuration is used, the host processor can request the current battery temperature by reading the *Temperature()*, and for internal temperature, *InternalTemperature()*.

The external thermistor circuit requires the use of an 10K NTC 103AT-type thermistor. Additional circuit information for connecting this thermistor to the fuel gauge is shown in [Chapter 7, *Reference Schematic*](#).

5.4.1 Overtemperature Indication

5.4.1.1 Overtemperature: Charge

If during charging, *Temperature()* reaches the threshold of **OT Chg** for a period of **OT Chg Time**, and *AverageCurrent()* > **Chg Current Threshold**, then the *Flags()* [OTC] bit is set. When *Temperature()* falls to **OT Chg Recovery**, the *Flags()* [OTC] bit is cleared.

If **OT Chg Time** = 0, then the feature is completely disabled.

5.4.1.2 Overtemperature: Discharge

If during discharging, *Temperature()* reaches the threshold of **OT Dsg** for a period of **OT Dsg Time**, and *AverageCurrent()* ≤ **-Dsg Current Threshold**, then the *Flags()* [OTD] bit is set. When *Temperature()* falls to **OT Dsg Recovery**, the *Flags()* [OTD] bit is cleared.

If **OT Dsg Time** = 0, then the feature is completely disabled.

5.5 Charging and Charge—Termination Indication

5.5.1 Detecting Charge Termination

For proper fuel gauge operation, the cell **Charging Voltage** must be specified by the user. The default value is specified in [Table 4-3, Data Flash Summary—Configuration Class](#).

The fuel gauge detects charge termination when:

- During two consecutive periods of 40 seconds, the $AverageCurrent() < Taper\ Current$.
- During the same two periods, the accumulated change in capacity must be > 0 .
- $Voltage() > Charging\ Voltage - Taper\ Voltage$.

When this occurs, the $Flags() [CHG]$ bit is cleared and the $Flags() [FC]$ bit is set. Also, if the **Op Config [RMFCC]** bit is set, then $RemainingCapacity()$ is set equal to $FullChargeCapacity()$.

5.5.2 Charge Inhibit and Suspend

The fuel gauge can indicate when battery temperature has fallen below or risen above predefined thresholds **Charge Inhibit Temp Low** or **Charge Inhibit Temp High**, respectively. In this mode, the $Flags() [CHG_INH]$ bit is set and the BAT_GD pin is deasserted to indicate this condition. The $[CHG_INH]$ bit is cleared and the BAT_GD pin is asserted once the battery temperature returns to the range **[Charge Inhibit Temp Low + Temp Hys, Charge Inhibit Temp High – Temp Hys]**.

When PFC = 1, the fuel gauge indicates when battery temperature has fallen below or risen above predefined thresholds **Suspend Low Temp** or **Suspend High Temp**, respectively. In this mode, the $Flags() [XCHG]$ bit is set to indicate this condition. The $[XCHG]$ bit is cleared once the battery temperature returns to the range **[Charge Inhibit Temp Low + Temp Hys, Charge Inhibit Temp High – Temp Hys]**.

The charging should not start when the temperature is below the **Charge Inhibit Temp Low** or above the **Charge Inhibit Temp High**. The charging can continue if the charging starts inside the window **[Charge Inhibit Temp Low, Charge Inhibit Temp High]** until the temperature is either below **Suspend Low Temp** or above the **Suspend High Temp**. Therefore, the window **[Charge Inhibit Temp Low, Charge Inhibit Temp High]** must be inside the window of **[Suspend Low Temp, Suspend High Temp]**.

5.6 Power Modes

The fuel gauge has different power modes: BAT INSERT CHECK, NORMAL, SNOOZE, SLEEP, and HIBERNATE. In NORMAL mode, the fuel gauge is fully powered and can execute any allowable task. In SNOOZE mode, both low-frequency and high-frequency oscillators are active. Although the SNOOZE mode has higher current consumption than the SLEEP mode, it is also a reduced power mode. In SLEEP mode, the fuel gauge turns off the high-frequency oscillator and exists in a reduced-power state, periodically taking measurements and performing calculations. In HIBERNATE mode, the fuel gauge is in a low-power state, but can be woken up by communication or certain IO activity. Finally, the BAT INSERT CHECK mode is a powered up, but low-power halted, state, where the fuel gauge resides when no battery is inserted into the system.

[Figure 5-1](#) and [Figure 5-2](#) show the relationship between these modes.

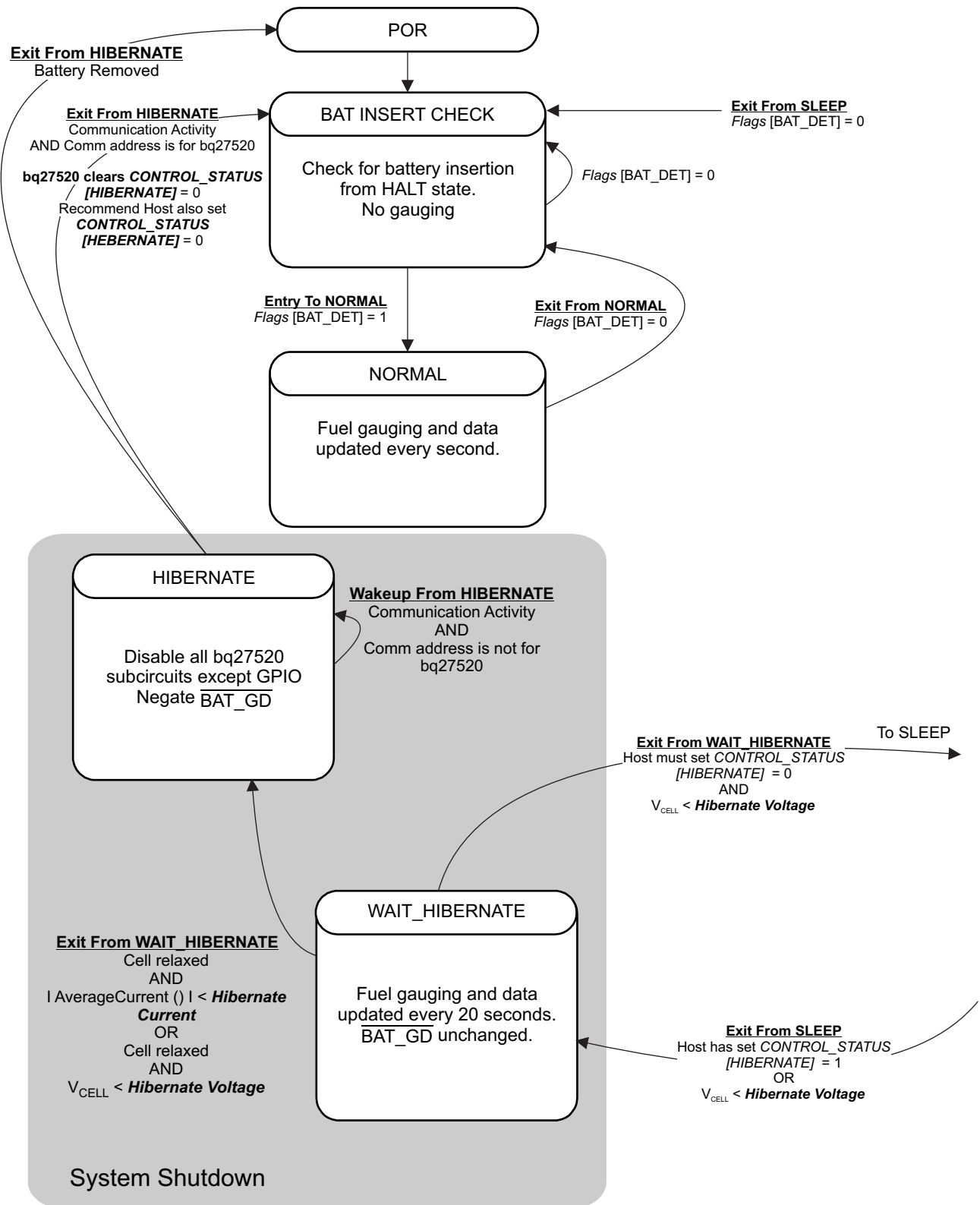


Figure 5-1. Power Mode Diagram for System Shutdown

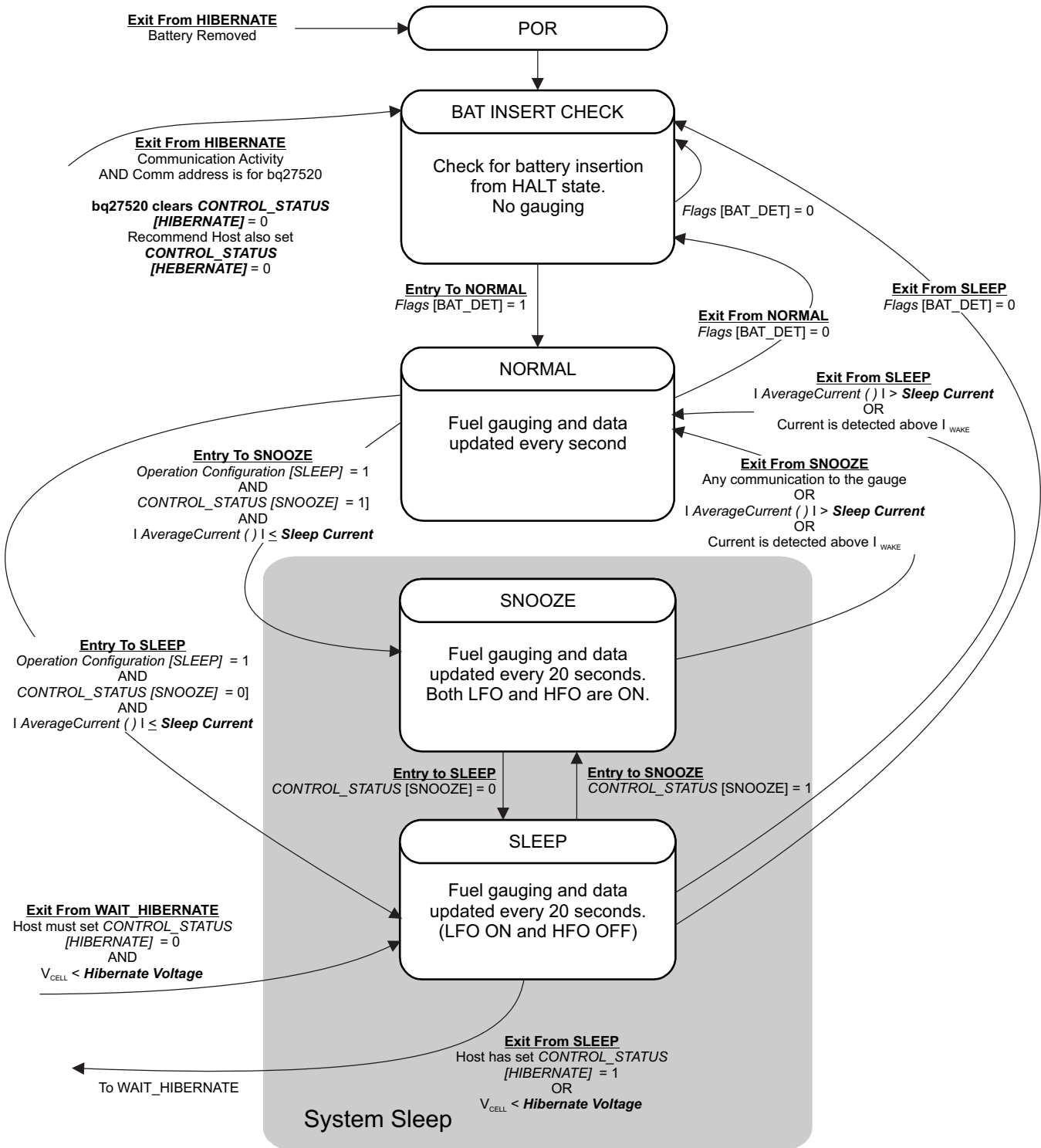


Figure 5-2. Power Mode Diagram for System Sleep

5.6.1 BAT INSERT CHECK Mode

This mode is a halted-CPU state that occurs when an adapter, or other power source, is present to power the fuel gauge (and system), yet no battery has been detected. If enabled via the **OpConfig B [BIE]** bit, the fuel gauge detects battery insertion either through use of the thermistor network or the BI/TOUT pin. Alternatively, the host can use the *BAT_INSERT* and *BAT_REMOVE* subcommands to inform the battery presence or removal status. When battery insertion is detected, a series of initialization activities begin which include: OCV measurement, setting the BAT_GD pin, and selecting the appropriate battery profiles.

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

5.6.2 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, 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 the NORMAL mode, the Impedance Track™ algorithm minimizes the time the fuel gauge remains in this mode.

5.6.3 SLEEP Mode

SLEEP mode is entered automatically if the feature is enabled (**Op Config [SLEEP]** bit = 1) and *AverageCurrent()* is below the programmable level **Sleep Current**. Once entry into SLEEP mode has been qualified, but prior to entering it, the fuel gauge performs a coulomb counter autocalibration to minimize offset.

During SLEEP mode, the fuel gauge periodically takes data measurements and updates its data set. However, a majority of its time is spent in an idle condition.

The fuel gauge exits the SLEEP mode if any entry condition is broken, specifically when either:

- *AverageCurrent()* rises above **Sleep Current**.
- A current in excess of I_{WAKE} through R_{SENSE} is detected.

In the event that a battery is removed from the system while a charger is present (and powering the gauge), Impedance Track™ updates are not necessary. Hence, the fuel gauge enters a state that checks for battery insertion and does not continue executing the Impedance Track™ algorithm.

5.6.4 SNOOZE Mode

Compared to the SLEEP mode, the SNOOZE mode has the high-frequency oscillator in operation, hence the communication delay associated with waking up from SLEEP mode can be eliminated. The SNOOZE mode is entered automatically if the feature is enabled (**CONTROL_STATUS [SNOOZE]** bit = 1) and *AverageCurrent()* is below the programmable level **Sleep Current**.

During SNOOZE mode, the fuel gauge periodically takes data measurements and updates its data set. However, a majority of its time is spent in an idle condition.

The fuel gauge exits the SNOOZE mode if any entry condition is broken, specifically when:

- Any communication activity with the gauge.
- *AverageCurrent()* rises above **Sleep Current**.
- A current in excess of I_{WAKE} through R_{SENSE} is detected.

5.6.5 HIBERNATE Mode

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

For normal entry to the HIBERNATE mode, the system must set the *CONTROL_STATUS [HIBERNATE]* bit by issuing a *SET_HIBERNATE* subcommand. The gauge does not enter the HIBERNATE mode until a valid OCV measurement is made and the magnitude of the average cell current has fallen below **Hibernate I**. Regardless of the *CONTROL_STATUS [HIBERNATE]* bit status, the gauge can also enter the HIBERNATE mode if *Voltage()* falls below **Hibernate V** and a valid OCV measurement has been taken. The gauge remains in the HIBERNATE mode until the system issues a direct I²C command to the gauge or a POR occurs. I²C communication that is not directed to the gauge does not wake the gauge.

For proper system-level coordination of the HIBERNATE mode with the use of a charger IC, see [Table 5-9, Pin Function Code Descriptions](#). It is important to prevent a charger from inadvertently charging the battery before an OCV reading can be taken. It is the system's responsibility to wake the fuel gauge after it has gone into the HIBERNATE mode. After waking, the gauge can proceed with the initialization of the battery information (OCV, profile selection, and so forth).

5.7 Application-Specific Information

5.7.1 Battery Profile Storage and Selection

The fuel gauge supports only one type of battery profile. This profile is stored in both the **Def0** and **Def1** profiles. When a battery pack is inserted for the first time, the default profile is copied into the **Packn** profiles. Then the Impedance Track™ algorithm begins gas gauging, regularly updating **Packn** as the battery is used.

In addition to the default profiles, the fuel gauge maintains two profiles: PACK0 and PACK1. These tables hold dynamic battery data, and keep track of the status for up to two of the most recent batteries used. In most cases, the fuel gauge can manage the information on two removable battery packs. When a battery pack is removed from host equipment, the fuel gauge maintains some of the battery information in case the battery is re-inserted. This way, the Impedance Track™ algorithm has a means of recovering battery-status information, thereby maintaining good state-of-charge (SOC) estimates.

When an existing pack is removed from the fuel gauge and a different (or same) pack is inserted, cell impedance is measured immediately after battery detection (see [Section 5.7.2, First OCV and Impedance Measurement](#)). The fuel gauge chooses the profile which is closest to the measured impedance, starting with the **Packn** profiles. That is, if the measured impedance matches **Pack0**, then the **Pack0** profile is used. If the measured impedance matches **Pack1**, then the **Pack1** profile is used. If the measured impedance does not match the impedance stored in either **Pack0** or **Pack1**, the battery pack is deemed new (none of the previously used packs). Either **Def0** or **Def1** profile is copied into either the **Pack0** or **Pack1** profile, overwriting the oldest **Packn** profile.

5.7.1.1 Reading Application Status

The **Application Status** data flash location contains cell profile status information, and can be read using the *ApplicationStatus()* extended command (0x6A). The bit configuration of this function or location is shown in [Table 5-12](#).

Table 5-12. ApplicationStatus() Bit Definitions

Application Configuration	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	—	—	—	—	—	—	—	LU_PROF

LU_PROF = Last profile used by fuel gauge. *Cell0* last used when cleared. *Cell1* last used when set. Default is 0.

5.7.2 First OCV and Impedance Measurement

Upon power-up or pack insertion an open-circuit voltage (OCV) measurement of the battery is made via the BAT pin. For best gauging results, the system load during the OCV measurement should not exceed a C/20 discharge rate of the battery. For this first critical measurement, both BAT_GD and SOC_INT pins are available for system synchronization. (See [Section 5.3.5](#), *Power Path Control With the BAT_GD Pin*, [Section 5.3.4](#), *SOC_INT Pin Behavior*, and [Section 2.1.6](#), *OCV_CMD: 0x000C*.)

Upon completion of the OCV voltage measurement, the *CONTROL_STATUS [OCVCMDCOMP]* is set. [Table 4-6](#), *Data Flash Summary—OCV Table*, and the first OCV voltage measurement determine the initial SOC of the inserted battery, while impedance is computed from subsequent voltage and current samples under load using:

$$Z(\text{SOC}) = (\text{OCV}(\text{SOC}) - V) / I.$$

This impedance is compared with the impedance of the dynamic profiles, *Packn*, and the default profiles, *Defn*, for the same SOC. Following all initialization functions, the *CONTROL_STATUS [INITCOMP]* bit is set.

5.8 Additional Data Flash Parameter Descriptions

5.8.1 TCA Set %

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

TCA Set % only affects the *[CHG]* bit but does not affect the charge termination process or the gauging function. The default value is set to 99%.

5.8.2 TCA Clear %

TCA Clear % is the Terminate Charge Alarm Clear Percentage threshold. **TCA Clear %** sets a *StateOfCharge()* percentage level at which the *Flags() [CHG]* bit is set.

TCA Clear % only affects the *[CHG]* bit but does not affect the charge termination process or the gauging function. The default value is set to 95%.

5.8.3 FC Set %

FC Set % is the Full Charge Set Percentage threshold. **FC Set %** sets a *StateOfCharge()* percentage threshold at which the *Flags() [FC]* bit is set. When **FC Set %** is a value other than -1, the *[FC]* bit is set based on the amount of passed charge detected by the gauge and not charge termination detection. If **FC Set %** is set to -1, the *[FC]* bit is set based on charge termination detection (see **Taper Current** and **Taper Voltage** in [Section 5.5.1](#)).

FC Set % only affects the *[FC]* bit which does not affect the charge termination process. The default value is set to 100%.

5.8.4 FC Clear %

FC Clear % is the Full Charge Clear Percentage threshold. **FC Clear %** sets a *StateOfCharge()* percentage threshold at which the *Flags() [FC]* bit is cleared.

FC Clear % only affects the *[FC]* bit register which does not affect the charge termination process. The default value is set to 98%.

5.8.5 DOD at EOC Delta Temperature

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

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

5.8.6 Default Temperature

This is the temperature used to initialize the *Temperature*() register until the host writes a different value if the **OpConfig B [WRTEMP]** bit is set.

5.8.7 Device Name

This is string data that can be a maximum of 7 characters. This field does not affect the operation, nor is it used by the part. It is read by using the extended data command: *DeviceName*() (0x63 through 0x69).

5.8.8 Data Flash Version

This location can be used to store the data flash configuration version. Version control of golden flash files used in production is recommended.

5.8.9 SOC1 Set Threshold

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

The default value is set to 10%.

5.8.10 SOC1 Clear Threshold

SOC1 Clear Threshold sets a *StateOfCharge*() percentage threshold used to indicate when *StateOfCharge*() rises above a defined *StateOfCharge*(). When *StateOfCharge*() rises above the **SOC1 Clear Threshold**, the State of Charge Initial [SOC1] bit in the *Flags*() register is cleared.

SOC1 Clear Threshold is normally set to 5% above the **SOC1 Set Threshold**. The default value is set to 15%.

5.8.11 Final Voltage and Final Volt Time

If *Voltage*() is below **Final Voltage** for at least **Final Volt Time** (in seconds), then *RemainingCapacity*() and *StateOfCharge*() are forced to 0. **Final Voltage** is usually set to the same value as **Terminate Voltage**.

5.8.12 Def Avg I Last Run and Def Avg P Last Run

These parameters are not used in the fuel gauge.

5.8.13 Max Res Factor

Max percentage (ratio) that an impedance value stored in the Ra table is allowed to change in a single update in the positive direction.

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

5.8.14 **Min Res Factor**

Max percentage (ratio) that an impedance value stored in the Ra table is allowed to change in a single update in the negative direction.

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

5.8.15 **Ra Filter**

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

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

5.8.16 **ResRelax Time**

This value is used for Impedance Track™ transient modeling of effective resistance. The resistance increases from zero to final value determined by the Ra table as defined by the exponent with time constant **ResRelax Time** during discharge simulation. Default value has been optimized for typical cell behavior, but could be increased if the gauge is being too conservative at low temperature.

5.8.17 **Max Sim Rate, Min Sim Rate**

Maximum and minimum limits for current used in simulation runs. The parameters are functions of *DesignCapacity()* (that is, **C/Max Sim Rate** or **C/Min Sim Rate**).

5.8.18 **Transient Factor Charge and Discharge**

When a battery is inserted and the system is powering up, it is possible that current may be flowing at the same time the gauge is initializing the SOC based on a voltage measurement. The gauge compensates for this current flow but the amount of compensation can be adjusted by changing the values of these data flash parameters. For most cases, the default values are recommended.

5.8.19 **Max IR Correct**

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

5.8.20 **Thermal Modeling**

At low temperatures the gauge could be overly conservative if it assumes the low temperature will persist. In fact, most systems will self-heat during operation. These thermal modeling parameters can be adjusted to predict the self-heating.

The **T Rise** constant reflects the level of system heating due to self-heating of the cell during discharge. This number can be measured empirically. It can be adjusted also through experimentation. In general, it can be increased if the gauge is too conservative at low temperature.

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. It can be adjusted also through experimentation. In general, it can be decreased if the gauge is too conservative at low temperature.

5.8.21 Cell 0 and 1 V at Chg Term

Records the voltage reading at the full charge taper termination. This voltage is used to calculate DODatEOC. Essentially this register is updated automatically to learn the true full level of the system or charger.

5.8.22 Calibration, Data, ID = 104

Most of the following values never require modification by the user. They are only modified by the Calibration commands in Calibration mode as explained in the application report: *Going to Production with the bq275xx* ([SLUA449](#)). For calibration using a host system, see *Host System Calibration Method* ([SLUA640](#)).

CC Gain

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

CC Delta

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

CC Offset

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

- (A) The first method is to calibrate **CC Offset** by putting the fuel gauge in Calibration mode and initiating the **CC Offset** function as part of the entire calibration suite. See the application note: *Going to Production with the bq275xx* ([SLUA449](#)) for more information on the Calibration mode. This is a short calibration that is not as accurate as the second method, **Board Offset**. Its primary purpose is to calibrate **CC Offset** enough so that it does not affect any other Coulomb Counter calibrations. This is only intended as a temporary calibration because the automatic calibration, **Board Offset**, is done the first time the I²C Data and Clock is low for more than 20 seconds, which is a much more accurate calibration.
- (B) During normal Gas Gauge Operation when the I²C clock and data lines are low for more than 5 seconds and *AverageCurrent()* is less than **Sleep Current** in mA, then an automatic **CC Offset** calibration is performed. This takes approximately 16 seconds and is much more accurate than the method in Calibration mode.

Board Offset

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

Int Temp Offset

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

Ext Temp Offset

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

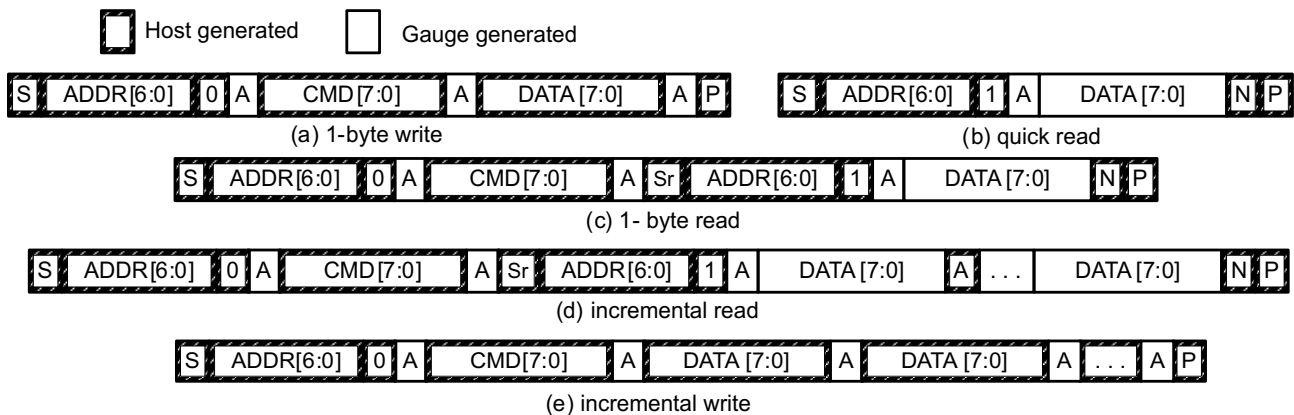
Pack V Offset

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

Communications

6.1 I²C Interface

The bq27520-G4 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 is, therefore, 0xAA or 0xAB for write or read, respectively.

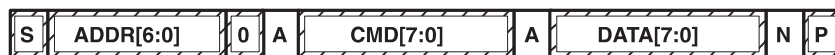


(S = Start, Sr = Repeated Start, A = Acknowledge, N = No Acknowledge, and P = Stop).

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 master. “Quick writes” function in the same manner and are a convenient means of sending multiple bytes to consecutive command locations (such as two-byte commands that require two bytes of data)

The following command sequences are not supported:

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



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

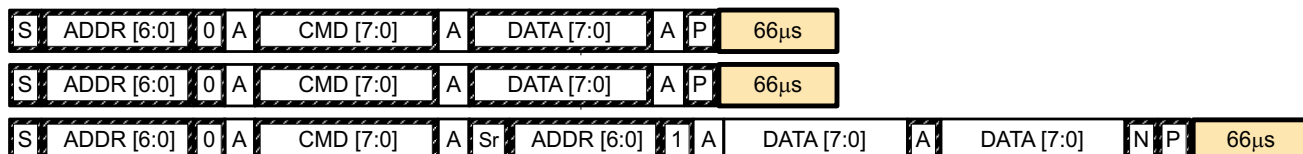


6.2 I²C Time Out

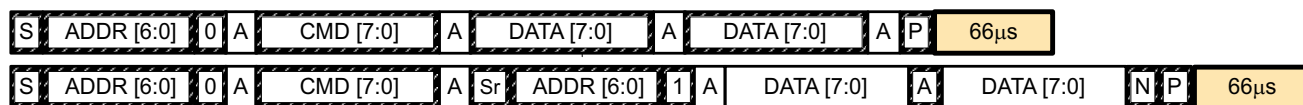
The I²C engine releases both SDA and SCL if the I²C bus is held low for 2 seconds. If the fuel gauge was holding the lines, releasing them frees them for the master to drive the lines. If an external condition is holding either of the lines low, the I²C engine enters the low-power sleep mode.

6.3 I²C Command Waiting Time

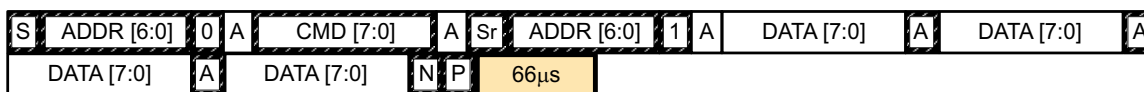
To ensure proper operation at 400 kHz, a $t_{(BUF)} \geq 66 \mu s$ bus free waiting time must be inserted between all packets addressed to the fuel gauge. In addition, if the SCL clock frequency (f_{SCL}) is > 100 kHz, use individual 1-byte write commands for proper data flow control. The following diagram shows the standard waiting time required between issuing the control subcommand the reading the status result. An *OCV_CMD* subcommand requires 1.2 seconds prior to reading the 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 should not issue all standard commands more than two times per second. Otherwise, the fuel 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 $100 \text{ kHz} < f_{SCL} \leq 400 \text{ kHz}$)



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



Waiting time inserted after incremental read

6.4 I²C Clock Stretching

A clock stretch can occur during all modes of fuel gauge operation. In SLEEP and HIBERNATE modes, a short clock stretch occurs on all I²C traffic as the device must wake-up to process the packet. In the other modes (BAT INSERT CHECK, NORMAL, SNOOZE) clock stretching only occurs for packets addressed for the fuel gauge. The majority of clock stretch periods are small as the I²C interface performs normal data flow control. However, less frequent yet more significant clock stretch periods may occur as blocks of Data Flash are updated. The following table summarizes the approximate clock stretch duration for various fuel gauge operating conditions.

Gauging Mode	Operating Condition or Comment	Approximate Duration
SLEEP HIBERNATE	Clock stretch occurs at the beginning of all traffic as the device wakes up.	5 ms
BAT INSERT CHECK, NORMAL, SNOOZE	Clock stretch occurs within the packet for flow control (after a start bit, ACK or first data bit).	100 μs
	Normal Ra table Data Flash updates.	24 ms
	Data Flash block writes.	72 ms
	Restored Data Flash block write after loss of power.	116 ms
	End of discharge Ra table Data Flash update.	144 ms

Reference Schematic

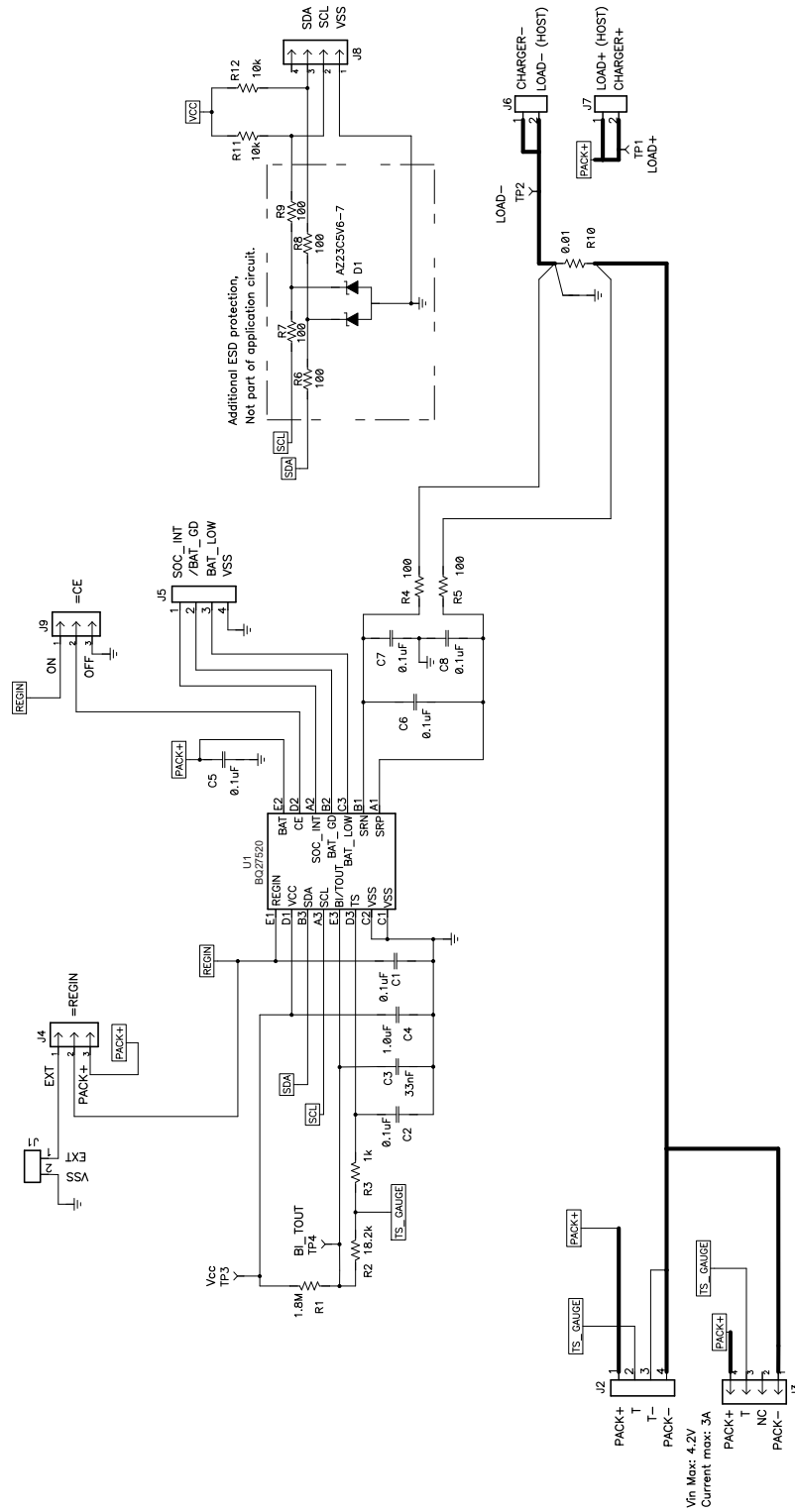


Figure 7-1. Reference Schematic

Open-Circuit Voltage Measurement Background

The accuracy of the Impedance Track™ (IT) algorithm strongly depends on the accuracy and validity of the open-circuit voltage (OCV) measurement taken by fuel gauges that are based on IT technology. This appendix describes the process of taking OCV measurements during different events.

A.1 Background

- **OCV Calculation:** OCV (open-circuit voltage) is normally a calculated value because a true measurement of OCV requires an unloaded and relaxed condition on the battery. Because such an unloaded and completely relaxed condition is not always possible in a real system, the fuel gauge uses measured voltage, current, and temperature (VIT) to compute the OCV and as a result of this calculation, the state of charge (SOC) of the battery is established or reestablished.
- **OCV Qualification Time (QT):** The time in which SOC_INT is asserted during an OCV measurement is approximately 165 ms. This is the timeframe in which we test if the VIT measurement is qualified for an OCV calculation. This is not the timeframe in which the actual VIT measurement is taken. During this time, the instantaneous current (adci) is measured. If $\text{abs(adci)} \geq \text{DesignCapacity}/18$, then the OCVFail bit is set. Otherwise, the VIT that we have just measured is qualified and the gauge proceeds with OCV calculation.
- **Current Measurement Time (CMT):** The time of current is measured – 1 s.
- **Voltage Measurement Time (VMT):** The time of voltage is measured – 125 ms.
- **Temperature Measurement Time (TMT):** The time of temperature is measured – 125 ms.

A.1.1 OCV Qualification and Calculation

OCV qualification and calculation (**QC**) happens under two conditions:

- OCV_CMD is sent by the host.
- Battery Insert (**BI**) event is detected.

NOTE: POR causes an immediate BI.

A.1.2 OCV Calculation Assumption

The current, voltage, and temperature must remain stable during QT, CMT, VMT, and TMT. In every case that stable VIT is mentioned, the desired stable condition for current is $<C/20$. If this is not true, error can be introduced into the OCV Calculation.

A.1.3 OCV Timing

The timing of each step in the OCV sequence is shown in [Figure A-1](#).

1. After a POR, voltage, current, and temperature are measured before updating the fuel gauge parameters.
2. Quick voltage and current measurements are taken to qualify OCV VIT conditions.
3. Voltage, current, and temperature are measured for subsequent fuel gauge parameters updates.

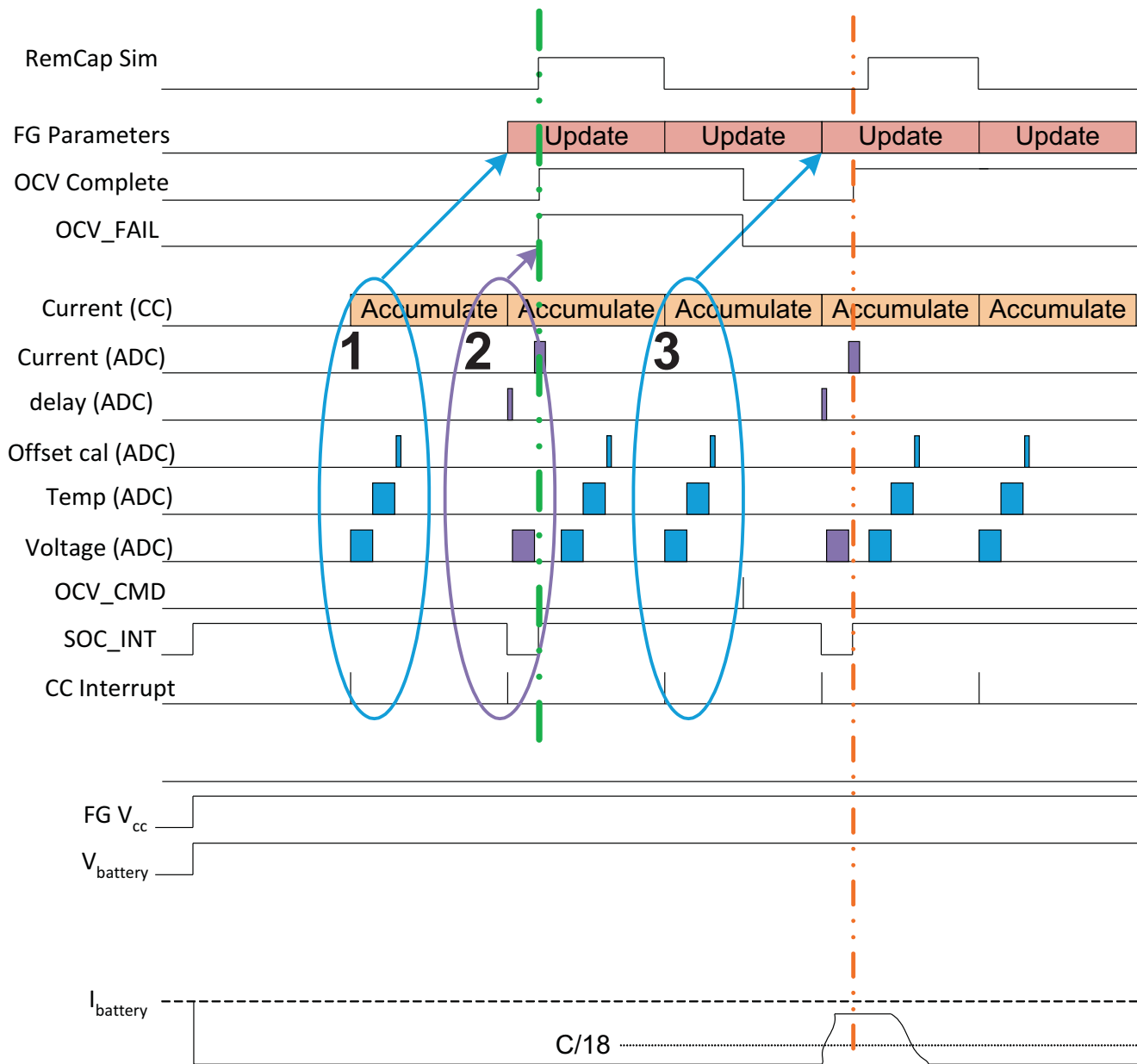


Figure A-1. OCV Timing Sequences

The green dashed lines indicate the completion of an OCV measurement that has failed due to high load detected in current (ADC) measurement, whereas the orange dashed lines indicate the completion of a successful OCV measurement, given that the load at the time of measurement was below C/18 rate.

The second OCV measurement (orange line) is a success by qualification standard. However, this is not the recommended-use case because the current is only lowered during the OCV_INT time (the qualification time). This makes the fuel gauge respond as if this were a pass condition; however, the actual result is not good because the actual VIT measurement used for OCV was taken under high load.

A.2 OCV Timing and OCV_CMD Use Recommendations

A.2.1 ACTIVE Mode (Fuel Gauge is not in SLEEP Mode)

The VIT measurement used for the OCV calculation is the last VIT measured before the OCV_CMD was received. The VIT value used for the OCV calculation needs to be a stable, not transient value. Before sending the OCV_CMD, the current must be stable and $<C/20$ for at least one second. The recommendations for the OCV_CMD used for active mode is that the VIT remains stable from two seconds before the OCV_CMD is sent until the end of SOC_INT (see [Figure A-2](#)).

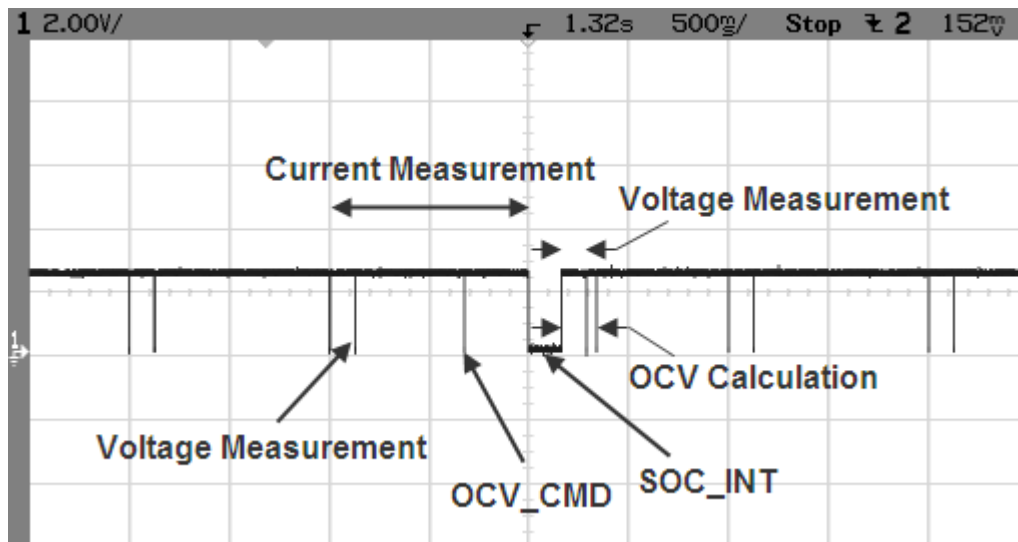


Figure A-2. OCV Calculation Based on OCV Command

A.2.2 SLEEP Mode

In SLEEP mode, the fuel gauge measures VIT every 20 s, instead of 1 s. The VIT measurement used for the OCV calculation is the last VIT measured before the OCV_CMD was received. Sleep current is usually below the OCV current-fail threshold. So, the recommendations for the OCV_CMD sent during SLEEP mode is that the VIT remains stable and below the sleep threshold from the time OCV_CMD is sent until the end of SOC_INT.

A.2.3 Initial OCV – POR

During POR, the VIT measurement used for the OCV calculation and qualification takes place between about 300 ms after POR until the end of SOC_INT. To achieve a good initial OCV measurement after POR, the recommendation is to keep VIT stable from POR until the end of SOC_INT (see [Figure A-3](#)).

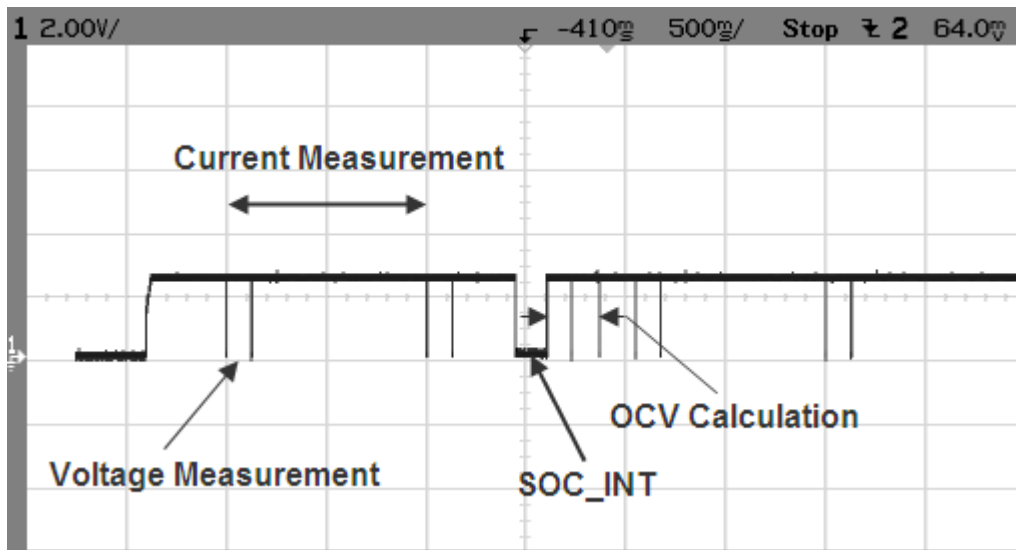


Figure A-3. Initial OCV Taken After POR

Glossary

ACK	Acknowledge character
ADC	Analog-to-digital converter
BCA	Board calibration
BI	Battery insert
CC	Coulomb counter
CCA	Coulomb counter calibration
CE	Chip enable
Charge Mode	Refers to a mode to where the gauge read <i>AverageCurrent()</i> > Chg Current Threshold for at least 1 second.
Clear	Refers to a bit in a register becoming a logic LOW or 0. The bqEvaluation software (EVSW) represents a clear bit with the color green .
cWh	Centiwatt-hour
CMT	Current measurement time
DF	Data flash
Discharge Mode	Refers to a mode where the gauge read <i>AverageCurrent()</i> < (-) Dsg Current Threshold for at least 1 second.
DOD	Depth of discharge in percent as related to Qmax. 100% corresponds to empty battery.
DOD0	Depth of discharge that was looked up in the DOD (OCV) table based on OCV measurement in relaxed state.
EOC	End of charge
FC	Fully charged
FCC	Full charge capacity. Total capacity of the battery compensated for present load current, temperature, and aging effects (reduction in chemical capacity and increase in internal impedance).
FIFO	First in, first out
Flag	This word usually represents a read-only status bit that indicates some action has occurred or is occurring. This bit typically cannot be modified. The flags are set and cleared automatically by the fuel gauge.
FVCA	Fast voltage and current acquisition
GPIO	General-purpose input output
HDQ	High-speed data queue
IC	Integrated circuit
ID	Identification
IO	Input or output
IT	Impedance Track™
I ² C	Inter-integrated circuit
LDO	Low dropout
LSB	Least significant bit
LT	Lifetime
MAC	Manufacturer access command or control command
mAh	Milliamp-hour
MSB	Most significant bit
mWh	Milliwatt-hour
NACK	Negative acknowledge character
NTC	Negative temperature coefficient
OCV	Open-circuit voltage. Voltage measured on fully-relaxed battery with no load applied.
OTC	Overtemperature in charge
OTD	Overtemperature in discharge
PFC	Pin function code
POR	Power-on reset
Qmax	Maximum chemical capacity

QC	Qualification and calculation
QT	Qualification time
Relaxation Mode	Refers to a mode to where the gauge read <i>AverageCurrent()</i> < Quit Current for at least 60 seconds.
RM	Remaining capacity
RW	Read or write
SCL	Serial clock: programmable serial clock used in the I ² C interface
SDA	Serial data: serial data bus in the I ² C interface
SE	Shutdown enable
Set	Refers to a bit in a register becoming a logic HIGH or 1. The bqEvaluation software (EVSW) represents a set bit with the color red .
SOC	State-of-charge in percent related to FCC
SOC1	State-of-charge initial
SOCF	State-of-charge final
System	The word system is sometimes used in this document. When used, it always means a host system that is consuming current from the battery pack.
TCA	Terminate charge alarm
TMT	Temperature measurement time
TS	Temperature status
TTE	Time-to-empty
TTF	Time-to-full
VIT	Voltage, current, temperature
VMT	Voltage measurement time

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com