bq27532-G1 Technical Reference Manual

User's Guide

Literature Number: SLUUB04
October 2014

Contents

Preface **8**

1 General Description **9**

2 Standard Data Commands **10**

2.1 ***Control()***: 0x00 and 0x01 11

2.1.1 CONTROL_STATUS: 0x0000 12

2.1.2 DEVICE_TYPE: 0x0001 13

2.1.3 FW_VERSION: 0x0002 13

2.1.4 HW_VERSION: 0x0003 13

2.1.5 MLC_ENABLE: 0x0004 13

2.1.6 MLC_DISABLE: 0x0005 13

2.1.7 CLEAR_IMAX_INT: 0x0006 13

2.1.8 PREV_MACWRITE: 0x0007 13

2.1.9 CHEM_ID: 0x0008 13

2.1.10 BOARD_OFFSET: 0x0009 13

2.1.11 CC_OFFSET: 0x000A 13

2.1.12 CC_OFFSET_SAVE: 0x000B 13

2.1.13 OCV_CMD: 0x000C 14

2.1.14 BAT_INSERT: 0x000D 14

2.1.15 BAT_REMOVE: 0x000E 14

2.1.16 BYPASS_ENABLE: 0x000F 14

2.1.17 BYPASS_DISABLE 14

2.1.18 SET_HIBERNATE: 0x0011 14

2.1.19 CLEAR_HIBERNATE: 0x0012 14

2.1.20 SET_SLEEP+ Mode: 0x0013 14

2.1.21 CLEAR_SLEEP+ Mode: 0x0014 14

2.1.22 ILIMIT_LOOP_ENABLE: 0x0015 15

2.1.23 ILIMIT_LOOP_DISABLE: 0x0016 15

2.1.24 SHIPMODE_ENABLE: 0x0017 15

2.1.25 SHIPMODE_DISABLE: 0x0018 15

2.1.26 CHG_ENABLE: 0x001A 15

2.1.27 CHG_DISABLE: 0x001B 15

2.1.28 GG_CHGRCTL_ENABLE: 0x001C 15

2.1.29 GG_CHGRCTL_DISABLE: 0x001D 15

2.1.30 SMOOTH_SYNC: 0x001E 15

2.1.31 DF_VERSION: 0x001F 15

2.1.32 SEALED: 0x0020 16

2.1.33 IT_ENABLE: 0x0021 16

2.1.34 RESET: 0x0041 16

2.2 ***AtRate()***: 0x02 and 0x03 16

2.3 ***AtRateTimeToEmpty()***: 0x04 and 0x05 16

2.4 ***Temperature()***: 0x06 and 0x07 16

2.5 ***Voltage()***: 0x08 and 0x09 16

2.6 ***Flags()***: 0x0A and 0x0B 16

2.7 ***NominalAvailableCapacity()***: 0x0C and 0x0D 17

2.8 ***FullAvailableCapacity()***: 0x0E and 0x0F 17

2.9 ***RemainingCapacity()***: 0x10 and 0x11 17
2.10 ***FullChargeCapacity()***: 0x12 and 0x13 17
2.11 ***AverageCurrent()***: 0x14 and 0x15 17
2.12 ***InternalTemperature()***: 0x16 and 0x17 18
2.13 ***ResScale***: 0x18 and 0x19 18
2.14 ***ChargingLevel():*** 0x1A and 0x1B 18
2.15 ***StateofHealth()***: 0x1C and 0x1D 18
2.16 ***CycleCount()***: 0x1E and 0x1F 18
2.17 ***StateOfCharge()***: 0x20 and 0x21 18
2.18 ***InstantaneousCurrentReading():*** 0x22 and 0x23 18
2.19 ***FineQPass()***: 0x24 and 0x25 19
2.20 ***FineQPassFract()***: 0x26 and 0x27 19
2.21 ***ProgChargingCurrent()***: 0x28 and 0x29 19
2.22 ***ProgChargingVoltage()***: 0x2A and 0x2B 19
2.23 ***LevelTaperCurrent():*** 0x2C and 0x2D 19
2.24 ***CalcChargingCurrent():*** 0x2E and 0x2F 19
2.25 ***CalcChargingVoltage():*** 0x30 and 0x31 19
2.26 ***InternalTemperature():*** 0x32 and 0x33 19
2.27 ***RemainingCapacityUnfiltered()***: 0x6C and 0x6D 19
2.28 ***RemainingCapacityFiltered()***: 0x6E and 0x6F 20
2.29 ***FullChargeCapacityUnfiltered()***: 0x70 and 0x71 20
2.30 ***FullChargeCapacityFiltered()***: 0x72 and 0x73 20
2.31 ***TrueSOC()***: 0x74 and 0x75 20
2.32 ***MaxCurrent()***: 0x76 and 0x77 20

3 Charger Interface 21
3.1 Charger Data Commands 21
3.1.1 ChargerStatus(): 0x32 22
3.1.2 Chrgr_Reg0(): 0x33 23
3.1.3 Chrgr_Reg1(): 0x34 23
3.1.4 Chrgr_Reg2(): 0x35 24
3.1.5 Chrgr_Reg3(): 0x36 24
3.1.6 Chrgr_Reg4(): 0x37 25
3.1.7 Chrgr_Reg5(): 0x38 26
3.1.8 Chrgr_Reg6(): 0x39 26
3.2 Charging Control 27
3.2.1 Charging Voltage and Current Standard Commands 27
3.2.2 Charging Voltage and Current Parameters Vary With Temperature() 27
3.2.3 Charging Voltage and Current Parameters Variance With StateofHealth() 28
3.2.4 Multi-Level Charging Algorithm 30
3.3 Data Flash Settings for Charger Operation 30
3.4 Indirect Charger Control 30

4 Extended Data Commands 32
4.1 ***DataFlashClass()***: 0x3E 32
4.2 ***DataFlashBlock()***: 0x3F 32
4.3 ***BlockData()***: 0x40 to 0x5F 32
4.4 ***BlockDataChecksum()***: 0x60 32
4.5 ***BlockDataControl()***: 0x61 33

5 Data Flash Interface 34
5.1 Accessing The Data Flash 34
5.2 Manufacturer Information Block 34
5.3 Access Modes 35
5.4 Sealing and Unsealing Data Flash 35
5.5 Data Flash Summary 36

5.6 Data Flash Parameter Update Example 42
5.6.1 Modify WRTEMP of OpConfigB Register 43
6 Functional Description 44
6.1 Fuel Gauging 44
6.2 Impedance Track™ Variables 44
6.2.1 Load Mode 44
6.2.2 Load Select 44
6.2.3 Reserve Cap-mAh 45
6.2.4 Reserve Cap-mWh 45
6.2.5 Dsg Current Threshold 45
6.2.6 Chg Current Threshold 45
6.2.7 Quit Current, DSG Relax Time, CHG Relax Time, and Quit Relax Time 46
6.2.8 Qmax 0 46
6.2.9 Update Status 0 46
6.2.10 Avg I Last Run 46
6.2.11 Avg P Last Run 46
6.2.12 Delta Voltage 46
6.2.13 Ra Tables 47
6.2.14 Fast Resistance Scaling 47
6.2.15 Fast Qmax Update 47
6.2.16 SOC Smoothing 48
6.2.17 Operation Configuration Registers 48
6.3 Detailed Pin Description 50
6.3.1 Battery Detection Using the BI/TOUT Pin 50
6.3.2 SOC_INT Pin 51
6.4 Temperature Measurement 51
6.5 Charging and Charge-Termination Indication 52
6.5.1 Detecting Charge Termination 52
6.6 Power Modes 52
6.6.1 BAT INSERT CHECK Mode 52
6.6.2 NORMAL Mode 52
6.6.3 SLEEP Mode 52
6.6.4 SLEEP+ Mode 53
6.6.5 HIBERNATE Mode 53
6.7 Power Control 56
6.7.1 Wake-up Comparator 56
6.7.2 Flash Updates 56
6.8 Autocalibration 56
6.9 Additional Data Flash Parameter Descriptions 57
6.9.1 Configuration Class 57
6.9.2 Gas Gauging Class 60
6.9.3 OCV Tables Class 68
6.9.4 Ra Tables Class 68
6.9.5 Calibration Class 69
6.9.6 Security Class 71
6.9.7 Charger Class 71
7 Application-Specific Information 76
7.1 Battery Profile Storage and Selection 76
7.1.1 Common Profile Aspects 76
7.1.2 Activities Upon Pack Insertion 76
8 Communications 77
8.1 I^2C Interface 77

8.2 I²C Time Out 78
8.3 I²C Command Waiting Time 78
8.4 I²C Clock Stretching 78
9 Reference Schematic 80
A Open-Circuit, Voltage Measurement Background 82
A.1 Background 82
A.1.1 OCV Qualification and Calculation 82
A.1.2 OCV Calculation Assumption 82
A.1.3 OCV Timing 82
A.2 OCV Timing and OCV_CMD Use Recommendations 84
A.2.1 ACTIVE Mode (Fuel Gauge is not in SLEEP Mode) 84
A.2.2 SLEEP Mode 84
A.2.3 Initial OCV – POR 84
B Glossary 86
Index 88

List of Figures

3-1. Charger Power-up 21
3-2. Charging Profile Example Based on SOH Range 29
3-3. Suggested Flow for Indirect Charger Control 31
6-1. Power Mode Diagram—System Sleep 54
6-2. Power Mode Diagram—System Shutdown 55
9-1. Charger Application Schematic 81
A-1. OCV Timing Sequences 83
A-2. OCV Calculation Based on OCV Command 84
A-3. Initial OCV Taken After POR 85

List of Tables

2-1. Standard Commands 10
2-2. ***Control()*** Subcommands 11
2-3. CONTROL_STATUS Bit Definitions 12
2-4. ***Flags()*** Bit Definitions 17
3-1. bq2425x Family Table 21
3-2. Charger Data Commands 22
3-3. ***ChargerStatus()*** Bit Definitions 22
3-4. ***Chrgr_Reg0()*** Bit Definitions 23
3-5. ***Chrgr_Reg1()*** Bit Definitions 23
3-6. ***Chrgr_Reg2()*** Bit Definitions 24
3-7. ***Chrgr_Reg3()*** Bit Definitions 25
3-8. ***Chrgr_Reg4()*** Bit Definitions 25
3-9. ***Chrgr_Reg5()*** Bit Definitions 26
3-10. ***Chrgr_Reg6()*** Bit Definitions 26
3-11. Default Data Flash Settings for Temperature Charging 27
3-12. Sample Data Flash Settings for State of Health Based Charging 29
3-13. Charger Options Bit Definitions 30
4-1. Extended Data Commands 32
5-1. Data Flash Access 35
5-2. Data Type Decoder 36
5-3. Data Flash Summary—Configuration Class 36
5-4. Data Flash Summary—System Data Class 37
5-5. Data Flash Summary—Gas (Fuel) Gauging Class 38
5-6. Data Flash Summary—OCV Tables 39
5-7. Data Flash Summary—Ra Tables 39
5-8. Data Flash Summary—Calibration Class 40
5-9. Data Flash Summary—Security Class 41
5-10. Data Flash Summary—Charger Class 41
5-11. OpConfig B Register Bit Definitions 42
5-12. Data Flash Summary—Configuration 42
6-1. Constant-Current Model Used When Load Mode = 0 44
6-2. Constant-Power Model Used When Load Mode = 1 45
6-3. Operation Configuration Bit Definitions 48
6-4. Operation Configuration B Bit Definitions 49
6-5. Operation Configuration C Bit Definitions 49
6-6. Operation Configuration D Bit Definitions 49

6-7. Operation Configuration E Bit Definitions 50
6-8. SOC_INT Pulse Conditions and Widths 51
6-9. I_{WAKE} Threshold Settings 56
6-10. Data Flash Parameter Scale and Unit Based on ***Design Energy Scale*** 58
6-11. Constant-Current Model Used When Load Mode = 0 60
6-12. Constant-Power Model Used When Load Mode = 1 61
6-13. Charger Register 00 Default Bit Descriptions 72
6-14. Charger Register 01 Default Bit Descriptions 72
6-15. Charger Register 02 Default Bit Descriptions 72
6-16. Charger Register 03 Default Bit Descriptions 72
6-17. Charger Register 04 Default Bit Descriptions 73
6-18. Charger Register 05 Default Bit Descriptions 73
6-19. Charger Register 06 Default Bit Descriptions 74
6-20. Charger Options Bit Definitions 74

Preface

This document is a detailed Technical Reference Manual (TRM) for using and configuring the bq27532-G1 battery fuel gauge. This TRM document is intended to complement but not supersede any information contained in the separate bq27532-G1 data sheet.

Refer to the *bq27532-G1 Data Sheet (SLUSBU6)*.

Formatting conventions used in this document:

Information Type	Formatting Convention	Example
Commands	*Italics* with parentheses and no breaking spaces	*RemainingCapacity()* command
Data Flash	*Italics*, **bold**, and breaking spaces	***Design Capacity*** data
Register bits and flags	Brackets and *italics*	*[QEN]* bit
Data Flash bits	Brackets, *italics*, and **bold**	***[TEMPS]*** 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 Support Center at (512) 434-1560. 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. *bq27532-G1, Battery Management Unit Impedance Track™ Fuel Gauge for bq2425x Charger* Data Sheet (SLUSBU6)
2. *Theory and Implementation of Impedance Track™ Battery Fuel-Gauging Algorithm in bq2750x Family* Application Report (SLUA450)
3. *How to Generate Golden Image for Single-Cell Impedance Track™ Devices* Application Report (SLUA544)
4. *Host System Calibration Method* Application Report (SLUA640)

Revision History

Version	Change Date	Description
—	October 2014	Initial Release

Impedance Track is a trademark of Texas Instruments.

General Description

The bq27532-G1 fuel gauge accurately predicts the battery capacity and other operational characteristics of a single, Li-based, rechargeable cell. It can be interrogated by a system processor to provide cell information, such as remaining capacity and state-of-charge (SOC), as well as the SOC interrupt signal to the host.

The fuel gauge can control a bq2425x Charger IC without the intervention from an application system processor. Using the fuel gauge and bq2425x chipset, batteries can be charged with the typical constant-current, constant-voltage (CCCV) profile or charged using a multi-level charging (MLC) algorithm.

Information is accessed through a series of commands, called *Standard Commands*. Further capabilities are provided by the additional *Extended Commands* set. Both sets of commands, indicated by the general format *Command()*, are used to read and write information contained within the device control and status registers, as well as its data flash locations. Commands are sent from system to gauge using the I^2C serial communications engine, and can be executed during application development, pack 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. Generally, they cannot 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 (see Section 5.5, *Data Flash Summary*).

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.

The device measures charging or discharging of the battery 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 computed, based on cell current, cell open-circuit voltage (OCV), and cell voltage under loading conditions.

The device external temperature sensing is optimized with the use of a high-accuracy, negative temperature coefficient (NTC) thermistor with R25 = 10.0 kΩ ±1%, B25/85 = 3435 kΩ ± 1% (such as Semitec NTC 103AT). The fuel gauge can also be configured to use its internal temperature sensor. When an external thermistor is used, a 18.2-kΩ pullup 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, SLEEP, 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 6.6, *Power Modes*.

Standard Data Commands

The fuel gauge uses a series of standard commands to enable system reading and writing of battery information. Each standard command has an associated command-code, as indicated in Table 2-1. Some commands consist of two bytes of data, so two consecutive I^2C transmissions must be executed both to initiate the command function, and to read or write the corresponding two bytes of data. Additional options for transferring data, such as spooling, are described in Chapter 8, *Communications*. Standard commands are accessible in NORMAL operation. Read-write permissions depend on the active access mode, SEALED or UNSEALED (for details on the SEALED and UNSEALED states, see Section 5.3, *Access Modes*).

Table 2-1. Standard Commands

Name	Command Code	Unit	SEALED Access	UNSEALED Access
Control()	0x00 and 0x01	hex	RW	RW
AtRate()	0x02 and 0x03	mA	RW	RW
AtRateTimeToEmpty()	0x04 and 0x05	Minutes	R	RW
Temperature()	0x06 and 0x07	0.1 K	RW	RW
Voltage()	0x08 and 0x09	mV	R	RW
Flags()	0x0A and 0x0B	hex	R	RW
NominalAvailableCapacity()	0x0C and 0x0D	mAh	R	RW
FullAvailableCapacity()	0x0E and 0x0F	mAh	R	RW
RemainingCapacity()	0x10 and 0x11	mAh	R	RW
FullChargeCapacity()	0x12 and 0x13	mAh	R	RW
AverageCurrent()	0x14 and 0x15	mA	R	RW
InternalTemperature()	0x16 and 0x17	0.1 K	R	RW
ResScale()	0x18 and 0x19	num	R	RW
ChargingLevel()	0x1A and 0x1B	num	R	RW
StateOfHealth()	0x1C and 0x1D	% / num	R	RW
CycleCount()	0x1E and 0x1F	Counter	R	R
StateOfCharge()	0x20 and 0x21	%	R	R
InstantaneousCurrentReading()	0x22 and 0x23	mA	R	RW
FineQPass()	0x24 and 0x25	mAh	R	RW
FineQPassFract()	0x26 and 0x27	num	R	RW
ProgChargingCurrent()	0x28 and 0x29	mA	R	R
ProgChargingVoltage()	0x2A and 0x2B	mV	R	R
LevelTaperCurrent()	0x2C and 0x2D	mA	R	RW
CalcChargingCurrent()	0x2E and 0x2F	mA	R	RW
CalcChargingVoltage()	0x30 and 0x31	mV	R	RW
ChargerStatus()	0x32	hex	R	RW
ChrgrReg0()	0x33	hex	RW	RW
ChrgrReg1()	0x34	hex	RW	RW
ChrgrReg2()	0x35	hex	RW	RW
ChrgrReg3()	0x36	hex	RW	RW
ChrgrReg4()	0x37	hex	RW	RW

Table 2-1. Standard Commands (continued)

Name	Command Code	Unit	SEALED Access	UNSEALED Access
ChrgrReg5()	0x38	hex	RW	RW
ChrgrReg6()	0x39	hex	RW	RW
RemainingChargeCapacityUnfiltered()	0x6C and 0x6D	mA	R	RW
RemainingChargeCapacityFiltered()	0x6E and 0x6F	mA	R	RW
FullChargeCapacityUnfiltered()	0x70 and 0x71	mAh	R	RW
FullChargeCapacityFiltered()	0x72 and 0x73	mAh	R	RW
TrueSOC()	0x74 and 0x75	%	R	RW
MaxCurrent()	0x76 and 0x77	mA	R	RW

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

Control Function	Control Data	SEALED Access	Description
CONTROL_STATUS	0x0000	Yes	Reports the status of HIBERNATE, Impedance Track™, etc.
DEVICE_TYPE	0x0001	Yes	Reports the device type, example: 0x0532 for bq27532-G1
FW_VERSION	0x0002	Yes	Reports the firmware version on the device type
HW_VERSION	0x0003	Yes	Reports the hardware version of the device type
MLC_ENABLE	0x0004	Yes	Charge profile is based on Multi-Level Charge profile
MLC_DISABLE	0x0005	Yes	Charge profile is solely based on charge temperature tables and, if enabled, State Of Health
CLEAR_IMAX_INT	0x0006	Yes	Clears the IMAX status bit and the interrupt signal from SOC_INT pin.
PREV_MACWRITE	0x0007	Yes	Returns previous MAC subcommand code
CHEM_ID	0x0008	Yes	Reports the chemical identifier of the Impedance Track™ configuration
BOARD_OFFSET	0x0009	No	Forces the device to measure and store the board offset
CC_OFFSET	0x000A	No	Forces the device to measure the internal CC offset
CC_OFFSET_SAVE	0x000B	No	Forces the device to store the internal CC offset
OCV_CMD	0x000C	Yes	Requests the gauge to take an OCV measurement
BAT_INSERT	0x000D	Yes	Forces the *Flags() [BAT_DET]* bit to set when the ***OpConfig B [BIE]*** bit is 0.
BAT_REMOVE	0x000E	Yes	Forces the *Flags() [BAT_DET]* bit to clear when the ***OpConfig B [BIE]*** bit is 0.
BYPASS_ENABLE	0x000F	Yes	Forces bypass mode to charger. The *ControlStatus() [BYPASS]* bit is 1.
BYPASS_DISABLE	0x0010	Yes	Disables bypass mode to charger. The *ControlStatus() [BYPASS]* bit is 0.
SET_HIBERNATE	0x0011	Yes	Forces the *CONTROL_STATUS [HIBERNATE]* bit to 1
CLEAR_HIBERNATE	0x0012	Yes	Forces the *CONTROL_STATUS [HIBERNATE]* bit to 0
SET_SLEEP+	0x0013	Yes	Forces the *CONTROL_STATUS [SNOOZE]* bit to 1
CLEAR_SLEEP+	0x0014	Yes	Forces the *CONTROL_STATUS [SNOOZE]* bit to 0
ILIMIT_LOOP_ENABLE	0x0015	Yes	When gauge is not connected to charger through I^2C, this command indicates to gauge that there is charger input current limiting loop active. Disables charge termination detection by gauge. The *Flags() [ILIMITED]* bit is 1.
ILIMIT_LOOP_DISABLE	0x0016	Yes	When gauge is not connected to charger through I^2C, this command indicates to gauge that battery charge current is not limited. Allows charge termination detection by gauge. The *Flags() [ILIMITED]* bit is 0.

Table 2-2. *Control()* Subcommands (continued)

Control Function	Control Data	SEALED Access	Description
SHIPMODE_ENABLE	0x0017	Yes	Commands the bq2425x to turn off BATFET[(1)] after a delay time programmed in data flash so that the system load does not draw power from the battery. The *ChargerStatus() [SHIPMODE]* bit is 1.
SHIPMODE_DISABLE	0x0018	Yes	Commands the bq2425x to disregard turning off BATFET before the delay time or commands BATFET to turn on if VIN had power during the SHIPMODE enabling process. The *ChargerStatus() [SHIPMODE]* bit is 0.
CHG_ENABLE	0x001A	Yes	Enable charger. Charge continues as dictated by the gauge charging algorithm. The *Flags() [CHG_DIS]* bit is 0.
CHG_DISABLE	0x001B	Yes	Disable charger (sets the $\overline{CE}$ bit of the bq2425x charger). The *Flags() [CHG_DIS]* bit is 1.
GG_CHGRCTL_ENABLE	0x001C	Yes	Enables the gas gauge to control the charger while continuously resetting the charger watchdog. The *Flags() [FGCHGCTL]* bit is 1.
GG_CHGRCTL_DISABLE	0x001D	Yes	The gas gauge stops resetting the charger watchdog. The *Flags() [FGCHGCTL]* bit is 0.
SMOOTH_SYNC	0x001E	Yes	Synchronizes *RemainingCapacityFiltered()* and *FullChargeCapacityFiltered()* with *RemainingCapacityUnfiltered()* and *FullChargeCapacityUnfiltered()*.
DF_VERSION	0x001F	Yes	Returns the ***Data Flash Version***
SEALED	0x0020	No	Places device in SEALED access mode
IT_ENABLE	0x0021	No	Enables the Impedance Track™ algorithm
RESET	0x0041	No	Forces a full reset of the fuel gauge

(1) BATFET (Q4) is internal to the charger (bq2425x).

2.1.1 CONTROL_STATUS: 0x0000

Instructs the fuel gauge to return the 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	BYPASS	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.

BYPASS = Status bit indicating that BYPASS is enabled. Active when set.

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

BCA = Status bit indicating the 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 the battery. True when set.

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

Low Byte

INITCOMP = Initialization completion bit indicating the initialization completed. This bit can only be set with the presence of the battery. True when set.

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

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

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

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

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

VOK = Status bit indicating the voltages are valid for Qmax. True when set.

QEN = Status bit indicating the Qmax updates 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 bq27532-G1 device type returns 0x0532.

2.1.3 FW_VERSION: 0x0002

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

2.1.4 HW_VERSION: 0x0003

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

2.1.5 MLC_ENABLE: 0x0004

Instructs the fuel gauge to base the charging profile on the Multi-Level Charging profile.

2.1.6 MLC_DISABLE: 0x0005

Instructs the fuel gauge to not use the Multi-Level Charging profile.

2.1.7 CLEAR_IMAX_INT: 0x0006

Clears the IMAX status bit and the interrupt signal from SOC_INT pin due to IMAX update.

2.1.8 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 0x001F. For subcommand codes greater than 0x001F, a value of 0x0007 is returned.

2.1.9 CHEM_ID: 0x0008

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

2.1.10 BOARD_OFFSET: 0x0009

Instructs the fuel gauge to compute the coulomb counter offset with an internal short and then without an internal short applied across the sensing resistor (SR) inputs. The difference between the two measurements is the board offset. After a delay of approximately 32 seconds, this offset value is returned to addresses 0x00 and 0x01 and written to data flash. The *CONTROL_STATUS [BCA]* bit is also set. The user must prevent any charge or discharge current from flowing during the process. This function is only available when the fuel gauge is UNSEALED. When SEALED, this command only reads back the board-offset value stored in data flash.

2.1.11 CC_OFFSET: 0x000A

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

2.1.12 CC_OFFSET_SAVE: 0x000B

Instructs the fuel gauge to save the coulomb counter calibration offset to data flash after calibration.

2.1.13 OCV_CMD: 0x000C

This command requests the gauge to take an OCV reading. This command can only be issued after the *CONTROL_STATUS [INITCOMP]* bit has been set, indicating the initialization has been completed. The OCV measurement takes place at the beginning of the next repeated 1-second firmware synchronization clock. During the same time period, the SOC_INT pulses. The host uses this signal to reduce the load current below the C/20 rate in 8 ms for a valid OCV reading.

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

2.1.14 BAT_INSERT: 0x000D

This command sets the *Flags() [BAT_DET]* bit when the battery insertion detection is disabled. When the ***OpConfig B [BIE]*** bit is 0, the battery insertion detection is disabled. The host informs the gauge of the presence of a battery with this command to set the *[BAT_DET]* bit.

2.1.15 BAT_REMOVE: 0x000E

This command clears the *Flags() [BAT_DET]* bit when the battery insertion detection is disabled. When the ***OpConfig B [BIE]*** bit is 0, the battery insertion detection is disabled. The host informs the gauge of the removal of a battery with this command to clear the *[BAT_DET]* bit.

2.1.16 BYPASS_ENABLE: 0x000F

This command ensures that the I^2C operation at BSCL and BSDA pins are in bypass mode. When bypass mode is enabled, all read or write communications from host directed to register addresses 0x33 through 0x39 of the gauge are passed through to register addresses 0x00 through 0x06 of bq2425x charger.

2.1.17 BYPASS_DISABLE

This command disables bypass mode. When bypass mode is disabled, all read or write communications from host directed to register addresses 0x33 through 0x39 of the gauge are masked before interacting with addresses 0x00 through 0x06 of bq2425x charger. When bypass mode is disabled writes to *[VBREG_x]* and *[ICHG_x]* bits are not allowed.

2.1.18 SET_HIBERNATE: 0x0011

Instructs the fuel gauge to set the *CONTROL_STATUS [HIBERNATE]* bit to 1. This allows the gauge to enter the HIBERNATE mode after the transition to SLEEP power state is detected. The *[HIBERNATE]* bit is automatically cleared upon exiting from the HIBERNATE mode.

2.1.19 CLEAR_HIBERNATE: 0x0012

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

2.1.20 SET_SLEEP+ Mode: 0x0013

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

2.1.21 CLEAR_SLEEP+ Mode: 0x0014

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

2.1.22 ILIMIT_LOOP_ENABLE: 0x0015

Indicates to the gauge that the charger is operating with a current limiting loop active. This prevents the gauge from detecting a false charge termination that may be due to low charge currents (that is, charging from USB port at 100 mA). This command should only be used when ***[IND_CHGCTL]*** bit of ***Charger Options*** register is set.

2.1.23 ILIMIT_LOOP_DISABLE: 0x0016

Indicates to the gauge that the charger is operating with full current capability (current limiting loops inactive). This permits proper charge termination when charge current reaches ***Taper Current*** threshold. This command should only be used when ***[IND_CHGCTL]*** bit of ***Charger Options*** register is set.

2.1.24 SHIPMODE_ENABLE: 0x0017

Commands the charger to turn off BATFET after a delay time programmed in data flash so that the system load does not draw power from the battery.

2.1.25 SHIPMODE_DISABLE: 0x0018

Commands the charger to disregard turning off BATFET before a delay time or commands BATFET to turn on if VBUS had power during the SHIPMODE enabling process.

2.1.26 CHG_ENABLE: 0x001A

Instructs the fuel gauge to continue operating according to the charging algorithm by releasing the forced logic high condition of the $\overline{CE}$ bit of the charger after a *CHG_DISABLE* command had been given.

2.1.27 CHG_DISABLE: 0x001B

Instructs the fuel gauge to disable charging by forcing $\overline{CE}$ bit of the charger to logic high.

2.1.28 GG_CHGRCTL_ENABLE: 0x001C

This command is part of the process for the system processor to identify if an input source to the charger is valid and that the fuel gauge is allowed to control the charger in host mode. This command is functional only when the ***Charger Options [CMD_NOT_REQ]*** bit in data flash is cleared. When ***[IND_CHGCTL]*** bit of ***Charger Options*** register is set, this command must be used to indicate to the gauge that the charger IC has a valid input and that it is ready to operate in charge mode.

2.1.29 GG_CHGRCTL_DISABLE: 0x001D

This command disables the fuel gauge from controlling the charger in host mode. The charger then operates in default mode. This command is functional only when the ***Charger Options [CMD_NOT_REQ]*** bit in data flash is cleared. When ***[IND_CHGCTL]*** bit of ***Charger Options*** register is set, this command must be used to indicate to the gauge that the charger IC does not have a valid input and that it is not ready to operate in charge mode.

2.1.30 SMOOTH_SYNC: 0x001E

Synchronizes *RemainingCapacityFiltered()* and *FullChargeCapacityFiltered()* with *RemainingCapacityUnfiltered()* and *FullChargeCapacityFiltered()*.

2.1.31 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.

2.1.32 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.

2.1.33 IT_ENABLE: 0x0021

This command forces the fuel gauge to begin the Impedance Track™ algorithm, sets the ***IT Enable*** to 0x01, and causes the *CONTROL_STATUS [VOK]* and *[QEN]* flags to be set. *[VOK]* is cleared if the voltages are not suitable for a Qmax update. This command is only available when the fuel gauge is UNSEALED.

2.1.34 RESET: 0x0041

This command instructs the fuel gauge to perform a full reset. This command 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 NORMAL mode.

2.3 *AtRateTimeToEmpty()*: 0x04 and 0x05

If the battery is discharged at the *AtRate()* value, then this read-word function returns an unsigned integer value of the predicted remaining operating time 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 1 second. Both the *AtRate()* and *AtRateTimeToEmpty()* commands must only be used in 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 as measured by the fuel gauge. If the ***OpConfig B [WRTEMP]*** bit = 1, a write command sets the temperature to be used for gauging calculations while a read command returns to temperature previously written. If the ***[WRTEMP]*** bit = 0 and the ***Op Config [TEMPS]*** bit = 0, a read command returns the internal temperature sensor value.

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 2-4. *Flags()* Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	–	–	FGCHGCTL	CALMODE	–	CHG_DIS	FC	CHG
Low Byte	ILIMITED	IMAX_INT	OCV_GD	MLC	BAT_DET	SOC1	–	DSG

High Byte

FGCHGCTL = Indicates that the gauge is in charge control mode.

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_DIS = Status of CHARGE_DISABLE and CHARGE_ENABLE commands. Charge is disabled when set.

FC = Full-charged condition reached. Set when charge termination condition is met. True when set.

CHG = The gauge is enabling charge by clearing $\overline{CE}$. True when set.

Low Byte

ILIMITED = Charger input current limit loop detected. Can be set by gauge reading non zero values from LOOP_STATUSx bits of bq2425x charger or the host sending the ILIMIT_LOOP_EN command. True when set.

IMAX_INT = Set when new IMAX value changes by more than ***Max Current Interrupt Step*** from the last value that was recorded at last SOC_INT interrupt.

OCV_GD = Good OCV measurement taken. True when set.

MLC = Indicates that Multi-Level Charge algorithm is enabled. True when set.

BAT_DET = Battery detected. True when set.

SOC1 = State-of-charge threshold 1 (***SOC1 Set Threshold***) reached. The flag is enabled when the ***OpConfig B [BL_INT]*** bit is set. True when 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) remaining battery capacity. 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 IT algorithm.

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

When the ***OpConfig C [SmoothEn]*** bit is cleared, this read-only command pair returns the compensated remaining battery capacity (*RemainingCapacityUnfiltered()*). When the ***[SmoothEn]*** bit is set, this command pair returns the filtered compensated battery capacity remaining (*RemainingCapacityFiltered()*). Units are mAh.

2.10 *FullChargeCapacity()*: 0x12 and 0x13

When the ***OpConfig C [SmoothEn]*** bit is cleared, this read-only command pair returns the compensated capacity of a fully charged battery (*FullChargeCapacityUnfiltered()*). When the ***[SmoothEn]*** bit is set, this command pair returns the filtered compensated capacity of a fully charged battery (*FullChargeCapacityFiltered()*) . Units are mAh. *FullChargeCapacity()* is updated at regular intervals, as specified by the IT 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. It is negative during discharge and positive during charge. It is updated every 1 second. Units are mA.

2.12 *InternalTemperature()*: 0x16 and 0x17

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

2.13 *ResScale*: 0x18 and 0x19

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 6.2.14, Fast Resistance Scaling.)

2.14 *ChargingLevel():* 0x1A and 0x1B

This read-word function returns the level of the bq2425x charging while using the gauge MLC algorithm.

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 that indicates the status of the SOH percentage. The meanings of the returned values are:

- 0x00: SOH not valid (initialization)
- 0x01: Instant SOH value ready
- 0x02: Initial SOH value ready
  - Calculation based on uncompensated Qmax
  - Updated at first grid point update after cell insertion
- 0x03: SOH value ready
  - Utilize the updated Qmax update
  - Calculation based on compensated Qmax
  - Updated after complete charge and relax is complete
- 0x04 to 0xFF: Reserved

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

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

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

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

2.18 *InstantaneousCurrentReading():* 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 1 second. Units are mA.

2.19 *FineQPass()*: 0x24 and 0x25

This read-only function returns an unsigned integer value of the amount of passed charge in the charging or discharging direction. The registers increment when the coulomb counter detects charge current and decrement when detecting discharge current. These registers cannot be cleared and continue to count beyond 0xFFFF and start at 0x0000 again. Units are mAh.

2.20 *FineQPassFract()*: 0x26 and 0x27

This read-only function returns an unsigned integer value of the amount of passed charge in the charging or discharging direction. The registers increment when the coulomb counter detects charge current and decrement when detecting discharge current. These registers cannot be cleared and continue to count beyond 0xFFFF and start at 0x0000 again. For every time that these registers count past 0xFFFF, there is an increment by 1 of *FineQPass()*. Units are num.

2.21 *ProgChargingCurrent()*: 0x28 and 0x29

This read-only function returns an unsigned integer value of the charging current that is programmed to the charger. This value is dependent on the offset (512 mA) and resolution (64 mA) of the programmable charge current of the charger and the *CalcChargingCurrent()* obtained through the fuel gauge charging algorithm.

2.22 *ProgChargingVoltage()*: 0x2A and 0x2B

This read-only function returns an unsigned integer value of the charging voltage that is programmed to the charger. This value is dependent on the offset (3504 mV) and resolution (16 mV) of the programmable charge voltage of the charger and the *CalcChargingVoltage()* obtained through the fuel gauge charging algorithm.

2.23 *LevelTaperCurrent():* 0x2C and 0x2D

This read-only command pair returns a signed integer value that indicates the target taper current for the present level of the MLC algorithm. This current is one of the requirements for the algorithm to change charging level. Units are mA.

2.24 *CalcChargingCurrent():* 0x2E and 0x2F

This read-only command pair returns a signed integer value that indicates the recommended charging current to be used by the charger. This value can vary with the configured charging algorithm. Units are mA.

2.25 *CalcChargingVoltage():* 0x30 and 0x31

This read-word function returns an unsigned integer value that indicates the recommended charging voltage to be used by the charger. This value can vary with the configured charging algorithm. Units are mV.

2.26 *InternalTemperature():* 0x32 and 0x33

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

2.27 *RemainingCapacityUnfiltered()*: 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.

2.28 *RemainingCapacityFiltered()*: 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.

2.29 *FullChargeCapacityUnfiltered()*: 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. *FullChargeCapacityUnfiltered( )* is updated at regular intervals, as specified by the Impedance Track™ algorithm.

2.30 *FullChargeCapacityFiltered()*: 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. *FullChargeCapacityFiltered( )* is updated at regular intervals, as specified by the Impedance Track™ algorithm.

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

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

2.32 *MaxCurrent()*: 0x76 and 0x77

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

Charger Interface

The bq27532-G1 fuel gauge can control any of the devices belonging to the bq2425x charger family. Which charger IC is used in the design depends on what detection scheme is used for input current limit. Table 3-1 summarizes the differences between the bq2425x devices. The charger default values are based on external resistors at charger IC and used only when the gauge is not in the charger-control mode and the charger is operating in the default mode.

Table 3-1. bq2425x Family Table

	bq24250	bq24251	bq24253[1]
I^2C Address	0x6A	0x6A	NA
USB Detection	EN1/EN2	D+/D–	D+/D– and EN1/EN2
Charging Temperature Profile	JEITA	JEITA	JEITA
Status Output	INT	$\overline{PG}$	$\overline{PG}$

(1) The bq24253 charger should not be selected given that it does not have I^2C control and it operates in standalone mode.

3.1 Charger Data Commands

The charger registers are mapped to a series of single-byte, charger data commands to enable system reading and writing of battery charger registers. The registers are powered up based on the flow chart in Figure 3-1.

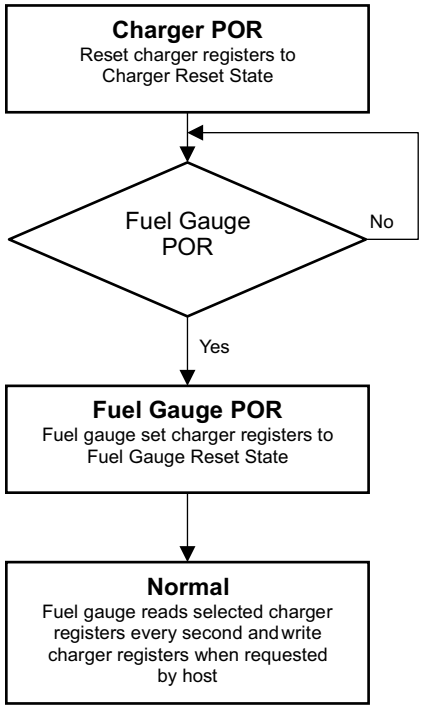


Figure 3-1. Charger Power-up

The fuel gauge can change the values of these registers during the system reset state. Three classes of system reset states are available:

- CHG (Charger) State: The fuel gauge retains the register value based on the charger reset state.
- DF (Data Flash) State: The fuel gauge uses the shadow data flash values as described in Section 5.5, *Data Flash Summary*, to determine the system reset state of these registers.
- FG (Fuel Gauge) State: The fuel gauge uses an embedded charging algorithm to determine the Fuel Gauge Reset State of these registers. The charging algorithm can be configured using the data flash as described in Section 5.5, *Data Flash Summary*.

Each bit in the Charger Data Commands can be read-write or read-only as defined by system access. It is important to note that system access can be different from the read-write access as defined in the charger hardware. The fuel gauge may block write access to the charger hardware when the bit function is controlled exclusively by the fuel gauge. For example, the *[VBREGx]* bits of *Chrgr_Reg2()* are controlled by the fuel gauge and cannot be modified by the system. The fuel gauge reads the corresponding registers of *Chrgr_Reg0()*, *Chrgr_Reg2()*, and *Chrgr_Reg4()* every second to mirror the charger status, input current limit detection and input current limit loop status. Other registers in the bq2425x charger are read when the registers are modified by the fuel gauge.

Table 3-2. Charger Data Commands

Name		Command Code	bq2425x Charger Memory Location	SEALED Access	UNSEALED Access	Refresh Rate
ChargerStatus()	CHGRSTAT	0x32	NA	R	R	Every second
Chrgr_Reg0()	CHGR0	0x33	0x00	R	R	Every second
Chrgr_Reg1()	CHGR1	0x34	0x01	RW	RW	Data Change
Chrgr_Reg2()	CHGR2	0x35	0x02	RW	RW	Every second
Chrgr_Reg3()	CHGR3	0x36	0x03	RW	RW	Data Change
Chrgr_Reg4()	CHGR4	0x37	0x04	RW	RW	Every second
Chrgr_Reg5()	CHGR5	0x38	0x05	RW	RW	Data Change
Chrgr_Reg6()	CHGR6	0x39	0x06	RW	RW	Data Change

3.1.1 *ChargerStatus(): 0x32*

This register provides the status of direct activity between the fuel gauge and the charger.

Table 3-3. *ChargerStatus()* Bit Definitions

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
WAIT_CMD	ERR	RSVD	WFAIL	RSVD	PROFILE	INIT	SHIPMODE

WAIT_CMD = Indicates that the gauge is waiting on *GG_CHGCNTRL_ENABLE* subcommand so that the gauge can control the charger in host mode. True when set.

ERR = Indicates I^2C communication error between gauge and charger. True when set.

RSVD = Bits 3 and 5 are reserved.

WFAIL = Indicates that a write intended to the charger has failed. True when set.

PROFILE = Indicates that charger profile has been updated or refreshed.

INIT = Indicates when the gauge is in the initializing process of the charge state machine.

SHIPMODE = Indicates that the *SHIPMODE_ENABLE* subcommand is active. *Chrgr_OpCtrl_Reg7() [BATFET_DIS]* is set after ***Shipmode Delay*** expires.

3.1.2 Chrgr_Reg0(): 0x33

This function returns the hex value corresponding to the charger register for status and faults.

Table 3-4. *Chrgr_Reg0()* Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	WD_FAULT	WD_EN	STAT_1	STAT_0	FAULT_3	FAULT_2	FAULT_1	FAULT_0
Charger Reset State	X	0/1[1]	X	X	X	X	X	X
Fuel Gauge Reset State	CHG	CHG	CHG	CHG	CHG	CHG	CHG	CHG
Fuel Gauge Access	R	RW	R	R	R	R	R	R

(1) The default is dependent on the specific charger.

WD_FAULT = 0 = No watchdog fault has occurred.
1 = Watchdog timeout has occurred.

WD_EN = 0 = Disables watchdog timer.
1 = Enables and resets the watchdog timer.

STAT[1:0] = Indicates the charger controller status
00 = Charger ready
01 = Charge in progress
10 = Charge done
11 = Charger fault

FAULT[3:0] = Indicates the faults that have occurred
0000 = Normal
0001 = Input OVP
0010 = Input UVLO
0011 = Sleep
0100 = Battery Temperature (TS) Fault
0101 = Battery OVP
0110 = Thermal Shutdown
0111 = Timer Fault
1000 = No Battery Connected
1001 = ISET Short
1010 = Input Fault and LDO Low
1011 to 1111 = Reserved

3.1.3 Chrgr_Reg1(): 0x34

This function returns the hex value corresponding to the charger register for input current limit and charger control bits.

Table 3-5. *Chrgr_Reg1()* Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RESET	IINLIMIT_2	IINLIMIT_1	IINLIMIT_0	EN_STAT	EN_TERM	$\overline{CE}$	HZ_MODE
Charger Reset State	X	X	X	X	1	1	0	0
Fuel Gauge Reset State	CHG	DF	DF	DF	DF	0	FG	DF
Fuel Gauge Access	RW	RW	RW	RW	RW	RW	RW	RW

RESET = 0 = No effect.
1 = Reset all charger registers to charger default values.

WD_EN = 0 = Disables watchdog timer.
1 = Enables and resets the watchdog timer.

IINLIMIT[2:0] = Sets the input current limit
000 = USB2.0 host with 100-mA current limit
001 = USB3.0 host with 150-mA current limit
010 = USB2.0 host with 500-mA current limit
011 = USB3.0 host with 900-mA current limit
100 = Charger with 1500-mA current limit
101 = Charger with 2000-mA current limit
110 = External ILIM current limit
111 = No input current limit with internal clamp at 3 A (PTM Mode)

EN_STAT = 0 = Disable STAT function
1 = Enable STAT function

EN_TERM = 0 = Disable charge termination
1 = Enable charge termination

$\overline{CE}$ = 0 = Enable charge
1 = Disable charge

HZ_MODE = 0 = Not in high impedance mode
1 = High impedance mode

3.1.4 Chrgr_Reg2(): 0x35

This function returns the hex value corresponding to the charger register for voltage regulation and input current limit detection.

Table 3-6. *Chrgr_Reg2()* Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	VBREG_5	VBREG_4	VBREG_3	VBREG_2	VBREG_1	VBREG_0	USBDET_EN_1	USBDET_EN_2
Charger Reset State	1	0	0	0	1	1	1	1
Fuel Gauge Reset State	FG	FG	FG	FG	FG	FG	CHG	CHG
Fuel Gauge Access	RW	RW	RW	RW	RW	RW	R	R

VBREG[5:0] = Sets the battery regulation voltage
VBREG_5 = Charge Voltage: 640 mV
VBREG_4 = Charge Voltage: 320 mV
VBREG_3 = Charge Voltage: 160 mV
VBREG_2 = Charge Voltage: 80 mV
VBREG_1 = Charge Voltage: 40 mV
VBREG_0 = Charge Voltage: 20 mV

USBDET_EN[1:0] = Provides the status of the D+/D– detection results for devices that include D+/D– pins or the status of EN1/EN0 pins on devices that don't have D+/D–
00 = DCP detected
01 = CDP detected
10 = SDP detected
11 = Apple/TT or non-standard adaptor detected

NOTE: The charger voltage is controlled by the gauge based on the charging algorithm.
Offset = 3500 mV; Steps of 20 mV; Range = 3500 to 4440 mV

3.1.5 Chrgr_Reg3(): 0x36

This function returns the hex value corresponding to the charger register for current regulation and termination current.

Table 3-7. *Chrgr_Reg3()* Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	ICHG_4	ICHG_3	ICHG_2	ICHG_1	ICHG_0	ITERM_2	ITERM_1	ITERM_0
Charger Reset State	1	1	1	1	1	0	0	0
Fuel Gauge Reset State	FG	FG	FG	FG	FG	0	0	0
Fuel Gauge Access	RW	RW	RW	RW	RW	RW	RW	RW

ICHG[4:0] = Sets the charge current regulation
ICHG_4 = Charge Current: 800 mA
ICHG_3 = Charge Current: 400 mA
ICHG_2 = Charge Current: 200 mA
ICHG_1 = Charge Current: 100 mA
ICHG_0 = Charge Current: 50 mA

ITERM[2:0] = Termination current sense threshold
ITERM_2 = Threshold = 100 mA
ITERM_1 = Threshold = 50 mA
ITERM_0 = Threshold = 25 mA

NOTE: The charger current is controlled by the gauge based on the charging algorithm. When charger operates in default mode, the charge current is based on external ISET configuration for charger.
Offset = 500 mA; Steps of 50 mA; Range = 500 to 2000 mA

3.1.6 Chrgr_Reg4(): 0x37

This function returns the hex value corresponding to the charger register for loop status, VINDPM and low charge control.

Table 3-8. *Chrgr_Reg4()* Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	LOOPSTAT_1	LOOPSTAT_0	LOW_CHG	DPDM_EN	CE_STATUS	VINDPM_2	VINDPM_1	VINDPM_0
Charger Reset State	X	X	0	0	X	0	1	0
Fuel Gauge Reset State	CHG	CHG	0	0	CHG	DF	DF	DF
Fuel Gauge Access	R	R	RW	RW	R	RW	RW	RW

LOOPSTAT[1:0] = Provides status of active regulation loop
00 = No loop is active
01 = VINDPM regulation loop is active
10 = Input current limit loop is active
11 = Thermal regulation loop is active

LOW_CHG = 0 = Normal charge current set by Register 0x03 of charger
1 = Low charge current setting of 330 mA

DPDM_EN = 0 = Bit returns to 0 after D+/D– detection is performed
1 = Force D+/D– detection

CE_STATUS = 0 = CE low
1 = CE high

VINDPM[2:0] = Sets the input VINDPM level
VINDPM[2] = Input VINDPM voltage: 320 mV
VINDPM[1] = Input VINDPM voltage: 160 mV
VINDPM[0] = Input VINDPM voltage: 80 mV

3.1.7 Chrgr_Reg5(): 0x38

This function returns the hex value corresponding to the charger register for TS status, safety timer setting, and SYSOFF setting.

Table 3-9. *Chrgr_Reg5()* Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	2XTMR_EN	TMR_1	TMR_0	SYSOFF	TS_EN	TS_STAT_2	TS_STAT_1	TS_STAT_0
Charger Reset State	1	0	1	X	1	X	X	X
Fuel Gauge Reset State	DF	DF	DF	CHG	0	CHG	HCG	CHG
Fuel Gauge Access	RW	RW	RW	RW	RW	R	R	R

2XTMR_EN = 0 = Timer not slowed at any time
1 = Timer slowed by 2× when in thermal regulation, VINDPM, or DPPM

TMR[1:0] = Safety timer time limit
00 = 0.75-hour fast charge
01 = 6-hour fast charge
10 = 9-hour fast charge
11 = Disable safety timers

SYSOFF = 0 = SYSOFF disabled
1 = SYSOFF enabled

TS_EN = 0 = TS function disabled
1 = TS function enabled

TS_STAT[2:0] = TS fault mode
000 = Normal, no TS fault
001 = TS temp > THOT
010 = TWARM < TS Temp < THOT
011 = TCOLD < TS Temp < TCOOL
100 = TS Temp < TCOLD
101 = TFREEZE < TS Temp < TCOLD
110 = TS Temp < TFREEZE
111 = TS open (TS disabled)

3.1.8 Chrgr_Reg6(): 0x39

This function returns the hex value corresponding to the charger register for input OVP.

Table 3-10. *Chrgr_Reg6()* Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	VOVP_2	VOVP_1	VOVP_0	CLR_VDP	FORCE_BATDET	FORCE_PTM	NA	NA
Charger Reset State	1	1	1	0	0	0	0	0
Fuel Gauge Reset State	DF	DF	DF	0	0	0	0	0
Fuel Gauge Access	RW	RW	RW	RW	RW	RW	R	R

VOVP[2:0] = Sets the input OVP level
000 = 6.0 V
001 = 6.5 V
010 = 7.0 V
011 = 8.0 V
100 = 9.0 V
101 = 9.5 V
110 = 10.0 V
111 = 10.5 V

CLR_VDP = 0 = Keep D+ voltage source on during DBP charging
1 = Turn off D+ voltage source to release D+ line

FORCE_BATDET = 0 = Enter the battery detection routine only if TERM is true or FORCE PTM is true
1 = Enter the battery detection routine

FORCE_PTM = 0 = PTM is disabled
1 = PTM is enabled

3.2 Charging Control

3.2.1 Charging Voltage and Current Standard Commands

Several standard commands are provided to monitor the charging process:

- *Voltage()* and *AverageCurrent()* measure the voltage at the BAT pin and the charging or discharging current of the battery through the sense resistor.
- *CalcChargingCurrent()* and *CalcChargingVoltage()* provide the optimal charging parameters based on the charging algorithm configurations in the data flash.
- *ProgChargingCurrent()* and *ProgChargingVoltage()* report the charging parameters that are programmed to the charger. *ProgChargingCurrent()* and *ProgChargingVoltage()* are based on the *CalcChargingCurrent()* and *CalcChargingVoltage()* with the consideration of charging parameters step resolution and offset. For example, when the optimal *CalcChargingVoltage()* is 3.535 V, the *ProgChargingVoltage()* value is shown as 3.520 V (based on 3.500-V offset and 20-mV step resolution). The *ProgChargingVoltage()* value is programmed into the charger.

3.2.2 Charging Voltage and Current Parameters Vary With Temperature()

The *CalcChargingCurrent()* and *CalcChargingVoltage()* can be set as a function of temperature. The function consists of five temperature thresholds and their corresponding charge currents and voltages. The profile is configured and stored in data flash. This allows for something as simple as stopping the charge when temperature is too high, and as complex as meeting the JEITA-like specifications.

When the temperature is between ***Charge_T***$_n$ and ***Charge_T***$_{n+1}$, the associated ***Charge_Current_T***$_n$ and ***Charge_Voltage_T***$_n$ stored in data flash are assigned as Charge_Voltage_Temp and Charge_Current_Temp and are used in the MLC algorithm to determine *CalcChargingCurrent()* and *CalcChargingVoltage()* values.

When temperature is below or equal to ***Charge_T0***, *CalcChargingCurrent()* and *CalcChargingVoltage()* values are set to zero and the $\overline{CE}$ bit is set to disable charging.

The default data flash setting is shown in Table 3-11 and an optional example setting in Charging Profile Example Based on Temperature Range.

NOTE: There is no interpolation within the temperature ranges.

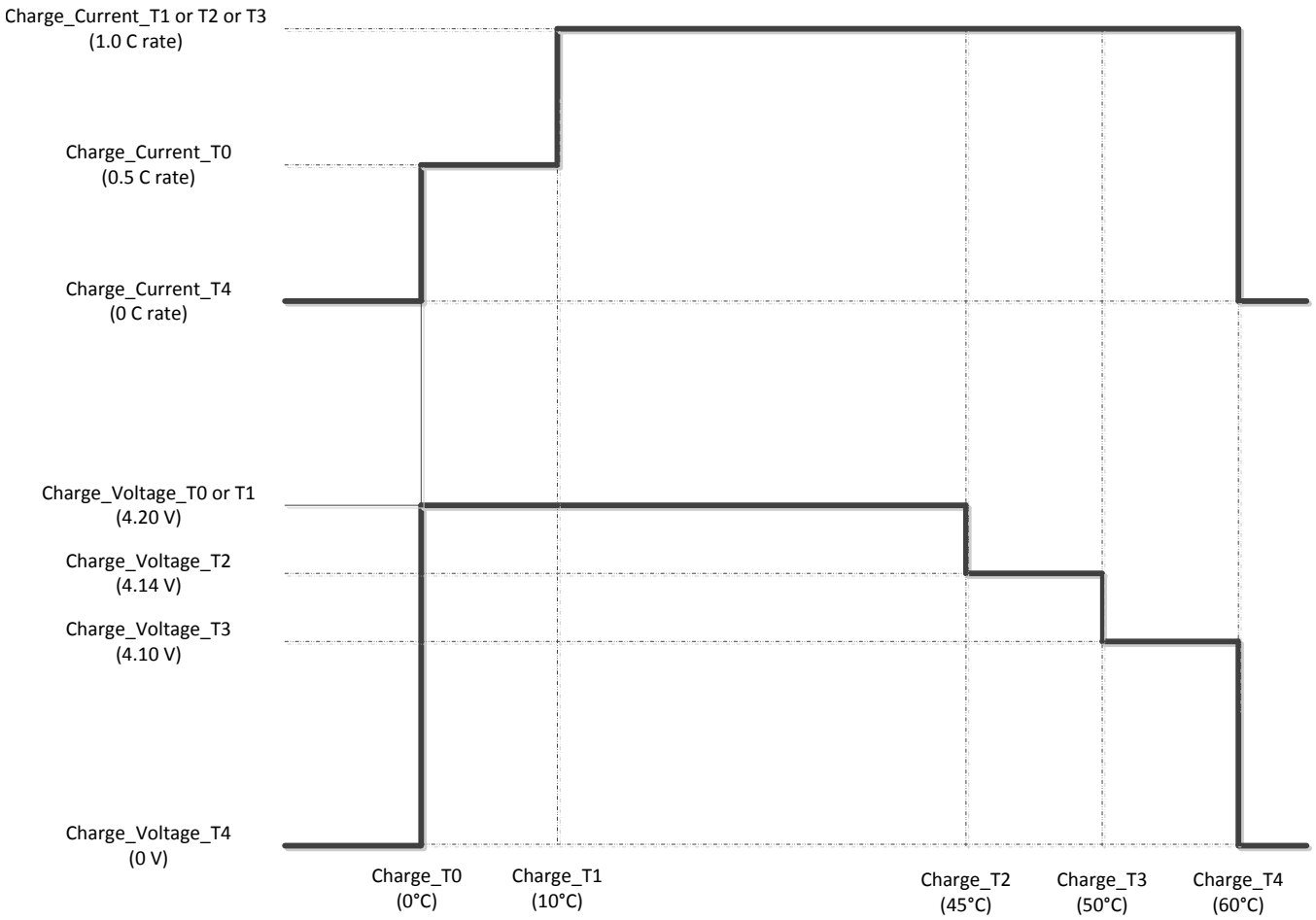
Table 3-11. Default Data Flash Settings for Temperature Charging

Data Flash	Effective Value	Example Setting	Unit
NA	T ≤ Charge_T0	NA	°C
Charge_T0	Charge_T1 ≥ T > Charge_T0	0	°C
Charge_T1	Charge_T2 ≥ T > Charge_T1	10	°C
Charge_T2	Charge_T3 ≥ T > Charge_T2	45	°C
Charge_T3	Charge_T4 ≥ T > Charge_T3	50	°C
Charge_T4	T > Charge_T4	60	°C
Charge_Current_T0	0.5C	50	% of C rate(1)
Charge_Current_T1	0.5C	50	% of C rate(1)
Charge_Current_T2	0.5C	50	% of C rate(1)
Charge_Current_T3	0.5C	50	% of C rate(1)

(1) C rate is the charge current based on ***Design Capacity***.

Table 3-11. Default Data Flash Settings for Temperature Charging (continued)

Data Flash	Effective Value	Example Setting	Unit
Charge_Current_T4	0C	0	% of C rate[(1)]
Charge_Voltage_T0	4200 mV	210	20mV
Charge_Voltage_T1	4200 mV	210	20mV
Charge_Voltage_T2	4140 mV	207	20mV
Charge_Voltage_T3	4100 mV	205	20mV
Charge_Voltage_T4	0 mV	0	20mV



Charging Profile Example Based on Temperature Range

3.2.3 Charging Voltage and Current Parameters Variance With StateofHealth()

The *CalcChargingCurrent()* and *CalcChargingVoltage()* can be set as a function of *StateofHealth()* (SOH) after the SOH status is updated to 3. The profile is configured and stored in the data flash. This allows higher charge voltage and current for new batteries to increase the run time and faster charging. It also allows reduced charge voltage and current as the battery ages to reduce degradation. Charging with SOH dependency is enabled by setting the ***[SOH_EN]*** bit in the ***Charger Options*** data flash register.

The voltage and current values are calculated with interpolation between the SOH threshold settings. The voltage and current values are fixed in the ranges between 0% - ***Charge SOH 0*** and ***Charge SOH 3*** - 100%. The fixed values are based on SOH 0 and SOH 3, respectively.

The default data flash setting is shown in Table 3-12 and an optional example setting in Figure 3-2.

Table 3-12. Sample Data Flash Settings for State of Health Based Charging

Data Flash	Effective Value	Example Setting	Unit
NA	SOH ≤ Charge SOH 0	NA	%
Charge SOH 0	SOH = Charge SOH 0	70	%
Charge SOH 1	SOH = Charge SOH 1	77	%
Charge SOH 2	SOH = Charge SOH 2	85	%
Charge SOH 3	Charge SOH 3 ≤ SOH ≤ 100%	92	%
Charge Current H0	70%	70	% of C rate(1)
Charge Current H1	100%	100	% of C rate(1)
Charge Current H2	100%	100	% of C rate(1)
Charge Current H3	100%	100	% of C rate(1)
Charge Voltage H0	4200 mV	210	20mV
Charge Voltage H1	4200 mV	210	20mV
Charge Voltage H2	4200 mV	210	20mV
Charge Voltage H3	4200 mV	210	20mV

(1) C rate is the charge current based on ***Design Capacity***.

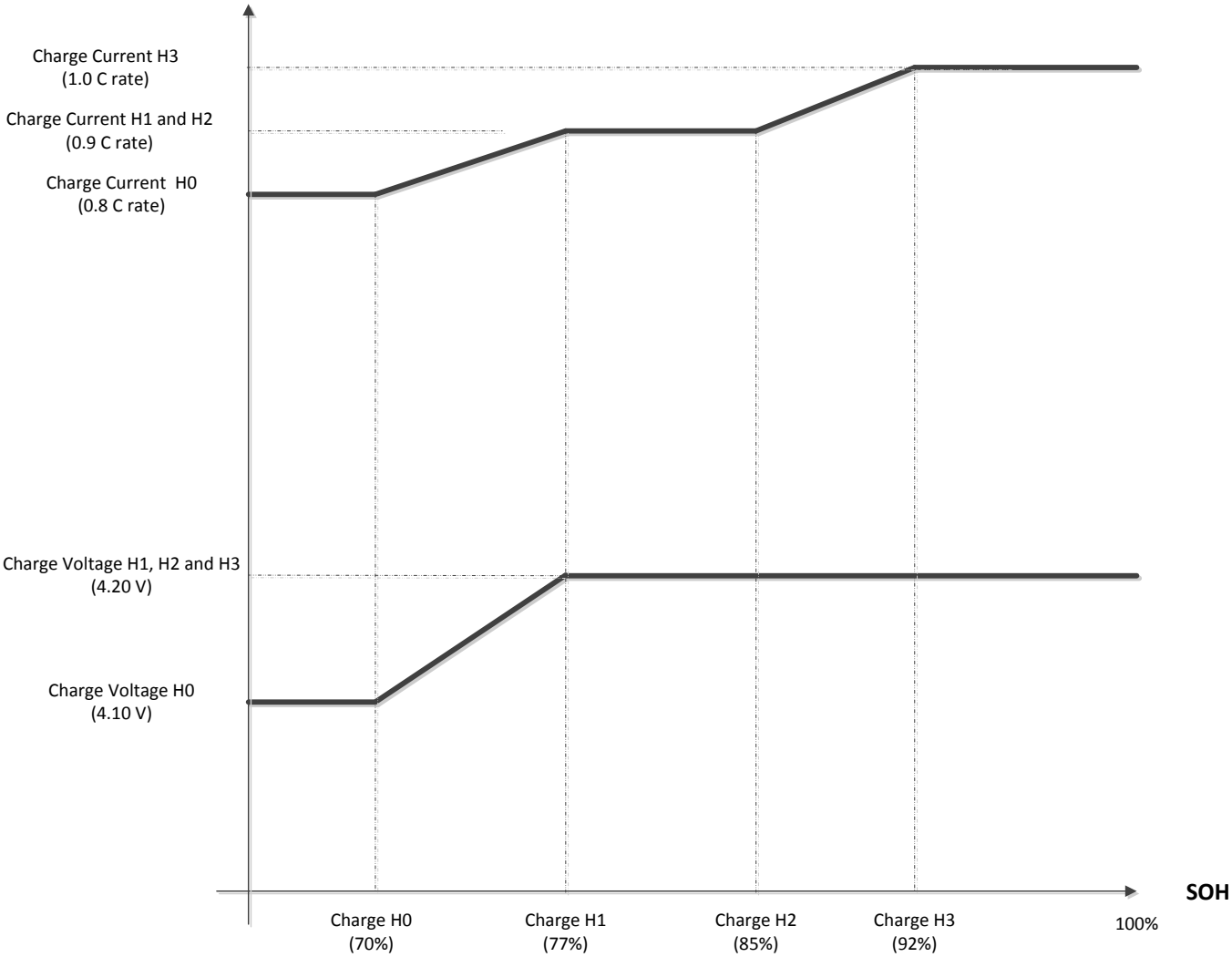


Figure 3-2. Charging Profile Example Based on SOH Range

3.2.4 Multi-Level Charging Algorithm

The *CalcChargingCurrent()* and *CalcChargingVoltage()* are directly controlled by the fuel gauge based on MLC algorithm as a function of *Temperature()* and *StateofHealth()*. Based on the algorithm, the fuel gauge instructs the charger to charge the battery progressively to a higher voltage with lower charging current. Users can choose to charge a battery with higher current at lower voltage for faster charging time, or lower current at higher voltage to reduce degradation. Charging using MLC is enabled by setting the ***[STEP_EN]*** bit in the ***Charger Options*** data flash register.

3.3 Data Flash Settings for Charger Operation

Table 3-13. Charger Options Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	IND_ CHGCTL	USB_IN_DEF	CMD_NOT_ REQ	CHGTRM_ HIZ	STEP_EN	SOH_EN	DEFAULT_ OVRD	BYPASS
Default	0	0	0	1	1	1	0	0
	0x1C							

IND_CHGCTL = Indirect Charge Control is enabled. See Section 3.4, Indirect Charger Control. True when set.

USB_IN_DEF = Sets the default input current limit for when the charger detects a USB-type input using PSEL or D+/D–.
0 = 100 mA
1 = 500 mA

CMD_NOT_REQ = *GG_CHGRCTL_ENABLE* subcommand is not required for the fuel gauge to control the charger and continuously reset the charger watchdog. True when set.

CHGTRM_HIZ = Charge termination Hi-Z. The charger is configured in Hi-Z mode upon a charge termination event. Hi-Z mode is removed once *Flags() [FC]* bit is cleared. True when set.

STEP_EN = Multi-level charge (MLC) enable. True when set.

SOH_EN = Charge profile based on State-of-Health enabled. True when set.

DEFAULT_OVRD = Allows the charger default values stored in the data flash to override the charge algorithm masks within the gauge during initialization. True when set.

BYPASS = Gas gauge bypass enable for charger control over I^2C. The gas gauge relays all reads and writes directed to the charger and does not autonomously reset the charger watchdog. True when set.

3.4 Indirect Charger Control

The fuel gauge can be used in applications that don't have a bq2425x charger in the system. Setting the ***[IND_CHGCTL]*** bit enables Indirect Charger Control mode in which the host must read the suggested charge voltage and current values based on the charging algorithm of the fuel gauge and then write the values accordingly to the particular charger in the system. Set the ***[INT_CHGPROF]*** bit so that SOC_INT is triggered rather than the host needing to poll constantly the voltage and current values. The charge voltage is read from *CalcChargingVoltage()* and the charge current is read from *CalcChargingCurrent()*. Figure 3-3 describes a flow that systems may follow to use the fuel gauge with any charger.

Below is a brief description of some of the states mentioned in Figure 3-3.

- **SYSTEM RESET:** The application system has not been able to determine yet whether a charge source is inserted or not.
- **Charger Source Present State:** Any time that *GG_CHGRCTL_ENABLE* is active.
- **Any Charging State:** Whenever charge occurs. LOOP_STATUS is a generic representation of any kind of charge current limiting loops being active for any given charger. A charge current limiting loop being active should be indicated to the gauge by the host using *ILIMIT_LOOP_ENABLE* command to eliminate chances of detecting a false battery full charge condition.

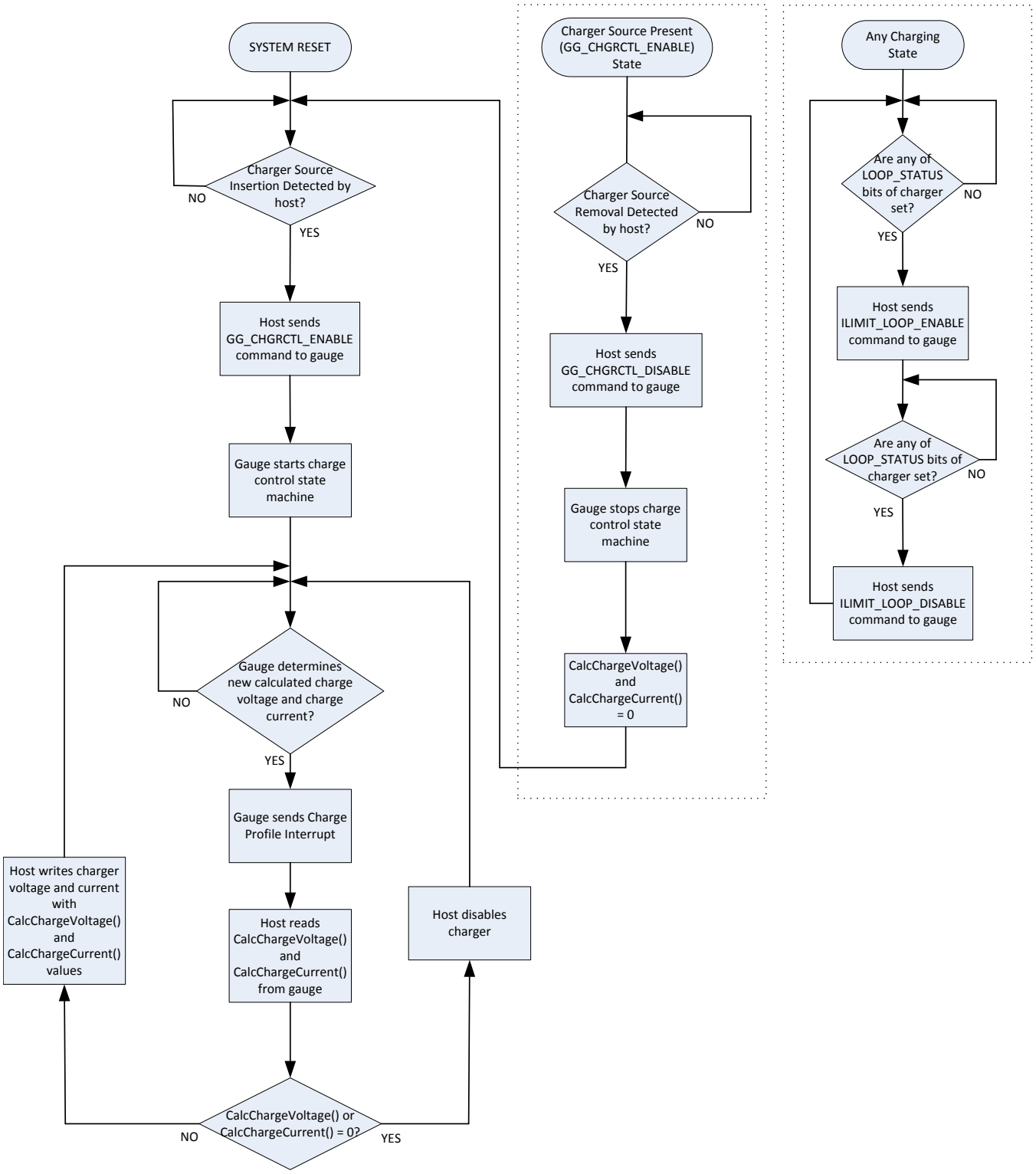


Figure 3-3. Suggested Flow for Indirect Charger Control

Extended Data Commands

Extended commands offer additional functionality beyond the standard set of commands. They are used in the same manner; however, unlike standard commands, extended commands are not limited to 2-byte words. The number of command bytes for a given extended command ranges in size from single to multiple bytes, as specified in Table 4-1.

Table 4-1. Extended Data Commands

Name		Command Code	Unit	SEALED Access[1] [2]	UNSEALED Access[1] [2]
Reserved	RSVD	0x3A to 0x3D	NA	R	R
DataFlashClass()[2]	DFCLS	0x3E	NA	NA	RW
DataFlashBlock()[2]	DFBLK	0x3F	NA	RW	RW
BlockData()	DFD	0x40 to 0x5F	NA	R	RW
BlockDataCheckSum()	DFDCKS	0x60	NA	RW	RW
BlockDataControl()	DFDCNTL	0x61	NA	NA	RW
Reserved	RSVD	0x62 to 0x6C	NA	R	R
Reserved	RSVD	0x78 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.

4.1 *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 the SEALED mode.

4.2 *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 is not available in the SEALED mode.

4.3 *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 command is not available in the SEALED mode.

4.4 *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*], for *x* being the least-significant byte) before being written to 0x60.

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

4.5 *BlockDataControl()*: 0x61

UNSEALED Access: This command controls the data flash access mode. Writing 0x00 to this command enables *BlockData()* to access general data flash. Writing a 0x01 to this command enables the SEALED mode operation of *DataFlashBlock()*.

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

Data Flash Interface

5.1 Accessing The Data Flash

The data flash is a non-volatile memory that contains initialization, default, cell status, calibration, configuration, and user information. The data flash can be accessed in several different ways, depending in which mode the 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 2, *Standard Data Commands*. These commands are available when the fuel gauge is either in the UNSEALED or SEALED mode.

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

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

Occasionally, a data flash class is larger than the 32-byte block size. In this case, the *DataFlashBlock()* command designates in which 32-byte block the desired locations reside. The correct command address is then given by 0x40 + offset *modulo* 32. For example, to access ***Terminate Voltage*** in the *Gas Gauging* class, *DataFlashClass()* is issued 80 (0x50) to set the class. Because the offset is 51, 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 + 51 *modulo* 32 = 0x40 + 19 = 0x40 + 0x13 = 0x53.

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

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

5.2 Manufacturer Information Block

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

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

The data flash location for ***Manufacturer Info Block*** is defined as having a Subclass = 57 and an Offset = 0 through 31 (32-byte block). The specification of Class = System Data is not needed to address Manufacturer Info Block, but is used instead for grouping purposes when viewing data flash info in the evaluation software.

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

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

5.3 Access Modes

The fuel gauge provides three security modes (FULL ACCESS, UNSEALED, and SEALED) that control the data flash access permissions, according to Table 5-1. Data flash refers to those data flash locations, specified in Section 5.5, *Data Flash Summary*, that are accessible to the user.

Table 5-1. Data Flash Access

Security Mode	Data Flash
FULL ACCESS	RW
UNSEALED	RW
SEALED	None

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

5.4 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()* command. The keys must be sent consecutively, with no other data being written to the *Control()* register between the keys.

NOTE: To avoid conflict, the keys must be greater than 0x0082.

The UNSEAL key and FULL-ACCESS key must be different.

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

5.5 Data Flash Summary

Table 5-3 through Table 5-10 summarize the data flash locations available to the user, including their default, minimum, and maximum values.

Table 5-2. Data Type Decoder

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

Table 5-3. Data Flash Summary—Configuration Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
36	Charge Termination	0	Charging Voltage	I2	0	44400	4200	mV
		2	Taper Current	I2	0	1000	100	mA
		6	Taper Voltage	I2	0	1000	100	mV
		9	TCA Set %	I1	-1	100	-1	%
		10	TCA Clear %	I1	-1	100	95	%
		11	FC Clear %	I1	–1	100	98	%
		12	FC Clear Volt	I2	0	32767	0	mV
		14	DODatEOC Delta T	I2	0	1000	50	0.1°C
48	Data	0	CC Threshold	I2	100	32767	900	mAh
		3	Design Capacity	I2	0	32767	1000	mAh
		5	Des Energy Scale	U1	0	255	1	num
		6	SOH LoadI	I2	–32767	0	–400	mA
		8	Default Temperature	I2	2732	3732	2982	0.1°C
		10	Device Name	S8	x	x	bq27532	string
		18	Data Flash Version	H2	0x0000	0xFFFF	0x0000	hex
49	Discharge	0	SOC1 Set Threshold	U2	0	5000	150	mAh
		2	SOC1 Clear Threshold	U2	0	5000	175	mAh
		4	Final Voltage	U2	0	4200	3100	mV
		6	Final Volt Time	U1	0	60	2	s
64	Registers	0	Op Config	H2	0x0000	0xFFFF	0x1971	flg
		2	SOC Delta	U1	0	25	1	%
		3	I^2C Timeout	U1	0	7	4	num
		4	OpConfig B	H1	0x00	0xFF	0x4A	flg
		5	OpConfig C	H1	0x00	0xFF	0x0C	flg
		6	OpConfig D	H1	0	0xFF	0xC3	flg
		7	OpConfig E	H1	0	0xFF	0x73	flg

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

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
68	Power	0	Flash Update OK Voltage	I2	0	4200	2800	mV
		2	Sleep Current	I2	0	100	10	mA
		9	Hibernate I	U2	0	700	8	mA
		11	Hibernate V	U2	2400	3000	2550	mV

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

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
57	Manufacturer Info	0	Block 0	U1	0x00	0xFF	0x00	hex
		1	Block 1	U1	0x00	0xFF	0x00	hex
		2	Block 2	U1	0x00	0xFF	0x00	hex
		3	Block 3	U1	0x00	0xFF	0x00	hex
		4	Block 4	U1	0x00	0xFF	0x00	hex
		5	Block 5	U1	0x00	0xFF	0x00	hex
		6	Block 6	U1	0x00	0xFF	0x00	hex
		7	Block 7	U1	0x00	0xFF	0x00	hex
		8	Block 8	U1	0x00	0xFF	0x00	hex
		9	Block 9	U1	0x00	0xFF	0x00	hex
		10	Block 10	U1	0x00	0xFF	0x00	hex
		11	Block 11	U1	0x00	0xFF	0x00	hex
		12	Block 12	U1	0x00	0xFF	0x00	hex
		13	Block 13	U1	0x00	0xFF	0x00	hex
		14	Block 14	U1	0x00	0xFF	0x00	hex
		15	Block 15	U1	0x00	0xFF	0x00	hex
		16	Block 16	U1	0x00	0xFF	0x00	hex
		17	Block 17	U1	0x00	0xFF	0x00	hex
		18	Block 18	U1	0x00	0xFF	0x00	hex
		19	Block 19	U1	0x00	0xFF	0x00	hex
		20	Block 20	U1	0x00	0xFF	0x00	hex
		21	Block 21	U1	0x00	0xFF	0x00	hex
		22	Block 22	U1	0x00	0xFF	0x00	hex
		23	Block 23	U1	0x00	0xFF	0x00	hex
		24	Block 24	U1	0x00	0xFF	0x00	hex
		25	Block 25	U1	0x00	0xFF	0x00	hex
		26	Block 26	U1	0x00	0xFF	0x00	hex
		27	Block 27	U1	0x00	0xFF	0x00	hex
		28	Block 28	U1	0x00	0xFF	0x00	hex
		29	Block 29	U1	0x00	0xFF	0x00	hex
		30	Block 30	U1	0x00	0xFF	0x00	hex
		31	Block 31	U1	0x00	0xFF	0x00	hex

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

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
80	IT Cfg	0	Load Select	U1	0	255	1	num
		1	Load Mode	U1	0	255	1	num
		17	Max Res Factor	U1	0	255	15	num
		18	Min Res Factor	U1	0	255	7	num
		20	Ra Filter	U2	0	1000	800	num
		37	Fast Qmax Start DOD %	U1	0	255	92	%
		38	Fast Qmax End DOD %	U1	0	255	96	%
		39	Fast Qm Start V Delta	I2	0	1000	125	mV
		41	Fast Qmax Current Threshold	I2	0	1000	4	C/Rate
		43	Fast Qmax Min Points	U1	0	255	2	num
		45	Min % Passed Chg for Qm	U1	1	100	37	%
		49	Qmax Filter	U1	0	255	96	num (mAh)
		50	Max % Default Qmax	U1	0	255	110	%
		51	Terminate Voltage	I2	–32768	32767	3200	mV
		53	Term V Delta	I2	0	4200	200	mV
		56	ResRelax Time	U2	0	65534	500	s
		60	T PreditAmbient Time Constant	U2	0	65535	2000	num
		62	User Rate-mA	I2	–32000	0	0	mA
		64	User Rate-mW	I2	–32000	0	0	mW (cW)
		66	Reserve Cap-mAh	I2	0	32000	0	mAh
		71	Min Delta Voltage	I2	–32000	32000	0	mV (num)
		73	Max Sim Rate	U1	0	255	1	C/rate
		74	Min Sim Rate	U1	0	255	20	C/rate
		75	Ra Max Delta	U2	0	65535	44	mΩ
		77	Trace Resistance	I2	0	2000	0	mΩ
		79	Qmax Max Delta %	U1	0	65535	5	%
		80	DeltaV Max dV	U2	0	65535	10	mV
		82	Max Res Scale	U2	0	32767	5000	num
		84	Min Res Scale	U2	0	32767	200	num
		86	Fast Scale Start SOC	U1	0	100	10	%
		87	Max Allowed Current	I2	-32768	-32767	8500	mAh
		89	Max Current Pulse Duration	U1	0	255	10	s
		90	Max Current Interrupt Step	I2	-32768	32767	500	mAh
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 5-5. Data Flash Summary—Gas (Fuel) Gauging Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
82	State	0	IT Enable	H1	0x0	0x3	0x0	hex
		1	Qmax Cell 0	I2	0	32767	1000	mAh
		3	Cycle Count 0	U2	0	65535	0	num
		5	Update Status 0	H1	0x0	0x3	0x0	hex
		6	Avg I Last Run	I2	–32768	32767	–299	mA
		8	Avg P Last Run	I2	–32768	32767	–1131	mW (cW)
		10	Delta Voltage	I2	–32768	32767	2	mV
		12	T Rise	U2	0	65535	50	Num
		14	T Relax Time Constant	U2	0	65535	1000	Num
		16	Cell0 V at Chg Term	I2	3800	4500	4200	mV

Table 5-6. Data Flash Summary—OCV Tables

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
83	OCVa0 Table	0	Chem ID	H2	0x0000	0xFFFF	0x0100	flags

Table 5-7. Data Flash Summary—Ra Tables

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
91	Pack0 Ra	0	Pack0 Ra status	H1	0x00	0x00	0xFF	hex
		1	Pack0 Ra Flag	H1	0	0xFF	0x55	hex
		2	Pack0 Ra Base R	I2	–200	200	41	num
		4	Pack0 Ra Gain	H1	0x0	0x4	0x0	num
		5	Pack0 Ra 1	I1	–128	127	2	num ($2^{-10}\Omega$)
		6	Pack0 Ra 2	I1	–128	127	-4	num ($2^{-10}\Omega$)
		7	Pack0 Ra 3	I1	–128	127	0	num ($2^{-10}\Omega$)
		8	Pack0 Ra 4	I1	–128	127	-2	num ($2^{-10}\Omega$)
		9	Pack0 Ra 5	I1	–128	127	2	num ($2^{-10}\Omega$)
		10	Pack0 Ra 6	I1	–128	127	6	num ($2^{-10}\Omega$)
		11	Pack0 Ra 7	I1	–128	127	7	num ($2^{-10}\Omega$)
		12	Pack0 Ra 8	I1	–128	127	5	num ($2^{-10}\Omega$)
		13	Pack0 Ra 9	I1	–128	127	8	num ($2^{-10}\Omega$)
		14	Pack0 Ra 10	I1	–128	127	15	num ($2^{-10}\Omega$)
		15	Pack0 Ra 11	I1	–128	127	30	num ($2^{-10}\Omega$)
		16	Pack0 Ra 12	I1	–128	127	54	num ($2^{-10}\Omega$)
		17	Pack0 Ra 13	I1	–128	127	87	num ($2^{-10}\Omega$)
		18	Pack0 Ra 14	I1	–128	127	115	num ($2^{-10}\Omega$)

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

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
92	Pack0 Rax	0	Pack0 Rax status	H1	0x00	0xFF	0xFF	hex
		1	Pack0 Rax Flag	H1	0	0xFF	0xFF	hex
		2	Pack0 Rax Base R	I2	–200	200	41	num
		4	Pack0 Rax Gain	H1	0x0	0x4	0x0	num
		5	Pack0 Rax 1	I1	–128	127	2	num ($2^{-10}\Omega$)
		6	Pack0 Rax 2	I1	–128	127	-4	num ($2^{-10}\Omega$)
		7	Pack0 Rax 3	I1	–128	127	0	num ($2^{-10}\Omega$)
		8	Pack0 Rax 4	I1	–128	127	-2	num ($2^{-10}\Omega$)
		9	Pack0 Rax 5	I1	–128	127	2	num ($2^{-10}\Omega$)
		10	Pack0 Rax 6	I1	–128	127	6	num ($2^{-10}\Omega$)
		11	Pack0 Rax 7	I1	–128	127	7	num ($2^{-10}\Omega$)
		12	Pack0 Rax 8	I1	–128	127	5	num ($2^{-10}\Omega$)
		13	Pack0 Rax 9	I1	–128	127	8	num ($2^{-10}\Omega$)
		14	Pack0 Rax 10	I1	–128	127	15	num ($2^{-10}\Omega$)
		15	Pack0 Rax 11	I1	–128	127	30	num ($2^{-10}\Omega$)
		16	Pack0 Rax 12	I1	–128	127	54	num ($2^{-10}\Omega$)
		17	Pack0 Rax 13	I1	–128	127	87	num ($2^{-10}\Omega$)
		18	Pack0 Rax 14	I1	–128	127	115	num ($2^{-10}\Omega$)

Table 5-8. Data Flash Summary—Calibration Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
104	Data	0	CC Gain	F4	1.00E–01	4.00E+01	0.4768	num (mΩ)
		4	CC Delta	F4	2.98E+04	1.19E+06	567744.56	num (mΩ)
		8	CC Offset	I2	–32768	32767	–1200	num (mA)
		10	Board Offset	I1	–128	127	0	num (µA)
		11	Int Temp Offset	I1	–128	127	0	num (°C)
		12	Ext Temp Offset	I1	–128	127	0	num (°C)
		13	Pack V Offset	I1	–128	127	0	num (mV)
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	(0.1°K)
		8	Ext a Coef 5	I2	–32768	32767	892	(0.1°K)
		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	(°K)
107	Current	1	Deadband	U1	0	255	5	mA
		2	CC Deadband	U1	0	255	17	num

Table 5-9. Data Flash Summary—Security Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
112	Codes	0	Sealed to Unsealed	H4	0x0000 0000	0xFFFF FFFF	0x3672 0414	hex
		4	Unsealed to Full	H4	0x0000 0000	0xFFFF FFFF	0xFFFF FFFF	hex

Table 5-10. Data Flash Summary—Charger Class

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
74	Temperature Table	0	Charge T0	I1	–40	125	0	°C
		1	Charge T1	I1	–40	125	10	°C
		2	Charge T2	I1	–40	125	45	°C
		3	Charge T3	I1	–40	125	50	°C
		4	Charge T4	I1	–40	125	60	°C
		5	Charge Current T0	U1	0	255	50	%C
		6	Charge Current T1	U1	0	255	50	%C
		7	Charge Current T2	U1	0	255	50	%C
		8	Charge Current T3	U1	0	255	50	%C
		9	Charge Current T4	U1	0	255	0	%C
		10	Charge Voltage T0	U2	0	222	210	20 mV
		12	Charge Voltage T1	U2	0	222	210	20 mV
		14	Charge Voltage T2	U2	0	222	207	20 mV
		16	Charge Voltage T3	U2	0	222	205	20 mV
		18	Charge Voltage T4	U2	0	222	0	20 mV
		20	Chg Temp Hys	I1	0	50	5	°C
75	State of Health Table	0	Charge SOH 0	U1	0	100	70	%
		1	Charge SOH 1	U1	0	100	77	%
		3	Charge SOH 2	U1	0	100	85	%
		4	Charge SOH 3	U1	0	100	92	%
		5	Charge Current H0	U1	0	255	70	%C
		6	Charge Current H1	U1	0	255	100	%C
		7	Charge Current H2	U1	0	255	100	%C
		8	Charge Current H3	U1	0	255	100	%C
		9	Charge Voltage H0	U2	0	222	210	20 mV
		11	Charge Voltage H1	U2	0	222	210	20 mV
		13	Charge Voltage H2	U2	0	222	210	20 mV
		15	Charge Voltage H3	U2	0	222	210	20 mV
76	Charger Info	1	Reg 00 Default	H1	0x00	0xFF	0x00	hex
		3	Reg 01 Default	H1	0x00	0xFF	0x4C	hex
		5	Reg 02 Default	H1	0x00	0xFF	0x00	hex
		7	Reg 03 Default	H1	0x00	0xFF	0x00	hex
		9	Reg 04 Default	H1	0x00	0xFF	0x02	hex
		11	Reg 05 Default	H1	0x00	0xFF	0x20	hex
		13	Reg 06 Default	H1	0x00	0xFF	0xE0	hex
		14	CDPLimit Default	H1	0x00	0xFF	0x40	hex
		15	NSTLimit Default	H1	0x00	0xFF	0x20	hex

Table 5-10. Data Flash Summary—Charger Class (continued)

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
78	Charger Control Configuration	0	Charger Options	H1	0x00	0xFF	0x20	flg
		1	Chg Disabled Regulation V	I2	3500	4440	4200	mV
		3	Temperature Filter	U1	0	255	232	num
		4	Force Update Time	U1	0	255	60	s
		5	Shipmode Delay	U1	0	255	30	s

5.6 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 ***[WRTEMP]*** bit of the ***OpConfigB*** register of the fuel gauge is changed from 0 to 1.

Some bq27532-G1 pins are configured via the ***Operation Configuration B*** register. This register is programmed and read via the methods described in Section 5.1, *Accessing The Data Flash*. See Section 5.5, *Data Flash Summary*, for the location (subclass and offset) of these configuration registers.

Table 5-11. OpConfig B Register Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	WRTEMP	BIE	BL_INT	GNDSEL	FCE	RSVD	RFACTSTEP	SIM_CTRL
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 5-12 are in decimal format. The example below has converted these to hexadecimal. For example, the *OpConfigB* subclass ID of 64 = 0x40.

Table 5-12. Data Flash Summary—Configuration

Subclass ID	Subclass	Offset	Name	Data Type	Value			Unit
					Min	Max	Default	
64	Registers	4	OpConfig B	H1	0x00	0xFF	0x4A	hex

5.6.1 Modify WRTEMP of OpConfigB Register

1. Unseal the device by using the *Control()* (0x00 and 0x01) command if the device is SEALED.
   (a) Write the first 2 bytes of the UNSEAL key using the *Control(0x0414)* command.
   (*wr 0x00 0x14 0x04*)
   (b) Write the second 2 bytes of the UNSEAL key using the *Control(0x3672)* command.
   (*wr 0x00 0x72 0x36*)
2. Write 0x00 using *BlockDataControl()* command (0x61) to enable block data flash control.
   (*wr 0x61 0x00*)
3. Write 0x40 (*OpConfigB* SubClass) using the *DataFlashClass()* command (0x3E) to access the registers subclass.
   (*wr 0x3E 0x40*)
4. 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, *OpConfigB* (offset = 4) is in the first block so use:
   (*wr 0x3F 0x00*)
5. To read the data of a specific offset use address 0x40 + mod(offset, 32).
   For example, *OpConfigB* (offset = 4) is located at 0x44, read 1 byte starting at 0x44 address.
   (*rd 0x44 old_OP_CONF_B_BYTE*)
   In our example, assume *WRTEMP*(MSB) is cleared.
6. To read the 1-byte checksum use the *BlockDataChecksum()* command (0x60).
   (*rd 0x60 OLD_checksum*)
7. In this example, set *WRTEMP* by setting the most-significant bit of *OP_CONF_B_BYTE*.
8. 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 *OpConfigB* (offset = 4) located at 0x44, use command: (*wr 0x44 new_OP_CONF_B_BYTE*)
9. 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:
   (a) temp = mod(255 – *OLD_checksum* – *old_OP_CONF_B_BYTE*, 256)
   (b) 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

6.1 Fuel Gauging

The fuel gauge measures the cell voltage, temperature, and current to determine battery SOC. The fuel gauge monitors charge and discharge activity by sensing the voltage across a small-value resistor (5 mΩ to 20 mΩ, typical) between the SRP and SRN pins and in series with the cell. By integrating the charge passing through the battery, the battery SOC is adjusted during battery charge or discharge.

The total battery capacity is found by comparing states of charge before and after applying the load with the amount of charge passed. When an application load is applied, the impedance of the cell is measured by comparing the OCV obtained from a predefined function for present SOC with the measured voltage under load. Measurements of OCV and charge integration determine chemical state of charge and chemical capacity (Qmax). The initial Qmax values are taken from a cell manufacturers' data sheet multiplied by the number of parallel cells. It is also used for the value in ***Design Capacity***. The fuel gauge acquires and updates the battery-impedance profile during normal battery usage. It uses this profile, along with SOC and the Qmax value, to determine *FullChargeCapacity()* and *StateOfCharge()*, specifically for the present load and temperature. *FullChargeCapacity()* is reported as capacity available from a fully charged battery under the present load and temperature until *Voltage()* reaches the ***Terminate Voltage***. *NominalAvailableCapacity()* and *FullAvailableCapacity()* are the uncompensated (no or light load) versions of *RemainingCapacity()* and *FullChargeCapacity()*, respectively.

The fuel gauge has a flag, *[SOC1]*, accessed by the *Flags()* function that warns when the battery SOC has fallen to critical level. When *RemainingCapacity()* falls below the first capacity threshold, specified in ***SOC1 Set Threshold***, the *[SOC1] (State of Charge Initial)* flag is set. The flag is cleared once *RemainingCapacity()* rises above ***SOC1 Clear Threshold***.

When the voltage is discharged to ***Terminate Voltage***, the SOC is set to 0.

6.2 Impedance Track™ Variables

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

6.2.1 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 6.2.2). When ***Load Mode*** is 0, the constant-current model is used (default). When 1, the constant-power model is used. The *CONTROL_STATUS [LDMD]* bit reflects the status of ***Load Mode***.

6.2.2 Load Select

Load Select defines the type of power or current model that computes load-compensated capacity in the Impedance Track™ algorithm. If ***Load Mode*** = 0 (constant-current model) then the options presented in Table 6-1 are available.

Table 6-1. Constant-Current Model Used When Load Mode = 0

Load Select Value	Current Model Used
0	Average discharge current from previous cycle: There is an internal register that records the average discharge current through each entire discharge cycle. The previous average is stored in this register.

Table 6-1. Constant-Current Model Used When Load Mode = 0 (continued)

Load Select Value	Current Model Used
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 *AverageCurrent()*
3	Current: based off of a low-pass-filtered version of *AverageCurrent()* (τ = 14 s)
4	Design capacity / 5: C Rate based off of ***Design Capacity*** /5 or a C/5 rate in mA.
5	AtRate (mA): Use whatever current is in *AtRate()*
6	User_Rate-mA: Use the value in ***User Rate-mA***. This mode provides a completely user-configurable method.

If ***Load Mode*** = 1 (constant-power model) then the options shown in Table 6-2 are available.

Table 6-2. Constant-Power Model Used When Load Mode = 1

Load Select Value	Power Model Used
0	Average discharge power from previous cycle: There is an internal register that records the average discharge power through each entire discharge cycle. The previous average is stored in this register.
1 (default)	Present average discharge power: This is the average discharge power from the beginning of this discharge cycle until present time.
2	Average current × voltage: based off the *AverageCurrent()* and *Voltage()*.
3	Current × voltage: based off of a low-pass-filtered version of *AverageCurrent()* (τ = 14 s) and *Voltage()*
4	Design energy / 5: C Rate based off of Design Energy /5 or a C/5 rate in mA.
5	AtRate (10 mW): Use whatever value is in *AtRate()*.
6	User_Rate-10mW: Use the value in ***User Rate-10mW***. This mode provides a completely user-configurable method.

6.2.3 Reserve Cap-mAh

Reserve Cap-mAh determines how much actual remaining capacity exists after reaching 0 *RemainingCapacity()*, before ***Terminate Voltage*** is reached. A no-load rate of compensation is applied to this reserve if the ***[RESCAP]*** bit in ***Op Config*** register is set.

6.2.4 Reserve Cap-mWh

Reserve Cap-mWh determines how much actual remaining capacity exists after reaching 0 *AvailableEnergy()*, before ***Terminate Voltage*** is reached. A no-load rate of compensation is applied to this reserve capacity if the ***[RESCAP]*** bit in ***Op Config*** register is set.

6.2.5 Dsg Current Threshold

This register is used as a threshold by many functions in the fuel gauge to determine if actual discharge current is flowing into or out of the cell. The default for this register is in Table 5-5, *Data Flash Summary—Gas (Fuel) Gauging*, 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.

6.2.6 Chg Current Threshold

This register is used as a threshold by many functions in the fuel gauge to determine if actual charge current is flowing into or out of the cell. The default for this register is in Table 5-5, *Data Flash Summary—Gas (Fuel) Gauging*, 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.

6.2.7 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 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 5-5, *Data Flash Summary—Gas (Fuel) Gauging*. It should be above the standby current of the system and below both the ***Dsg Current Threshold*** and ***Chg Current Threshold***.

Either of the following criteria must be met to enter 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. 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 rate when attempting to go into relaxation mode.

Quit Relax Time specifies the minimum time required for *AverageCurrent()* to remain above the ***Quit Current*** threshold before exiting the relaxation mode.

6.2.8 Qmax 0

Generically called *Qmax*, this dynamic variable contains the maximum chemical capacity of the cell profile, and is determined by comparing states of charge before and after applying the load with the amount of charge passed. It also corresponds 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* 0** field for the default cell profile. The Impedance Track™ algorithm updates this value and maintains it for the cell profile.

6.2.9 Update Status 0

Bit 0 (0x01) of the ***Update Status* 0** register indicates that the fuel gauge has learned new Qmax parameters and is accurate. The remaining bits are reserved. Bit 0 is user-configurable; however, it is also a status flag that can be set by the fuel gauge. Bit 0 should never be modified except when creating a golden image file as explained in application note SLUA334, *Preparing Optimized Default Flash Constants for Specific Battery Types*. Bit 0 is updated as needed by the fuel gauge.

6.2.10 Avg I Last Run

The fuel gauge logs the current averaged from the beginning to the end of each discharge cycle. It stores this average current from the previous discharge cycle in this register. This register should not be modified. It is only updated by the fuel gauge when required.

6.2.11 Avg P Last Run

The fuel gauge logs the power averaged from the beginning to the end of each discharge cycle. It stores this average power from the previous discharge cycle in this register. To get a correct average power reading the fuel gauge continuously multiplies instantaneous current times *Voltage()* to get power. It then logs this data to derive the average power. This register should not be modified. It is only updated by the fuel gauge when required.

6.2.12 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 the 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.

6.2.13 Ra Tables

These tables contain encoded data and 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.

6.2.14 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***)

6.2.15 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 0*** ≠ 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.

6.2.16 SOC Smoothing

Rapid changes in operating conditions, such as temperature or discharge current, can lead to sudden changes in the immediate calculation of *RemainingCapacity( )*, *FullChargeCapacity( )*, and *StateOfCharge( )* by the algorithm. 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 [RLXSMEN]*** 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.

6.2.17 Operation Configuration Registers

Some pin configurations and algorithm settings are configured via the ***Operation Configuration*** data flash register, as indicated in Table 6-3. This register is programmed/read via the methods described in Section 5.1. The register is located at subclass = 64, offset = 0.

Table 6-3. Operation Configuration Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
High Byte	RESCAP	RSVD	INT_BREM	SOCHOLD1	SOCHOLD99	IWAKE	RSNS1	RSNS0
Default	0	0	0	1	1	0	0	1
	0x19							
Low Byte	INT_FOCV	INT_CHGFLT	SLEEP	RMFCC	SOCI_POL	RSVD	RSVD	TEMPS
Default	0	1	1	1	0	0	0	1
	0x71							

High Byte

RESCAP = No-load rate of compensation is applied to the reserve capacity calculation. True when set.

RSVD = Bit 6 is reserved.

INT_BREM = Battery removal interrupt bit. The SOC_INT pulses 1 ms when the battery removal interrupt is enabled. True when set. See Section 6.3.2.

SOCHOLD1 = Holds SOC to 1% during discharge until Terminate Voltage is reached. True when set.

SOCHOLD99 = Holds SOC to 99% during charge until charge termination is reached. True when set.

IWAKE/RSNS1/RSNS0 = These bits configure the current wake function (see Table 6-9).

Low Byte

INT_FOCV = Indication of the measurement of the OCV during the initialization. The SOC_INT pulses during the first measurement if this bit is set. True when set. See Section 6.3.2.

INT_CHGFLT = Charge fault interrupt enable bit. True when set. See Section 6.3.2.

SLEEP = The fuel gauge can enter SLEEP mode, 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 6.5.1, *Detecting Charge Termination*)

SOCI_POL = SOC interrupt polarity is active-low. True when cleared.

RSVD = Bit 2 and 1 are reserved.

TEMPS = Selects external thermistor for *Temperature()* measurements. True when set.

Some pin configurations and algorithm settings are configured via the ***Operation Configuration B*** data flash register, as indicated in Table 6-4. This register is programmed and read via the methods described in Section 5.1, *Accessing the Data Flash*. The register is located at subclass = 64, offset = 11.

Table 6-4. Operation Configuration B Bit Definitions

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

WRTEMP = Enables the temperature write. The temperature could be written by the host. True when set.

BIE = Battery insertion detection enable. When the battery insertion detection is disabled, the gauge relies on the host command to set the *Flags() [BAT_DET]* bit. True when set.

BL_INT = Battery low interrupt enable. True when set. See Section 6.3.2.

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.

FCE = Fast convergence enable. Configures algorithm to use fast convergence method. Default is 1 and is the recommended setting for all applications. True when set.

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

SIM_CTRL = 0 = Dynamic Simulation Step Enabled
1 = Voltage Consistency Enabled

Table 6-5. Operation Configuration C Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RSVD	RSVD	RSVD	RSVD	INT_CHGPROF	SlpWkChg	DeltaVOpt1	DeltaVOpt0
Default	0	0	0	0	1	1	0	0
	0x0C							

RSVD = Bits 7, 6, 5 and 4 are reserved.

INT_CHGPROF = Enables interrupt for any changes in charge profile. True when set. See Section 6.3.2.

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 a used as a compensation factor for calculating for *RemainingCapacity()* under pulsed loads.
00 = Standard DeltaV. Average variance from steady state voltage that determines end-of-discharge voltage.
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.

Table 6-6. Operation Configuration D Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RLXSMEN	SMTHEN	SOC_STATE	SOC_OCV	RSVD	RSVD	CHGDODEOC	AMB_PRED
Default	1	1	0	0	0	0	1	1
	0xC3							

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

SMTHEN = Enables SOC smoothing algorithm. True when set.

SOC_STATE = Enables SOC_INT pin function to generate a pulse due to an Impedance Track™ algorithm state change. (See Table 6-8, SOC_INT Pulse Conditions and Widths)

SOC_OCV = Enables SOC_INT pin function to generate a pulse due to OCV command. (See Table 6-8, SOC_INT Pulse Conditions and Widths)

RSVD = Bits 3 and 2 are reserved.

CHGDODEOC = Enables DoD at End-of-Charge recalculation during charging only. True when set. The default setting is recommended.

AMB_PRED = Enables predicting ambient temperature based on thermal model.

Table 6-7. Operation Configuration E Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	SMSYNEN	IMAXRESVEN	IMAXINTEN	IMAXEN	DSGFASTQM	CHGFASTQM	RMHOLD100	RMHOLD0
Default	0	1	1	1	0	0	1	1
	0x73							

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

IMAXRESVEN = Enables usage of ***Reserve Capacity*** in the *MaxCurrent()* calculation.

IMAXINTEN = Enables interrupt for any changes in *MaxCurrent()*.

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

DSGFASTQM = Enables end of discharge (DSG) related Fast Qmax function. See Section 6.2.15, *Fast Qmax Update*, for additional details. Use the defaults for most applications.

CHGFASTQM = Enables end of charge (CHG) related Fast Qmax function. See Section 6.2.15, *Fast Qmax Update*, for additional details. Use the defaults for most applications.

RMHOLD100 = Enables holding SOC to 100% when the battery has been overcharged until the battery is discharged below 100%.

RMHOLD0 = Enables holding SOC to 0% when the battery has been overdischarged until the battery is charged above 0%.

6.3 Detailed Pin Description

6.3.1 Battery 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 the host can inform the gauge of the battery status via the *BAT_INSERT* and *BAT_REMOVE* subcommands.

6.3.2 SOC_INT Pin

The SOC_INT pin generates a pulse of different pulse widths under various conditions as indicated by the Table 6-8. After initialization only one SOC_INT pulse is generated within any given one-second time slot and therefore, may indicate multiple event conditions.

Table 6-8. SOC_INT Pulse Conditions and Widths

	Enable Condition	Pulse Width	Comment
SOC_Delta Point	***SOC_Delta*** ≠ 0	1 ms	During charge, when the SOC is greater than (>) the points, 100% – n × ***SOC_Delta*** and 100%; During discharge, when the SOC reaches (≤) the points, 100% – n × ***SOC_Delta*** and 0%; where n is an integer starting from 0 to the number generating SOC no less than 0%
SOC1 Set	***[BL_INT]*** bit in ***OpConfig B*** is set.	1 ms	When RSOC reaches the ***SOC1 Set*** or ***Clear Threshold*** set in the data flash.
SOC1 Clear	***[BL_INT]*** bit in ***OpConfig B*** is set.	1 ms	
Battery State Change	(***SOC Delta***) ≠ 0 and ***OpConfig D [SOC_STATE]*** = 1	1 ms	Upon detection of a state change in battery charging and discharging. Relaxation is not included.
Battery Removal	***[INT_BREM]*** bit is set in ***Op Config*** and ***OpConfig B [BIE]*** bit is set	1 ms	This function is disabled when ***[BIE]*** bit is cleared.
MaxCurrent Update	***[IMAXINTEN]*** bit is set in ***Op Config E***	1 ms	When *MaxCurrent()* is updated.
OCV Command	After initialization	About 165 ms. Same as the OCV command execution time period	SOC_INT pulses for the OCV command after the OCV command execution.
OCV Command	***[INT_FOCV]*** bit is set in ***Op Config***	About 165 ms. Same as the OCV command execution time period	This command generates the SOC_INT pulse during the initialization.
Charge Fault	***[INT_CHGFLT]*** bit is set in ***Op Config***	1 ms	Indicates any fault change in *Chrgr_Reg0()*
Charge Profile	***[INT_CHGPROF]*** bit is set in ***OpConfig B***	1 ms	Indicates changes in *ProgChargingVoltage()*, *ProgChargingCurrent()* and *Chrgr_Reg1[CE]* bits.

6.4 Temperature Measurement

The fuel gauge 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 also measure internal temperature via its on-chip temperature sensor, but only if the ***[TEMPS]*** bit of the ***Op Config*** register is cleared. The ***[GNDSEL]*** bit of the ***OpConfig B*** register selects the ground reference of the ADC converter for temperature measurement.

Regardless of which sensor is used for measurement, a system processor can request the current battery temperature by calling the *Temperature()* function (see Chapter 2, *Standard Data Commands*, for specific information).

The thermistor circuit requires the use of an external NTC 103AT-type thermistor. Additional circuit information for connecting this thermistor to the fuel gauge is shown in Chapter 9, *Reference Schematic*.

6.5 Charging and Charge-Termination Indication

6.5.1 Detecting Charge Termination

For proper fuel gauge operation, the cell charging voltage must be specified by the user. The default value for this variable is ***Charging Voltage***, see Section 5.5, *Data Flash Summary*.

The fuel gauge detects charge termination when:

- During 2 consecutive periods of ***Current Taper Window***, *the AverageCurrent()* is < ***Taper Current***
- During the same periods, the accumulated change in capacity > ***Min Taper Charge*** / ***Current Taper Window***, and
- *Voltage()* > ***Charging Voltage – Taper Voltage***, and ILIMITED flag, LOOP_STAT_1, and LOOP_STAT_0 are clear

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

6.6 Power Modes

The fuel gauge has different power modes:

- BAT INSERT CHECK: 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.
- NORMAL: In NORMAL mode, the fuel gauge is fully powered and can execute any allowable task.
- SLEEP: 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.
- SLEEP+: In SLEEP+ mode, both low-frequency and high-frequency oscillators are active. Although the SLEEP+ mode has higher current consumption than the SLEEP mode, it is also a reduced power mode.
- HIBERNATE: In HIBERNATE mode, the fuel gauge is in a low power state, but can be woken up by communication or certain I/O activity.

The relationship between these modes is shown in Figure 6-1.

6.6.1 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. When battery insertion is detected, a series of initialization activities begin, which include: OCV measurement, setting the *Flags() [BAT_DET]* bit, 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.

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

6.6.3 SLEEP Mode

SLEEP mode is entered automatically if the feature is enabled (***Op Config [SLEEP]*** = 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 SLEEP mode if any entry condition is broken, specifically when:

- *AverageCurrent()* rises above ***Sleep Current***, or
- 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.

6.6.4 SLEEP+ Mode

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

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 SLEEP+ mode if any entry condition is broken, specifically when:

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

6.6.5 HIBERNATE Mode

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.

Before the fuel gauge can enter HIBERNATE mode, the system must set the *CONTROL_STATUS [HIBERNATE]* bit. The gauge waits to enter HIBERNATE mode until it has taken a valid OCV measurement and the magnitude of the average cell current has fallen below ***Hibernate Current***. The gauge can also enter HIBERNATE mode if the cell voltage falls below ***Hibernate Voltage*** and a valid OCV measurement has been taken. The gauge remains in HIBERNATE mode until the system issues a direct I^2C command to the gauge or a POR occurs. Any I^2C communication that is not directed to the gauge does not wake the gauge.

It is the responsibility of the system to wake the fuel gauge after it has gone into HIBERNATE mode. After waking, the gauge can proceed with the initialization of the battery information (OCV, profile selection, etc.)

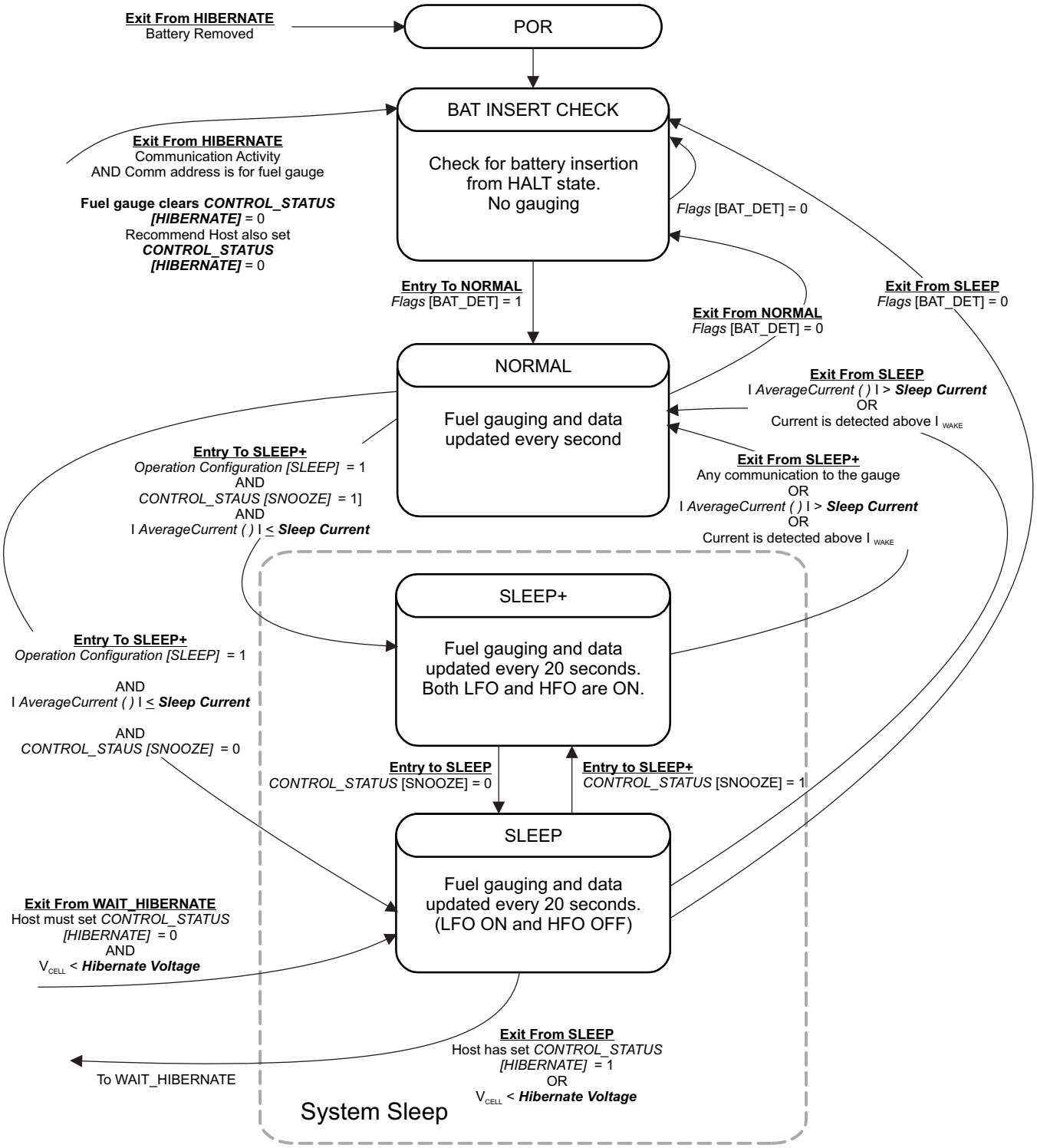


Figure 6-1. Power Mode Diagram—System Sleep

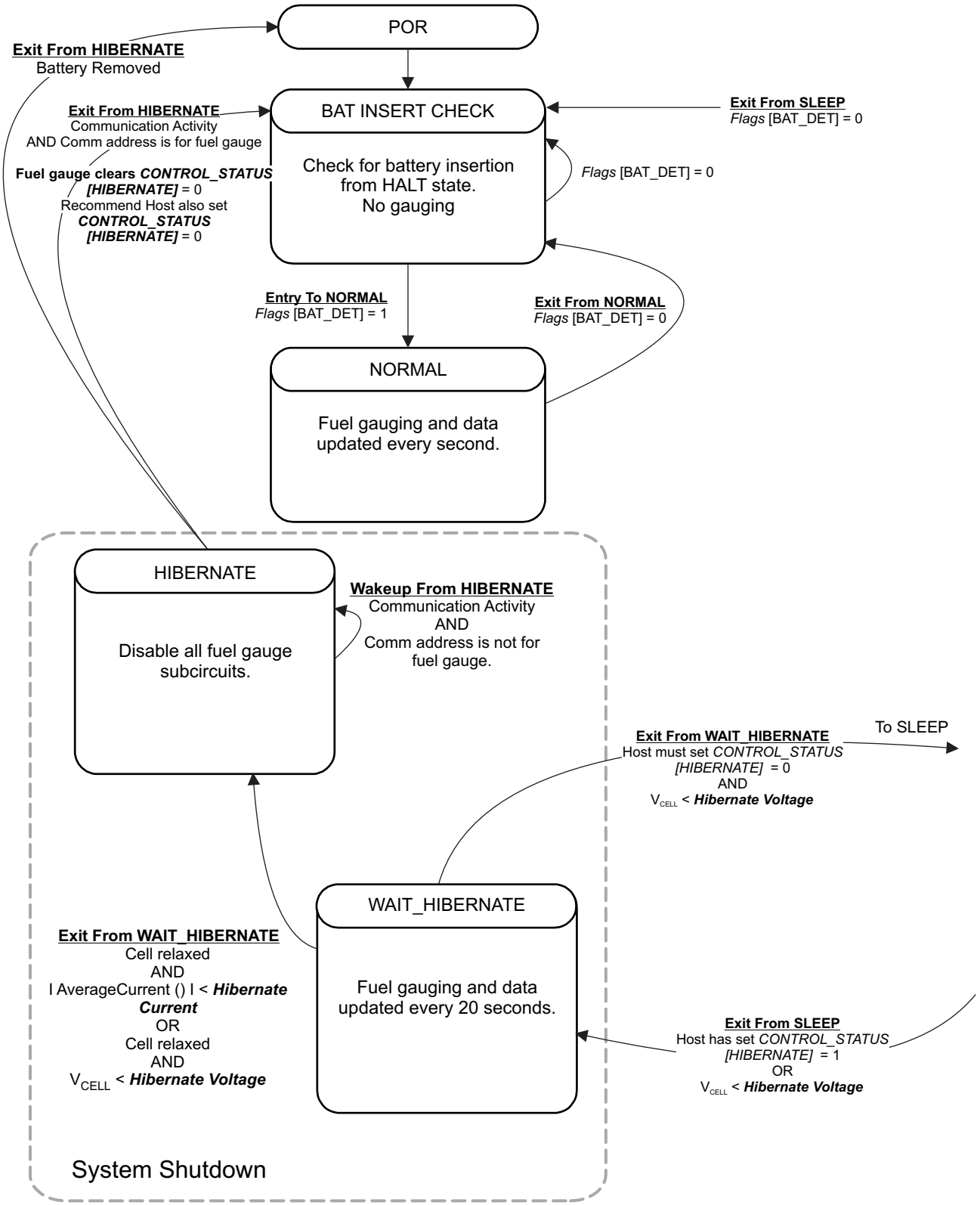


Figure 6-2. Power Mode Diagram—System Shutdown

6.7 Power Control

6.7.1 Wake-up Comparator

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

Table 6-9. 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, just the actual voltage threshold when calculating the configuration. The voltage thresholds are typical values under room temperature.

6.7.2 Flash Updates

Data flash can only be updated if *Voltage( )* ≥ ***Flash Update OK Voltage***. Flash programming current can cause an increase in LDO dropout. The value of ***Flash Update OK Voltage*** must be selected such that the V_{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 I^2C clock significantly. See Section 8.4, *I^2C Clock Stretching*, for more information.

6.8 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 ≤ 5°C or *Temperature()* ≥ 45°C.

The fuel gauge also performs a single offset when:

- The condition of *AverageCurrent()* ≤ 100 mA, and
- {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.

6.9 Additional Data Flash Parameter Descriptions

6.9.1 Configuration Class

6.9.1.1 Charge Termination Subclass

6.9.1.1.1 Charging Voltage

The fuel gauge uses this value along with ***Taper Voltage*** to detect charge termination. During Primary Charge Termination detection, one of the requirements is that ***Voltage*** must be above (***Charging Voltage*** – ***Taper Voltage***) for the gauge to start trying to qualify a termination. This value depends on the battery that is charged using the charger. The default is 4.2 V.

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

The fuel gauge uses ***Charging Voltage*** along with ***Taper Voltage*** to detect charge termination. During Primary Charge Termination detection, one of the three requirements is that ***Voltage*** must be above (***Charging Voltage*** – ***Taper Voltage***) for the gauge to start trying to qualify a termination. This value depends on the battery that is charged using the charger. The default is 4.2 V.

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

- During two consecutive periods of ***Current Taper Windows***:
  Iavg1 < ***Taper Current*** and Iavg2 < ***Taper Current***
- During the same periods: Accumulated change in capacity < ***Min Taper Capacity*** per ***Current Taper Window***
- ***Voltage*** > ***Charging Voltage*** – ***Taper Voltage***, and ILIMITED flag, LOOP_STAT_1, and LOOP_STAT_0 are clear

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

Normal Settings:

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

6.9.1.1.3 Full Charge Clear %, Full Charge Clear Voltage

Once charge is terminated, the gauge keeps the charger from recharging the battery until SOC is below ***FC Clear %*** or the battery voltage is below ***FC Clear Volt***. Recharge based on SOC is disabled by setting ***FC Clear %*** to –1. Recharge based on voltage is disabled by setting ***FC Clear Volt*** to 0.

6.9.1.1.4 DOD at EOC Delta Temperature

During relaxation and charge start, **REMCAP** = *FullChargeCapacity()* – *Qstart()*. But as temperature decreases, *Qstart()* can become much smaller than the old *FullChargeCapacity()* resulting in overestimation of **REMCAP**. To improve accuracy, *FullChargeCapacity()* is updated when the temperature change from the previous *FullChargeCapacity()* update is more than ***DODatEOC Delta T***.

6.9.1.2 Data Subclass

6.9.1.2.1 Cycle Count Threshold (CC Threshold)

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

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

6.9.1.2.2 Design Capacity

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

This value is set based on the battery specification. See the data sheet from the battery manufacturer.

6.9.1.2.3 Des 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 and less than 14500 mAh, a setting of 10 is recommended.

Table 6-10. Data Flash Parameter Scale and Unit Based on *Design Energy Scale*

Data Flash	Design Energy Scale = 1 (default)	Design Energy Scale = 10
Design Energy	mWh	cWh
Avg P Last Run	mW	cW
User Rate-mW/cW	mWh	cWh
T Rise	No scale (20 typical)	Scaled by x 10 (2 typical)

6.9.1.2.4 State Of Health Load I

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

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

6.9.1.2.5 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.

6.9.1.2.6 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).

6.9.1.2.7 Data Flash Version

The pack manufacturer can use this location to store the data flash configuration version.

6.9.1.3 Discharge Subclass

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

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

These values are user preference.

SOC1 Set Threshold

SOC1 Set Threshold sets a *StateOfCharge( )* percentage threshold that indicates 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 *[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.

SOC1 Clear Threshold

SOC1 Clear Threshold sets a *StateOfCharge( )* percentage threshold that indicates when *StateOfCharge( )* rises above a defined *StateOfCharge( )*. When *StateOfCharge( )* rises above the ***SOC1 Clear Threshold***, the *[SOC1]* bit in the *Flags( )* register is cleared.

SOC1 Clear Threshold is normally set to 5% above the ***SOC1 Set Threshold***.

6.9.1.3.2 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***.

6.9.1.3.3 Default Average Current Last Run and Default Average Power Last Run

These parameters are not used in the fuel gauge.

6.9.1.4 Full Reset Counter

The ***Full Reset Counter*** is not used in the fuel gauge.

6.9.1.5 Registers Subclass

6.9.1.5.1 State Of Charge Delta

See Table 6-8, *SOC_INT Pulse Conditions and Widths*, for the description of this parameter.

6.9.1.5.2 I^2C Timeout

See Section 8.2, *I^2C Time Out*, for the description of this parameter.

6.9.1.5.3 Clock Control Register

See Section 8.4, *I^2C Clock Stretching*, for the description of this parameter.

6.9.1.6 Power Subclass

6.9.1.6.1 Flash Update OK Voltage

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

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

6.9.1.6.2 Sleep Current

Sleep Current sets a current threshold that determines if the fuel gauge can enter SLEEP mode. When *AverageCurrent( )* < ***Sleep Current*** or > –***Sleep Current***, the fuel gauge enters SLEEP mode if the feature is enabled (***Op Config [SLEEP]*** = 1).

Sleep Current should be below any normal application currents.

6.9.1.6.3 Hibernate Current

Hibernate I sets the current threshold that the fuel gauge uses as a possible condition to enter HIBERNATE mode. If the *[HIBERNATE]* bit in the *Control_Status( )* register is set, the gauge is allowed to enter HIBERNATE mode if the |*AverageCurrent( )*| is below ***Hibernate I*** and the cell is relaxed (an OCV measurement has been taken).

Hibernate I should be below any normal application currents. See the bq27532-G1 data sheet (SLUSBU6) for more details on HIBERNATE mode.

6.9.1.6.4 Hibernate Voltage

Hibernate V sets the voltage threshold the fuel gauge uses, as a possible condition to enter HIBERNATE mode. If the gauge has taken a valid OCV measurement (cell is relaxed) and *Voltage( )* is less than ***Hibernate V***, the gauge enters HIBERNATE mode. Setting ***Hibernate V*** to 0, disables this method of entry into HIBERNATE mode.

Hibernate V should be below any normal application voltages. See the bq27532-G1 data sheet SLUSBU6) for more details on HIBERNATE mode.

6.9.2 Gas Gauging Class

6.9.2.1 IT Cfg Subclass

6.9.2.1.1 Load Select

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

If ***Load Mode*** = 0 (constant-current model), then the options presented in Table 6-11 are available.

Table 6-11. Constant-Current Model Used When Load Mode = 0

Load Select Value	Current Model Used
0	Average discharge current from the previous cycle: There is an internal register that records the average discharge current through each entire discharge cycle. The previous average is stored in this register.
1 (default)	Present average discharge current: This is the average discharge current from the beginning of this discharge cycle until present time.

Table 6-11. Constant-Current Model Used When Load Mode = 0 (continued)

Load Select Value	Current Model Used
2	Average current: based off the *AverageCurrent( )*
3	Current: based off of a low-pass-filtered version of *AverageCurrent( )* (τ = 14 s)
4	Design capacity / 5: C Rate based off of Design Capacity /5 or a C / 5 rate in mA.
5	Use the value specified by *AtRate( )*
6	Use the value in ***User_Rate-mA:*** This gives a completely user-configurable method.

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

Table 6-12. Constant-Power Model Used When Load Mode = 1

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

6.9.2.1.2 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 6.9.2.1.1, *Load Select*). When ***Load Mode*** is 0, the constant-current model is used (default). When Load Mode is 1, the constant-power model is used. The *CONTROL_STATUS [LDMD]* bit reflects the status of ***Load Mode***.

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

6.9.2.1.3 Maximum Resistance Factor

Maximum 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.

6.9.2.1.4 Minimum Resistance Factor

Maximum 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 7. 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 30% from the current resistance value in the negative direction.

6.9.2.1.5 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% of the previous Ra value + 20% of the learned Ra value = the new Ra value).

6.9.2.1.6 Fast Qmax Start DOD%

DOD > ***Fast Qmax Start DOD%*** is one of the conditions to be met for the Fast Qmax Update algorithm to start trying to qualify Fast Qmax DOD samples. See Section 6.2.15.

6.9.2.1.7 Fast Qmax End DOD%

One condition for which the Fast Qmax DOD point will be qualified when discharge stops is when DOD > ***Fast Qmax End DOD%***. See Section 6.2.15.

6.9.2.1.8 Fast Qm Start V Delta

Voltage < (***Terminate Voltage*** + ***Fast Qmax Start V Delta***) is one of the conditions to be met for the Fast Qmax Update algorithm to start trying to qualify Fast Qmax DOD samples. See Section 6.2.15.

6.9.2.1.9 Fast Qmax Current Threshold

Current < C / ***Fast Qmax Current Threshold*** is one of the conditions to be met for the Fast Qmax Update algorithm to start trying to qualify Fast Qmax DOD samples. See Section 6.2.15.

6.9.2.1.10 Fast Qmax Min Points

One condition for which the Fast Qmax DOD point will be qualified when discharge stops is when number of Fast Qmax measurements > ***Fast Qmax Min Points***. See Section 6.2.15.

6.9.2.1.11 Minimum Percentage Passed Charge for Qmax

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.

6.9.2.1.12 Qmax Update Filter

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

6.9.2.1.13 Terminate Voltage

Terminate Voltage stores the voltage for the end of discharge where *RemainingCapacity( )* is set to 0 mAh. ***Terminate Voltage*** is used in the Impedance Track™ algorithm to help compute *RemainingCapacity( )*.

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

6.9.2.1.14 Terminate Voltage Delta

Term V Delta stores the offset that is added to ***Terminate Voltage*** to create a voltage threshold for the *Voltage( )* register for triggering fast resistance scaling provided that the *[FConvEn]* bit is set in ***Pack Configuration B***. The fast resistance scaling algorithm is meant to improve accuracy at the end of discharge. This is one of the triggering conditions for fast resistance scaling; the other condition is when *StateOfCharge( )* goes below ***Fast Scale Start SOC***. If SOC goes below ***Fast Scale Start SOC*** before *Voltage( )* goes below (***Terminate Voltage*** + ***Term V Delta***), fast resistance scaling is already enabled.

For most battery applications, it is recommended to keep (***Terminate Voltage*** + ***Term V Delta***) below 3.4 volts.

6.9.2.1.15 Simulation Res Relax Time

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

The default value of 500 seconds is sufficient for most applications.

6.9.2.1.16 T Predict Ambient Time Constant

Sets the time into charge or discharge to initiate ambient temperature prediction. A general guideline is to program this value to twice the value of ***Thermal Time Constant***.

6.9.2.1.17 User Defined Rate—mA or mW

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

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

6.9.2.1.18 Reserve Capacity—mAh

Reserve Cap-mAh is used to determine the amount of capacity, in mAh, that will be left in the battery when the fuel gauge reports *RemainingCapacity( )* = 0 mAh when ***Load Mode*** = 0 (constant-current model). The ***Reserve Cap-mAh*** parameter allows for a controlled shutdown after the gauge reports *RemainingCapacity( )* = 0 mAh.

Carefully select ***Reserve Cap-mAh*** based upon the system requirements.

6.9.2.1.19 Minimum Delta Voltage

This is the minimum ***Delta Voltage*** that is saved during discharge cycles.

6.9.2.1.20 Maximum Simulation Rate

Maximum IT simulation rate (inversed). 2 implies C / 2.

6.9.2.1.21 Minimum Simulation Rate

Minimum IT simulation rate (inversed). 20 implies C / 20.

6.9.2.1.22 Ra Maximum Delta

The maximum jump allowed during updates of a Ra table grid point. Manually change ***Ra Max Delta*** to 15% of the grid 4 Ra value after an optimization cycle is completed.

Calculate and modify ***Ra Max Delta*** when creating the golden file.

6.9.2.1.23 Trace Resistance

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

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

6.9.2.1.24 Qmax Maximum Delta %

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

6.9.2.1.25 DeltaV Maximum Delta Voltage

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

6.9.2.1.26 Maximum and Minimum Resistance Scale

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

6.9.2.1.27 Fast Scale Start State Of Charge

Fast Scale Start SOC is the threshold on *StateOfCharge( )*. When SOC falls below this threshold, fast resistance scaling is enabled provided that the ***[FConvEn]*** bit in ***Pack Configuration B*** is set. This is the other condition which might trigger fast resistance scaling; the first condition is defined in the ***Term V Delta*** flash parameter. If the ***Term V Delta*** flash parameter is reached before SOC falls below ***Fast Scale Start SOC***, fast resistance scaling is already enabled.

6.9.2.1.28 Maximum Allowed Current

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

6.9.2.1.29 Max Current Pulse Duration

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

6.9.2.1.30 Max Current Interrupt Step

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

6.9.2.2 Current Thresholds Subclass

6.9.2.2.1 Discharge and Charge Detection Thresholds

Dsg Current Threshold:

This register is used as a threshold in the gauge to determine if actual discharge current is flowing out of the battery. This is independent of the *Flags( ) [DSG]* bit, which indicates whether the gauge is in discharge mode or charge mode. If the gauge is charging, then the *[DSG]* bit is 0; and at any other time, the *[DSG]* bit is set to 1. The Impedance Track™ algorithm in the gauge requires more definitive information about whether current is flowing in either the charge or discharge direction. ***Dsg Current Threshold*** is used for this purpose. This default threshold 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.

This register is used as a threshold by many functions in the fuel gauge to determine if actual discharge current is flowing out of the battery.

The *[DSG]* bit in *Flags( )* is the method for determining charging or discharging. If the fuel gauge detects charging or relaxation, then *[DSG]* is 0; and at any other time (*AverageCurrent( )* ≤ ***Dsg Current Threshold***), the *[DSG]* bit is set to 1.

Chg Current Threshold:

This register is used as a threshold in the gauge to determine if actual charge current is flowing into the battery. This is independent of the *Flags( ) [DSG]* bit, which indicates whether the gauge is in discharge mode or charge mode. If the gauge is charging, then the *[DSG]* bit is 0 and any other time, the *[DSG]* bit is set to 1. The Impedance Track™ algorithm in the gauge requires more definitive information about whether current is flowing in either the charge or discharge direction. ***Chg Current Threshold*** is used for this purpose. This default threshold 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.

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

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

6.9.2.2.2 Quit Current

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

- *AverageCurrent( )* > (–)***Quit Current*** and then goes within (±)***Quit Current*** for ***Dsg Relax Time***.
- *AverageCurrent( )* < ***Quit Current*** and then goes within (±)***Quit Current*** for ***Dsg Relax Time***.

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

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

6.9.2.2.3 Discharge and Charge Relax Times

Dsg Relax Time:

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

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

Chg Relax Time:

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

This is application specific, the default is 60 seconds.

6.9.2.2.4 Quit Relax Time

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

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

6.9.2.2.5 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.

6.9.2.2.6 Maximum IR Correct

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

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

This is particular to handheld applications.

6.9.2.3 State Subclass

6.9.2.3.1 IT Enable

See Section 2.1.33, *IT_ENABLE: 0x0021*, for the description of ***IT Enable***.

6.9.2.3.2 Qmax Cell 0

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

6.9.2.3.3 Cycle Count 0

These are the number of Qmax updates the battery has experienced.

Initially, set ***Cycle Count*** to 0 for fresh battery cell.

6.9.2.3.4 Update Status 0

Because this is a system-side gauge, the ***Update Status 0*** register can be represented by the bits below:

x	x	x	x	x	x	Bit 1	Bit 0

Two bits in this register are important:

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

At the beginning of a learning cycle when creating a golden file, ***IT Enable*** starts at 0x00. When IT is enabled with the *IT_ENABLE* subcommand being sent to *Control( )*, ***IT Enable*** automatically changes to 0x01. After the charge and relaxation portion of the learning cycle are complete, ***Update Status 0*** should have become 0x01. Finally, after the discharge and relaxation portion of the learning cycle, ***Update Status 0*** becomes 0x02 if the learning cycle was successfully completed. A golden file can then be generated if ***Update Status 0*** was successfully set to 0x02 by the gauge. When the golden file is created, bit 1 of ***IT Enable*** is cleared, leaving ***Update Status 0*** = 0x02.

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

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

6.9.2.3.5 Average Current Last Run

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

This register should never need to be modified, it is only updated by the fuel gauge when the gauge exits the discharge mode.

6.9.2.3.6 Average Power Last Run

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

This register should never need to be modified. It is only updated by the fuel gauge when the gauge exits discharge mode.

6.9.2.3.7 Pulse Delta Voltage

The ***Delta Voltage*** value is the maximum difference of *Voltage( )* during short load spikes and normal load, so the Impedance Track™ algorithm can calculate remaining capacity for pulse loads. The ***Delta Voltage*** value is automatically updated by the gauge during operation as voltage spikes are detected. It can be initialized to a higher value if large spikes are typical for the system. Allowable values are limited by *Max Delta V* and *Min Delta V*.

During the IT simulations, the target voltage of the empty battery is ***Terminate Voltage*** + ***Delta Voltage***. This feature allows ***Terminate Voltage*** to be set at the minimum operating voltage of the system with confidence that the 0% point will be reached at a sufficiently high voltage to prevent voltage spikes from crashing the system while still extracting maximum run time from the battery when spikes are small.

6.9.2.3.8 Thermal Rise Factor

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

6.9.2.3.9 Thermal Time Constant

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

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

6.9.2.3.10 Cell0 V at Chg Term

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

Cell0 V at Chg Term defaults to 4200 mV, but can be initialized to the nominal charging voltage of the system.

6.9.3 OCV Tables Class

6.9.3.1 Chemistry Identification

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

It defaults to 0100 when you program the default flash image which can be obtained from the Texas Instruments website.

6.9.4 Ra Tables Class

This data is automatically updated during device operation. Do not make changes except for reading the values from another pre-learned pack for creating *Golden Image Files*. Profiles have format *Pack0 R_a* M where M is the number indicating state of charge to which the value corresponds.

Pack0 R_a flag

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

R_a0 or **R_a0x**

The two profiles for the cell ensure that at any given time at least one profile is enabled and is being used while attempts can be made to update the alternate profile without interference. Having two profiles also helps reduce stress on the flash memory. At the beginning of the subclass (profile) is a flag called ***Pack0 R_a flag***. This flag is a status flag that indicates the validity of the table data associated with this flag and whether this particular table is enabled or disabled.

Each flag has two bytes:

1. The least-significant byte (LSB) indicates whether the table is currently enabled or disabled. It has the following options:
   (a) 0x00: means the table had a resistance update in the past; however, it is not the currently enabled table for the cell. (The alternate table for the cell must be enabled at this time.)
   (b) 0xFF: This means that the values in this table are default values. These table resistance values have never been updated, and this table is not the currently enabled table for the cell. (The

alternate table for the indicated cell must be enabled at this time.)

(c) 0x55: This means that this table is enabled for the indicated cell. (The alternate table must be disabled at this time.)

2. The most-significant byte (MSB) indicates the status of the data in this particular table. The possible values for this byte are:

(a) 0x00: The data associated with this flag has a resistance update and the *Qmax Pack* is updated.

(b) 0x05: The resistance data associated with this flag is updated and the pack is no longer discharging (this is prior to a *Qmax Pack* update).

(c) 0x55: The resistance data associated with this flag is updated and the pack is still discharging. (Qmax update attempt not possible until discharging stops.)

(d) 0xFF: The resistance data associated with this flag is all default data.

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

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

Pack0 R_a0 – Pack0 R_a14

The ***Ra Table*** class has 17 values for each R_a subclass. Each of these values represent a resistance value normalized at 0°C for the associated *Qmax Pack*-based SOC grid point as found by the following rules:

For ***Pack0 R_aM*** where:

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

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

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

- Watch for negative values in the **Ra Table** class. Negative numbers in profiles should never be anywhere in this class.

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

6.9.5 Calibration Class

6.9.5.1 Data Subclass

Most of the following values never require modification by the user. They are only modified by the calibration commands in calibration mode. For calibration using a host system, see *Host System Calibration Method* (SLUA640).

6.9.5.1.1 CC Gain

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

6.9.5.1.2 CC Delta

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

6.9.5.1.3 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-second 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 *Host System Calibration Method Application Report* (SLUA640) 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 performed the first time the I^2C 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 I^2C clock and data lines are low for more than 5 seconds and *AverageCurrent( )* is less than ***Sleep Current*** in mA, then an automatic ***CC Offset*** calibration is performed. This takes approximately 16 seconds and is much more accurate than the method in calibration mode.

6.9.5.1.4 Board Offset

Board Offset is the second offset register. It primarily calibrates 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. The simplified ground circuit design in the fuel gauge requires a separate board offset for each tested device.

6.9.5.1.5 Internal Temperature 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.

6.9.5.1.6 External Temperature Offset

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

6.9.5.1.7 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.

6.9.5.2 Temperature Model Subclass

These parameters are configured to work with a 103AT thermistor and the internal temperature sensor and do not need to be modified.

6.9.5.3 Current Subclass

6.9.5.3.1 Deadband

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

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

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

6.9.6 Security Class

Sealed to Unsealed

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

The default code is set to 0x3672 0414.

Unsealed to Full

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

The default code is set to 0xFFFF FFFF.

6.9.7 Charger Class

6.9.7.1 Temperature Table Subclass

See Section 3.2.2, *Charging Voltage and Current Parameters Vary With Temperature()*, for additional details.

6.9.7.2 State of Health Table Subclass

See Section 3.2.3, *Charging Voltage and Current Parameters Variance With StateofHealth()*, for additional details.

6.9.7.3 Charger Information Subclass

Section 6.9.7.3.1 through Section 6.9.7.3.7 describe the default configuration for each charger register by the fuel gauge when it is in control.

6.9.7.3.1 Charger Register 00 Default

This function returns the hex value corresponding to the Input Source Control Register of the charger. See Section 3.1.2, for additional information.

Table 6-13. Charger Register 00 Default Bit Descriptions

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

RSVD = Bits 7:0 are reserved.

6.9.7.3.2 Charger Register 01 Default

This function returns the hex value corresponding to the Power-On Configuration Register of the charger. See Section 3.1.3, for additional information.

Table 6-14. Charger Register 01 Default Bit Descriptions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RSVD	IINLIMIT_2	IINLIMIT_1	IINLIMIT_0	EN_STAT	RSVD	RSVD	HZ_MODE
Default	0	1	0	0	1	1	0	0
	0x48							

RSVD = Bit 7 is reserved.

IINLIMIT[2:0] = Sets the input current limit
000 = USB2.0 host with 100-mA current limit
001 = USB3.0 host with 150-mA current limit
010 = USB2.0 host with 500-mA current limit
011 = USB3.0 host with 900-mA current limit
100 = Charger with 1500-mA current limit
101 = Charger with 2000-mA current limit
110 = External ILIM current limit
111 = No input current limit with internal clamp at 3-A (PTM Mode)

EN_STAT = 0 = Disable STAT function
1 = Enable STAT function

RSVD = Bits 2:1 are reserved.

HZ_MODE = 0 = Not in high impedance mode
1 = High impedance mode

6.9.7.3.3 Charger Register 02 Default

Table 6-15. Charger Register 02 Default Bit Descriptions

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

RSVD = Bits 7:0 are reserved.

6.9.7.3.4 Charger Register 03 Default

This function returns the hex value corresponding to the Pre-Charge/Termination Current Control Register of the charger. See Section 3.1.5, for additional information.

Table 6-16. Charger Register 03 Default Bit Descriptions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
Default	0	0	0	0	0	0	0	0
	0x10							

RSVD = Bits 7:0 are reserved.

6.9.7.3.5 Charger Register 04 Default

This function returns the hex value corresponding to the Voltage Control Register of the charger. See Section 3.1.6, for additional information.

Table 6-17. Charger Register 04 Default Bit Descriptions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	RSVD	RSVD	LOW_CHG	DPDM_EN	RSVD	VINDPM_2	VINDPM_1	VINDPM_0
Default	0	0	0	0	0	0	1	0
	0x02							

RSVD = Bits 7:6 are reserved.

LOW_CHG = 0 = Normal charge current set by Register 0x03 of charger
1 = Low charge current setting of 330 mA

DPDM_EN = 0 = Bit returns to 0 after D+/D– detection is performed
1 = Forces D+/D– detection

RSVD = Bit 3 is reserved.

VINDPM[2:0] = Sets the input VINDPM level
VINDPM[2] = Input VINDPM voltage: 320 mV
VINDPM[1] = Input VINDPM voltage: 160 mV
VINDPM[0] = Input VINDPM voltage: 80 mV

6.9.7.3.6 Charger Register 05 Default

This function returns the hex value corresponding to the Voltage Control Register of the charger. See Section 3.1.7, for additional information.

Table 6-18. Charger Register 05 Default Bit Descriptions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	2XTMR_EN	TMR_1	TMR_2	RSVD	RSVD	RSVD	RSVD	RSVD
Default	0	0	1	0	0	0	0	0
	0x0A							

2XTMR_EN = 0 = Timer not slowed at any time
1 = Timer slowed by 2× when in thermal regulation, VINDPM, or DPPM

TMR[1:0] = Safety timer time limit
00 = 0.75-hour fast charge
01 = 6-hour fast charge
10 = 9-hour fast charge
11 = Disable safety timers

RSVD = Bits 4:0 are reserved.

6.9.7.3.7 Charger Register 06 Default

This function returns the hex value corresponding to the IR Compensation/Thermal Regulation Control Register of the charger. See Section 3.1.8, for additional information.

Table 6-19. Charger Register 06 Default Bit Descriptions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	VOVP_2	VOVP_1	VOVP_0	CLR_VDP	FORCE_BAT DET	FORCE_PTM	RSVD	RSVD
Default	1	1	1	0	0	0	0	0
	0xE0							

VOVP[2:0] = Sets the input OVP level
000 = 6.0 V
001 = 6.5 V
010 = 7.0 V
011 = 8.0 V
100 = 9.0 V
101 = 9.5 V
110 = 10.0 V
111 = 10.5 V

CLR_VDP = 0 = Keep D+ voltage source on during DBP charging
1 = Turn off D+ voltage sourceto release D+ line

FORCE_BATDET = 0 = Enter the battery detection routine only if TERM is true or FORCE PTM is true
1 = Enter the battery detection routine

FORCE_PTM = 0 = PTM is disabled
1 = PTM is enabled

RSVD = Bits 1:0 are reserved.

6.9.7.4 Charger Control Configuration Subclass

6.9.7.4.1 Charger Control Options

See Table 6-20, for details on ***Charger Options***.

Table 6-20. Charger Options Bit Definitions

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Byte	IND_CHG CTL	USB_IN_DEF	CMD_NOT_ REQ	CHGTRM_ HIZ	STEP_EN	SOH_EN	DEFAULT_ OVRD	BYPASS
Default	0	0	1	0	0	0	0	0
	0x20							

IND_CHGCTL = Indirect Charge Control is enabled. See Section 3.4, Indirect Charger Control. True when set.

USB_IN_DEF = Sets the default input current limit for when the charger detects a USB-type input using PSEL or D+/D–.
0 = 100 mA
1 = 500 mA

CMD_NOT_REQ = *GG_CHGRCTL_ENABLE* subcommand is not required for the fuel gauge to control the charger and continuously reset the charger watchdog. True when set.

CHGTRM_HIZ = Charge termination Hi-Z. The charger is configured in Hi-Z mode upon a charge termination event. Hi-Z mode is removed once *Flags() [FC]* bit is cleared. True when set.

STEP_EN = Multi-level charge (MLC) enable. True when set.

SOH_EN = Charge profile based on State-of-Health enabled. True when set.

DEFAULT_OVRD = Allows the charger default values stored in the data flash to override the charge algorithm masks within the gauge during initialization. True when set.

BYPASS = Gas gauge bypass enable for charger control over I^2C. The gas gauge relays all reads and writes directed to the charger and does not autonomously reset the charger watchdog. True when set.

6.9.7.4.2 Charge Disabled Regulation Voltage

Chg Disabled Regulation V is the regulation voltage the charger is configured to when the battery is not being charged.

6.9.7.4.3 Temperature Filter

Temperature Filter is the coefficient used to filter measured temperature for temperature considerations in charge control (for example, JEITA table).

6.9.7.4.4 Force Update Time

Force Update Time is the interval at which the fuel gauge re-writes the present charge voltage and current settings to the charger (0 to disable).

6.9.7.4.5 Shipmode Delay

Shipmode Delay is the time between receiving the *SHIPMODE_ENABLE* command to configuring the charger to enable shipmode.

Application-Specific Information

7.1 Battery Profile Storage and Selection

7.1.1 Common Profile Aspects

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.

7.1.2 Activities Upon Pack Insertion

7.1.2.1 First OCV and Impedance Measurement

The fuel gauge measures its first open-circuit voltage (OCV) via the BAT pin. The *CONTROL_STATUS [OCVCMDCOMP]* bit is set once the OCV measurement is completed. Depending on the load current, the *CONTROL_STATUS [OCVFAIL]* bit indicates whether the OCV reading is valid. From the OCV(SOC) table, the SOC of the inserted battery is found. Then the impedance of the inserted battery is calculated from the measured voltage and the load current:

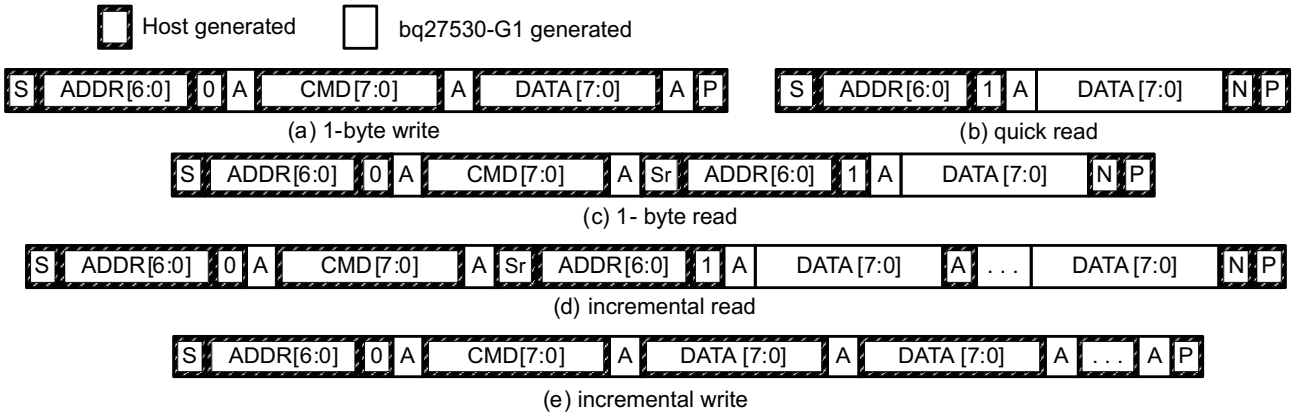
$$Z(SOC) = (OCV(SOC) – V) \div I.$$

This impedance is compared with the impedance of the dynamic profile, ***Pack***, for the same SOC. The *CONTROL_STATUS [INITCOMP]* bit is set afterwards and the OCV command could be issued.

Communications

8.1 I²C Interface

The fuel gauge supports the standard I²C read, incremental read, quick read, 1-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.

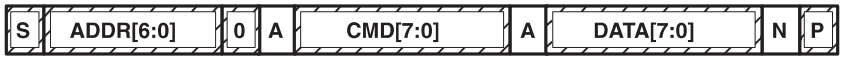


(S = Start, Sr = Repeated Start, A = Acknowledge, N = No Acknowledge , and P = Stop).

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):

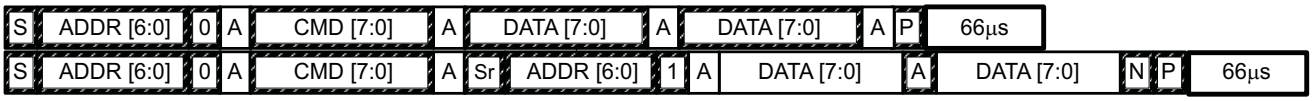
8.2 I²C Time Out

The I²C engine releases both SDA and SCL if the I²C bus is held low for the time defined by ***I²C Timeout*** times 0.5 second. If the fuel gauge was holding the lines, releasing them frees the master to drive the lines. If an external condition is holding either of the lines low, the I²C engine enters the low-power SLEEP mode.

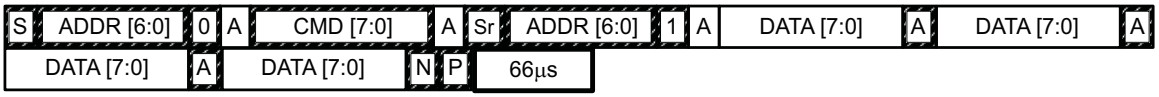
8.3 I²C Command Waiting Time

To make sure the correct results of a command with the 400-kHz I²C operation, a proper waiting time should be added between issuing command and reading results. For subcommands, the following diagram shows the waiting time required between issuing the control command, and reading the status with the exception of checksum and OCV commands. A 100-ms waiting time is required between the checksum command and reading result, and a 1.2-second waiting time is required between the OCV command and result. For read-write standard commands, a minimum of 2 seconds is required to get the result updated. For read-only standard commands, there is no waiting time required, but the host should not issue all standard commands more than two times per second. Otherwise, the gauge could result in a reset issue due to the expiration of the watchdog timer.

The I²C clock stretch could happen in a typical application. A maximum 80-ms clock stretch could be observed during the flash updates. There is up to 270-ms clock stretch after the OCV command is issued.



Waiting time between control subcommand and reading results



Waiting time between continuous reading results

8.4 I²C Clock Stretching

I²C clock stretches can occur during all modes of fuel gauge operation. In the FULLSLEEP (SLEEP or SLEEP+) and HIBERNATE modes, a clock stretch occurs on all I²C bus traffic as the device must wake-up to process the packet. The data flash ***Clock Control Register*** can be configured with a value of 0x09 to minimize the wake-up clock stretch period to approximately 32 µs; while a ***Clock Control Register*** value of 0x00 configures a longer wake-up clock stretch period of approximately 4 ms used in previous bq2750x devices. Any other value used in the ***Clock Control Register*** may result in unpredictable clock stretch timing.

In NORMAL and SLEEP modes, clock stretching only occurs for packets addressed for the fuel gauge. The timing of stretches varies as interactions between the communicating host and the gauge are asynchronous. The I²C clock stretches may occur after start bits, the ACK/NAK bit and first data bit transmit on a host read cycle. The majority of clock stretch periods are small (≤ 4 ms) as the I²C interface peripheral and CPU firmware perform normal data flow control. However, less frequent but more significant clock stretch periods may occur when data flash (DF) is being written by the CPU to update the resistance (Ra) tables and other DF parameters such as ***Qmax***. Due to the organization of DF, updates need to be written in data blocks consisting of multiple data bytes.

An Ra table update requires erasing a single page of DF, programming the updated Ra table and a flag. The potential I²C clock stretching time is 24 ms maximum. This includes 20-ms page erase and 2-ms row programming time (×2 rows). The Ra table updates occur during the discharge cycle and at up to 15 resistance grid points that occur during the discharge cycle.

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

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

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

Reference Schematic

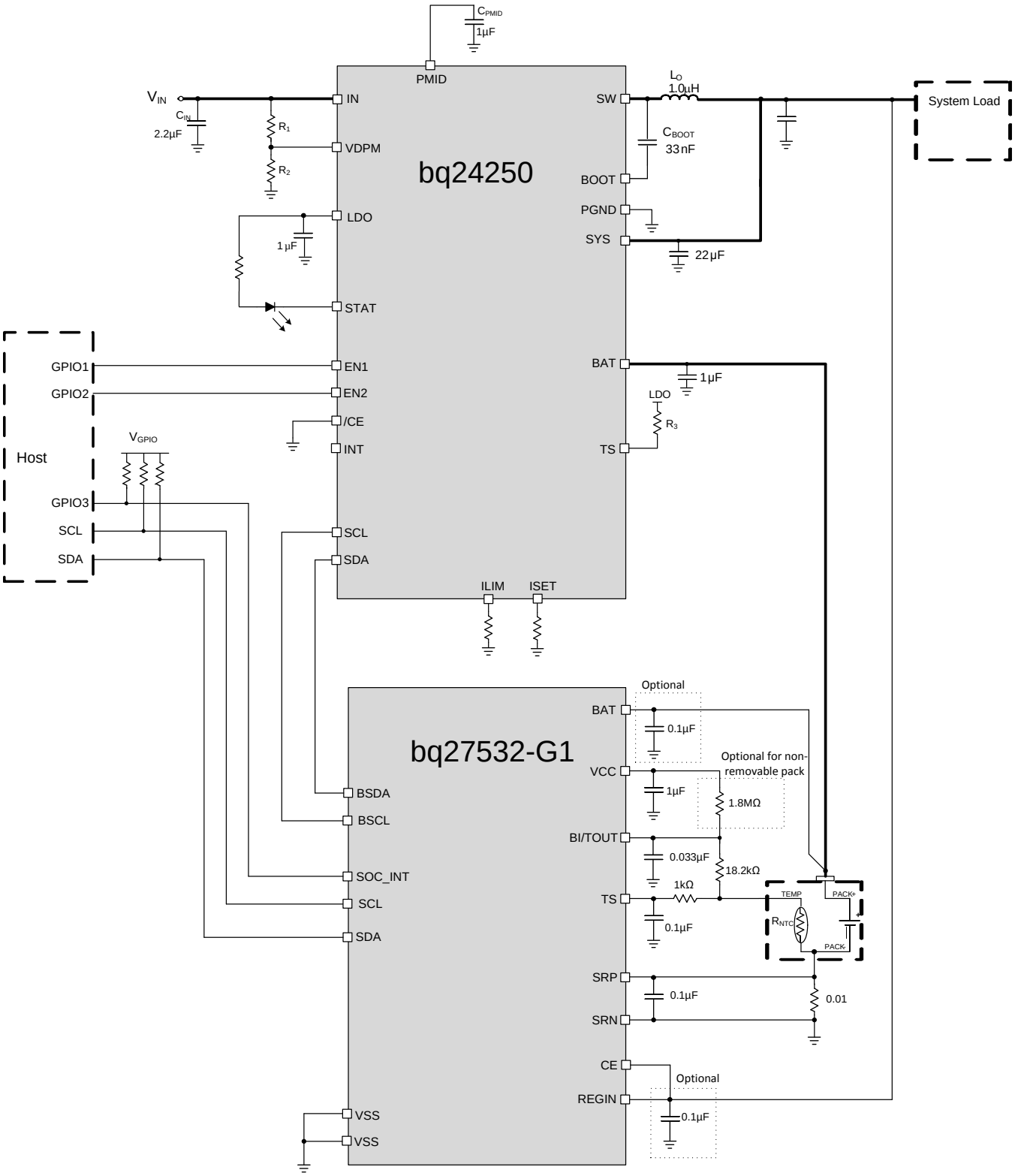


Figure 9-1. Charger Application Schematic

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 abs(adci) ≥ 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 second.
- **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 rate. 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 parameter updates.

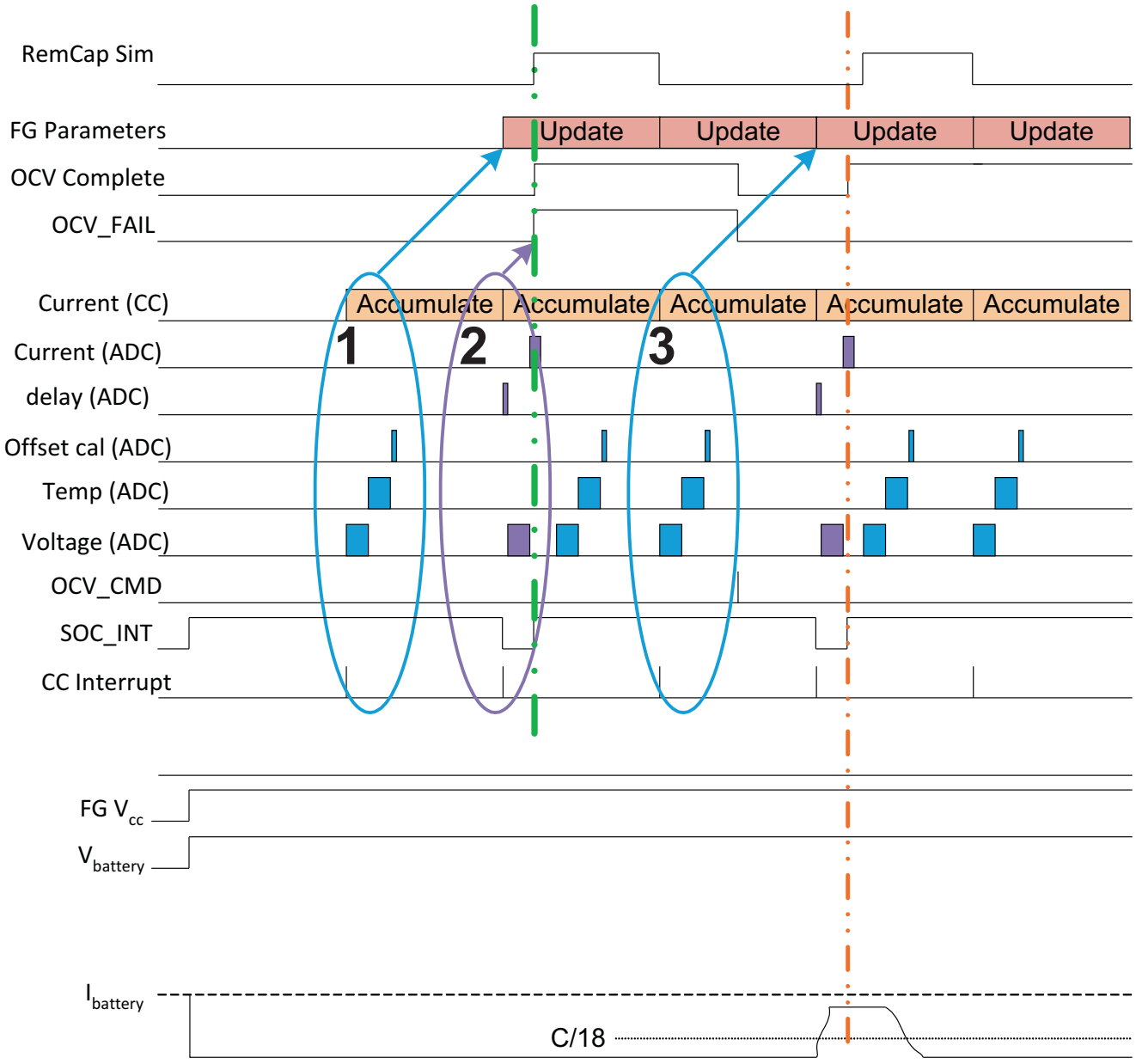


Figure A-1. OCV Timing Sequences

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 rate 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).

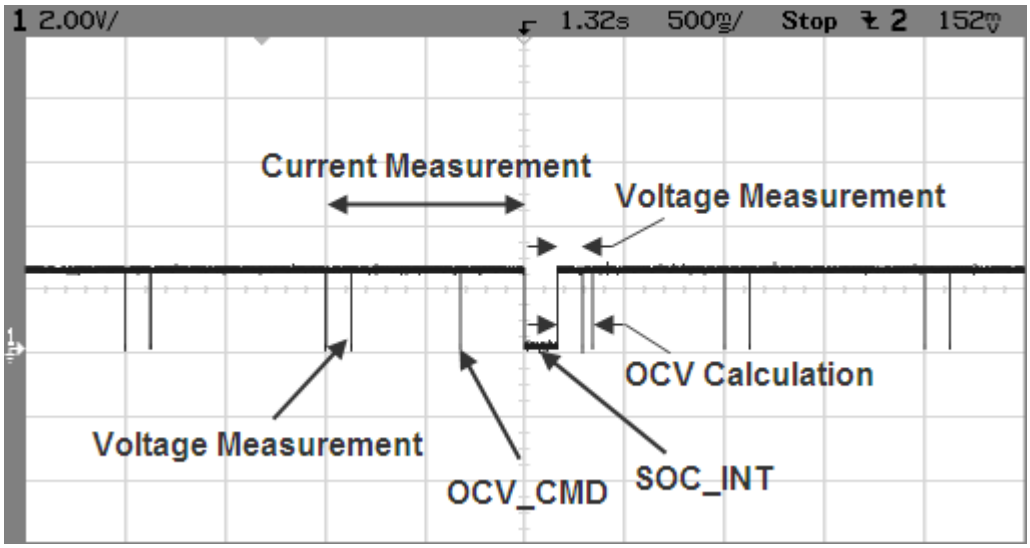


Figure A-2. OCV Calculation Based on OCV Command

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).

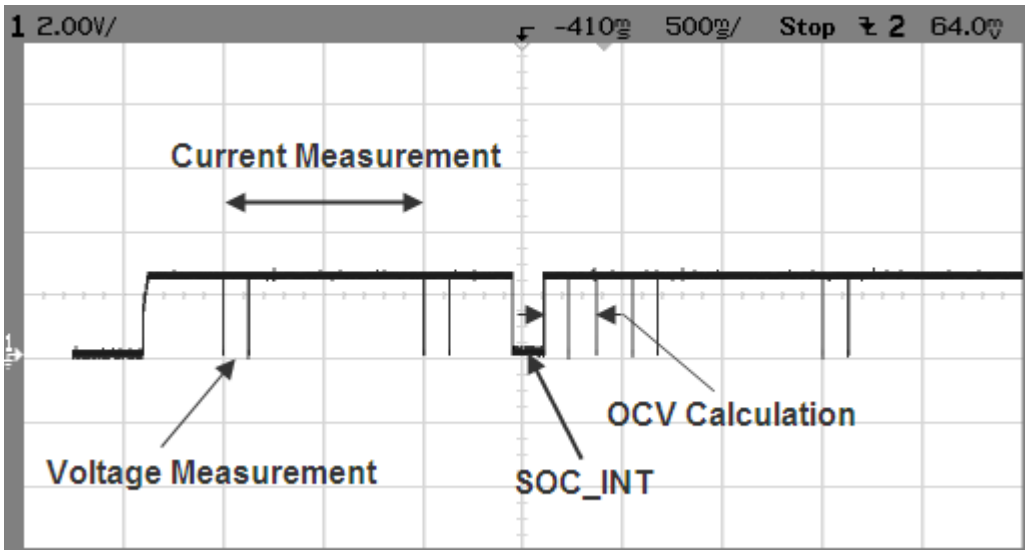


Figure A-3. Initial OCV Taken After POR

Glossary

ACK	Acknowledge character
ADC	Analog-to-digital converter
BCA	Board calibration
BI	Battery insert
CC	Coulomb counter
CCA	Coulomb counter calibration
CCCV	Constant-current, constant-voltage
CE	Chip enable
Charge Mode	Refers to a mode where the gauge reads *AverageCurrent( )* > Chg ***Current*** 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.
CMT	Current Measurement Time
cWh	Centiwatt-hour
DF	Data flash
Discharge Mode	Refers to a mode where the gauge reads *AverageCurrent( )* < ***(–)Dsg Current*** for at least 1 second.
DOD	Depth of discharge in % 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.
DPM	Dynamic power management
EOC	End of charge
FC	Fully charged
FCC	Full charge capacity. Total capacity of the battery compensated for present load current, temperature, and aging effects (reduction in chemical capacity and increase in internal impedance).
FIFO	First in, first out
Flag	This word usually represents a read-only status bit that indicates some action occurred or is occurring. This bit typically cannot be modified. The flags are set and cleared automatically by the gauge.
FVCA	Fast voltage and current acquisition
GPIO	General-purpose input-output
HDQ	High-speed data queue
IC	Integrated circuit
ID	Identification
IINDPM	DPM mode: input current regulation for the charger (bq2425x)
IO	Input or output
IT	Impedance Track™
I^2C	Inter-integrated circuit
LDO	Low dropout
LSB	Least significant bit
LT	Lifetime
MAC	Manufacturer access command or control command
mAh	Milliamp-hour
MLC	Multi-level charge
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
OTG	On-The-Go
POR	Power-on reset
PSEL	Power source selection input of the charger (bq2425x)
QC	Qualification and calculation
Qmax	Maximum chemical capacity
QT	Qualification Time
Relaxation Mode	Refers to a mode where the gauge reads *AverageCurrent( )* < ***Quit Current*** for at least 60 seconds.
RM	Remaining capacity
RW	Read or write
SCL	Serial clock: programmable serial clock used in the I^2C interface
SDA	Serial data: serial data bus in the I^2C interface
SE	Shutdown enable
Set	Refers to a bit in a register becoming a logic HIGH or 1. The bqEvaluation software (EVSW) represents a set bit with the color red.
SOC	State-of-charge in % related to FCC
SOC1	State-of-charge initial
SOCF	State-of-charge final
SOH	State-of-Health
SR	Sensing resistor
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
VINDPM	V_{IN} DPM setting
VIT	Voltage, current, and 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		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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