bq34z100-G1

Technical Reference

TEXAS INSTRUMENTS

Literature Number: SLUUBW5
July 2018

Contents

Preface

Read this First

This manual discusses the modules and peripherals of the bq34z100-G1 device, and how each is used to build a complete battery pack gas gauge and protection solution. For electrical specifications, see the *bq34z100-G1 Wide Range Fuel Gauge with Impedance Track™ Technology Data Sheet* (SLUSBZ5). For application information, see the data sheet as well as the application reports, *bq34z100-G1 High Cell Count and High Capacity Applications* (SLUA760) and *Using I^{2}C Communications With the bq34110, bq35100, and bq34z100-G1 Series of Gas Gauges* (SLUA790).

Notational Conventions

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

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

Trademarks

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

Glossary

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

Introduction

The bq34z100-G1 device is an Impedance Track™ fuel gauge for Li-Ion, PbA, NiMH, and NiCd batteries, and works independently of battery series-cell configurations. Batteries from 3 V to 65 V can be easily supported through an external voltage translation circuit that is controlled automatically to reduce system power consumption.

- Supports Li-Ion, LiFePO4, PbA, NiMH, and NiCd Chemistries
- Capacity Estimation Using Patented Impedance Track™ Technology for Batteries from 3 V to 65 V
  - Aging Compensation
  - Self-Discharge Compensation
- Supports Battery Capacities up to 29 Ah with Standard Configuration Options
- Supports Charge and Discharge Currents up to 32 A with Standard Configuration Options
- External NTC Thermistor Support
- Supports Two-Wire I^{2}C and HDQ Single-Wire Communication Interfaces with Host System
- SHA-1/HMAC Authentication
- One- or Four-LED Direct Display Control
- Five-LED and Higher Display Through Port Expander
- Reduced Power Modes (Typical Battery Pack Operating Range Conditions)
  - Normal Operation: < 145-µA Average
  - Sleep: < 84-µA Average
  - Full Sleep: < 30-µA Average
- Package: 14-Pin TSSOP

The bq34z100-G1 device accurately predicts the battery capacity and other operational characteristics of a single cell or multiple rechargeable cell blocks, which are voltage balanced when resting. The device supports various Li-Ion, Lead Acid (PbA), Nickel Metal Hydride (NiMH), and Nickel Cadmium (NiCd) chemistries, and can be interrogated by a host processor to provide cell information, such as remaining capacity, full charge capacity, and average current.

Information is accessed through a series of commands called Standard Commands (see *Section 2.1*). Further capabilities are provided by the additional Extended Commands set (see *Section 2.2*). Both sets of commands, indicated by the general format *Command()*, are used to read and write information contained within the bq34z100-G1 device's control and status registers, as well as its data flash locations. Commands are sent from host to gauge using the bq34z100-G1 serial communications engines, HDQ and I^{2}C, and can be executed during application development, pack manufacture, or end-equipment operation.

Cell information is stored in the bq34z100-G1 in non-volatile flash memory. Many of these data flash locations are accessible during application development and pack manufacture. They cannot, generally, be accessed directly during end-equipment operation. Access to these locations is achieved by using the bq34z100-G1 device's companion evaluation software, through individual commands, or through a sequence of data-flash-access commands. To access a desired data flash location, the correct data flash subclass and offset must be known.

The bq34z100-G1 provides 32 bytes of user-programmable data flash memory. This data space is accessed through a data flash interface. For specifics on accessing the data flash, refer to *Section 2.2.33*.

The key to the bq34z100-G1 device's high-accuracy gas gauging prediction is Texas Instrument's proprietary Impedance Track algorithm. This algorithm uses voltage measurements, characteristics, and properties to create state-of-charge predictions that can achieve accuracy with as little as 1% error across a wide variety of operating conditions.

The bq34z100-G1 measures charge/discharge activity by monitoring the voltage across a small-value series sense resistor connected in the low side of the battery circuit. When an application's load is applied, cell impedance is measured by comparing its Open Circuit Voltage (OCV) with its measured voltage under loading conditions.

The bq34z100-G1 can use an NTC thermistor (default is Semitec 103AT or Mitsubishi BN35-3H103FB-50) for temperature measurement, or can also be configured to use its internal temperature sensor. The bq34z100-G1 uses temperature to monitor the battery-pack environment, which is used for fuel gauging and cell protection functionality.

To minimize power consumption, the bq34z100-G1 has three power modes: NORMAL, SLEEP, and FULL SLEEP. The bq34z100-G1 passes automatically between these modes, depending upon the occurrence of specific events.

Multiple modes are available for configuring from one to 16 LEDs as an indicator of remaining state of charge. More than four LEDs require the use of one or two inexpensive SN74HC164 shift register expanders.

A SHA-1/HMAC-based battery pack authentication feature is also implemented on the bq34z100-G1. When the IC is in UNSEALED mode, authentication keys can be (re)assigned. A scratch pad area is used to receive challenge information from a host and to export SHA-1/HMAC encrypted responses. See the *Section 4.1* section for further details.

Chapter 2

Data Commands

2.1 Standard Data Commands

The bq34z100-G1 uses a series of 2-byte standard commands to enable host reading and writing of battery information. Each standard command has an associated command-code pair, as indicated in Table 2-1. Because each command consists of two bytes of data, two consecutive HDQ/I^{2}C transmissions must be executed to initiate the command function and to read or write the corresponding two bytes of data. Standard commands are accessible in NORMAL operation. Also, two block commands are available to read Manufacturer Name and Device Chemistry. Read/Write permissions depend on the active access mode.

Table 2-1. Commands

NAME		COMMAND CODE	UNIT	SEALED ACCESS	UNSEALED ACCESS
Control()	CNTL	0x00/0x01	N/A	R/W	R/W
StateOfCharge()	SOC	0x02	%	R	R
MaxError()	ME	0x03	%	R	R
RemainingCapacity()	RM	0x04/0x05	mAh	R	R
FullChargeCapacity()	FCC	0x06/0x07	mAh	R	R
Voltage()	VOLT	0x08/0x09	mV	R	R
AverageCurrent()	AI	0x0A/0x0B	mA	R	R
Temperature()	TEMP	0x0C/0x0D	0.1ºK	R	R
Flags()	FLAGS	0x0E/0x0F	N/A	R	R
Current()	I	0x10/0x11	mA	R	R
FlagsB()	FLAGSB	0x12/0x13	N/A	R	R

2.1.1 Control(): 0x00/0x01

Issuing a *Control()* command requires a subsequent two-byte subcommand. These additional bytes specify the particular control function desired. The *Control()* command allows the host to control specific features of the bq34z100-G1 during normal operation, and additional features when the bq34z100-G1 is in different access modes, as described in Table 2-2.

Table 2-2. Control() Subcommands

CNTL FUNCTION	CNTL DATA	SEALED ACCESS	DESCRIPTION
CONTROL_STATUS	0x0000	Yes	Reports the status of key features.
DEVICE_TYPE	0x0001	Yes	Reports the device type of 0x100 (indicating bq34z100-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
RESET_DATA	0x0005	Yes	Returns reset data
PREV_MACWRITE	0x0007	Yes	Returns previous *Control()* command code
CHEM_ID	0x0008	Yes	Reports the chemical identifier of the Impedance Track configuration
BOARD_OFFSET	0x0009	Yes	Forces the device to measure and store the board offset
CC_OFFSET	0x000A	Yes	Forces the device to measure the internal CC offset

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

CNTL FUNCTION	CNTL DATA	SEALED ACCESS	DESCRIPTION
CC_OFFSET_SAVE	0x000B	Yes	Forces the device to store the internal CC offset
DF_VERSION	0x000C	Yes	Reports the data flash version on the device
SET_FULLSLEEP	0x0010	Yes	Set the [FULLSLEEP] bit in the control register to 1
STATIC_CHEM_CHKSUM	0x0017	Yes	Calculates chemistry checksum
SEALED	0x0020	No	Places the device in SEALED access mode
IT_ENABLE	0x0021	No	Enables the Impedance Track algorithm
CAL_ENABLE	0x002D	No	Toggle CALIBRATION mode enable
RESET	0x0041	No	Forces a full reset of the bq34z100-G1
EXIT_CAL	0x0080	No	Exit CALIBRATION mode
ENTER_CAL	0x0081	No	Enter CALIBRATION mode
OFFSET_CAL	0x0082	No	Reports internal CC offset in CALIBRATION mode

2.1.1.1 CONTROL_STATUS: 0x0000

Instructs the fuel gauge to return status information to Control addresses 0x00/0x01. The status word includes the following information.

Table 2-3. CONTROL_STATUS Flags

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	RSVD	FAS	SS	CALEN	CCA	BCA	CSV	RSVD
Low Byte	RSVD	RSVD	FULLSLEEP	SLEEP	LDMD	RUP_DIS	VOK	QEN

Legend: RSVD = Reserved

FAS: Status bit that indicates the bq34z100-G1 is in FULL ACCESS SEALED state. Active when set.

SS: Status bit that indicates the bq34z100-G1 is in the SEALED State. Active when set.

CALEN: Status bit that indicates the bq34z100-G1 calibration function is active. True when set.
Default is 0.

CCA: Status bit that indicates the bq34z100-G1 Coulomb Counter Calibration routine is active. Active when set.

BCA: Status bit that indicates the bq34z100-G1 Board Calibration routine is active. Active when set.

CSV: Status bit that indicates a valid data flash checksum has been generated. Active when set.

FULLSLEEP: Status bit that indicates the bq34z100-G1 is in FULL SLEEP mode. True when set. The state can only be detected by monitoring the power used by the bq34z100-G1 because any communication will automatically clear it.

SLEEP: Status bit that indicates the bq34z100-G1 is in SLEEP mode. True when set.

LDMD: Status bit that indicates the bq34z100-G1 Impedance Track algorithm using constant-power mode. True when set. Default is 0 (CONSTANT CURRENT mode).

RUP_DIS: Status bit that indicates the bq34z100-G1 Ra table updates are disabled. True when set.

VOK: Status bit that indicates cell voltages are OK for Qmax updates. True when set.

QEN: Status bit that indicates the bq34z100-G1 Qmax updates are enabled. True when set.

2.1.1.2 DEVICE TYPE: 0x0001

Instructs the fuel gauge to return the device type to addresses 0x00/0x01.

2.1.1.3 FW_VERSION: 0x0002

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

2.1.1.4 HW_VERSION: 0x0003

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

2.1.1.5 RESET_DATA: 0x0005

Instructs the fuel gauge to return the number of resets performed to addresses 0x00/0x01.

2.1.1.6 PREV_MACWRITE: 0x0007

Instructs the fuel gauge to return the previous command written to addresses 0x00/0x01. The value returned is limited to less than 0x0020.

2.1.1.7 CHEM ID: 0x0008

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

2.1.1.8 BOARD_OFFSET: 0x0009

Instructs the fuel gauge to calibrate board offset. During board offset calibration the [BCA] bit is set.

2.1.1.9 CC_OFFSET: 0x000A

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

2.1.1.10 CC_OFFSET_SAVE: 0x000B

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

2.1.1.11 DF_VERSION: 0x000C

Instructs the fuel gauge to return the data flash version to addresses 0x00/0x01.

2.1.1.12 SET_FULLSLEEP: 0x0010

Instructs the fuel gauge to set the FULLSLEEP bit in the Control Status register to 1. This allows the gauge to enter the FULL SLEEP power mode after the transition to SLEEP power state is detected. In FULL SLEEP mode, less power is consumed by disabling an oscillator circuit used by the communication engines. For HDQ communication, one host message will be dropped. For I^2C communications, the first I^2C message will incur a 6-ms–8-ms clock stretch while the oscillator is started and stabilized. A communication to the device in FULL SLEEP will force the part back to the SLEEP mode.

2.1.1.13 STATIC_CHEM_DF_CHKSUM: 0x0017

Instructs the fuel gauge to calculate chemistry checksum as a 16-bit unsigned integer sum of all static chemistry data. The most significant bit (MSB) of the checksum is masked yielding a 15-bit checksum. This checksum is compared with the value stored in the data flash Static Chem DF Checksum. If the value matches, the MSB will be cleared to indicate a pass. If it does not match, the MSB will be set to indicate a failure.

2.1.1.14 SEALED: 0x0020

Instructs the fuel gauge to transition from UNSEALED state to SEALED state. The fuel gauge should always be set to SEALED state for use in customer's end equipment.

2.1.1.15 IT ENABLE: 0x0021

Forces the fuel gauge to begin the Impedance Track algorithm, sets Bit 2 of ***UpdateStatus*** and causes the [VOK] and [QEN] flags to be set in the CONTROL STATUS register. [VOK] is cleared if the voltages are not suitable for a Qmax update. Once set, [QEN] cannot be cleared. This command is only available when the fuel gauge is UNSEALED and is typically enabled at the last step of production after the system test is completed.

2.1.1.16 CAL_ENABLE: 0x002D

Instructs the fuel gauge to enable entry and exit to CALIBRATION mode.

2.1.1.17 RESET: 0x0041

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

2.1.1.18 EXIT_CAL: 0x0080

Instructs the fuel gauge to exit CALIBRATION mode.

2.1.1.19 ENTER_CAL: 0x0081

Instructs the fuel gauge to enter CALIBRATION mode.

2.1.1.20 OFFSET_CAL: 0x0082

Instructs the fuel gauge to perform offset calibration.

2.1.2 StateOfCharge(): 0x02

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

2.1.3 MaxError(): 0x03

This read-only command returns an unsigned integer value of the expected margin of error, in %, in the state-of-charge calculation, with a range of 1% to 100%. *MaxError()* is incremented internally by 0.05% for every increment of ***CycleCount*** after the last ***QMAX*** update. *MaxError()* is incremented in the display by 1% for each increment of ***CycleCount***.

Table 2-4. *MaxError()* **Updates**

Event	*MaxError()* Setting
Full reset	Set to 100%
QMAX and Ra table update	Set to 1%
QMAX update	Set to 3%
Ra table update	Set to 5%

If *MaxError()* exceeds the value programmed in ***Max Error Limit***, then *[CF]* in *ControlStatus()* is set. Only when *MaxError()* returns below this value will *[CF]* be cleared.

2.1.4 RemainingCapacity(): 0x04/0x05

This read-only command pair returns the compensated battery capacity remaining. Unit is 1 mAh per bit.

2.1.5 FullChargeCapacity(): 0x06/07

This read-only command pair returns the compensated capacity of the battery when fully charged with units of 1 mAh per bit. However, if ***PackConfiguration [SCALED]*** is set then the units have been scaled through the calibration process. The actual scale is not set in the device and ***SCALED*** is just an indicator flag. *FullChargeCapacity()* is updated at regular intervals under the control of the Impedance Track algorithm.

2.1.6 Voltage(): 0x08/0x09

This read-word command pair returns an unsigned integer value of the measured battery voltage in mV with a range of 0 V to 65535 mV.

2.1.7 AverageCurrent(): 0x0A/0x0B

This read-only command pair returns a signed integer value that is the average current flowing through the sense resistor. It is updated every 1 second with units of 1 mA per bit. However, if ***PackConfiguration [SCALED]*** is set then the units have been scaled through the calibration process. The actual scale is not set in the device and ***SCALED*** is just an indicator flag.

2.1.8 Temperature(): 0x0C/0x0D

This read-only command pair returns an unsigned integer value of the temperature, in units of 0.1ºK, measured by the gas gauge and has a range of 0 to 6553.5ºK. The source of the measured temperature is configured by the *[TEMPS]* bit in the ***Pack Configuration*** register.

Table 2-5. Temperature Sensor Selection

TEMPS	TEMPERATURE() SOURCE
0	Internal Temperature Sensor
1	TS Input (default)

2.1.9 Flags(): 0x0E/0x0F

This read-only command pair returns the contents of the Gas Gauge Status register, depicting current operation status.

Table 2-6. Flags Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	OTC	OTD	BATHI	BATLOW	CHG_INH	XCHG	FC	CHG
Low Byte	OCVTAKEN	RSVD	RSVD	CF	RSVD	SOC1	SOCF	DSG

Legend: **RSVD** = Reserved

OTC: Overtemperature in Charge condition is detected. True when set.

OTD: Overtemperature in Discharge condition is detected. True when set.

BATHI: Battery High bit that indicates a high battery voltage condition. Refer to the data flash ***Cell BH*** parameters for threshold settings. True when set.

BATLOW: Battery Low bit that indicates a low battery voltage condition. Refer to the data flash ***Cell BL*** parameters for threshold settings. True when set.

CHG_INH: Charge Inhibit: unable to begin charging. Refer to the data flash ***[Charge Inhibit Temp Low, Charge Inhibit Temp High]*** parameters for threshold settings. True when set.

XCHG: Charging not allowed.

FC: Full charge is detected. FC is set when charge termination is reached and ***FC Set%*** = –1 (see *Section 3.7* for details) or *StateOfCharge()* is larger than ***FC Set%*** and ***FC Set%*** is not –1. True when set.

CHG: (Fast) charging allowed. True when set.

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

CF: Condition Flag indicates that the gauge needs to run through an update cycle to optimize accuracy.

SOC1: State-of-Charge Threshold 1 reached. True when set.

SOCF: State-of-Charge Threshold Final reached. True when set.

DSG: Discharging detected. True when set.

2.1.10 FlagsB(): 0x12/0x13

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

Table 2-7. Flags B Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	SOH	LIFE	FIRSTDOD	RSVD	RSVD	DODEOC	DTRC	RSVD
Low Byte	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD

Legend: **RSVD** = Reserved

SOH: *StateOfHealth()* calculation is active.

LIFE: Indicates that LiFePO4 RELAX is enabled.

FIRSTDOD: Set when RELAX mode is entered and then cleared upon valid DOD measurement for QMAX update or RELAX exit.

DODEOC: DOD at End-of-Charge is updated.

DTRC: Indicates *RemainingCapacity()* has been changed due to change in temperature.

2.1.11 Current(): 0x10/0x11

This read-only command pair returns a signed integer value that is the current flow through the sense resistor. It is updated every 1 s with units of 1 mA; however, if ***PackConfiguration [SCALED]*** is set, then the units have been scaled through the calibration process. The actual scale is not set in the device and ***SCALED*** is just an indicator flag.

2.2 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 2-8. For details on the SEALED and UNSEALED states, refer to *Section 2.2.33.3*.

Table 2-8. Extended Commands

NAME		COMMAND CODE	UNIT	SEALED ACCESS[1], [2]	UNSEALED ACCESS[1], [2]
AverageTimeToEmpty()	ATTE	0x18/0x19	Minutes	R	R
AverageTimeToFull()	ATTF	0x1A/0x1B	Minutes	R	R

[1] SEALED and UNSEALED states are entered via commands to CNTL 0x00/0x01.
[2] In SEALED mode, data flash *cannot* be accessed through commands 0x3E and 0x3F.

Table 2-8. Extended Commands (continued)

NAME		COMMAND CODE	UNIT	SEALED ACCESS[1], [2]	UNSEALED ACCESS[1], [2]
PassedCharge()	PCHG	0x1C/0x1D	mAh	R	R
DoD0Time()	DoD0T	0x1E/0x1F	Minutes	R	R
AvailableEnergy()	AE	0x24/0x25	10 mW/h	R	R
AveragePower()	AP	0x26/0x27	10 mW	R	R
Serial Number	SERNUM	0x28/0x29	N/A	R	R
Internal_Temperature()	INTTEMP	0x2A/0x2B	0.1°K	R	R
CycleCount()	CC	0x2C/0x2D	Counts	R	R
StateOfHealth()	SOH	0x2E/0x2F	%	R	R
ChargeVoltage()	CHGV	0x30/0x31	mV	R	R
ChargeCurrent()	CHGI	0x32/0x33	mA	R	R
PackConfiguration()	PKCFG	0x3A/0x3B	N/A	R	R
DesignCapacity()	DCAP	0x3C/0x3D	mAh	R	R
DataFlashClass() (2)	DFCLS	0x3E	N/A	N/A	R/W
DataFlashBlock() (2)	DFBLK	0x3F	N/A	R/W	R/W
Authenticate()/BlockData()	A/DF	0x40…0x53	N/A	R/W	R/W
AuthenticateCheckSum()/BlockData( )	ACKS/DFD	0x54	N/A	R/W	R/W
BlockData()	DFD	0x55…0x5F	N/A	R	R/W
BlockDataCheckSum()	DFDCKS	0x60	N/A	R/W	R/W
BlockDataControl()	DFDCNTL	0x61	N/A	N/A	R/W
GridNumber()	GN	0x62	N/A	R	R
LearnedStatus()	LS	0x63	N/A	R	R
DoD@EoC()	DEOC	0x64/0x65	N/A	R	R
QStart()	QS	0x66/0x67	mAh	R	R
TrueRC()	TRC	0x68/0x69	mAh	R	R
TrueFCC()	TFCC	0x6A/0x6B	mAh	R	R
StateTime()	ST	0x6C/0x6D	s	R	R
QMaxPassedQ	QPC	0x6E/0x6F	mAh	R	R
DOD0()	DOD0	0x70/0x71	HEX#	R	R
QmaxDOD0()	QD0	0x72/0x73	N/A	R	R
QmaxTime()	QT	0x74/0x75	h/16	R	R
Reserved	RSVD	0x76...0x7F	N/A	R	R

2.2.1 AverageTimeToEmpty(): 0x18/0x19

This read-only command pair returns an unsigned integer value of the predicted remaining battery life at the present rate of discharge (using *AverageCurrent()*), in minutes. A value of 65535 indicates that the battery is not being discharged.

2.2.2 AverageTimeToFull(): 0x1A/0x1B

This read-only command pair returns an unsigned integer value of predicted remaining time until the battery reaches full charge, in minutes, based upon *AverageCurrent()*. The computation should account for the taper current time extension from the linear TTF computation based on a fixed *AverageCurrent()* rate of charge accumulation. A value of 65535 indicates the battery is not being charged.

2.2.3 PassedCharge(): 0x1C/0x1D

This read-only command pair returns a signed integer, indicating the amount of charge passed through the sense resistor since the last IT simulation in mAh.

2.2.4 DOD0Time(): 0x1E/0x1F

This read-only command pair returns the time since the last DOD0 update.

2.2.5 AvailableEnergy(): 0x24/0x25

This read-only command pair returns an unsigned integer value of the predicted charge or energy remaining in the battery. The value is reported in units of mWh.

2.2.6 AveragePower(): 0x26/0x27

This read-word command pair returns an unsigned integer value of the average power of the current discharge. A value of 0 indicates that the battery is not being discharged. The value is reported in units of mW.

2.2.7 SerialNumber(): 0x28/0x29

This read-only command pair returns the assigned pack serial number programmed in ***Serial Number***.

2.2.8 InternalTemperature(): 0x2A/0x2B

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

2.2.9 CycleCount(): 0x2C/0x2D

This read-only command pair 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*.

2.2.10 StateOfHealth(): 0x2E/0x2F

This read-only command pair returns an unsigned integer value, expressed as a percentage of the ratio of predicted FCC (25°C, SOH current rate) over the *DesignCapacity()*. The FCC (25°C, SOH current rate) is the calculated full charge capacity at 25°C and the SOH current rate that is specified in the data flash (State of Health Load). The range of the returned SOH percentage is 0x00 to 0x64, indicating 0% to 100%, correspondingly.

2.2.11 ChargeVoltage(): 0x30/0x31

This read-only command pair returns the recommended charging voltage output from the JEITA charging profile. It is updated automatically based on the present temperature range.

2.2.12 ChargeCurrent(): 0x32/0x33

This read-only command pair returns the recommended charging current output from the JEITA charging profile. It is updated automatically based on the present temperature range.

2.2.13 PackConfiguration(): 0x3A/0x3B

This read-only command pair allows the host to read the configuration of selected features of the device pertaining to various features.

2.2.14 DesignCapacity(): 0x3C/0x3D

This read-only command pair returns theoretical or nominal capacity of a new pack. The value is stored in ***Design Capacity*** and is expressed in mAh.

2.2.15 DataFlashClass(): 0x3E

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

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

2.2.16 DataFlashBlock(): 0x3F

UNSEALED Access: If ***BlockDataControl*** has been set to 0x00, this command directs which data flash block will be accessed by the *BlockData()* command. Writing a 0x00 to *DataFlashBlock()* specifies the *BlockData()* command will transfer authentication data. Issuing a 0x01 instructs the *BlockData()* command to transfer ***Manufacturer Data***.

SEALED Access: This command directs which data flash block will be accessed by the *BlockData()* command. Writing a 0x00 to *DataFlashBlock()* specifies that the *BlockData()* command will transfer authentication data. Issuing a 0x01 instructs the *BlockData()* command to transfer ***Manufacturer Data***.

2.2.17 AuthenticateData/BlockData(): 0x40…0x53

UNSEALED Access: This data block has a dual function: It is used for the authentication challenge and response and is part of the 32-byte data block when accessing data flash.

SEALED Access: This data block has a dual function: It is used for authentication challenge and response, and is part of the 32-byte data block when accessing the ***Manufacturer Data***.

2.2.18 AuthenticateChecksum/BlockData(): 0x54

UNSEALED Access: This byte holds the authentication checksum when writing the authentication challenge to the device, and is part of the 32-byte data block when accessing data flash.

SEALED Access: This byte holds the authentication checksum when writing the authentication challenge to the device, and is part of the 32-byte data block when accessing ***Manufacturer Data***.

2.2.19 BlockData(): 0x55…0x5F

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

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

2.2.20 BlockDataChecksum(): 0x60

UNSEALED Access: This byte contains the checksum on the 32 bytes of block data read or written to data flash.

SEALED Access: This byte contains the checksum for the 32 bytes of block data written to ***Manufacturer Data***.

2.2.21 BlockDataControl(): 0x61

UNSEALED Access: This command is used to control 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()*.

2.2.22 GridNumber(): 0x62

This read-only command returns the active grid point. This data is only valid during DISCHARGE mode when [R_DIS] = 0. If [R_DIS] = 1 or not discharging, this value is not updated.

2.2.23 LearnedStatus(): 0x63

This read-only command returns the learned status of the resistance table.

Table 2-9. LearnedStatus(): 0x63

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSVD	RSVD	RSVD	RSVD	Qmax	ITEN	CF1	CF0

Legend: **RSVD** = Reserved

QMax (Bit 3): QMax updates in the field.
0 = QMax has not been updated in the field.
1 = QMax updated in the field.
ITEN (Bit 2): IT enable
0 = IT is disabled.
1 = IT is enabled.
CF1, CF0 (Bit 1–0): QMax Status
0,0 = Battery is OK.
0,1 = QMax is first updated in the learning cycle.

2.2.24 Dod@Eoc(): 0x64/0x65

This read-only command pair returns the depth of discharge (DOD) at the end of charge.

2.2.25 QStart(): 0x66/0x67

This read-only command pair returns the initial capacity calculated from IT simulation.

2.2.26 TrueRC(): 0x68/0x69

This read-only command pair returns the True remaining capacity from IT simulation without the effects of the smoothing function.

2.2.27 TrueFCC(): 0x6A/0x6B

This read-only command pair returns the True full charge capacity from IT simulation without the effects of the smoothing function.

2.2.28 StateTime(): 0x6C/0x6D

This read-only command pair returns the time past since last state change (DISCHARGE, CHARGE, REST).

2.2.29 QmaxPassedQ(): 0x6E/0x6F

This read-only command pair returns the passed capacity since the last Qmax DOD update.

2.2.30 DOD0(): 0x70/0x71

This unsigned integer indicates the depth of discharge during the most recent OCV reading.

2.2.31 QmaxDod0(): 0x72/0x73

This read-only command pair returns the DOD0 saved to be used for next QMax update of Cell 1. The value is only valid when [VOK] = 1.

2.2.32 QmaxTime(): 0x74/0x75

This read-only command pair returns the time since the last Qmax DOD update.

2.2.33 Data Flash Interface

NOTE: For the data flash summary, see *Data Flash Summary*

2.2.33.1 Accessing Data Flash

The bq34z100-G1 data flash is a non-volatile memory that contains bq34z100-G1 initialization, default, cell status, calibration, configuration, and user information. The data flash can be accessed in several different ways, depending on in what mode the bq34z100-G1 is operating and what data is being accessed.

Commonly accessed data flash memory locations, frequently read by a host, are conveniently accessed through specific instructions described in *Chapter 2*. These commands are available when the bq34z100-G1 is either in UNSEALED or SEALED modes.

Most data flash locations, however, can only be accessible in UNSEALED mode by use of the bq34z100-G1 evaluation software or by data flash block transfers. These locations should be optimized and/or fixed during the development and manufacture processes. They become part of a Golden Image File and can then be written to multiple 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 host or changed directly. This is accomplished by sending the set-up command *BlockDataControl()* (code 0x61) with data 0x00. Up to 32 bytes of data can be read directly from the *BlockData()* command locations 0x40…0x5F, externally altered, then re-written to the *BlockData()* command space. Alternatively, specific locations can be read, altered, and re-written 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()* (command number 0x60).

Occasionally, a data flash class will be larger than the 32-byte block size. In this case, the *DataFlashBlock()* command is used to designate which 32-byte block in which 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 48, 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 + 48 *modulo* 32 = 0x40 + 16 = 0x40 + 0x10 = 0x50; for example, to modify ***[VOLTSEL]*** in ***Pack Configuration*** from 0 to 1 to enable the external voltage measurement option.

NOTE: The subclass ID and Offset values are in decimal format in the documentation and in bqSTUDIO. The example below shows these values converted to hexadecimal. For example, the ***Pack Configuration*** subclass is d64 = 0x40.

1. Unseal the device using the *Control()* (0x00/0x01) 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 (*Pack Configuration* Subclass) using the *DataFlashClass()* command (0x3E) to access the Registers subclass.

   (*wr 0x3E 0x40*)
4. Write the block offset location using *DataFlashBlock()* command (0x3F). To access data located at offset 0 to 31, use offset = 0x00. To access data located at offset 32 to 63, use offset = 0x01, and so on, as necessary.

   For example, *Pack Configuration* (offset = 0) 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, *Pack Configuration* (offset = 0) is located at 0x40 and 0x41; however, ***[VOLTSEL]*** is in the MSB so only 0x40 needs to be read. Read 1 byte starting at the 0x40 address.

   (*rd 0x40 old_Pack_Configuration_MSB*)

   In this example, assume ***[VOLTSEL]*** = 0 (default).
6. To read the 1-byte checksum, use the *BlockDataChecksum()* command (0x60).

   (*rd 0x60 OLD_checksum*)
7. In this example, set ***[VOLTSEL]*** by setting Bit 3 of *old_Pack_Configuration_MSB* to create *new_Pack_Configuration_MSB*.
8. The new value for *new_Pack_Configuration_MSB* can be written by writing to the specific offset location.

   For example, to write 1-byte *new_Pack_Configuration_MSB* to ***Pack Configuration*** (offset=0) located at 0x40, use command (*wr 0x4B new_Pack_Configuration_MSB*).
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_Pack_Configuration_MSB*), 256)

   b. NEW_checksum = 255 – mod (temp + *new_Pack_Configuration_MSB*, 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, the gauge will automatically become sealed again after RESET.
11. If not previously sealed, then seal the gauge by using *Control*(0x0020).

    (*wr 0x00 0x20 0x00*)

Reading and writing subclass data are block operations 32 bytes in length. Data can be written in shorter block sizes, however. Blocks can be shorter than 32 bytes in length. Writing these blocks back to data flash will not overwrite data that extend beyond the actual block length.

NOTE: None of the data written to memory is bounded by the bq34z100-G1: The values are not rejected by the gas gauge. Writing an incorrect value may result in hardware failure due to firmware program interpretation of the invalid data. The data written is persistent, so a power-on reset resolves the fault.

2.2.33.2 Manufacturer Information Block

The bq34z100-G1 contains 32 bytes of user-programmable data flash storage: ***Manufacturer Info Block***. The method for accessing these memory locations is slightly different, depending on if the device is in UNSEALED or SEALED modes.

When in UNSEALED mode and when an “0x00” has been written to *BlockDataControl()*, accessing the Manufacturer Info Block is identical to accessing general data flash locations. First, a *DataFlashClass()* command is used to set 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.

As an example, the data flash location for ***Manufacturer Info Block*** is defined as having a Subclass = 58 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 bq34z100-G1 evaluation software.

When in SEALED mode or when “0x01” *BlockDataControl()* does not contain “0x00”, data flash is no longer available in the manner used in UNSEALED mode. Rather than issuing subclass information, a designated ***Manufacturer Information Block*** is selected with the *DataFlashBlock()* command. Issuing a 0x01, 0x02, or 0x03 with this command causes the corresponding information block (A, B, or C, respectively) to be transferred to the command space 0x40…0x5F for editing or reading by the host. Upon successful writing of checksum information to *BlockDataChecksum()*, the modified block is returned to data flash.

NOTE: ***Manufacturer Info Block A*** is “read only” when in SEALED mode.

2.2.33.3 Access Modes

The bq34z100-G1 provides three security modes that control data flash access permissions according to Table 2-10. Public Access refers to those data flash locations specified in Chapter 7 that are accessible to the user. Private Access refers to reserved data flash locations used by the bq34z100-G1 system. Care should be taken to avoid writing to Private data flash locations when performing block writes in FULL ACCESS mode by following the procedure outlined in *Section 2.2.33.1*.

Table 2-10. Data Flash Access

SECURITY MODE	DF—PUBLIC ACCESS	DF—PRIVATE ACCESS
BOOTROM	N/A	N/A
FULL ACCESS	R/W	R/W
UNSEALED	R/W	R/W
SEALED	R	N/A

Although FULL ACCESS and UNSEALED modes appear identical, FULL ACCESS mode allows the bq34z100-G1 to directly transition to BOOTROM mode and also write access keys. UNSEALED mode does not have these abilities.

2.2.33.4 Sealing/Unsealing Data Flash Access

The bq34z100-G1 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 bq34z100-G1 via the *Control()* command (these keys are unrelated to the keys used for SHA-1/HMAC authentication). The keys must be sent consecutively, with no other data being written to the *Control()* register in between. Note that to avoid conflict, the keys must be different from the codes presented in the CNTL DATA column of Table 2-2 subcommands.

When in SEALED mode, the [SS] bit of *Control Status()* is set, but when the UNSEAL keys are correctly received by the bq34z100-G1, the [SS] bit is cleared. When the full access keys are correctly received, then the *Flags() [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 bytes entered through the *Control()* command is the reverse of what is read from the part. For example, if the 1st and 2nd word of the UnSeal Key 0 returns 0x1234 and 0x5678, then *Control()* should supply 0x3412 and 0x7856 to unseal the part.

Fuel Gauging

3.1 Overview

The bq34z100-G1 measures the cell voltage, temperature, and current to determine the battery SOC based in the Impedance Track algorithm (refer to *Theory and Implementation of Impedance Track Battery Fuel-Gauging Algorithm Application Report* [SLUA450] for more information). The bq34z100-G1 monitors charge and discharge activity by sensing the voltage across a small-value resistor (5 mΩ to 20 mΩ typ.) between the SRP and SRN pins and in-series with the cell. By integrating charge passing through the battery, the cell's 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 the 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* value is taken from a cell manufacturers' data sheet multiplied by the number of parallel cells. The parallel value is also used for the value programmed in ***Design Capacity***. The bq34z100-G1 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.

During normal battery usage there could be instances where a small rise of SOC for a short period of time could occur at the beginning of discharge. The ***[RSOC_HOLD]*** option in ***Pack Configuration C*** prevents SOC rises during discharge. SOC will be held until the calculated value falls below the actual state.

The bq34z100-G1 has two flags accessed by the *Flags()* function that warn when the battery's SOC has fallen to critical levels. 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***. All units are in mAh.

When *RemainingCapacity()* falls below the second capacity threshold, ***SOCF Set Threshold***, the *[SOCF] (State of Charge Final)* flag is set, serving as a final discharge warning. If ***SOCF Set Threshold*** = –1, the flag is inoperative during discharge. Similarly, when *RemainingCapacity()* rises above ***SOCF Clear Threshold*** and the [SOCF] flag has already been set, the *[SOCF]* flag is cleared. All units are in mAh.

The bq34z100-G1 includes charge efficiency compensation that makes use of four *Charge Efficiency* factors to correct for energy lost due to heat. This is commonly used in NiMH and Lead-Acid chemistries and is not always linear with respect to state-of-charge.

3.2 Impedance Track Variables

The bq34z100-G1 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.

3.2.1 Load Mode

Load Mode is used to select either the constant current or constant power model for the Impedance Track algorithm as used in ***Load Select***. See the *Section 3.2.2* section. When ***Load Mode*** is 0, the *Constant Current Model* is used (default). When Load Mode is 1, the *Constant Power Model* is used. The [LDMD] bit of CONTROL_STATUS reflects the status of Load Mode.

3.2.2 Load Select

Load Select defines the type of power or current model to be used to compute load-compensated capacity in the Impedance Track algorithm. If ***Load Mode*** = 0 (Constant Current), then the options presented in Table 3-1 are available.

Table 3-1. 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. However, if this is the first cycle of the gauge, then the present average current is 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 the *AverageCurrent()*
3	Current: based on a low-pass-filtered version of *AverageCurrent()* (τ=14s)
4	Design Capacity/5: C Rate based off of Design Capacity /5 or a C/5 rate in mA.
6	Use the value in ***User_Rate-mA***: This gives a completely user configurable method.

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

Table 3-2. Constant-Power Model Used When Load Mode = 1

LOAD SELECT VALUE	POWER MODEL USED
0 (default)	Average discharge power from previous cycle: There is an internal register that records the average discharge power through each entire discharge cycle. The previous average is stored in 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 on a low-pass-filtered version of *AverageCurrent()* (τ=14s) and *Voltage()*
4	Design Energy/5: C Rate based off of Design Energy /5 or a C/5 rate in mA.
6	Use the value in ***User_Rate-mW/cW***. This gives a completely user-configurable method.

3.2.3 Reserve Cap-mAh

Reserve Cap-mAh determines how much actual remaining capacity exists after reaching 0 *RemainingCapacity()* before ***Terminate Voltage*** is reached. A loaded rate or no-load rate of compensation can be selected for Reserve Cap by setting the [RESCAP] bit in the Pack Configuration register.

3.2.4 Reserve Cap-mWh/cWh

Reserve Cap-mWh determines how much actual remaining capacity exists after reaching 0 *AvailableEnergy()* before ***Terminate Voltage*** is reached. A loaded rate or no-load rate of compensation can be selected for Reserve Cap by setting the [RESCAP] bit in the Pack Configuration register.

3.2.5 Design Energy Scale

Design Energy Scale is used to select the scale/unit of a set of data flash parameters. The value of ***Design Energy Scale*** can be between 1 and 10 only.

When using Design Energy Scale > 1, the value for each of the parameters in Table 3-3 must be adjusted to reflect the new units. See *Section 3.8*.

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

DATA FLASH PARAMETER	DESIGN ENERGY SCALE = 1 (default)	DESIGN ENERGY SCALE >1
Design Energy	mWh	Scaled by Design Energy Scale
Reserve Energy-mWh/cWh	mWh	Scaled by Design Energy Scale

Table 3-3. Data Flash Parameter Scale/Unit-Based on Design Energy Scale (continued)

DATA FLASH PARAMETER	DESIGN ENERGY SCALE = 1 (default)	DESIGN ENERGY SCALE >1
Avg Power Last Run	mW	Scaled by Design Energy Scale
User Rate-mW/cW	mWh	Scaled by Design Energy Scale
T Rise	No Scale	***Scaled by Design Energy Scale***

3.2.6 Dsg Current Threshold

This register is used as a threshold by many functions in the bq34z100-G1 to determine if actual discharge current is flowing into or out of the cell. The default for this register 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.

3.2.7 Chg Current Threshold

This register is used as a threshold by many functions in the bq34z100-G1 to determine if actual charge current is flowing into or out of the cell. The default for this register 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.

3.2.8 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 bq34z100-G1 enters RELAX 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 that should be above the standby current of the host system.

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

1. |*AverageCurrent()*| < |***Quit Current***| for Dsg Relax Time
2. |*AverageCurrent()*| > |***Quit Current***| for Chg Relax Time

After about 6 minutes in RELAX mode, the device attempts to take accurate OCV readings. An additional requirement of dV/dt < 4 μV/s is required for the device to perform Qmax updates. These updates are used in the Impedance Track algorithms. It is critical that the battery voltage be relaxed during OCV readings, and that the current is not higher than C/20 when attempting to go into RELAX mode.

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

3.2.9 Qmax

Qmax Cell 0 contains the maximum chemical capacity of the cell and is determined by comparing states of charge before and after applying the load with the amount of charge passed. It also corresponds to capacity at low rate of discharge, such as C/20 rate. For high accuracy, this value is periodically updated by the device during operation.

Based on the battery cell capacity information, the initial value of chemical capacity should be entered in the ***Qmax Cell 0*** data flash parameter. The Impedance Track algorithm will update this value and maintain it internally in the gauge.

3.2.10 Update Status

The Update Status register indicates the status of the Impedance Track algorithm.

Table 3-4. Update Status Definitions

UPDATE STATUS	STATUS
0x02	Qmax and Ra data are learned, but Impedance Track is not enabled. This should be the standard setting for a Golden Image File.

Table 3-4. Update Status Definitions (continued)

UPDATE STATUS	STATUS
0x04	Impedance Track is enabled but Qmax and Ra data are not yet learned.
0x05	Impedance Track is enabled and only Qmax has been updated during a learning cycle.
0x06	Impedance Track is enabled. Qmax and Ra data are learned after a successful learning cycle. This should be the operation setting for end equipment.

This register should only be updated by the device during a learning cycle or when the *IT_ENABLE()* subcommand is received. Refer to the *Preparing Optimized Default Flash Constants for Specific Battery Types Application Report* (SLUA334B).

3.2.11 Avg I Last Run

The device 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 never be modified. It is only updated by the device when required.

3.2.12 Avg P Last Run

The device 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 device continuously multiplies instantaneous current times *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 device when the required.

3.2.13 Cell Delta Voltage

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

3.2.14 Ra Tables

This data is automatically updated during device operation. No user changes should be made except for reading the values from another pre-learned pack for creating Golden Image Files. Profiles have format *Cell0 R_a M*, where M is the number that indicates state-of-charge to which the value corresponds.

3.2.15 StateOfCharge() Smoothing

When operating conditions change (such as temperature, discharge current, and resistance, and so on), it can lead to large changes of compensated battery capacity and battery capacity remaining. These changes can result in large changes of *StateOfCharge()*. When ***[SmoothEn]*** is enabled in ***Pack Configuration C***, the smoothing algorithm injects gradual changes of battery capacity when conditions vary. This results in a gradual change of *StateOfCharge()* and can provide a better end-user experience for *StateOfCharge()* reporting.

The *RemainingCapacity()*, *FullChargeCapacity()*, and *StateOfCharge()* are modified depending on ***[SmoothEn]***, as below.

[SmoothEn]	*RemainingCapacity()*	*FullChargeCapacity()*	*StateOfCharge()*
0	*TrueRC()*	*TrueFCC()*	*TrueRC()/ TrueFCC()*
1	*FilteredRC()*	*FilteredFCC()*	*FilteredRC()/FilteredFCC()*

3.2.16 Charge Efficiency

Tracking state-of-charge during the charge phase is relatively easy with chemistries such as Li-Ion where essentially none of the applied energy from the charger is lost to heat. However, lead-acid and NiMH chemistries may demonstrate significant losses to heat during charging. Therefore, to more accurately track state of charge and Time-to-Full during the charge phase, the bq34z100-G1 uses four charge-efficiency factors to compensate for charge acceptance. These factors are ***Charge Efficiency***, ***Charge Eff Reduction Rate***, ***Charge Effi Drop Off***, and ***Charge Eff Temperature Compensation***.

The bq34z100-G1 applies the ***Charge Efficiency*** when *RelativeStateOfCharge()* is less than the value stored in ***Charge Efficiency Drop Off***. When *RelativeStateOfCharge()* is > or equal to the value coded in ***Charge Efficiency Drop Off***, ***Charge Efficiency*** and ***Charge Efficiency Reduction Rate*** determine the charge efficiency rate. ***Charge Efficiency Reduction Rate*** defines the percent efficiency reduction per percentage point of *RelativeStateOfCharge()* over ***Charge Efficiency Drop Off***. The ***Charge Efficiency Reduction Rate*** has units of 0.1%. The bq34z100-G1 also adjusts the efficiency factors for temperature. ***Charge Efficiency Temperature Compensation*** defines the percent efficiency reduction per degree C over 25°C. ***Charge Efficiency Temperature Compensation*** has units of 0.01%.

Applying the four factors:

Effective Charge Efficiency % = ***Charge Efficiency*** – ***Charge Eff Reduction Rate*** [*RSOC()* – Charge Effi Drop Off] – Charge Eff Temperature Compensation [Temperature – 25°C] (1)

Where: *RSOC()* ≥ ***Charge Efficiency*** and ***Temperature*** ≥ 25°C (2)

3.2.17 Lifetime Data Logging

The Lifetime Data Logging function helps development and diagnosis with the fuel gauge.

NOTE: IT_ENABLE must be enabled (Command 0x0021) for lifetime data logging functions to be active.

The fuel gauge logs the lifetime data as specified in the ***Lifetime Data*** and ***Lifetime Temp Samples*** data flash subclasses. The data log recordings are controlled by the ***Lifetime Resolution*** data flash subclass.

The Lifetime Data Logging can be started by setting the ***IT_ENABLE*** bit and setting the ***LTUpdate Time*** register to a non-zero value.

Once the Lifetime Data Logging function is enabled, the measured values are compared to what is already stored in the data flash. If the measured value is higher than the maximum or lower than the minimum value stored in the data flash by more than the "Resolution" set for at least one parameter, the entire Data Flash Lifetime Registers are updated after at least ***LTUpdateTime***.

LTUpdateTime sets the minimum update time between DF writes. When a new maximum or minimum is detected, an LT Update window of [update time] second is enabled and the DF writes occur at the end of this window. Any additional max/min value detected within this window will also be updated. The first new max/min value detected after this window will trigger the next LT Update window.

Internal to the fuel gauge, there exists a RAM maximum/minimum table in addition to the DF maximum/minimum table. The RAM table is updated independent of the resolution parameters. The DF table is updated only if at least one of the RAM parameters exceeds the DF value by more than the resolution associated with it. When DF is updated, the entire RAM table is written to DF. Consequently, it is possible to see a new maximum or minimum value for a certain parameter even if the value of this parameter never exceeds the maximum or minimum value stored in the data flash for this parameter value by the resolution amount.

The Life Time Data Logging of one or more parameters can be reset or restarted by writing new default (or starting) values to the corresponding data flash registers through sealed or unsealed access as described below. However, when using unsealed access, new values will only take effect after device reset.

The logged data can be accessed as RW in UNSEALED mode from the Lifetime Data Subclass (Subclass ID = 59) of data flash. Lifetime data may be accessed (RW) when sealed using a process identical ***Manufacturer Info Block B***. The *DataFlashBlock* command code is 4. Note only the first 32 bytes of lifetime data (not resolution parameters) can be RW when sealed. See *Section 2.2.33.2* for sealed access. The logging settings such as Temperature Resolution, Voltage Resolution, Current Resolution, and Update Time can be configured only in UNSEALED mode by writing to the Lifetime Resolution Subclass (SubclassID = 66) of the data flash.

The Lifetime resolution registers contain the parameters that set the limits related to how much a data parameter must exceed the previously logged maximum/minimum value to be updated in the lifetime log. For example, V must exceed MaxV by more than Voltage Resolution to update MaxV in the data flash.

3.3 Device Configuration

The bq34z100-G1 has many features that can be enabled, disabled, or modified through settings in the Pack Configuration registers. These registers are programmed/read via the methods described in *Section 2.2.33.1*.

3.3.1 Pack Configuration Register

Table 3-5. Pack Configuration Register Bits

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	RESCAP	CAL_EN	SCALED	RSVD	VOLTSEL	IWAKE	RSNS1	RSNS0
Low Byte	RFACT STEP	SLEEP	RMFCC	NiDT	NiDV	QPCCLEAR	GNDSEL	TEMPS

Legend: **RSVD** = Reserved

RESCAP: No-load rate of compensation is applied to the reserve capacity calculation. True when set. Default is 0.

CAL_EN: When enabled, entering CALIBRATION mode is permitted. For special use only. Default = 0.

SCALED: Scaled Capacity and/or Current bit. The mA, mAh, and cWh settings and reports will take on a value that is artificially scaled. This setting has no actual effect within the gauge. It is the responsibility of the host to reinterpret the reported values. Scaled current measurement is achieved by calibrating the current measurement to a value lower than actual.

VOLTSEL: This bit selects between the use of an internal or external battery voltage divider. The internal divider is for single cell use only. Default is 0.

1 = External

0 = Internal

IWAKE/RSNS1/RSNS0: These bits configure the current wake function (see Table 6-1). Default is 0/0/1.

RFACTSTEP: Enables Ra step up/down to Max/Min Res Factor before disabling Ra updates. Default is 1.

SLEEP: The fuel gauge can enter sleep, if operating conditions allow. True when set. Default is 1.

RMFCC: RM is updated with the value from FCC on valid charge termination. True when set. Default is 1.

NiDT: Performs primary charge termination using the ΔT/Δt algorithm. See *Section 3.7*. This bit is only acted upon when a NiXX Chem ID is used.

NiDV: Performs primary charge termination using the –ΔV algorithm. See *Section 3.7*. This bit is only acted upon when a NiXX Chem ID is used.

QPCCLEAR: Upon exit from RELAX where a DOD update occurred, the QMAX Passed Charge is cleared.

GNDSEL: The ADC ground select control. The VSS pin is selected as ground reference when the bit is clear. Pin 10 is selected when the bit is set.

TEMPS: Selects external thermistor for *Temperature()* measurements. True when set. Uses internal temp when clear. Default is 1.

3.3.2 Pack Configuration B Register

Table 3-6. Pack Configuration B Register Bits

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHGDoDEoC	RSVD	VconsEN	RSVD	JEITA	LFPRelax	DoDWT	FConvEN

Legend: **RSVD** = Reserved

CHGDoDEoC: Enable DoD at EoC during charging only. True when set. Default is 1. Default setting is recommended.

VconsEN: Enable voltage measurement consistency check. True when set. Default is 1. Default setting is recommended.

JEITA: Enables *ChargingVoltage()* and *ChargingCurrent()* to report data per the JEITA charging algorithm. When disabled, the values programmed in ***Cell Charge Voltage*** T2–T3 and ***Charge Current*** T2–T3 are reported.

LFPRelax: Enables Lithium Iron Phosphate RELAX mode

DoDWT: Enable Dod weighting for LiFePO4 support when chemical ID 400 series is selected. True when set. Default is 1.

FConvEN: Enable fast convergence algorithm. Default is 1. Default setting is recommended.

3.3.3 Pack Configuration C Register

Table 3-7. Pack Configuration C Register Bits

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SOH_DISP	RSOC_HOLD	FF_NEAR_EDV	SleepWakeCHG	LOCK_0	RELAX_JUMP_OK	RELAX_SMOOTH_OK	SMOOTH

SOH_DISP: Enables State-of-Health Display

RSOC_HOLD: **RSOC_HOLD** enables RSOC Hold Feature preventing RSOC from increasing during discharge.
NOTE: It is recommended to disable **RSOC_HOLD** when SOC Smoothing is enabled (SMOOTH = 1).

FF_NEAR_EDV: Enables Fast Filter Near EDV

SleepWakeCHG: Enable for faster sampling in SLEEP mode. Default setting is recommended.

LOCK_0: Keep *RemainingCapacity()* and *RelativeStateOfCharge()* jumping back during relaxation after 0 is reached during discharge.

RELAX_JUMP_OK: Allows RSOC jump during RELAX mode if [SMOOTH =1]

RELAX_SMOOTH_OK: Smooth RSOC during RELAX mode if [SMOOTH =1]

SMOOTH: Enabled RSOC Smoothing

3.4 Voltage Measurement and Calibration

The device is shipped with a factory configuration for the default case of the 1-series Li-Ion cell. This can be changed by setting the VOLTSEL bit in the Pack Configuration register and by setting the number of series cells in the data flash configuration section.

Multi-cell applications, with voltages up to 65535 mV, may be gauged by using the appropriate input scaling resistors such that the maximum battery voltage, under all conditions, appears at the BAT input as approximately 900 mV. The actual gain function is determined by a calibration process and the resulting voltage calibration factor is stored in the data flash location ***Voltage Divider***.

For single-cell applications, an external divider network is not required. Inside the IC, behind the BAT pin is a nominal 5:1 voltage divider with 88 KΩ in the top leg and 22 KΩ in the bottom leg. This internal divider network is enabled by clearing the VOLTSEL bit in the Pack Configuration register. This ratio is optimum for directly measuring a single Li-Ion cell where charge voltage is limited to 4.5 V.

For higher voltage applications, an external resistor divider network should be implemented as per the reference designs in this document. The quality of the divider resistors is very important to avoid gauging errors over time and temperature. It is recommended to use 0.1% resistors with 25-ppm temperature coefficient. Alternately, a matched network could be used that tracks its dividing ratio with temperature and age due to the similar geometry of each element. Calculation of the series resistor can be made per the equation below.

NOTE: Exceeding **Vin max mV** results in a measurement with degraded linearity.

The bottom leg of the divider resistor should be in the range of 15 KΩ to 25 K, using 16.5 KΩ:

Rseries = 16500 Ω (Vin max mV – 900 mV)/900 mV

For all applications, the ***Voltage Divider*** value in data flash will be used by the firmware to calibrate the total divider ratio. The nominal value for this parameter is the maximum expected value for the stack voltage. The calibration routine adjusts the value to force the reported voltage to equal the actual applied voltage.

3.4.1 1S Example

For stack voltages under 4.5 V max, it is not necessary to provide an external voltage divider network. The internal 5:1 divider should be selected by clearing the VOLTSEL bit in the Pack Configuration register. The default value for ***Voltage Divider*** is 5000 (representing the internal 5000:1000 mV divider) when no external divider resistor is used, and the default number of series cells = 1. In the 1-S case, there is usually no requirement to calibrate the voltage measurement, since the internal divider is calibrated during factory test to within 2 mV.

3.4.2 7S Example

In the multi-cell case, the hardware configuration is different. An external voltage divider network is calculated using the Rseries formula above. The bottom leg of the divider should be in the range of 15 KΩ to 25 KΩ.

3.4.3 Autocalibration

The device provides an autocalibration feature that will measure the voltage offset error across SRP and SRN from time-to-time as operating conditions change. It subtracts the resulting offset error from normal sense resistor voltage, V_{SR}, for maximum measurement accuracy.

The gas gauge performs a single offset calibration when:

1. The interface lines stay low for a minimum of ***Bus Low Time*** and
2. V_{SR} > ***Deadband***.

The gas gauge also performs a single offset when:

1. The condition of *AverageCurrent()* ≤ ***Autocal Min Current*** and
2. {Voltage change since last offset calibration ≥ ***Delta Voltage***} or {temperature change since last offset

calibration is greater than ***Delta Temperature*** for ≥ ***Autocal Time***}.

Capacity and current measurements should continue at the last measured rate during the offset calibration when these measurements cannot be performed. If the battery voltage drops more than ***Cal Abort*** during the offset calibration, the load current has likely increased considerably; hence, the offset calibration will be aborted.

3.5 Temperature Measurement

The bq34z100-G1 can measure temperature via the on-chip temperature sensor or via the TS input, depending on the setting of the [TEMPS] bit *PackConfiguration()*. The bit is set by using the *PackConfiguration()* function, described in *Section 2.2*.

Temperature measurements are made by calling the *Temperature()* function (see *Section 2.1* for specific information).

When an external thermistor is used, REG25 (pin 7) is used to bias the thermistor and TS (pin 11) is used to measure the thermistor voltage (a pull-down circuit is implemented inside the device). The device then correlates the voltage to temperature, assuming the thermistor is a Semitec 103AT or similar device.

3.6 Overtemperature Indication

3.6.1 Overtemperature: Charge

If during charging, *Temperature()* reaches the threshold of ***OT Chg*** for a period of ***OT Chg Time*** and *AverageCurrent()* > ***Chg Current Threshold***, then the *[OTC]* bit of *Flags()* is set. Note: If ***OT Chg Time*** = 0, then the feature is completely disabled.

When *Temperature()* falls to ***OT Chg Recovery***, the *[OTC]* of *Flags()* is reset.

3.6.2 Overtemperature: Discharge

If during discharging *Temperature()* reaches the threshold of ***OT Dsg*** for a period of ***OT Dsg Time***, and *AverageCurrent()* ≤ –***Dsg Current Threshold***, then the *[OTD]* bit of *Flags()* is set. If ***OT Dsg Time*** = 0, then the feature is completely disabled.

When *Temperature()* falls to OT Dsg Recovery, the *[OTD]* bit of *Flags()* is reset.

3.7 Charging and Charge Termination Indication

For proper bq34z100-G1 operation, the battery per cell charging voltage must be specified by the user in ***Cell Charging Voltage***. The default value for this variable is ***Charging Voltage*** = 4200 mV. This parameter should be set to the recommended charging voltage for the entire battery stack divided by the number of series cells.

The device detects valid charge termination in one of three ways:

A. Current Taper method:

   1. During two consecutive periods of ***Current Taper Window***, the *AverageCurrent()* is less than ***Taper Current*** AND
   2. During the same periods, the accumulated change in capacity > 0.25 mAh /***Taper Current Window*** AND
   3. *Voltage()* is > ***Charging Voltage*** – ***Charging Taper Voltage***. When this occurs, the [CHG] bit of *Flags()* is cleared. Also, if the [RMFCC] bit of Pack Configuration is set, and *RemainingCapacity()* is set equal to *FullChargeCapacity()*.

B. Delta Temperature (ΔT/Δt) method—For ΔT/Δt, the bq34z100-G1 detects an increase in temperature over many seconds. The ΔT/Δt setting is programmable in the temperature step, ***Delta Temp*** (0°C – 25.5°C), and the time step, ***Delta Temp Time*** (0 s–1000 s). Typical settings for 1°C/minute include 2°C/120 s and 3°C/180 s (default). Longer times may be used for increased slope resolution.

   In addition to the ΔT/Δt timer, a holdoff timer starts when the battery is charged at more than ***Holdoff Current*** (default is 240 mA), and the temperature is above ***Holdoff Temp***. Until this timer expires, ΔT/Δt detection is suspended. If *Current()* drops below ***Holdoff Current*** or *Temperature()* below

Holdoff Temp, the holdoff timer resets and restarts only when the current and temperature conditions are met again.

C. Negative Delta Voltage (–ΔV) method—For negative delta voltage, the bq34z100-G1 detects a charge termination when the pack voltage drops during charging by ***Cell Negative Delta Volt*** for a period of ***Cell Negative Delta Time*** during which time *Voltage()* must be greater than ***Cell Negative Qual Volt***.

When either condition occurs, the *Flags()[CHG]* bit is cleared. Also, if the *[RMFCC]* bit of Pack Configuration is set, and *RemainingCapacity()* is set equal to *FullChargeCapacity()*.

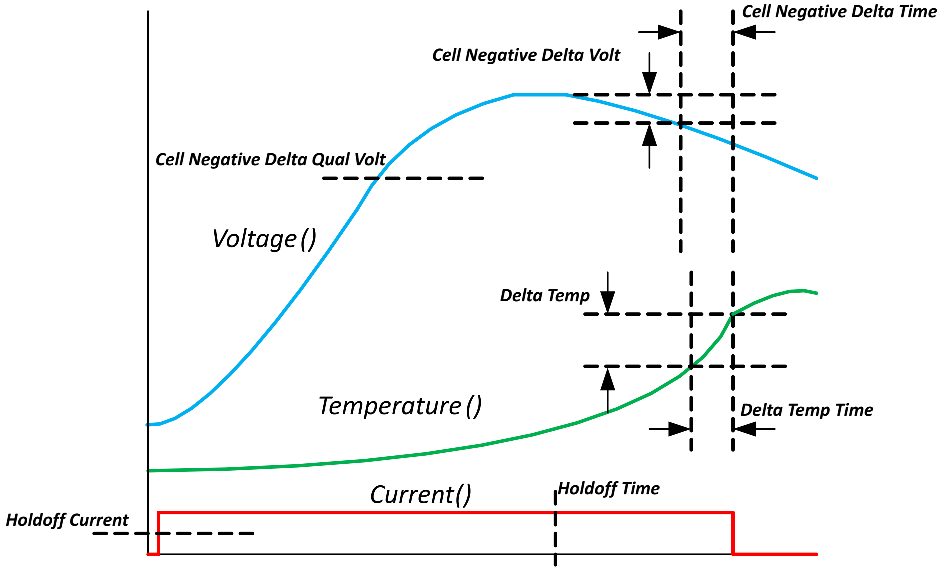

Figure 3-1. NiXX Termination

3.8 SCALED Mode

The device supports high current and high capacity batteries above 32.76 Amperes and 29 Ampere-Hours indirectly by scaling the actual sense resistor value compared with the calibrated value stored in the device. The need for this is due to the standardization of a 2-byte data command having a maximum representation of +/–32767. When ***[SCALED]*** is set in the ***Pack Configuration*** register, this indicates that the current and capacity data is scaled.

It is important to know that setting the SCALED flag does not actually change anything in the operation of the gauge. It serves as a notice to the host that the various reported values should be reinterpreted based on the scale used. Because the flag has no actual effect, it can be used to represent other scaling values. See *Section 3.2.5*.

NOTE: It is recommended to only scale by a value between 1 and 10 to optimize resolution and accuracy while still extending the data range.

3.9 LED Display

The device supports multiple options for using one to 16 LEDs as an output device to display the remaining state of charge, or, if Pack Configuration C [SOH_DISP] is set, then state-of-health. The LED/COMM Configuration register determines the behavior.

Table 3-8. LED/COMM Configuration Bits

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EXT_LED3	EXT_LED2	EXT_LED1	EXT_LED0	LED_ON	LED_Mode2	LED_Mode1	LED_Mode0

Bits 0, 1, 2 are a code for one of five modes. 0 = No LED, 1 = Single LED, 2 = Four LEDs, 3 = External LEDs with I^{2}C comm, 4 = External LEDs with HDQ comm.

Setting Bit 3, LED_ON, will cause the LED display to be always on, except in Single LED mode where it is not applicable. When clear (default), the LED pattern will only be displayed after holding an LED display button for one to two seconds. The button applies 2.5 V from REG25 pin 7 to VEN pin 2. The ***LED Hold Time*** parameter may be used to configure how long the LED display remains on if LED_ON is clear. ***LED Hold Time*** configures the update interval for the LED display if LED_ON is set.

Bits 4, 5, 6, and 7 are a binary code for number of external LEDs. Code 0 is reserved. Codes 1 through 15 represents 2~16 external LEDs. So, number of External LEDs is 1 + Value of the 4-bit binary code. Display of Remaining Capacity *RemainingCapacity()*or *StateOfHealth()* will be evenly divided among the selected number of LEDs.

Single LED mode—Upon detecting an A/D value representing 2.5 V on the VEN pin, Single LED mode will toggle the LED as duty cycle on within a period of 1 s where each 1% of RSOC is a 7.8125-ms high time. So, for example, 10% RSOC or SOH will have the LED on for 78.1 ms and off for 921.9 ms. 90% RSOC or SOH will have the LED on for 703.125 ms and off for 296.875 ms. Any value > 90% will display as 90%.

Four-LED mode—Upon detecting an A/D value representing 2.5 V on the VEN pin, Four-LED mode will display the RSOC or SOH by driving pins RC2(LED1), RC0(LED2), RA1(LED3),RA2(LED4) in a proportional manner where each LED represents 25% of the remaining state-of-charge. For example, if RSOC or SOH = 67%, three LEDs will be illuminated.

External LED mode—Upon detecting an A/D value representing 2.5 V on the VEN pin, External LED mode will transmit the RSOC into an SN74HC164 (for 2–8 LEDs) or two SN74HC164 devices (for 9–16 LEDs) using a bit-banged approach with RC2 as Clock and RC0 as Data. LEDs will be lit for a number of seconds as defined in a data flash parameter. Refer to the *SN54HC164, SN74HC164 8-Bit Parallel-Out Serial Shift Registers Data Sheet* (SCLS115E) for details on these devices.

Extended commands are available to turn the LEDs on and off for test purposes.

3.10 Alert Signal

Based on the selected LED mode, various options are available for the hardware implementation of an Alert signal. Software configuration of the Alert Configuration register determines which alert conditions will assert the $\overline{\text{ALERT}}$ pin.

Table 3-9. $\overline{\text{ALERT}}$ Signal Pins

MODE	DESCRIPTION	$\overline{\text{ALERT}}$ PIN	$\overline{\text{ALERT}}$ PIN NAME	CONFIG REGISTER HEX CODE	COMMENT
0	No LED	1	P2	0	
1	Single LED	1	P2	1	
2	4 LED	11	P6	2	Filter and FETs are required to eliminate temperature sense pulses.
3	5-LED Expander with I^{2}C Host Comm	12	P5	43	
3	10-LED Expander with I^{2}C Host Comm	12	P5	93	
4	5-LED Expander with HDQ Host Comm	13	P4	44	
4	10-LED Expander with HDQ Host Comm	13	P4	94	

The port used for the Alert output will depend on the mode setting in ***LED/Comm Configuration*** as defined in Table 3-9. The default mode is 0. The $\overline{\text{ALERT}}$ pin will be asserted by driving LOW. However, note that in LED/COM mode 2, pin TS/P6, which has a dual purpose as temperature sense pin, will be driven low except when temperature measurements are made each second. See the reference schematic in the *bq34z100-G1 Wide Range Fuel Gauge with Impedance Track™ Technology Data Sheet* (SLUSBZ5) for filter implementation details if host alert sensing requires a continuous signal.

The $\overline{\text{ALERT}}$ pin will be a logical OR of the selected bits in the new configuration register when asserted in the Flags register. The default value for Alert Configuration register is 0.

Table 3-10. Alert Configuration Register Bit Definitions

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High Byte	OTC	OTD	BAT_HIGH	BATLOW	CHG_INH	XCHG	FC	CHG
Low Byte	OCVTAKEN	RSVD	RSVD	CF	RSVD	RCA	EOD	DSG

Legend: **RSVD** = Reserved

OTC: Over-Temperature in Charge condition is detected. $\overline{\text{ALERT}}$ is enabled when set.

OTD: Over-Temperature in Discharge condition is detected. $\overline{\text{ALERT}}$ is enabled when set.

BAT_HIGH: Battery High bit that indicates a high battery voltage condition. Refer to the data flash ***CELL BH*** parameters for threshold settings. $\overline{\text{ALERT}}$ is enabled when set.

BATLOW: Battery Low bit that indicates a low battery voltage condition. Refer to the data flash ***CELL BL*** parameters for threshold settings. $\overline{\text{ALERT}}$ is enabled when set.

CHG_INH: Charge Inhibit: unable to begin charging. Refer to the data flash ***[Charge Inhibit Temp Low, Charge Inhibit Temp High]*** parameters. $\overline{\text{ALERT}}$ is enabled when set.

XCHG: Charging disallowed $\overline{\text{ALERT}}$ is enabled when set.

FC: Full charge is detected. FC is set when charge termination is reached and ***FC Set%*** = –1 (see *Section 3.7* for details) or *StateOfCharge()* is larger than ***FC Set%*** and ***FC Set%*** is not –1. $\overline{\text{ALERT}}$ is enabled when set.

CHG: (Fast) charging allowed. $\overline{\text{ALERT}}$ is enabled when set.

OCVTAKEN: Cleared on entry to RELAX mode and set to 1 when OCV measurement is performed in RELAX mode. $\overline{\text{ALERT}}$ is enabled when set.

CF: Condition Flag set. $\overline{\text{ALERT}}$ is enabled when set.

RCA: Remaining Capacity Alarm reached. $\overline{\text{ALERT}}$ is enabled when set.

EOD: End-of-Discharge Threshold reached. $\overline{\text{ALERT}}$ is enabled when set.

DSG: Discharging detected. $\overline{\text{ALERT}}$ is enabled when set.

Communications

4.1 Authentication

The bq34z100-G1 can act as a SHA-1/HMAC authentication slave by using its internal engine. Sending a 160-bit SHA-1 challenge message to the device will cause the IC to return a 160-bit digest, based upon the challenge message and hidden plain-text authentication keys. When this digest matches an identical one generated by a host or dedicated authentication master (operating on the same challenge message and using the same plain text keys), the authentication process is successful.

The device contains a default plain-text authentication key of 0x0123456789ABCDEFFEDCBA987654321. If using the device's internal authentication engine, the default key can be used for development purposes, but should be changed to a secret key and the part immediately sealed before putting a pack into operation.

4.2 Key Programming

When the device's SHA-1/HMAC internal engine is used, authentication keys are stored as plain-text in memory. A plain-text authentication key can only be written to the device while the IC is in UNSEALED mode. Once the IC is UNSEALED, a 0x00 is written to *BlockDataControl()* to enable the authentication data commands. Next, subclass ID and offset are specified by writing 0x70 and 0x00 to *DataFlashClass()* and *DataFlashBlock()*, respectively. The device is now prepared to receive the 16-byte plain-text key, which must begin at command location 0x4C. The key is accepted once a successful checksum has been written to *BlockDataChecksum()* for the entire 32-byte block (0x40 through 0x5F), not just the 16-byte key.

4.3 Executing an Authentication Query

To execute an authentication query in UNSEALED mode, a host must first write 0x01 to the *BlockDataControl()* command to enable the authentication data commands. If in SEALED mode, 0x00 must be written to *DataFlashBlock().*

Next, the host writes a 20-byte authentication challenge to the *AuthenticateData()* address locations (0x40 through 0x53). After a valid checksum for the challenge is written to *AuthenticateChecksum()*, the device uses the challenge to perform its own SHA-1/HMAC computation in conjunction with its programmed keys. The resulting digest is written to *AuthenticateData()*, overwriting the pre-existing challenge. The host may then read this response and compare it against the result created by its own parallel computation.

4.4 HDQ Single-Pin Serial Interface

The HDQ interface is an asynchronous return-to-one protocol where a processor sends the command code to the device. With HDQ, the least significant bit (LSB) of a data byte (command) or word (data) is transmitted first. Note that the DATA signal on pin 12 is open-drain and requires an external pull-up resistor. The 8-bit command code consists of two fields: the 7-bit HDQ command code (bits 0–6) and the 1-bit R/W field (MSB Bit 7). The R/W field directs the device either to:

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

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

The return-to-one data bit frame of HDQ consists of three distinct sections. The first section is used to start the transmission by either the host or by the device taking the DATA pin to a logic-low state for a time $t_{STRH,B}$. The next section is for data transmission where the data is valid for a time t_{DSU} after the negative edge used to start communication. The data is held until a time t_{DV}, allowing the host or device time to sample the data bit. The final section is used to stop the transmission by returning the DATA pin to a logic-high state by at least a time t_{SSU} after the negative edge used to start communication. The final logic-high state is held until the end of $t_{CYCH,B}$, allowing time to ensure the transmission was stopped correctly.

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

The device is shipped in the I^2C mode. TI provides tools can be used to switch from I^2C to HDQ communications.

4.5 I^2C Interface

The gas gauge supports the standard I^2C read, incremental read, one-byte write quick read, and functions. The 7-bit device address (ADDR) is the most significant 7 bits of the hex address and is fixed as 1010101. The 8-bit device address is therefore 0xAA or 0xAB for write or read, respectively.

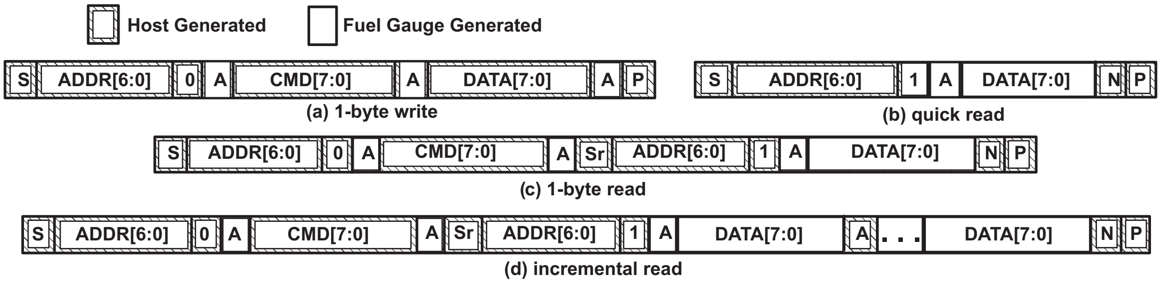
Host Generated
Fuel Gauge Generated
S ADDR[6:0] 0 A CMD[7:0] A DATA[7:0] A P
(a) 1-byte write
S ADDR[6:0] 1 A DATA[7:0] N P
(b) quick read
S ADDR[6:0] 0 A CMD[7:0] A Sr ADDR[6:0] 1 A DATA[7:0] N P
(c) 1-byte read
S ADDR[6:0] 0 A CMD[7:0] A Sr ADDR[6:0] 1 A DATA[7:0] A . . . DATA[7:0] N P
(d) incremental read

Figure 4-1. Supported I^2C formats: (a) 1-byte write, (b) quick read, (c) 1 byte-read, and (d) incremental read (S = Start, Sr = Repeated Start, A = Acknowledge, N = No Acknowledge, and P = Stop).

The “quick read” returns data at the address indicated by the address pointer. The address pointer, a register internal to the I^2C communication engine, increments whenever data is acknowledged by the device or the I^2C master. “Quick writes” function in the same manner and are a convenient means of sending multiple bytes to consecutive command locations (such as 2-byte commands that require two bytes of data).

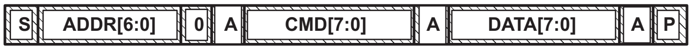

Figure 4-2. Attempt To Write a Read-Only Address (Nack After Data Sent By Master)

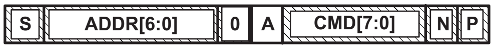

Figure 4-3. Attempt To Read An Address Above 0x7F (Nack Command)

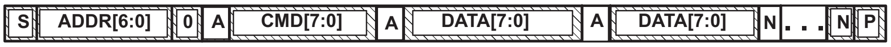
S ADDR[6:0] 0 A CMD[7:0] A DATA[7:0] A DATA[7:0] N . . . N P

Figure 4-4. Attempt At Incremental Writes (nack All Extra Data Bytes Sent)

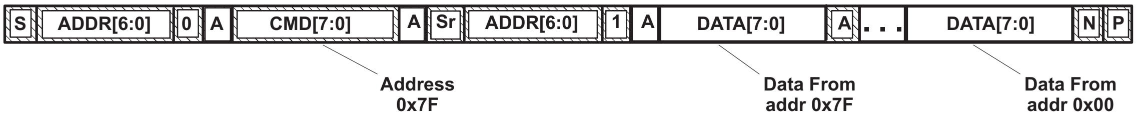

Figure 4-5. Incremental Read at the Maximum Allowed Read Address

The I^{2}C engine releases both SDA and SCL if the I^{2}C bus is held low for ***Bus Low Time***. If the gas 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 I^{2}C engine enters the low-power SLEEP mode.

4.6 Switching Between I^{2}C and HDQ Modes

Texas Instruments ships the bq34z100-G1 device in I^{2}C mode (factory default); however, this mode can be changed to HDQ mode if needed.

NOTE: To make changes in the data flash, the device must be in I^{2}C mode.

4.6.1 Converting to HDQ Mode

Using the Battery Management Studio (bqSTUDIO) tool to configure the bq34z100-G1 to HDQ mode, a write to the Control command [0x00] of [0x7C40] is required.

To configure HDQ mode with bqSTUDIO:

1. Navigate to the *Registers* screen. HDQ mode is configured by writing data [0x7C40] to Control command [0x00].
2. Click on the **Control value** field.
3. Write 0x7C40 into the text field and click **OK**. Because the change in communication protocol involves writing a flag for the new protocol to data flash, it takes about 200 ms to complete. During this time, communications are disabled. Once the command takes effect, the bqSTUDIO will no longer communicate with the gauge.
4. Close bqSTUDIO. Change communication connections from the gauge to the HDQ port of either the EV2300 or EV2400 device. Run bqSTUDIO. The bqSTUDIO auto-detection only works for devices that operate in I^{2}C mode.

   When the bq34z100-G1 device is in HDQ mode, it will not be detected.
5. Select bq34z100-G1 manually. Click **OK** to all messages that indicate that the device is not detected or not responsive. When the *Registers* screen starts, it will take a period of time from when bqSTUDIO first tries to communicate with the device in I^{2}C before trying HDQ mode.

Once it is complete, the *Registers* screen will display data as it had done initially when it was in I^{2}C mode. The refresh is noticeably slower, due to the slow speed of HDQ.

Use the *Registers* screen only while the bq34z100-G1 is in HDQ mode. All other functions will not be supported in Battery Management Studio.

4.6.2 Converting to I^{2}C Mode

Texas Instruments ships the bq34z100-G1 device in I^{2}C mode, which is required when updating data flash. However, this mode can be changed to HDQ mode if needed.

To configure the device to use I^{2}C mode when presently in the HDQ mode, a write to the Control command [0x00] of [0x29E7] is required. Use the Battery Management Studio (bqSTUDIO) tool, as follows:

1. Click on the **Control value** field. Write [0x29E7] in the text field and click **OK**. Once the command takes effect, bqSTUDIO will no longer communicate with the gauge.
2. Close bqSTUDIO. Change communication connections from the gauge to the I^{2}C port of either the EV2300 or EV2400 device. Run bqSTUDIO.

Device Functional Modes

The device has three power modes: NORMAL mode, SLEEP mode, and FULL SLEEP mode.

- In NORMAL mode, the device is fully powered and can execute any allowable task.
- In SLEEP mode, the gas gauge exists in a reduced-power state, periodically taking measurements and performing calculations.
- In FULL SLEEP mode, the high frequency oscillator is turned off, and power consumption is further reduced compared to SLEEP mode.

5.1 NORMAL Mode

The gas 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. Determinations to change states are also made. This mode is exited by activating a different power mode.

5.2 SLEEP Mode

SLEEP mode is entered automatically if the feature is enabled (***Pack Configuration [SLEEP]*** = 1) and *Average Current()* is below the programmable level ***Sleep Current***. Once entry to sleep has been qualified but prior to entry to SLEEP mode, the device performs an ADC autocalibration to minimize offset. Entry to SLEEP mode can be disabled by the *[SLEEP]* bit of *Pack Configuration()*, where 0 = disabled and 1 = enabled. During SLEEP mode, the device periodically wakes to take data measurements and updates the data set, after which it then returns directly to SLEEP. The device exits SLEEP if any entry condition is broken, a change in protection status occurs, or a current in excess of I_{WAKE} through R_{SENSE} is detected.

5.3 FULL SLEEP Mode

FULL SLEEP mode is entered automatically when the device is in SLEEP mode and the timer counts down to 0 (*Full Sleep Wait Time* > 0). FULL SLEEP mode is disabled when *Full Sleep Wait Time* is set to 0.

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

The gauge exits the FULL SLEEP mode when there is any communication activity. Therefore, the execution of SET_FULLSLEEP sets *[FULLSLEEP]* bit, but the EVSW might still display the bit clear. The FULL SLEEP mode can be verified by measuring the current consumption of the gauge. In this mode, the high frequency oscillator is turned off. The power consumption is further reduced compared to the SLEEP mode.

While in FULL SLEEP mode, the fuel gauge can suspend serial communications as much as 4 ms by holding the communication line(s) low. This delay is necessary to correctly process host communication since the fuel gauge processor is mostly halted. For HDQ communication one host message will be dropped.

Power Control

6.1 Reset Functions

When the device detects either a hardware or software reset ($\overline{\text{MRST}}$ pin is driven low or the *[RESET]* bit of *Control()* is initiated, respectively), it determines the type of reset and increments the corresponding counter. This information is accessible by issuing the command *Control()* function with the RESET_DATA subcommand.

As shown in Figure 6-1, if a partial reset was detected, a RAM checksum is generated and compared against the previously stored checksum. If the checksum values do not match, the RAM is reinitialized (a “Full Reset”). The stored checksum is updated every time RAM is altered.

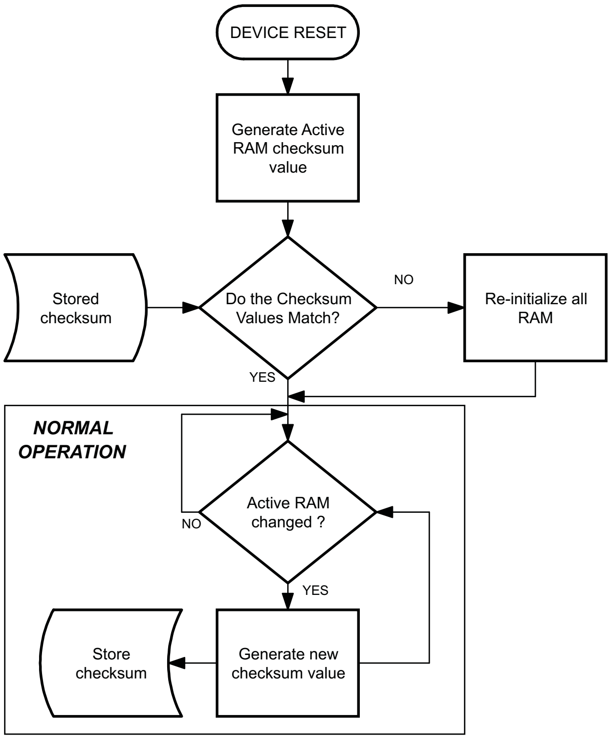

Figure 6-1. Partial Reset Flow Diagram

6.2 Wake-Up Comparator

The wake up comparator is used to indicate a change in cell current while the device is in SLEEP mode. *PackConfiguration()* uses bits [RSNS1–RSNS0] to set the sense resistor selection. *PackConfiguration()* 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 charge or discharge directions. A setting of 0x00 of RSNS1..0 disables this feature.

Table 6-1. I_{WAKE} t=Threshold Settings[1]

RSNS1	RSNS0	IWAKE	Vth(SRP–SRN)
0	0	0	Disabled
0	0	1	Disabled
0	1	0	+1.25 mV or –1.25 mV
0	1	1	+2.5 mV or –2.5 mV
1	0	0	+2.5 mV or –2.5 mV
1	0	1	+5 mV or –5 mV
1	1	0	+5 mV or –5 mV
1	1	1	+10 mV or –10 mV

[1] The actual resistance value vs. the setting of the sense resistor is not important. Only the actual voltage threshold when calculating the configuration is important.

6.3 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 should be selected such that the device V_{CC} voltage does not fall below its minimum of 2.4 V during Flash write operations. The default value of 2800 mV is appropriate; however, for more information.

Data Flash Summary

Table 7-1 summarizes the data flash locations available to the user, including the default, minimum, and maximum values.

Table 7-1. Data Flash Summary

CLASS	SUBCLASS	SUBCLASS ID	OFFSET	TYPE	NAME	MIN	MAX	DEFAULT	UNIT
Configuration	Safety	2	0	I2	OT Chg	0	1200	550	0.1°C
Configuration	Safety	2	2	U1	OT Chg Time	0	60	2	s
Configuration	Safety	2	3	I2	OT Chg Recovery	0	1200	500	0.1°C
Configuration	Safety	2	5	I2	OT Dsg	0	1200	600	0.1°C
Configuration	Safety	2	7	U1	OT Dsg Time	0	60	2	s
Configuration	Safety	2	8	I2	OT Dsg Recovery	0	1200	550	0.1°C
Configuration	Charge Inhibit Cfg	32	0	I2	Chg Inhibit Temp Low	–400	1200	0	0.1°C
Configuration	Charge Inhibit Cfg	32	2	I2	Chg Inhibit Temp High	–400	1200	450	0.1°C
Configuration	Charge Inhibit Cfg	32	4	I2	Temp Hys	0	100	50	0.1°C
Configuration	Charge	34	0	I2	Suspend Low Temp	–400	1200	–50	0.1°C
Configuration	Charge	34	2	I2	Suspend High Temp	–400	1200	550	0.1°C
Configuration	Charge	34	4	U1	Pb EFF Efficiency	0	100	100	%
Configuration	Charge	34	5	F4	Pb Temp Comp	0	0.078125	0.01953125	%
Configuration	Charge	34	9	U1	Pb Drop Off Percent	0	100	96	%
Configuration	Charge	34	10	F4	Pb Reduction Rate	0	1.25	0.125	%
Configuration	Charge Termination	36	0	I2	Taper Current	0	1000	100	mA
Configuration	Charge Termination	36	2	I2	Min Taper Capacity	0	1000	25	mAh
Configuration	Charge Termination	36	4	I2	Cell Taper Voltage	0	1000	100	mV
Configuration	Charge Termination	36	6	U1	Current Taper Window	0	60	40	s
Configuration	Charge Termination	36	7	I1	TCA Set %	–1	100	99	%
Configuration	Charge Termination	36	8	I1	TCA Clear %	–1	100	95	%
Configuration	Charge Termination	36	9	I1	FC Set %	–1	100	100	%
Configuration	Charge Termination	36	10	I1	FC Clear %	–1	100	98	%
Configuration	Charge Termination	36	11	I2	DODatEOC Delta T	0	1000	100	0.1°C
Configuration	Charge Termination	36	13	I2	NiMH Delta Temp	0	255	30	0.1°C
Configuration	Charge Termination	36	15	U2	NiMH Delta Temp Time	0	1000	180	s
Configuration	Charge Termination	36	17	U2	NiMH Hold Off Time	0	1000	100	s
Configuration	Charge Termination	36	19	I2	NiMH Hold Off Current	0	32000	240	mA

Table 7-1. Data Flash Summary (continued)

CLASS	SUBCLASS	SUBCLASS ID	OFFSET	TYPE	NAME	MIN	MAX	DEFAULT	UNIT
Configuration	Charge Termination	36	21	I2	NiMH Hold Off Temp	0	1000	250	0.1°C
Configuration	Charge Termination	36	23	U1	NiMH Cell Negative Delta Volt	0	100	17	mV
Configuration	Charge Termination	36	24	U1	NiMH Cell Negative Delta Time	0	255	16	s
Configuration	Charge Termination	36	25	I2	NiMH Cell Neg Delta Qual Volt	0	32767	4200	mV
Configuration	Data	48	2	U2	Manufacture Date	0	65535	0	Day + Mo*32 + (Yr -1980)*256
Configuration	Data	48	4	H2	Serial Number	0	ffff	1	hex
Configuration	Data	48	6	U2	Cycle Count	0	65535	0	Counts
Configuration	Data	48	8	I2	CC Threshold	100	32767	900	mAh
Configuration	Data	48	10	U1	Max Error Limit	0	100	100	%
Configuration	Data	48	11	I2	Design Capacity	0	32767	1000	mAh
Configuration	Data	48	13	I2	Design Energy	0	32767	5400	mWh
Configuration	Data	48	15	I2	SOH Load I	–32767	0	–400	mA
Configuration	Data	48	17	U2	Cell Charge Voltage T1-T2	0	4600	4200	mV
Configuration	Data	48	19	U2	Cell Charge Voltage T2-T3	0	4600	4200	mV
Configuration	Data	48	21	U2	Cell Charge Voltage T3-T4	0	4600	4100	mV
Configuration	Data	48	23	U1	Charge Current T1-T2	0	100	10	%
Configuration	Data	48	24	U1	Charge Current T2-T3	0	100	50	%
Configuration	Data	48	25	U1	Charge Current T3-T4	0	100	30	%
Configuration	Data	48	26	I1	JEITA T1	–128	127	–10	°C
Configuration	Data	48	27	I1	JEITA T2	–128	127	10	°C
Configuration	Data	48	28	I1	JEITA T3	–128	127	45	°C
Configuration	Data	48	29	I1	JEITA T4	–128	127	55	°C
Configuration	Data	48	30	U1	Design Energy Scale	0	255	1	Num
Configuration	Data	48	31	S12	Device Name	x	x	bq34z100-G1	—
Configuration	Data	48	43	S12	Manufacturer Name	x	x	Texas Inst.	—
Configuration	Data	48	55	S5	Device Chemistry	x	x	LION	—
Configuration	Discharge	49	0	U2	SOC1 Set Threshold	0	65535	150	mAh
Configuration	Discharge	49	2	U2	SOC1 Clear Threshold	0	65535	175	mAh
Configuration	Discharge	49	4	U2	SOCF Set Threshold	0	65535	75	mAh
Configuration	Discharge	49	6	U2	SOCF Clear Threshold	0	65535	100	mAh
Configuration	Discharge	49	8	I2	Cell BL Set Volt Threshold	0	5000	0	mV
Configuration	Discharge	49	10	U1	Cell BL Set Volt Time	0	60	0	s
Configuration	Discharge	49	11	I2	Cell BL Clear Volt Threshold	0	5000	5	mV
Configuration	Discharge	49	13	I2	Cell BH Set Volt Threshold	0	5000	4300	mV
Configuration	Discharge	49	15	U1	Cell BH Volt Time	0	60	2	s
Configuration	Discharge	49	16	I2	Cell BH Clear Volt Threshold	0	5000	5	mV
Configuration	Discharge	49	21	U1	Cycle Delta	0	255	5	0.01%
Configuration	Manufacturer Data	56	0	H2	Pack Lot Code	0	ffff	0	hex
Configuration	Manufacturer Data	56	2	H2	PCB Lot Code	0	ffff	0	hex
Configuration	Manufacturer Data	56	4	H2	Firmware Version	0	ffff	0	hex
Configuration	Manufacturer Data	56	6	H2	Hardware Revision	0	ffff	0	hex

Table 7-1. Data Flash Summary (continued)

CLASS	SUBCLASS	SUBCLASS ID	OFFSET	TYPE	NAME	MIN	MAX	DEFAULT	UNIT
Configuration	Manufacturer Data	56	8	H2	Cell Revision	0	ffff	0	hex
Configuration	Manufacturer Data	56	10	H2	DF Config Version	0	ffff	0	hex
Configuration	Lifetime Data	59	0	I2	Lifetime Max Temp	0	1400	300	0.1°C
Configuration	Lifetime Data	59	2	I2	Lifetime Min Temp	–600	1400	200	0.1°C
Configuration	Lifetime Data	59	4	I2	Lifetime Max Chg Current	–32767	32767	0	mA
Configuration	Lifetime Data	59	6	I2	Lifetime Max Dsg Current	–32767	32767	0	mA
Configuration	Lifetime Data	59	8	U2	Lifetime Max Pack Voltage	0	65535	320	20 mV
Configuration	Lifetime Data	59	10	U2	Lifetime Min Pack Voltage	0	65535	350	20 mV
Configuration	Lifetime Temp Samples	60	0	U2	LT Flash Cnt	0	65535	0	Counts
Configuration	Registers	64	0	H2	Pack Configuration	0	ffff	161	flags
Configuration	Registers	64	2	H1	Pack Configuration B	0	ff	ff	flags
Configuration	Registers	64	3	H1	Pack Configuration C	0	ff	30	flags
Configuration	Registers	64	4	H1	LED_Comm Configuration	0	ff	0	flags
Configuration	Registers	64	5	H2	Alert Configuration	0	ffff	0	flags
Configuration	Registers	64	7	U1	Number of series cell	0	100	1	Num
Configuration	Lifetime Resolution	66	0	U1	LT Temp Res	0	255	10	0.1°C
Configuration	Lifetime Resolution	66	1	U1	LT Cur Res	0	255	100	mA
Configuration	Lifetime Resolution	66	2	U1	LT V Res	0	255	1	20 mV
Configuration	Lifetime Resolution	66	3	U2	LT Update Time	0	65535	60	s
Configuration	LED Display	67	0	U1	LED Hold Time	0	255	4	Num
Configuration	Power	68	0	I2	Flash Update OK Cell Volt	0	4200	2800	mV
Configuration	Power	68	2	I2	Sleep Current	0	100	10	mA
Configuration	Power	68	11	U1	FS Wait	0	255	0	s
System Data	Manufacturer Info	58	0	H1	Block A 0	0	ff	0	hex
System Data	Manufacturer Info	58	1	H1	Block A 1	0	ff	0	hex
System Data	Manufacturer Info	58	2	H1	Block A 2	0	ff	0	hex
System Data	Manufacturer Info	58	3	H1	Block A 3	0	ff	0	hex
System Data	Manufacturer Info	58	4	H1	Block A 4	0	ff	0	hex
System Data	Manufacturer Info	58	5	H1	Block A 5	0	ff	0	hex
System Data	Manufacturer Info	58	6	H1	Block A 6	0	ff	0	hex
System Data	Manufacturer Info	58	7	H1	Block A 7	0	ff	0	hex
System Data	Manufacturer Info	58	8	H1	Block A 8	0	ff	0	hex
System Data	Manufacturer Info	58	9	H1	Block A 9	0	ff	0	hex
System Data	Manufacturer Info	58	10	H1	Block A 10	0	ff	0	hex
System Data	Manufacturer Info	58	11	H1	Block A 11	0	ff	0	hex
System Data	Manufacturer Info	58	12	H1	Block A 12	0	ff	0	hex

Table 7-1. Data Flash Summary (continued)

CLASS	SUBCLASS	SUBCLASS ID	OFFSET	TYPE	NAME	MIN	MAX	DEFAULT	UNIT
System Data	Manufacturer Info	58	13	H1	Block A 13	0	ff	0	hex
System Data	Manufacturer Info	58	14	H1	Block A 14	0	ff	0	hex
System Data	Manufacturer Info	58	15	H1	Block A 15	0	ff	0	hex
System Data	Manufacturer Info	58	16	H1	Block A 16	0	ff	0	hex
System Data	Manufacturer Info	58	17	H1	Block A 17	0	ff	0	hex
System Data	Manufacturer Info	58	18	H1	Block A 18	0	ff	0	hex
System Data	Manufacturer Info	58	19	H1	Block A 19	0	ff	0	hex
System Data	Manufacturer Info	58	20	H1	Block A 20	0	ff	0	hex
System Data	Manufacturer Info	58	21	H1	Block A 21	0	ff	0	hex
System Data	Manufacturer Info	58	22	H1	Block A 22	0	ff	0	hex
System Data	Manufacturer Info	58	23	H1	Block A 23	0	ff	0	hex
System Data	Manufacturer Info	58	24	H1	Block A 24	0	ff	0	hex
System Data	Manufacturer Info	58	25	H1	Block A 25	0	ff	0	hex
System Data	Manufacturer Info	58	26	H1	Block A 26	0	ff	0	hex
System Data	Manufacturer Info	58	27	H1	Block A 27	0	ff	0	hex
System Data	Manufacturer Info	58	28	H1	Block A 28	0	ff	0	hex
System Data	Manufacturer Info	58	29	H1	Block A 29	0	ff	0	hex
System Data	Manufacturer Info	58	30	H1	Block A 30	0	ff	0	hex
System Data	Manufacturer Info	58	31	H1	Block A 31	0	ff	0	hex
Gas Gauging	IT Cfg	80	0	U1	Load Select	0	255	1	Num
Gas Gauging	IT Cfg	80	1	U1	Load Mode	0	255	0	Num
Gas Gauging	IT Cfg	80	10	I2	Res Current	0	1000	10	mA
Gas Gauging	IT Cfg	80	14	U1	Max Res Factor	0	255	50	Num
Gas Gauging	IT Cfg	80	15	U1	Min Res Factor	0	255	1	Num
Gas Gauging	IT Cfg	80	17	U2	Ra Filter	0	1000	500	Num
Gas Gauging	IT Cfg	80	47	U1	Min PassedChg NiMH-LA 1st Qmax	1	100	50	%
Gas Gauging	IT Cfg	80	49	U1	Maximum Qmax Change	0	255	100	%
Gas Gauging	IT Cfg	80	53	I2	Cell Terminate Voltage	1000	3700	3000	mV
Gas Gauging	IT Cfg	80	55	I2	Cell Term V Delta	0	4200	200	mV
Gas Gauging	IT Cfg	80	58	U2	ResRelax Time	0	65534	500	s
Gas Gauging	IT Cfg	80	62	I2	User Rate-mA	–32767	32767	0	mA
Gas Gauging	IT Cfg	80	64	I2	User Rate-Pwr	–32767	32767	0	mW/cW
Gas Gauging	IT Cfg	80	66	I2	Reserve Cap-mAh	0	9000	0	mAh
Gas Gauging	IT Cfg	80	68	I2	Reserve Energy	0	14000	0	mWh/cWh
Gas Gauging	IT Cfg	80	72	U1	Max Scale Back Grid	0	15	4	Num
Gas Gauging	IT Cfg	80	73	U2	Cell Min DeltaV	0	65535	0	mV
Gas Gauging	IT Cfg	80	75	U1	Ra Max Delta	0	255	15	%

Table 7-1. Data Flash Summary (continued)

CLASS	SUBCLASS	SUBCLASS ID	OFFSET	TYPE	NAME	MIN	MAX	DEFAULT	UNIT
Gas Gauging	IT Cfg	80	76	I2	Design Resistance	1	32767	42	mΩ
Gas Gauging	IT Cfg	80	78	U1	Reference Grid	0	14	4	—
Gas Gauging	IT Cfg	80	79	U1	Qmax Max Delta %	0	100	10	mAh
Gas Gauging	IT Cfg	80	80	U2	Max Res Scale	0	32767	32000	Num
Gas Gauging	IT Cfg	80	82	U2	Min Res Scale	0	32767	1	Num
Gas Gauging	IT Cfg	80	84	U1	Fast Scale Start SOC	0	100	10	%
Gas Gauging	IT Cfg	80	89	I2	Charge Hys V Shift	0	2000	40	mV
Gas Gauging	IT Cfg	80	91	I2	Smooth Relax Time	1	32767	1000	s
Gas Gauging	Current Thresholds	81	0	I2	Dsg Current Threshold	0	2000	60	mA
Gas Gauging	Current Thresholds	81	2	I2	Chg Current Threshold	0	2000	75	mA
Gas Gauging	Current Thresholds	81	4	I2	Quit Current	0	1000	40	mA
Gas Gauging	Current Thresholds	81	6	U2	Dsg Relax Time	0	8191	60	s
Gas Gauging	Current Thresholds	81	8	U1	Chg Relax Time	0	255	60	s
Gas Gauging	Current Thresholds	81	9	U2	Cell Max IR Correct	0	1000	400	mV
Gas Gauging	State	82	0	I2	Qmax Cell 0	0	32767	1000	mAh
Gas Gauging	State	82	2	U2	Cycle Count	0	65535	0	Num
Gas Gauging	State	82	4	H1	Update Status	0	6	0	Num
Gas Gauging	State	82	5	I2	Cell V at Chg Term	0	5000	4200	mV
Gas Gauging	State	82	7	I2	Avg I Last Run	–32768	32767	–299	mA
Gas Gauging	State	82	9	I2	Avg P Last Run	–32768	32767	–1131	mWh
Gas Gauging	State	82	11	I2	Cell Delta Voltage	–32768	32767	2	mV
Gas Gauging	State	82	13	I2	T Rise	0	32767	20	Num
Gas Gauging	State	82	15	I2	T Time Constant	0	32767	1000	Num
Ra Table	R_a0	88	0	H2	R_a0 Flag	0	ffff	ff55	Hex
Ra Table	R_a0	88	2	I2	R_a0 0	0	32767	105	Num
Ra Table	R_a0	88	4	I2	R_a0 1	0	32767	100	Num
Ra Table	R_a0	88	6	I2	R_a0 2	0	32767	113	Num
Ra Table	R_a0	88	8	I2	R_a0 3	0	32767	143	Num
Ra Table	R_a0	88	10	I2	R_a0 4	0	32767	98	Num
Ra Table	R_a0	88	12	I2	R_a0 5	0	32767	97	Num
Ra Table	R_a0	88	14	I2	R_a0 6	0	32767	108	Num
Ra Table	R_a0	88	16	I2	R_a0 7	0	32767	89	Num
Ra Table	R_a0	88	18	I2	R_a0 8	0	32767	86	Num
Ra Table	R_a0	88	20	I2	R_a0 9	0	32767	85	Num
Ra Table	R_a0	88	22	I2	R_a0 10	0	32767	87	Num
Ra Table	R_a0	88	24	I2	R_a0 11	0	32767	90	Num
Ra Table	R_a0	88	26	I2	R_a0 12	0	32767	110	Num
Ra Table	R_a0	88	28	I2	R_a0 13	0	32767	647	Num
Ra Table	R_a0	88	30	I2	R_a0 14	0	32767	1500	Num
Ra Table	R_a0x	89	0	H2	R_a0x Flag	0	ffff	ffff	Hex
Ra Table	R_a0x	89	2	I2	R_a0x 0	0	32767	105	Num
Ra Table	R_a0x	89	4	I2	R_a0x 1	0	32767	100	Num
Ra Table	R_a0x	89	6	I2	R_a0x 2	0	32767	113	Num
Ra Table	R_a0x	89	8	I2	R_a0x 3	0	32767	143	Num
Ra Table	R_a0x	89	10	I2	R_a0x 4	0	32767	98	Num
Ra Table	R_a0x	89	12	I2	R_a0x 5	0	32767	97	Num
Ra Table	R_a0x	89	14	I2	R_a0x 6	0	32767	108	Num

Table 7-1. Data Flash Summary (continued)

CLASS	SUBCLASS	SUBCLASS ID	OFFSET	TYPE	NAME	MIN	MAX	DEFAULT	UNIT
Ra Table	R_a0x	89	16	I2	R_a0x 7	0	32767	89	Num
Ra Table	R_a0x	89	18	I2	R_a0x 8	0	32767	86	Num
Ra Table	R_a0x	89	20	I2	R_a0x 9	0	32767	85	Num
Ra Table	R_a0x	89	22	I2	R_a0x 10	0	32767	87	Num
Ra Table	R_a0x	89	24	I2	R_a0x 11	0	32767	90	Num
Ra Table	R_a0x	89	26	I2	R_a0x 12	0	32767	110	Num
Ra Table	R_a0x	89	28	I2	R_a0x 13	0	32767	647	Num
Ra Table	R_a0x	89	30	I2	R_a0x 14	0	32767	1500	Num
Calibration	Data	104	0	F4	CC Gain	1.00E-01	4.00E+01	0.4768	mΩ
Calibration	Data	104	4	F4	CC Delta	2.98E+04	1.19E+06	567744.56	mΩ
Calibration	Data	104	8	I2	CC Offset	–32768	32767	–1200	Num
Calibration	Data	104	10	I1	Board Offset	–128	127	0	Num
Calibration	Data	104	11	I1	Int Temp Offset	–128	127	0	0.1°C
Calibration	Data	104	12	I1	Ext Temp Offset	–128	127	0	0.1°C
Calibration	Data	104	14	U2	Voltage Divider	0	65535	5000	mV
Calibration	Current	107	1	U1	Deadband	0	255	5	mA
Security	Codes	112	0	H4	Sealed to Unsealed	0	ffffffff	36720414	hex
Security	Codes	112	4	H4	Unsealed to Full	0	ffffffff	ffffffff	hex
Security	Codes	112	8	H4	Authen Key3	0	ffffffff	1234567	hex
Security	Codes	112	12	H4	Authen Key2	0	ffffffff	89abcdef	hex
Security	Codes	112	16	H4	Authen Key1	0	ffffffff	fedcba98	hex
Security	Codes	112	20	H4	Authen Key0	0	ffffffff	76543210	hex

Gas Gauge Timing Considerations

A.1 Gauging Effects on I^{2}C Transactions

The sources for gauge operations to impact I^{2}C transactions are interrupt latency, command decode, and the execution of critical code sections. These result in variability to the clock stretches observed on the I^{2}C bus. There is a slight variability in interrupt latency because the timing of communications is asynchronous to gauge processes. The execution path between a halted gauge and one with an active process can be slightly different. Command decoding adds several µs of variability to clock stretches; that is, some commands are decoded more quickly than others. The impact of critical sections is to impact clock stretching on the I^{2}C bus. The most critical sections only add 5 µs to 10 µs of clock stretch. However, if the gauge has initiated an update to its flash data memory, it is possible to see a series of 2-ms to 20-ms clock stretches. With no synchronizing mechanism between the host and gauge, these stretches will appear as random occurrences to the host system.

A.2 HDQ Bus Effects on Gauging

Bus cycles consumed by HDQ communication must not exceed the same budget as for I^{2}C. HDQ uses all the same interrupts and process operations that I^{2}C uses. The gauge firmware abstracts communication to be hardware-layer independent. However, it should be noted that each byte exchange for HDQ requires about 3.6 ms (address + data), so only about 280 transactions per second are possible. This communication rate would require ((140 + 100) µs × 280) equals 67 ms of CPU time per second. Most hosts only require reading 20 bytes in any one second. That data rate demands only around 5 ms of CPU time.

A.3 Gauging Effects on HDQ Transactions

HDQ, like I^{2}C, runs in the highest priority process. However, it cannot clock stretch, so flow control between the host and gauge is not possible. For HDQ, there is concern for timing violations for the host read and write requests. The interrupt latency plus critical section plus command decode will not cause timing violations. However, gauge- or host-initiated data flash memory updates can cause timing violations. Host firmware should plan for these violations. It is recommended on timing-violation detection for the host to delay 70 ms before restarting a transaction to allow for a typical single-page flash update time. Three retries should be allowed to account for a possible flash write recovery and a possible two-page flash write. The restarted transaction should be preceded by the "break" signal to assure proper synchronization of the data frame.

The gauge provides a FULLSLEEP power mode feature to minimize power consumption by the gauge when the host system is in a standby mode: low power consumption. This mode disables a clock used by the HDQ communication peripheral. Because of this, it is likely when the gauge is in FULLSLEEP mode that a first communication to the gauge will time out. A signal on the HDQ pin will start the clock running, but it takes several milliseconds to stabilize and during that time the HDQ peripheral will not be able to read bits transmitted by the host. A reread or write of the register at 4-ms intervals will be successful by the third attempt.

NOTE: A delay interval greater than 1 second may allow the gauge to reenter FULLSLEEP. Often, this is not an issue, because the host system having exited its standby mode will be drawing sufficient current, such that the gauge will have already not returned to FULLSLEEP until the FULLSLEEP conditions are satisfied.

A.4 Manufacturer Timing Notes

During pack manufacturing, the gauge may be requested by the host to update certain flash constants that are reflective of specific characteristics of the cell and its system requirements. When the manufacturer host system requests updates, it must take into account the flash update times. The gauge firmware ACKs a properly received data flash write request. This has the effect to immediately release the bus. The bus is released before the flash write activity is completed. The host should delay 250 ms after receiving an ACK from the gauge to ensure all flash write operations have completed.

There are other MAC commands that are also requests for the gauge to perform certain operations. Commands that request specific information about the gauge should allow about 2 ms for the gauge to prepare data. Commands that request the gauge to compute checksums should allow more time. 20 ms is recommended.

HDQ Communication Basics

This appendix compares the communication timing specifications for the different battery fuel gauge and battery monitor products. It discusses some of the interface requirements necessary for robust HDQ communication, such as the need for a break prior to each communication, how to reliably read 16-bit dynamic values over the 8-bit bus, and the need for some noise filtering on the HDQ line. It also discusses the firmware strategy required for implementing the HDQ interface using a discrete I/O port, and describes the option of using a UART to handle the HDQ bit timing requirement.

B.1 Basic HDQ Protocol

HDQ communication between a host device and slave device uses a single-wire, open-drain interface. The communication protocol is asynchronous return-to-one referenced to V_{SS}. A passive pullup resistance is required to pull the HDQ line to a high state when neither the host nor the slave is pulling the line low during the two-way communication over the single-wire interface. The interface uses a command-based protocol where the host sends a command byte to the HDQ slave device. The command directs the slave either to store the next 8 bits of data received to a register specified by the command byte (write command), or to output the 8 bits of data from a register specified by the command byte (read command). Command and data bytes consist of a stream of bits that have a maximum transmission rate of 5 Kb/s. The least-significant bit of a command or data byte is transmitted first. The first 7 bits of the command word are the register address, and the last command bit transmitted is the read/write (R/W) bit. Figure B-1 illustrates a typical HDQ read cycle.

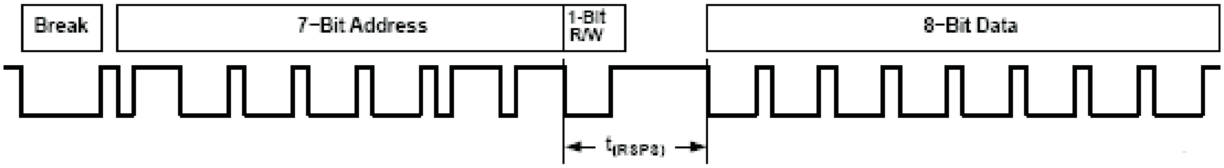

Figure B-1. Typical HDQ Read Cycle

The HDQ line may remain high for an indefinite period of time between each bit of address or between each bit of data on a write cycle. After the last bit of address is sent on a read cycle, the HDQ slave starts outputting the data after the specified response time, $t_{(RSPS)}$. The response time is measured from the fall time of the command R/W bit to the fall time of the first data bit returned by the slave and therefore includes the entire bit time for the R/W bit. (The response time is *not* the time after the last command bit before the first data bit of the response begins.) Because the minimum response time is equal to the minimum bit cycle time, this means that the first data bit may begin as soon as the command R/W bit time ends.

This communication protocol may be referred to as HDQ8 to distinguish it from the HDQ16 protocol used by some devices like the bq2060 and bq2063. The bit timing of HDQ16 is identical to that of HDQ8, except that 16 bits of data are written or read instead of 8 bits. The HDQ16 command word is still a 7-bit address plus a R/W bit.

B.2 Break

The HDQ communication engine is reset if the HDQ line is held low for longer than the 190-µs minimum break time, $t_{(B)}$. If the host does not get an expected response from the HDQ slave or if the host needs to restart a communication before it is complete, the host can hold the HDQ line low and generate a break to reset the communication engine. The next communication can begin after the 40-µs minimum break recovery time, $t_{(BR)}$. Figure B-2 illustrates break timing.

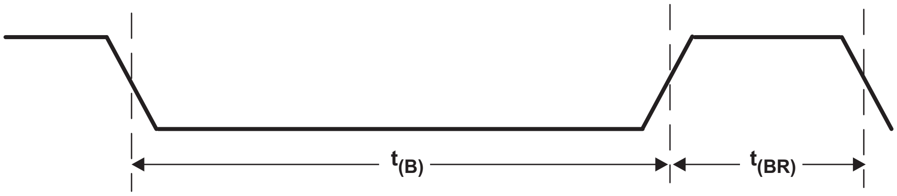

Figure B-2. HDQ Break and Break Recovery Timing

If the HDQ line is disconnected and reconnected, unpredictable bit values may be input to the HDQ engine, leaving it in a non-reset state. It may be possible for the HDQ line to be disconnected during the middle of a communication. If HDQ communications are interrupted or unexpected transients occur on the HDQ line and there are no periodic breaks to reset the communication engine, it is possible for the HDQ slave engine to perform an incorrect command, including an unintended write.

The most robust design practice for reliable HDQ communication is to precede each communication with a break. This ensures that the engine is reset before each communication, and minimizes the risk of unintended writes and/or reading incorrect data. Figure B-1 illustrates a typical HDQ read cycle following a break.

B.3 Basic Timing

Figure B-3 and Figure B-4 show the HDQ bit timing.

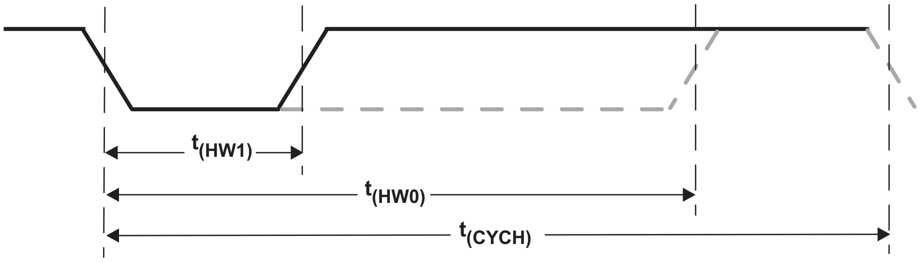

Figure B-3. Host-Transmitted HDQ Bit

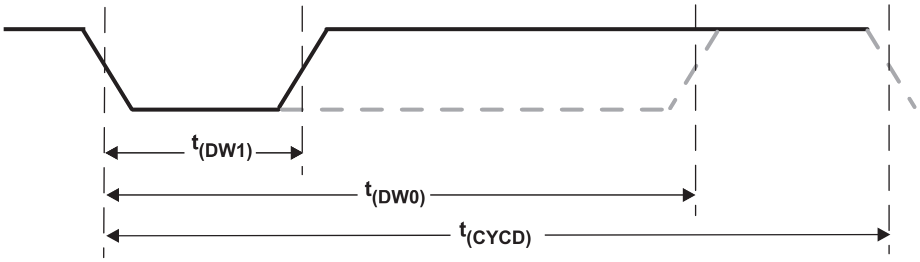

Figure B-4. Slave-Transmitted HDQ Data Bit

B.4 Reading 16-Bit Words

Reading 16-bit dynamic values from an 8-bit HDQ device requires special care to ensure that a register update occurring during the process of reading the two 8-bit bytes of the 16-bit word does not cause an inappropriate value read by the host. For example, if the 16-bit word is an incrementing or decrementing counter, it is possible that a carry or borrow to the high byte could occur after one data byte is read, but before the other data byte is read. In this case, the word read by the host might be as much as 256 counts in error due to the low-high byte rollover that occurred during the data read process. To prevent any system issues, any 16-bit values read by the host should be read with the following protocol:

1. Read high byte (H0)
2. Read low byte (L0)
3. Read high byte again (H1)
4. If H1 = H0, then the valid 16-bit result is H0:L0.
5. Otherwise, read low byte again (L1) and the valid 16-bit result is H1:L1.

This procedure is sufficient to ensure that the 16-bit word is read correctly if the 3-byte to 4-byte reads occur in less time than the update rate of the register value.

B.5 Host Processor Interrupts Using Discrete I/O Port for HDQ

If the host implements the HDQ communication using a discrete processor I/O port, the timing of the transmitted HDQ data and the sampling of the received HDQ data depends on the host processor timing of the transitions on the HDQ line. If the HDQ communication routine is interrupted during a communication, it may cause the transmitted times to stretch, and the received data may not be interpreted correctly.

One solution is to disable interrupts during HDQ communication critical times. When the host is sending the address or data, there is no restriction on the time between each bit, so host interrupts can be enabled during the high time between bits. Interrupts may need to be disabled during the low bit times to ensure that the bit low times meet the required HDQ timing constraints. After the last address bit (a R/W bit) is sent on a read, interrupts also must be disabled to ensure that the received data is sampled correctly. Interrupts must be disabled during the entire receipt of the 8 data bit times.

If disabling interrupts during HDQ communication critical times is not feasible, another approach is to leave interrupts enabled, but to have all enabled interrupts set a flag that can be read to determine whether an interrupt occurred during an HDQ communication. The strategy would be to stop the communication with a break and then retry the communication if an interrupt occurred during an HDQ communication. This method requires that there be a reasonable probability of completing an HDQ communication without an interrupt.

B.6 Using UART Interface to HDQ

An implementation option for HDQ is to use a UART. An advantage to using the UART is that if the UART is handling the communication and storing the results in a buffer, host processor interrupts during the communication do not affect the timing of the HDQ communication with the HDQ slave. Use of the UART for HDQ communication requires that each word sent to or received from the UART is only a single bit of the HDQ data or address. The procedure is to set the UART baud rate to 57,600 with no parity and two stop bits. This yields a data word with 11 bits total (start bit, eight data bits, and two stop bits). At a baud rate of 57,600 (17.3 µs per bit), this is a total communication time of 190.9 µs and meets the required HDQ bit timing of 190 µs minimum. If data of 0xFE is sent to the UART, the transmitted data is low for 34.6 µs and then high for the remaining bit time and is interpreted by the HDQ slave as a 1. If data of 0xC0 is sent to the UART, the transmitted data is low for 121.5 µs and then high for the remaining bit time and is interpreted by the HDQ slave as a 0. When data is sent to the host from the HDQ slave, the received data could be interpreted as either 0xFE or 0xFC if a logic 1 is sent, or either 0xF0, 0xE0, 0xC0, 0x80, or 0x00 if a logic 0 is sent. A simple test of the received data determines the received data bit. If the received data is greater than 0xF8, the data bit should be interpreted as a logic 1, and if less than or equal to 0xF8, the data bit should be interpreted as a logic 0. This analysis assumes the UART samples the received data approximately half-way through each of the 17.3-µs UART bit times, and that capacitive loading on the HDQ line may delay the rise time of the data a few microseconds.

Note that the TX and RX of the UART must be tied together because HDQ is a single-wire interface. In case the TX output is not an open-drain output, it needs to be converted to an open-drain output, as shown in Figure B-5.

NOTE: Any data sent out to the HDQ slave is also received by the UART; therefore, if 8 bits of an address are sent and then 8 bits of data from the HDQ slave are received from that address, the UART inputs 16 bytes of data into the UART data buffer. The host needs to skip the first 8 bytes, which contain the command word sent to the HDQ slave and use the second 8 bytes of data.

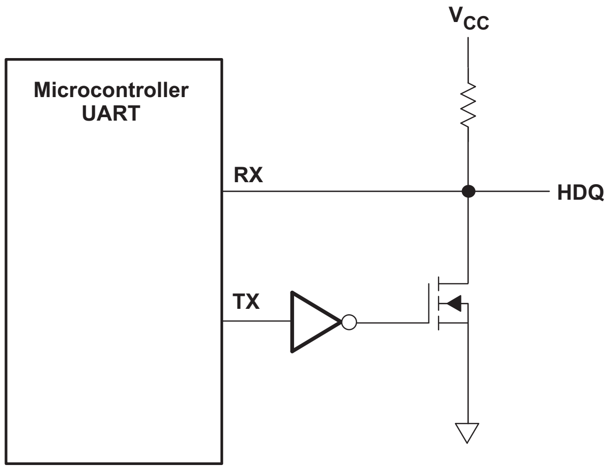

Figure B-5. HDQ Communication Using UART Without Open-Drain Output

Procedures to Seal and Unseal the Gauge

C.1 Unseal the Gauge to UNSEALED Mode

The procedure to unseal the gauge to UNSEALED mode is as follows:

1. Read the *CONTROL_STATUS* register and verify that the *[FAS]* and *[SS]* bits are set, indicating that the battery is in the SEALED state.
2. Send the low-byte Unseal Key (the low byte is 0414; the default Unseal Key is 0x36720414).
3. Send the high-byte Unseal Key (the high byte is 3672; the default Unseal Key is 0x36720414).
4. Wait 100 ms.
5. Read the *CONTROL_STATUS* register and verify that the *[FAS]* bit is set and the *[SS]* bit is cleared, indicating that the battery is in the UNSEALED state. If the *[SS]* bit is set, then return to Step 2. In this state, the fuel gauge allows data flash modification and complete operation of the UNSEAL state with the exception of modifying fuel gauge firmware or reading or modifying the security keys (Unseal Key or Full Access Key).
6. Retry three times before stopping the sequence.

C.2 Unseal the Gauge to FULL ACCESS Mode

The procedure to unseal the gauge to FULL ACCESS mode is as follows:

1. Read the *CONTROL_STATUS* register and verify that the *[FAS]* bit is set and the *[SS]* bit is cleared, indicating that the battery is in the UNSEALED state.
2. Send the low-byte Full Access Key (the low byte is FFFF; the default Full Access Key is 0xFFFFFFFF).
3. Send the high-byte Full Access Key (the high byte is FFFF; the default Full Access Key is 0xFFFFFFFF).
4. Wait 100 ms.
5. Read the *CONTROL_STATUS* register and verify that the *[FAS]* and *[SS]* bits are cleared, indicating that the battery is in the FULL ACCESS mode. If the *[FAS]* bit is set, but the *[SS]* bit is cleared, then return to Step 2. In this state, the fuel gauge allows full access to data flash, commands, and the ability to update firmware.
6. Retry three times before stopping the sequence.

C.3 Seal the Gauge

The procedure to seal the gauge is as follows:

1. Send the *Control()* command 0x0020 (send 0x0020 to 0x00/0x01).
2. Wait 200 ms.
3. Read the *CONTROL_STATUS* register and verify that the *[FAS]* and *[SS]* bits are set, indicating that the battery is in the SEALED state. If either the *[FAS]* or the *[SS]* bit is cleared, return to Step 1.
4. Retry three times before stopping the sequence.

Impedance Track Gauge Configuration

D.1 Introduction

The gauge uses a combination of open circuit voltage (OCV) measurements and coulomb counting to determine the absolute state of charge (SOC).

There are typically five necessary steps, as follows, to complete prior to using a gauge with a battery:

1. Determine the ChemID.
2. Perform a learning cycle.
3. Test the gauge and optimize.
4. Finalize the golden file.
5. Program and test the PCB.

D.2 Determining ChemID

A proper ChemID or profile for a given battery must be determined for the Impedance Track algorithm.

To determine ChemID for a given battery, do the following:

a. Look up the cell/pack in the TI database to see if there is an existing ChemID already available. To do this, use bqStudio with the latest chemistry plugin (http://www.ti.com/lit/zip/sluc564).

b. If an existing ChemID cannot be found, use the *Gauging Parameter Calculator Chemistry* matching tool (GPCCHEM) to submit logs and to get a closely matched ChemID. For more details, see the *Simple Guide to Chemical ID Selection Tool (GPC) Application Note* (SLVA725).

c. If there is no match from either (1) or (2), cells can be sent to TI to characterize and generate a custom ChemID.

D.3 Learning Cycle

For the most optimum gauging, the pack/cell must undergo a learning cycle with the gauge. The gauge indicates the progress of the learning cycle with the following control flag bits:

1. ***VOK***: This bit tracks when the gauge measures the battery's voltage. This bit is normally set when charging/discharging starts, and is cleared when discharging stops. The gauge has detected that the battery voltage has stabilized, and has taken the OCV measurement. It is a good way to track when OCV measurements occur.
2. ***RUP_DIS***: If set, the gauge cannot determine its current state and calculations, and will not update the battery resistance tables with its presently measured data. It is cleared when a good OCV measurement is taken, as the gauge acknowledges the absolute state of charge based on this value.
3. ***FC***: This flag indicates whether or not the gauge detects the battery as "full."
4. ***Update Status***: This data flash value contains the current status of the learning cycle.

Figure D-1 shows the voltage and current.

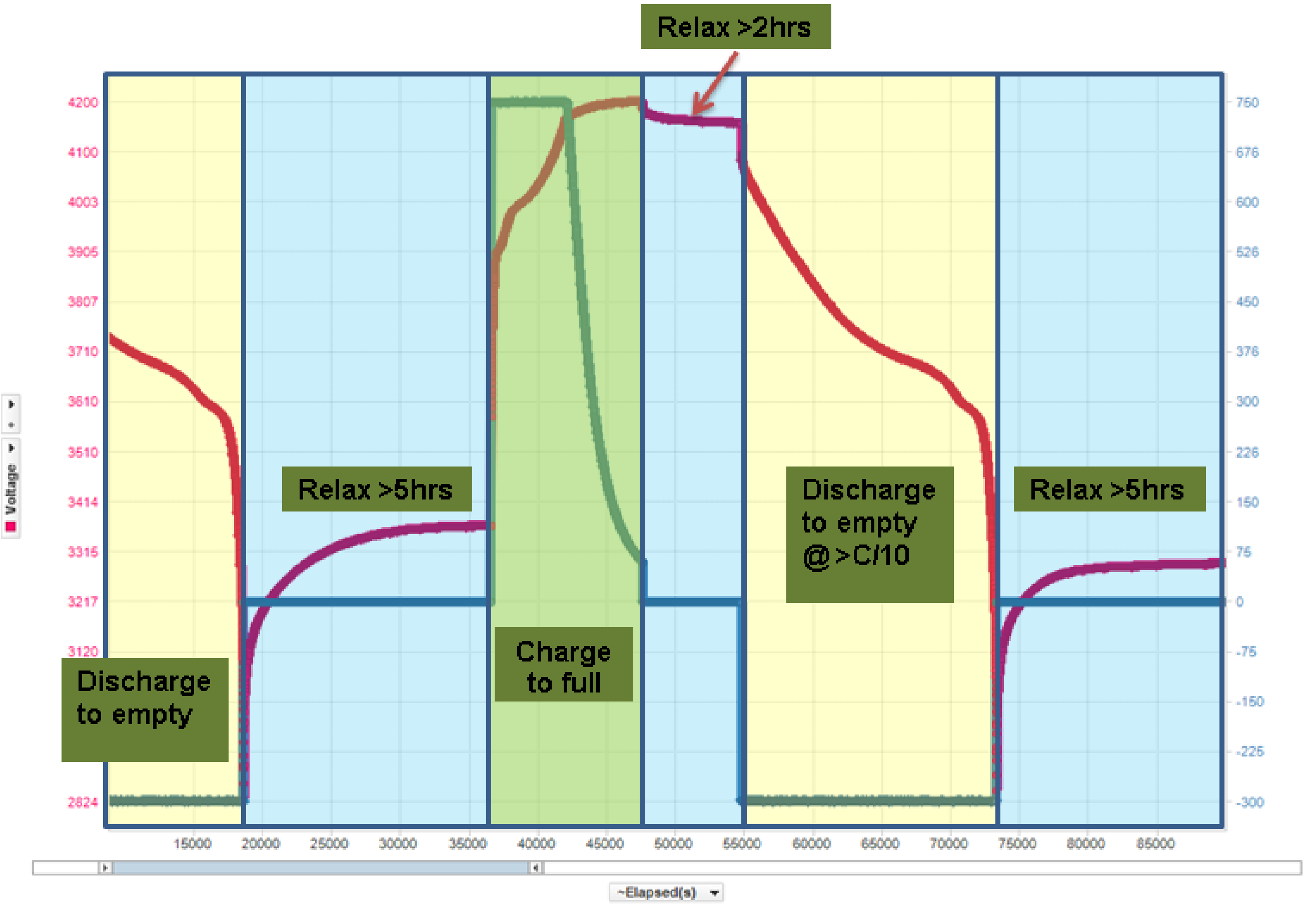

Figure D-1. Voltage/Current

Figure D-2 shows the value of the control flag bits and the ***Update Status*** for each state.

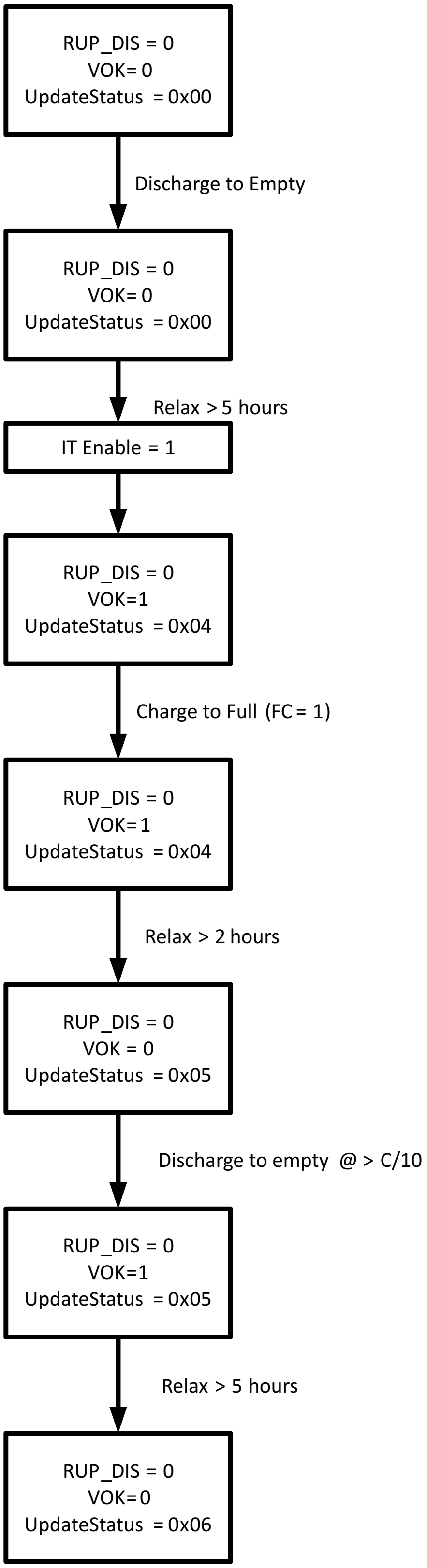

Figure D-2. Update Status Learning Cycle Diagram

D.4 Common Problems Seen During the Learning Cycle

To diagnose a failed learning cycle, it is critical to set bqStudio to log the data RAM during the cycle every ~5–10 seconds. It is also advisable to auto-export the data flash on a less frequent basis (every ~1–10 minutes). This way, all of the information is collected to determine the point of failure.

Tips for the learning cycle are as follows:

1. Ensure that the battery is at a low SOC and relaxed when running *IT_ENABLE*. This is typically not a problem, but it is important.
2. During the charge, ensure that the gauge detects the FULL CHARGE condition. If after the learning cycle, ***Update Status*** has not been updated to 05 (that is, it is still 04), then this may be the problem. The gauge detects the full charge condition with three criteria:
   a. Battery voltage is within 0.1 V of the ***Charging Voltage*** as defined in bqStudio.
   b. Battery current is below ***Taper Current***, as defined in bqStudio.
   c. Battery current stays below this ***Taper Current*** and above the ***Quit Current*** for over 40 seconds.

   This means that the battery must be charging with a significant current below the ***Taper Current*** for almost a minute. If the charger cuts off before or just after the current drops below the ***Taper Current***, as indicated in bqStudio, then the gauge does not detect the "full" condition.
3. When the battery is fully charged, wait long enough for the ***VOK*** bit to be cleared. This is generally two hours. Logging data RAM shows if this has occurred. ***Update Status*** will stay 04 if this does not happen. If ***VOK*** is never set, then this means that it did not start with an empty battery or did not fully charge the battery or the charge cycle was disturbed in some way (see Step 7) for charging information.
4. When discharging starts, ensure that the ***VOK*** bit is set. The gauge has the data flash parameter ***Dsg Threshold*** that determines the minimum current needed to enter the DISCHARGE state. If it discharges with less than this current, the ***VOK*** bit will not be set. The gauge will never go into the DISCHARGE state, and the Ra tables will never be updated. ***Update Status*** will be 05. Either decrease ***Dsg Threshold*** in the data flash, or increase the discharge rate.
5. During the discharge, the resistance table is *never* updated. If during a learning cycle the data flash log shows that the resistance table never changed, this indicates that the discharge load was too light. The gauge needs to measure a significant voltage drop across the internal battery impedance before it can measure the impedance. If the load is too light, the measurement fails, and it will never get any resistance table updates.
6. During discharge, the resistance table may update for a while and then stop. When this happens, RUP_DIS is set. This indicates that the Chemistry ID choice is incorrect. This means that the gauge has measured a resistance value that just does not make sense (that is, is negative). A chemistry cycling is needed to identify the correct chemistry profile.
7. General charge/discharge profile information: Most learning cycles fail because there is something incorrect with the charge/discharge profile. The following are suggestions:
   a. Ensure charger cutoff upon charge completion: Most users do not have a battery cycling automation setup, so a bench power supply is used to change a battery overnight. This is not recommended. The system does follow a CC/CV profile, but there is no cutoff. Therefore, when the gauge recognizes a full charge and tries to take an OCV measurement, it actually measures the supply voltage and disrupts the system.
   b. CC/CV charge profile: In line with the above, ensure that a CC/CV profile charger with reasonable values is used: C/2 fast charge rate, C/100 to C/10 taper current.
   c. Continuous charge profile: While not strictly necessary, it is advisable to ensure that the charging profile is continuous. If the charging cycle stops, then the cycle *may* fail. If the battery discharges for any reason during this time, the cycle will fail.
   d. After charge, relax at least two hours with NO load/charger. Wait long enough to see ***VOK***. Two hours is generally enough.
   e. Discharge C/5 constant current. Use a C/5 current. This is preferred, and there can be some error; however, if the current drifts too high or too low, the cycle can fail. Too low is around C/10; too high is around C/3 to C/2. Smaller cells (<800 mAh) are much less forgiving in this regard.
   f. Continuous discharge: This is absolutely necessary. If the discharge ever stops before reaching

the terminate voltage, the cycle will fail.

g. Termination voltage: Ensure that when terminate voltage is reached, the load is removed. Let the cell relax. If the load stays attached to the battery and causes the battery voltage to drop well below terminate, then not only does the learning cycle fail, but it also damages the battery.

8. If Impedance Track is not enabled (*IT_ENABLE* = 0), ***Qmax*** updates can still occur if a relax period lasts more than 5 hours.

D.5 Test Gauge and Optimize

Once learning is completed, the fuel gauge can be tested in an actual application environment to start optimizing the performance across temperature and load corners. TI offers the GPCRA0 tool as part of the Gauge Parameter Calculator to help users. For more details, see the *Simple Guide to GPC Golden GG Maker Tool Application Note* (SLUUBC9). For very low temperature operations (typically below 5°C, system temperature distribution can deviate from lab conditions and so resistance temperature compensation parameters need to be adjusted. TI offers the GPCRB tool as part of the Gauge Parameter Calculator to help achieving this. For more details, see the *Golden GG Maker and Resistance Temperature Application Note* (SLUUBD0).

D.6 Finalize Golden File

After all the optimizations have been performed and validation is done, the golden file can be created and exported from bqStudio.

D.7 Program and Test the PCB

Once the final golden file is ready, it can be programmed to the battery pack PCB and then tested directly with the battery pack.

NOTE: A .gg.csv file is not sufficient for production programming, as it does not contain the entire data flash image. Typical file formats used for production are .dfi, .df.fs, .bq.fs, or .srec.

Revision History

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

Date	Revision	Notes
July 2018	*	Initial Release

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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