# bq27541-G1 单节锂离子电池电量计，用于电池组集成

## 1 特性

- 用于高达 14500 毫安时 容量的 1 串联 (1sXp) 锂离子应用的电池电量计
- 微控制器外设提供：
  - 准确的电池电量计量支持高达 14500 毫安时
  - 用于电池温度报告的内部或者外部温度传感器
  - 安全哈希算法 (SHA)-1 / 哈希消息认证码 (HMAC) 认证
  - 使用寿命的数据记录
  - 64 字节非易失性暂用闪存
- 基于已获专利的 Impedance Track™ 技术的电池电量计量
  - 用于电池续航能力精确预测的电池放电模拟曲线
  - 可针对电池老化、电池自放电以及温度/速率低效情况进行自动调节
  - 低值感应电阻器（5mΩ 至 20mΩ）
- 高级电量计量特性
  - 内部短路检测
  - 电池端子断开侦测
- 高速 1 线 (HDQ) 和用于与主机系统通信的 I²C 接口格式
- 小型 12 引脚 2.5mm x 4mm 小外形尺寸无引线 (SON) 封装

## 2 应用范围

- 智能电话
- 平板电脑
- 数码相机与视频摄像机
- 手持式终端
- MP3 或多媒体播放器

## 3 说明

德州仪器 (TI) bq27541-G1 锂离子电池 电量 计是一款微控制器外设，此外设能够提供针对单节锂离子电池组的电量计量。 此器件只需很少的系统微控制器固件开发即可实现精确电池电量计量。 此 电量 计驻留在电池组内部，或者驻留在具有一个嵌入式电池（不可拆卸）的系统的主板上。

此 电量 计使用已经获得专利的 Impedance Track™ 算法来进行电量计量，并提供诸如剩余电量 (mAh)，充电状态 (%)，续航时间（分钟），电池电压 (mV)，和温度 (°C) 等信息。 它还提供针对内部短路或者电池端子断开事件的检测功能。
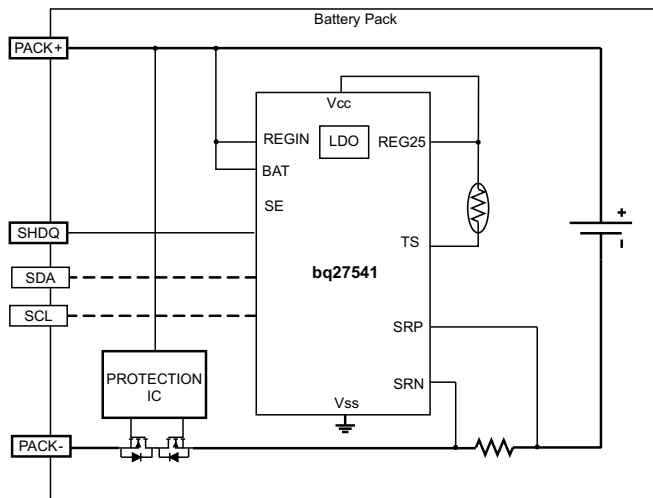
此 电量 计还特有针对安全电池组认证（使用 SHA-1/HMAC 认证算法）的集成支持功能。

### 器件信息[1]

| 产品型号 | 封装 | 封装尺寸（标称值） |
|---|---|---|
| bq27541-G1 | SON (12) | 2.50mm x 4.00mm |

(1) 要了解所有可用封装，请见数据表末尾的可订购产品附录。

## 4 典型应用

English Data Sheet: SLUSAL6

## 目录

## 5 修订历史记录

**Changes from Revision C (October 2012) to Revision D**          **Page**

**Changes from Revision B (June 2012) to Revision C**          **Page**

### Device Comparison Table

| PRODUCTION PART NO.[1] | PACKAGE[2] | $T_A$ | COMMUNICATION FORMAT | TAPE AND REEL QUANTITY |
|---|---|---|---|---|
| bq27541DRZR-G1 | SON-12 | –40°C to 85°C | I²C, HDQ[1] | 3000 |
| bq27541DRZT-G1 | | | | 250 |

(1) bq27541-G1 is shipped in I²C mode
(2) For the most current package and ordering information, see the Package Option Addendum at the end of this document; or, see the TI website at www.ti.com.

## 6 Pin Configurations and Functions

**bq27541 Pin Diagram**
**(Top View)**
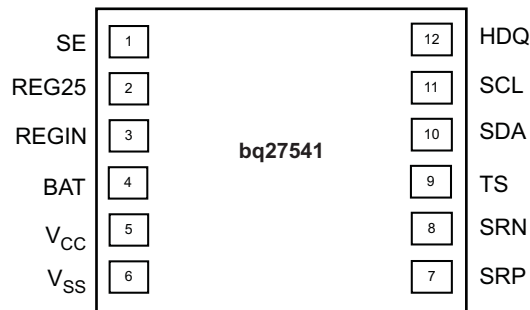
### Pin Functions

| PIN | | TYPE[1] | DESCRIPTION |
|---|---|---|---|
| NAME | NUMBER | | |
| BAT | 4 | I | Cell-voltage measurement input. ADC input. Decouple with 0.1-µF capacitor. |
| HDQ | 12 | I/O | HDQ serial communications line (Slave). Open-drain. Use with 10-kΩ pullup resistor (typical) or leave floating when it is not used. |
| REG25 | 2 | P | 2.5-V output voltage of the internal integrated LDO. Connect a minimum 0.47-µF ceramic capacitor. |
| REGIN | 3 | P | The input voltage for the internal integrated LDO. Connect a 0.1-µF ceramic capacitor. |
| SCL | 11 | I | Slave I²C serial communications clock input line for communication with system (Slave). Open-drain I/O. Use with 10-kΩ pullup resistor (typical). |
| SDA | 10 | I/O | Slave I²C serial communications data line for communication with system (Slave). Open-drain I/O. Use with 10-kΩ pullup resistor (typical). |
| SE | 1 | O | Shutdown Enable output. Push-pull output. Leave floating when it is not used. |
| SRN | 8 | IA | Analog input pin connected to the internal coulomb counter with a Kelvin connection where SRN is nearest the PACK– connection. Connect to 5-mΩ to 20-mΩ sense resistor. |
| SRP | 7 | IA | Analog input pin connected to the internal coulomb counter with a Kelvin connection where SRP is nearest the CELL– connection. Connect to 5-mΩ to 20-mΩ sense resistor |
| TS | 9 | IA | Pack thermistor voltage sense (use 103AT-type thermistor). ADC input |
| $V_{CC}$ | 5 | P | Processor power input. The minimum 0.47-µF capacitor connected to REG25 should be close to $V_{CC}$. |
| $V_{SS}$ | 6 | P | Device ground |

(1) I/O = Digital input/output, IA = Analog input, P = Power connection

# 7 Specifications

## 7.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)[1]

|  |  | VALUE | | UNIT |
|---|---|---|---|---|
|  |  | **MIN** | **MAX** |  |
| $V_I$ | Regulator input, REGIN | −0.3 | 24 | V |
| $V_{CC}$ | Supply voltage range | −0.3 | 2.75 | V |
| $V_{IOD}$ | Open-drain I/O pins (SDA, SCL, HDQ) | −0.3 | 6 | V |
| $V_{BAT}$ | BAT input (pin 4) | −0.3 | 6 | V |
| $V_I$ | Input voltage range to all others (pins 1, 7, 8, 9) | −0.3 | $V_{CC}$ + 0.3 | V |
| $T_F$ | Functional temperature range | −40 | 100 | °C |

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

## 7.2 Handling Ratings

|  |  |  | **MIN** | **MAX** | **UNIT** |
|---|---|---|---|---|---|
| $T_{stg}$ | Storage temperature range | | −65 | 150 | °C |
| $V_{(ESD)}$ | Electrostatic discharge | Human Body Model (HBM), BAT pin | | 1.5 | kV |
|  |  | Human Body Model (HBM), per ANSI/ESDA/JEDEC JS-001, all pins[1] | | 2 | |
|  |  | Charge Device Model (CDM), per JEDEC specification JESD22-C101, all pins[2] | −250 | 250 | V |

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

## 7.3 Recommended Operating Conditions

$T_A$ = -40°C to 85°C; typical values at $T_A$ = 25°C and $V_{(REGIN)}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| | | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_I$ | Supply voltage, REGIN | No operating restrictions | 2.7 | | 5.5 | V |
| | | No FLASH writes | 2.45 | | 2.7 | |
| $I_{CC}$ | Normal operating mode current [1] | Fuel gauge in NORMAL mode. $I_{LOAD}$ > *Sleep Current* | | 131 | | µA |
| $I_{(SLP)}$ | Low-power operating mode current[1] | Fuel gauge in SLEEP mode. $I_{LOAD}$ < *Sleep Current* | | 60 | | µA |
| $I_{(FULLSLP)}$ | Low-power operating mode current[1] | Fuel gauge in FULLSLEEP mode. $I_{LOAD}$ < *Sleep Current* | | 21 | | µA |
| $I_{(HIB)}$ | Hibernate operating mode current [1] | Fuel gauge in HIBERNATE mode. Available in $I^2C$ mode only. $I_{LOAD}$ < *Hibernate Current* | | 6 | | µA |
| $V_{OL}$ | Output voltage low (HDQ, SDA, SCL, SE) | $I_{OL}$ = 3 mA | | | 0.4 | V |
| $V_{OH(PP)}$ | Output high voltage (SE) | $I_{OH}$ = −1 mA | $V_{CC}$ − 0.5 | | | V |
| $V_{OH(OD)}$ | Output high voltage (HDQ, SDA, SCL) | External pullup resistor connected to $V_{CC}$ | $V_{CC}$ − 0.5 | | | V |
| $V_{IL}$ | Input voltage low (HDQ, SDA, SCL) | | −0.3 | | 0.6 | V |
| $V_{IH}$ | Input voltage high (HDQ, SDA, SCL) | | 1.2 | | 6 | V |
| $V_{(A1)}$ | Input voltage range (TS) | | $V_{SS}$ − 0.125 | | 2 | V |
| $V_{(A2)}$ | Input voltage range (BAT) | | $V_{SS}$ − 0.125 | | 5 | V |
| $V_{(A3)}$ | Input voltage range (SRP, SRN) | | $V_{SS}$ − 0.125 | | 0.125 | V |
| $I_{lkg}$ | Input leakage current (I/O pins) | | | | 0.3 | µA |
| $t_{PUCD}$ | Power-up communication delay | | | 250 | | ms |

(1) Specified by design. Not tested in production.

## 7.4 Thermal Information

over operating free-air temperature range (unless otherwise noted)

| | THERMAL METRIC[1] | bq27541-G1 DRZ (12 PINS) | UNIT |
|---|---|---|---|
| $R_{\theta JA}$ | Junction-to-ambient thermal resistance | 64.1 | °C/W |
| $R_{\theta JCtop}$ | Junction-to-case (top) thermal resistance | 59.8 | |
| $R_{\theta JB}$ | Junction-to-board thermal resistance | 52.7 | |
| $\psi_{JT}$ | Junction-to-top characterization parameter | 0.3 | |
| $\psi_{JB}$ | Junction-to-board characterization parameter | 28.3 | |
| $R_{\theta JCbot}$ | Junction-to-case (bottom) thermal resistance | 2.4 | |

(1) For more information about traditional and new thermal metrics, see the *IC Package Thermal Metrics* application report, SPRA953.

## 7.5 Power-on Reset

$T_A$ = −40°C to 85°C, $C_{(REG)}$ = 0.47 µF, 2.45 V < $V_{(REGIN)}$ = $V_{BAT}$ < 5.5 V; typical values at $T_A$ = 25°C and $V_{(REGIN)}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_{IT+}$ | Positive-going battery voltage input at $V_{CC}$ | | 2.05 | 2.20 | 2.31 | V |
| $V_{HYS}$ | Power-on reset hysteresis | | | 115 | | mV |

## 7.6 2.5-V LDO Regulator[1]

$T_A$ = −40°C to 85°C, $C_{(REG)}$ = 0.47 µF, 2.45 V < $V_{(REGIN)}$ = $V_{BAT}$ < 5.5 V; typical values at $T_A$ = 25°C and $V_{(REGIN)}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| PARAMETER | | TEST CONDITION | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $V_O$ | Regulator output voltage, REG25 | 2.7 V ≤ $V_{(REGIN)}$ ≤ 5.5 V, $I_{OUT}$ ≤ 16mA | $T_A$ = −40°C to 85°C | 2.4 | 2.5 | 2.6 | V |
| | | 2.45 V ≤ $V_{(REGIN)}$ < 2.7 V (low battery), $I_{OUT}$ ≤ 3mA | | 2.4 | | | V |
| $I_{OS}$ [2] | Short circuit current limit | $V_{(REG25)}$ = 0 V | $T_A$ = −40°C to 85°C | | | 250 | mA |

(1) LDO output current, $I_{OUT}$, is the total load current. LDO regulator should be used to power internal fuel gauge only.
(2) Specified by design. Not production tested.

## 7.7 Internal Temperature Sensor Characteristics

$T_A$ = −40°C to 85°C, $C_{(REG)}$ = 0.47 µF, 2.45 V < $V_{(REGIN)}$ = $V_{BAT}$ < 5.5 V; typical values at $T_A$ = 25°C and $V_{(REGIN)}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $G_{(TEMP)}$ | Temperature sensor voltage gain | | | −2.0 | | mV/°C |

## 7.8 Internal Clock Oscillators

$T_A$ = −40°C to 85°C, $C_{(REG)}$ = 0.47 µF, 2.45 V < ($V_{(REGIN)}$ = $V_{BAT}$) < 5.5 V; typical values at $T_A$ = 25°C and $V_{(REGIN)}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $f_{(OSC)}$ | Operating frequency | | | 8.389 | | MHz |
| $f_{(LOSC)}$ | Operating frequency | | | 32.768 | | kHz |

## 7.9 Integrating ADC (Coulomb Counter) Characteristics

$T_A$ = −40°C to 85°C, $C_{(REG)}$ = 0.47 µF, 2.45 V < $V_{(REGIN)}$ = $V_{BAT}$ < 5.5 V; typical values at $T_A$ = 25°C and $V_{(REGIN)}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_{IN(SR)}$ | Input voltage range, $V_{(SRN)}$ and $V_{(SRP)}$ | $V_{SR}$ = $V_{(SRN)}$ − $V_{(SRP)}$ | −0.125 | | 0.125 | V |
| $t_{CONV(SR)}$ | Conversion time | Single conversion | | 1 | | s |
| | Resolution | | 14 | | 15 | bits |
| $V_{OS(SR)}$ | Input offset | | | 10 | | µV |
| $I_{NL}$ | Integral nonlinearity error | | | ±0.007 | ±0.034 | FSR |
| $Z_{IN(SR)}$ | Effective input resistance[1] | | 2.5 | | | MΩ |
| $I_{lkg(SR)}$ | Input leakage current[1] | | | | 0.3 | µA |

(1) Specified by design. Not production tested.

## 7.10 ADC (Temperature and Cell Voltage) Characteristics

$T_A$ = −40°C to 85°C, $C_{(REG)}$ = 0.47 µF, 2.45 V < $V_{(REGIN)}$ = $V_{BAT}$ < 5.5 V; typical values at $T_A$ = 25°C and $V_{(REGIN)}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_{IN(ADC)}$ | Input voltage range | | −0.2 | | 1 | V |
| $t_{CONV(ADC)}$ | Conversion time | | | | 125 | ms |
| | Resolution | | 14 | | 15 | bits |
| $V_{OS(ADC)}$ | Input offset | | | 1 | | mV |
| $Z_{(ADC1)}$ | Effective input resistance (TS) [1] | | | 5 | | kΩ |
| $Z_{(ADC2)}$ | Effective input resistance (BAT)[1] | bq27541-G1 not measuring cell voltage | 8 | | | MΩ |
| | | bq27541-G1 measuring cell voltage | | 100 | | kΩ |
| $I_{lkg(ADC)}$ | Input leakage current[1] | | | | 0.3 | µA |

(1) Specified by design. Not production tested.

## 7.11 Data Flash Memory Characteristics

$T_A$ = –40°C to 85°C, $C_{(REG)}$ = 0.47 µF, 2.45 V < $V_{(REGIN)}$ = $V_{BAT}$ < 5.5 V; typical values at $T_A$ = 25°C and $V_{(REGIN)}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{DR}$ | Data retention[1] | | 10 | | | Years |
| | Flash programming write-cycles [1] | | 20,000 | | | Cycles |
| $t_{WORDPROG}$ | Word programming time[1] | | | | 2 | ms |
| $I_{CCPROG}$ | Flash-write supply current[1] | | | 5 | 10 | mA |

(1) Specified by design. Not production tested.

## 7.12 HDQ Communication Timing Characteristics

$T_A$ = –40°C to 85°C, $C_{REG}$ = 0.47 µF, 2.45 V < $V_{REGIN}$ = $V_{BAT}$ < 5.5 V; typical values at $T_A$ = 25°C and $V_{REGIN}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{(CYCH)}$ | Cycle time, host to bq27541-G1 | | 190 | | | µs |
| $t_{(CYCD)}$ | Cycle time, bq27541-G1 to host | | 190 | 205 | 250 | µs |
| $t_{(HW1)}$ | Host sends 1 to bq27541-G1 | | 0.5 | | 50 | µs |
| $t_{(DW1)}$ | bq27541-G1 sends 1 to host | | 32 | | 50 | µs |
| $t_{(HW0)}$ | Host sends 0 to bq27541-G1 | | 86 | | 145 | µs |
| $t_{(DW0)}$ | bq27541-G1 sends 0 to host | | 80 | | 145 | µs |
| $t_{(RSPS)}$ | Response time, bq27541-G1 to host | | 190 | | 950 | µs |
| $t_{(B)}$ | Break time | | 190 | | | µs |
| $t_{(BR)}$ | Break recovery time | | 40 | | | µs |
| $t_{(RISE)}$ | HDQ line rising time to logic 1 (1.2V) | | | | 950 | ns |
| $t_{(TRND)}$ | Turnaround time (time from the falling edge of the last transmitted bit of 8-bit data and the falling edge of the next Break signal) | | 210 | | | µs |



(a) Break and Break Recovery

(b) HDQ line rise time

(c) Host Transmitted Bit

(d) Gauge Transmitted Bit

(e) Gauge to Host Response
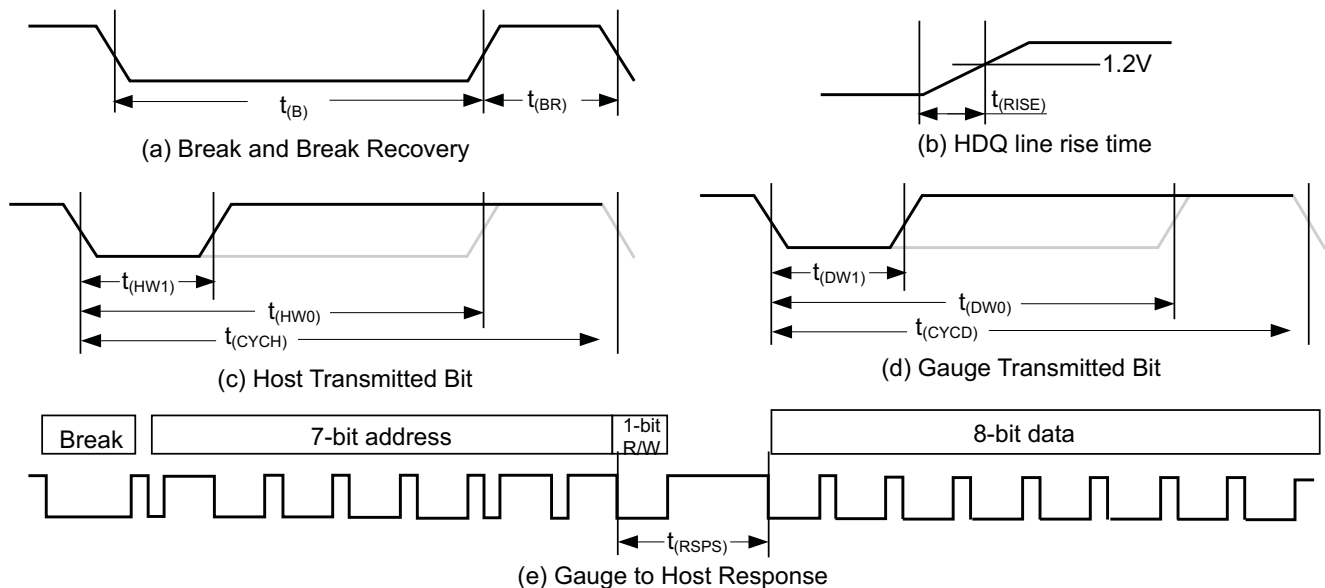
**Figure 1. Timing Diagrams**

## 7.13 I²C-Compatible Interface Timing Characteristics

$T_A$ = −40°C to 85°C, $C_{REG}$ = 0.47μF, 2.45 V < $V_{REGIN}$ = $V_{BAT}$ < 5.5 V; typical values at $T_A$ = 25°C and $V_{REGIN}$ = $V_{BAT}$ = 3.6 V (unless otherwise noted)

| | PARAMETER | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_r$ | SCL/SDA rise time | | | | 300 | ns |
| $t_f$ | SCL/SDA fall time | | | | 300 | ns |
| $t_{w(H)}$ | SCL pulse width (high) | | 600 | | | ns |
| $t_{w(L)}$ | SCL pulse width (low) | | 1.3 | | | μs |
| $t_{su(STA)}$ | Setup for repeated start | | 600 | | | ns |
| $t_{d(STA)}$ | Start to first falling edge of SCL | | 600 | | | ns |
| $t_{su(DAT)}$ | Data setup time | | 1000 | | | ns |
| $t_{h(DAT)}$ | Data hold time | | 0 | | | ns |
| $t_{su(STOP)}$ | Setup time for stop | | 600 | | | ns |
| $t_{BUF}$ | Bus free time between stop and start | | 66 | | | μs |
| $f_{SCL}$ | Clock frequency | | | | 400 | kHz |
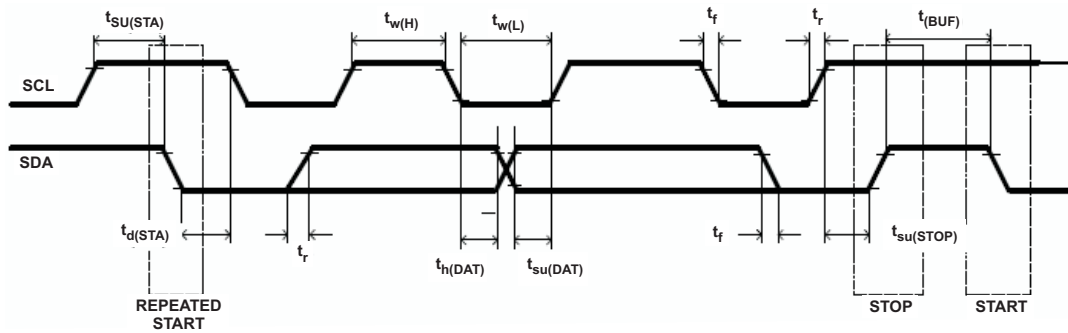


**Figure 2. I²C-Compatible Interface Timing Diagrams**

# 8 Detailed Description

## 8.1 Overview

The bq27541-G1 fuel gauge accurately predicts the battery capacity and other operational characteristics of a single Li-based rechargeable cell. It can be interrogated by a system processor to provide cell information, such as state-of-charge (SOC) and time-to-empty (TTE).

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

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

The fuel gauge provides 64 bytes of user-programmable data flash memory, partitioned into two (2) 32-byte blocks: **Manufacturer Info Block A** and **Manufacturer Info Block B**. This data space is accessed through a data flash interface. For specifics on accessing the data flash, see Manufacturer Information Block. The key to the high-accuracy gas gauging prediction is Texas Instruments proprietary Impedance Track™ algorithm. This algorithm uses cell measurements, characteristics, and properties to create state-of-charge predictions that can achieve less than 1% error across a wide variety of operating conditions and over the lifetime of the battery.

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

The fuel gauge external temperature sensing is optimized with the use of a high-accuracy negative temperature coefficient (NTC) thermistor with R25 = 10 kΩ ± 1% and B25/85 = 3435 kΩ ± 1% (such as Semitec 103AT) for measurement. The fuel gauge can also be configured to use its internal temperature sensor. The fuel gauge uses temperature to monitor the battery-pack environment, which is used for fuel gauging and cell protection functionality.

To minimize power consumption, the fuel gauge has different power modes: NORMAL, SLEEP, FULLSLEEP, and HIBERNATE. The fuel gauge passes automatically between these modes, depending upon the occurrence of specific events, though a system processor can initiate some of these modes directly. More details can be found in Device Functional Modes.

---

**NOTE**

**The following formatting conventions are used in this document:**

**Commands:**  *italics* with parentheses( ) and no breaking spaces, for example: *RemainingCapacity( )*
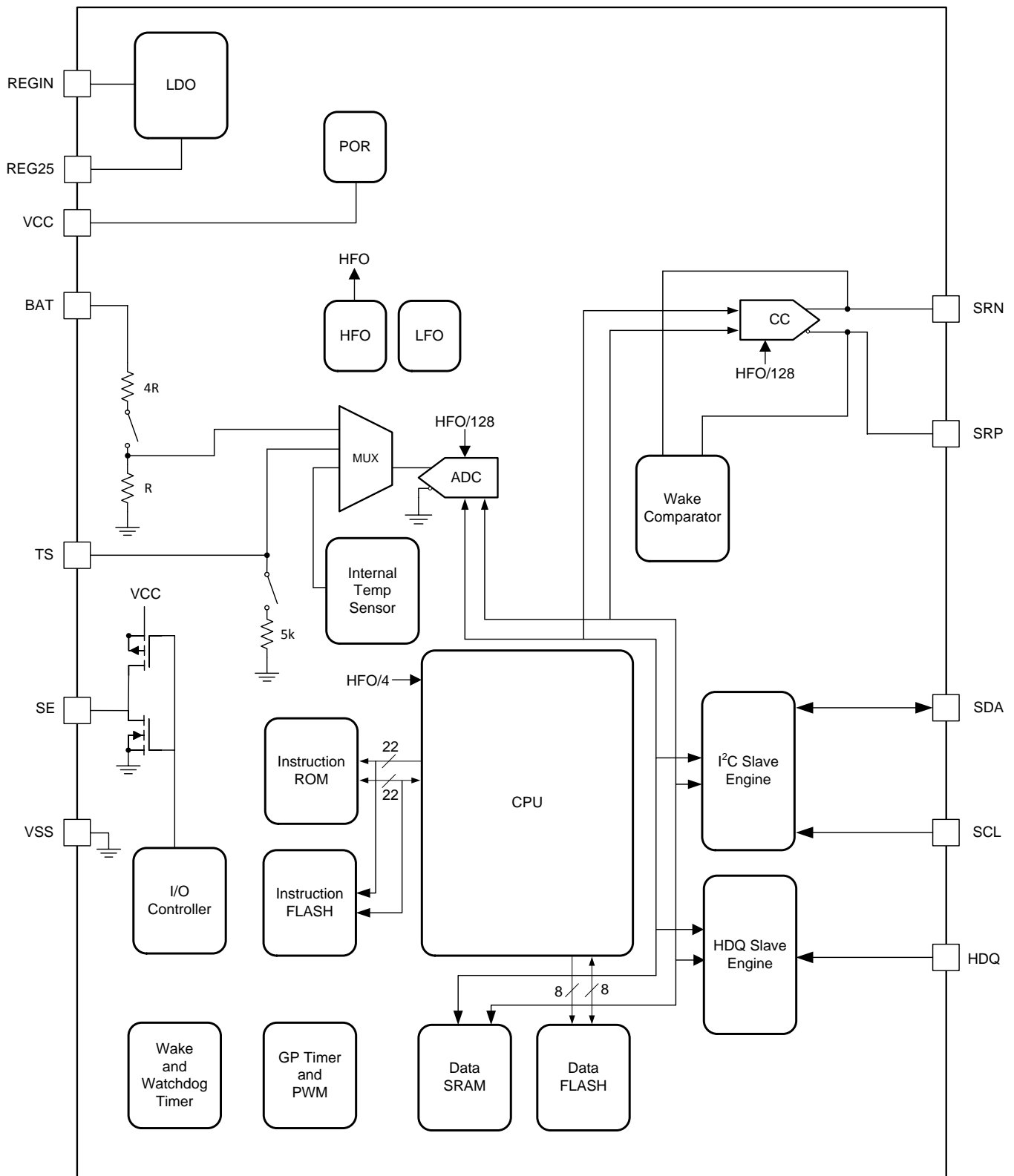
**Data Flash:**  *italics*, **bold**, and breaking spaces, for example: ***Design Capacity***

**Register Bits and Flags:**  *italics* with brackets[ ], for example: *[TDA]*

**Data Flash Bits:**  *italics*, **bold**, and brackets[ ], for example: ***[LED1]***

**Modes and States:**  All capitals, for example: UNSEALED mode

---

## 8.2 Functional Block Diagram

## 8.3 Feature Description

### 8.3.1 Fuel Gauging

The fuel gauge measures the cell voltage, temperature, and current to determine battery SOC based on Impedance Track™ algorithm (Please refer to Application Report SLUA450, *Theory and Implementation of Impedance Track Battery Fuel-Gauging Algorithm*, for more information). The fuel gauge monitors charge and discharge activity by sensing the voltage across a small-value resistor (5 mΩ to 20 mΩ, typical) between the SRP and SRN pins and in series with the cell. By integrating charge passing through the battery, the battery SOC is adjusted during battery charge or discharge.

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

The fuel gauge has two flags accessed by the *Flags( )* function that warns when the battery 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 fuel gauge has two additional flags accessed by the *Flags()* function that warns of internal battery conditions. The fuel gauge monitors the cell voltage during relaxed conditions to determine if an internal short has been detected. When this condition occurs, *[ISD]* will be set. The fuel gauge also has the capability of detecting when a tab has been disconnected in a 2-cell parallel system by actively monitoring the *SOH*. When this conditions occurs, *[TDD]* will be set.

### 8.3.2 Impedance Track™ Variables

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

#### 8.3.2.1 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 Load Select). When **Load Mode** is 0, the *Constant Current Model* is used (default). When Load Mode is 1, the *Constant Power Model* is used. The *CONTROL_STATUS [LDMD]* bit the status of **Load Mode**.

#### 8.3.2.2 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 1 are available.

## Feature Description (continued)

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

| LOAD SELECT VALUE | CURRENT MODEL USED |
|---|---|
| 0 | Average discharge current from previous cycle: There is an internal register that records the average discharge current through each entire discharge cycle. The previous average is stored in this register. |
| 1 (default) | Present average discharge current: This is the average discharge current from the beginning of this discharge cycle until present time. |
| 2 | Average current: based off the *AverageCurrent( )* |
| 3 | Current: based off of a low-pass-filtered version of *AverageCurrent( )* (τ = 14s) |
| 4 | Design capacity / 5: C Rate based off of Design Capacity /5 or a C / 5 rate in mA. |
| 5 | Use the value specified by *AtRate( )* |
| 6 | Use the value in **User_Rate-mA:** This gives a completely user-configurable method. |

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

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

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

### 8.3.2.3  Reserve Cap-mAh

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

### 8.3.2.4  Reserve Energy

**Reserve Energy** determines how much actual remaining capacity exists after reaching 0 *RemainingCapacity( )* which is equivalent to 0 remaining power, before **Terminate Voltage** is reached when **Load Mode** = 1 is selected. A loaded rate or no-load rate of compensation can be selected for *Reserve Cap* by setting **[RESCAP]** bit in the **Pack Configuration** data flash register.

### 8.3.2.5  Design Energy Scale

**Design Energy Scale** is used to select the scale and unit of a set of data flash parameters. The value of **Design Energy Scale** can be either 1 or 10 only, other values are not supported. For battery capacities larger than 6 Ah, **Design Energy Scale** = 10 is recommended.

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

| DATA FLASH | DESIGN ENERGY SCALE = 1 (Default) | DESIGN ENERGY SCALE = 10 |
|---|---|---|
| *Design Energy* | mWh | cWh |
| *Reserve Energy* | mWh | cWh |
| *Avg Power Last Run* | mW | cW |
| *User Rate-Pwr* | mWh | cWh |
| *T Rise* | No Scale (20 typical) | Scaled by ×10 (2 typical) |

### 8.3.2.6  Dsg Current Threshold

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

### 8.3.2.7  Chg Current Threshold

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

### 8.3.2.8  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 fuel gauge enters relaxation mode from a current flowing mode in either the charge direction or the discharge direction. The value of Quit Current is set to a default value that should be above the standby current of the system.

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

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

After about 6 minutes in relaxation mode, the fuel gauge attempts to take accurate OCV readings. An additional requirement of dV/dt < 1 µV/s is required for the fuel gauge to perform Qmax updates. These updates are used in the Impedance Track™ algorithms. It is critical that the battery voltage be relaxed during OCV readings and that the current is not higher than C/20 when attempting to go into relaxation mode.

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

### 8.3.2.9  Qmax

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

### 8.3.2.10  Update Status

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

**Table 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. |
| 0x04 | Impedance Track™ is enabled but Qmax and Ra data are not 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 fuel gauge during a learning cycle or when *IT_ENABLE* subcommand is received. Refer to the application note *How to Generate Golden Image for Single-Cell Impedance Track™Device* (SLUA544) for learning cycle details.

### 8.3.2.11  Avg I Last Run

The fuel gauge logs the current averaged from the beginning to the end of each discharge cycle. It stores this average current from the previous discharge cycle in this register. This register should never need to be modified. It is only updated by the fuel gauge when required.

### 8.3.2.12 Avg P Last Run

The fuel gauge logs the power averaged from the beginning to the end of each discharge cycle. It stores this average power from the previous discharge cycle in this register. To get a correct average power reading the fuel gauge continuously multiplies instantaneous current times *Voltage( )* to get power. It then logs this data to derive the average power. This register should never need to be modified. It is only updated by the fuel gauge when required.

### 8.3.2.13 Delta Voltage

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

### 8.3.2.14 Ra Tables and Ra Filtering Related Parameters

These tables contain encoded data and are automatically updated during device operation. The fuel gauge has a filtering process to eliminate unexpected fluctuations in Ra values while the Ra values are being updated. The DF parameters **RaFilter**, **RaMaxDelta**, **MaxResfactor**, and **MinResfactor** control the Filtering process of Ra values. **RaMaxDelta** limits the change in Ra values to an absolute magnitude. **MinResFactor** and **MaxResFactor** parameters are cumulative filters which limit the change in Ra values to a scale on a per discharge cycle basis. These values are Data Flash configurable. No further user changes should be made to Ra values except for reading or writing the values from a prelearned pack (part of the process for creating golden image files).

### 8.3.2.15 MaxScaleBackGrid

MaxScaleBackGrid parameter limits the resistance grid point after which back scaling will not be performed. This variable ensures that the resistance values in the lower resistance grid points remain accurate while the battery is at a higher DoD state.

### 8.3.2.16 Max DeltaV, Min DeltaV

Maximum or minimum value allowed for **Delta Voltage**, which will be subtracted from simulated voltage during remaining capacity simulation.

### 8.3.2.17 Qmax Max Delta %

Maximum change of Qmax during one update, as percentage of **Design Capacity**. If the gauges attempts to change Qmax exceeds this limit, changed value will be capped to old value ± **DesignCapacity** × **QmaxMaxDelta** / 100.

### 8.3.2.18 Fast Resistance Scaling

When Fast Resistance Scaling is enabled by setting the *[FConvEn]* bit in *Pack Configuration B*, the algorithm improves accuracy at the end of discharge. The *RemainingCapacity()* and *StateOfCharge()* should smoothly converge to 0. The algorithm starts convergence improvements when cell voltage goes below (**Terminate Voltage** + **Term V Delta**) or *StateofCharge()* goes below **Fast Scale Start SOC**. For most applications, the default value of **Term V Delta** and **Fast Scale Start SOC** are recommended. Also it is recommended to keep (**Terminate Voltage** + **Term V Delta**) below 3.6V for most battery applications.

### 8.3.2.19 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 | UnfilteredRM() | UnfilteredFCC() | UnfilteredRM() / UnfilteredFCC() |
| 1 | FilteredRM() | FilteredFCC() | FilteredRM() / FilteredFCC() |

### 8.3.2.20 DeltaV Max Delta

Maximum change of Delta V value. If attempted change of the value exceeds this limit, change value will be capped to old value ±**DeltaV Max Delta**.

### 8.3.2.21 Lifetime Data Logging Parameters

The Lifetime Data logging function helps development and diagnosis with the fuel gauge. Note that IT_ENABLE needs to 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 Update 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, a 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 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 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 Manufacturer Information Block 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 which 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.

### 8.3.3 Detailed Configuration Register Descriptions

#### 8.3.3.1 Pack Configuration Register

Some bq27541-G1 pins are configured via the **Pack Configuration** data flash register, as indicated in Table 5. This register is programmed and read via the methods described in *Accessing the Data Flash*. The register is located at subclass = 64, offset = 0.

**Table 5. Pack Configuration Bit Definition**

|           | bit7   | bit6      | bit5   | bit4   | bit3  | bit2   | bit1   | bit0   |
|-----------|--------|-----------|--------|--------|-------|--------|--------|--------|
| High Byte | RESCAP | CALEN     | INTPOL | INTSEL | RSVD  | IWAKE  | RSNS1  | RSNS0  |
| Default = | 0      | 0         | 0      | 1      | 0     | 0      | 0      | 1      |
|           | 0x11   |           |        |        |       |        |        |        |
| Low Byte  | GNDSEL | RFACTSTEP | SLEEP  | RMFCC  | SE_PU | SE_POL | SE_EN  | TEMPS  |
| Default = | 0      | 1         | 1      | 1      | 0     | 1      | 1      | 1      |
|           | 0x77   |           |        |        |       |        |        |        |

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

CALEN = Calibration mode is enabled.

INTPOL = Polarity for Interrupt pin. (See Interrupt Mode)

INTSEL = Interrupt pin select: 0 = SE pin, 1 = HDQ pin. (See Interrupt Mode)

RSVD = Reserved. Must be 0.

IWAKE/RSNS1/RSNS0 = These bits configure the current wake function (See Wake-Up Comparator).

GNDSEL = The ADC ground select control. The $V_{SS}$ (pin 6) is selected as ground reference when the bit is clear. Pin 7 is selected when the bit is set.

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

SLEEP = The fuel gauge can enter sleep, if operating conditions allow. True when set. (See Sleep Mode)

RMFCC = RM is updated with the value from FCC, on valid charge termination. True when set. (See Detection Charge Termination)

SE_PU = Pullup enable for SE pin. True when set (push-pull). (See Shutdown Mode)

SE_POL = Polarity bit for SE pin. SE is active high when set (makes SE high when gauge is ready for shutdown). (See Shutdown Mode)

SE_EN = Indicates if set the shutdown feature is enabled. True when set. (See Shutdown Mode for details.)

TEMPS = Selects external thermistor for *Temperature( )* measurements. True when set. (See Temperature Measurement and the TS Input)

### 8.3.3.2 Pack Configuration B Register

Some bq27541-G1 pins are configured via the **Pack Configuration B** data flash register, as indicated in Table 6. This register is programmed and read via the methods described in *Accessing the Data Flash*. The register is located at subclass = 64, offset = 2.

#### Table 6. Pack Configuration B Bit Definition

|            | bit7      | bit6    | bit5    | bit4    | bit3  | bit2     | bit1   | bit0    |
|------------|-----------|---------|---------|---------|-------|----------|--------|---------|
|            | ChgDoDEoC | SE_TDD  | VconsEN | SE_ISD  | RSVD  | LFPRelax | DoDWT  | FConvEn |
| **Default =** | 1      | 0       | 1       | 0       | 0     | 1        | 1      | 1       |
|            | 0x67      |         |         |         |       |          |        |         |

ChgDoDEoC = Enable DoD at EoC recalculation during charging only. True when set. Default setting is recommended.

SE_TDD = Enable Tab Disconnection Detection. True when set. (See Tab Disconnection Detection)

VconsEN = Enable voltage consistency check. True when set. Default setting is recommended.

SE_ISD = Enable Internal Short Detection. True when set. (See Internal Short Detection)

RSVD = Reserved. Must be 0

LFPRelax = Enable LiFePO$_4$ long relaxation mode. True when set.

DoDWT = Enable DoD weighting feature of gauging algorithm. This feature can improve accuracy during relaxation in a flat portion of the voltage profile, especially when using LiFePO$_4$ chemistry. True when set.

FConvEn = Enable fast convergence algorithm. Default setting is recommended. (See Fast Resistance Scaling)

### 8.3.3.3 Pack Configuration C Register

Some bq27541-G1 algorithm settings are configured via the **Pack Configuration C** data flash register, as indicated in Table 7. This register is programmed and read via the methods described in *Accessing the Data Flash*. The register is located at subclass = 64, offset = 3.

#### Table 7. Pack Configuration C Bit Definition

|            | bit7  | bit6  | bit5         | bit4     | bit3      | bit2  | bit1  | bit0  |
|------------|-------|-------|--------------|----------|-----------|-------|-------|-------|
|            | RSVD  | RSVD  | RelaxRCJumpOK | SmoothEn | SleepWkChg | RSVD  | RSVD  | RSVD  |
| **Default =** | 0  | 0     | 0            | 1        | 1         | 0     | 0     | 0     |
|            | 0x18  |       |              |          |           |       |       |       |

RSVD = Reserved. Must be 0.

RelaxRCJumpOK = Allow SOC to change due to temperature change during relaxation when SOC smoothing algorithm is enabled. True when set.

SmoothEn = Enable SOC smoothing algorithm. True when set. (See StateOfCharge Smoothing)

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

### 8.3.4  System Control Function

The fuel gauge provides system control functions which allows the fuel gauge to enter shutdown mode in order to power-off with the assistance of external circuit or provides interrupt function to the system. Table 8 shows the configurations for SE and HDQ pins.

**Table 8. SE and HDQ Pin Functions**

| [INTSEL] | COMMUNICATION MODE | SE PIN FUNCTION | HDQ PIN FUNCTION |
|---|---|---|---|
| 0 (default) | I²C | Interrupt Mode [1] | Not Used |
| | HDQ | | HDQ Mode[2] |
| 1 | I²C | Shutdown Mode | Interrupt Mode |
| | HDQ | | HDQ Mode[2] |

(1)  [SE_EN] bit in **Pack Configuration** can be enabled to use *[SE]* and *[SHUTDWN]* bits in *CONTROL_STATUS()* function; The SE pin shutdown function is disabled.
(2)  HDQ pin is used for communication and HDQ Host Interrupt Feature is available.

### 8.3.4.1  Shutdown Mode

In the shutdown mode, the SE pin is used to signal external circuit to power-off the fuel gauge. This feature is useful to shutdown the fuel gauge in a deeply discharged battery to protect the battery. By default, the Shutdown Mode is in normal state. By sending the SET_SHUTDOWN subcommand or setting the *[SE_EN]* bit in **Pack Configuration** register, the *[SHUTDWN]* bit is set and enables the shutdown feature. When this feature is enabled and *[INTSEL]* is set, the SE pin can be in normal state or shutdown state. The shutdown state can be entered in HIBERNATE mode (only if HIBERNATE mode is enabled due to low cell voltage), all other power modes will default SE pin to normal state. Table 9 shows the SE pin state in normal or shutdown mode. The CLEAR_SHUTDOWN subcommand or clearing *[SE_EN]* bit in the **Pack Configuration** register can be used to disable shutdown mode.

The SE pin will be high impedance at power-on reset (POR), the *[SE_POL]* does not affect the state of SE pin at POR. Also, *[SE_PU]* configuration changes will only take effect after POR. In addition, the *[INTSEL]* only controls the behavior of the SE pin; it does not affect the function of *[SE]* and *[SHUTDWN]* bits.

**Table 9. SE Pin State**

| | | SHUTDOWN Mode [INTSEL] = 1 and ([SE_EN] or [SHUTDOWN] = 1) | |
|---|---|---|---|
| *[SE_PU]* | *[SE_POL]* | NORMAL State | SHUTDOWN State |
| 0 | 0 | High Impedance | 0 |
| 0 | 1 | 0 | High Impedance |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

### 8.3.4.2  Interrupt Mode

By utilizing the interrupt mode, the system can be interrupted based on detected fault conditions as specified in Table 12. The SE or HDQ pin can be selected as the interrupt pin by configuring the *[INTSEL]* bit based on . In addition, the pin polarity and pullup (SE pin only) can be configured according to the system needs as described in Table 10 or Table 11.

**Table 10. SE Pin in Interrupt Mode ([INTSEL] = 0)**

| *[SE_PU]* | *[INTPOL]* | INTERRUPT CLEAR | INTERRUPT SET |
|-----------|------------|-----------------|---------------|
| 0 | 0 | High Impedance | 0 |
| 0 | 1 | 0 | High Impedance |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Table 11. HDQ Pin in Interrupt Mode ([INTSEL] = 1)**

| *[INTPOL]* | INTERRUPT CLEAR | INTERRUPT SET |
|------------|-----------------|---------------|
| 0 | High Impedance | 0 |
| 1 | 0 | High Impedance |

**Table 12. Interrupt Mode Fault Conditions**

| INTERRUPT CONDITION | Flags() STATUS BIT | ENABLE CONDITION | COMMENT |
|---------------------|--------------------|------------------|---------|
| SOC1 Set/Clear | *[SOC1]* | Always | The SOC1 Set/Clear interrupt is based on the*[SOC1]* Flag condition when *RemainingCapacity()* reaches the SOC1 Set or Clear threshold in the Data Flash. |
| Over Temperature Charge | *[OTC]* | **OT Chg Time** ≠ 0 | The *[OTC]* Flag is set/clear based on conditions specified in Over-Temperature: Charge. |
| Over Temperature Discharge | *[OTD]* | **OT Dsg Time** ≠ 0 | The *[OTD]* Flag is set/clear based on conditions specified in Over-Temperature: Discharge. |
| Battery High | *[BATHI]* | Always | The *[BATHI]* Flag is set/clear based on conditions specified in Battery Level Indication. |
| Battery Low | *[BATLOW]* | Always | The *[BATLOW]* Flag is set/clear based on conditions specified in Battery Level Indication. |
| Internal Short Detection | *[ISD]* | **[SE_ISD]** = 1 in **Pack Configuration B** | The *[SE_ISD]* Flag is set/clear based on conditions specified in Internal Short Detection. |
| Tab disconnection detection | *[TDD]* | **[SE_TDD]** = 1 in **Pack Configuration B** | The *[TDD]* Flag is set/clear based on conditions specified in Tab Disconnection Detection. |

### 8.3.4.3  Battery Level Indication

The fuel gauge can indicate when battery voltage has fallen below or risen above predefined thresholds. The *[BATHI]* bit of *Flags()* is set high to indicate *Voltage()* is above the **BH Set Volt Threshold** for a predefined duration set in the **BH Volt Time**. This flag returns to low once battery voltage is below or equal the **BH Clear Volt threshold**. It is recommended that the **BH Set Volt Threshold** is configured higher than the **BH Clear Volt threshold** to provide proper voltage hysteresis.

The *[BATLOW]* bit of *Flags()* is set high to indicate *Voltage()* is below the **BL Set Volt Threshold** for predefined duration set in the **BL Volt Time**. This flag returns to low once battery voltage is above or equal the **BL Clear Volt threshold**. It is recommended that the **BL Set Volt Threshold** is configured lower than the **BL Clear Volt threshold** to provide proper voltage hysteresis.

The *[BATHI]* and *[BATLOW]* flags can be configured to control the interrupt pin (SE or HDQ) by enabling interrupt mode. See Interrupt Mode for details.

### 8.3.4.4 Internal Short Detection

The fuel gauge can indicate detection of an internal battery short if the **[SE_ISD**] bit in **Pack Configuration B** is set. The device compares the self-discharge current calculated in relaxation mode and *AverageCurrent()* measured in the system. The self-discharge rate is measured at 1-hour interval. When battery *SelfDischargeCurrent()* is less than the predefined (**–Design Capacity / ISD Current** threshold), the *[ISD]* of *Flags()* is set high. The *[ISD]* of *Flags()* can be configured to control interrupt pin (SE or HDQ) by enabling interrupt mode. See Interrupt Mode for details.

### 8.3.4.5 Tab Disconnection Detection

The fuel gauge can indicate tab disconnection by detecting change of *StateOfHealth()*. This feature is enabled by setting **[SE_TDD]** bit in **Pack Configuration B**. The *[TDD]* of *Flags()* is set when the ratio of current *StateOfHealth()* divided by the previous *StateOfHealth()* reported is less than **TDD SOH Percent**. The *[TDD]* of *Flags()* can be configured to control an interrupt pin (SE or HDQ) by enabling interrupt mode. See Interrupt Mode for details.

### 8.3.5 Temperature Measurement and the TS Input

The fuel gauge measures the battery temperature via the TS input to supply battery temperature status information to the fuel gauging algorithm and charger-control sections of the gauge. Alternatively, the gauge can also measure internal temperature via its on-chip temperature sensor, but only if the **[TEMPS]** bit of **Pack Configuration** register is cleared.

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

The thermistor circuit requires the use of an external 10-kΩ thermistor with negative temperature coefficient (NTC) thermistor with R25 = 10 kΩ ± 1% and B25/85 = 3435 kΩ ± 1% (such as Semitec 103AT) that connects between the $V_{CC}$ and TS pins. Additional circuit information for connecting the thermistor to the bq27541 is shown in Typical Applications.

### 8.3.6 Over-Temperature Indication

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

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

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

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

### 8.3.7 Charging and Charge Termination Indication

### 8.3.7.1 Detection Charge Termination

For proper fuel gauge operation, the cell charging voltage must be specified by the user. The default value for this variable is in the data flash **Charging Voltage**.

The fuel gauge detects charge termination when:
- During 2 consecutive periods of **Current Taper Window**, the *AverageCurrent( )* is **< Taper Current**
- During the same periods, the accumulated change in capacity > 0.25 mAh / **Current Taper Window**
- *Voltage( )* **> Charging Voltage – Taper Voltage**

When this occurs, the *[CHG]* bit of *Flags( )* is cleared. Also, if the **[RMFCC]** bit of **Pack Configuration** is set, *RemainingCapacity( )* is set equal to *FullChargeCapacity( )*. When **TCA_Set** is set to –1, it disables the use of the charger alarm threshold. In that case, TerminateCharge is set when the taper condition is detected. When **FC_Set** is set to –1, it disables the use of the full charge detection threshold. In that case, the *[FC]* bit is not set until the taper condition is met.

### 8.3.7.2  Charge Inhibit

The fuel gauge can indicate when battery temperature has fallen below or risen above predefined thresholds (**Charge Inhibit Temp Low** and **Charge Inhibit Temp High**, respectively). In this mode, the *[CHG_INH]* of *Flags( )* is made high to indicate this condition, and is returned to its low state, once battery temperature returns to the range **[Charge Inhibit Temp Low + Temp Hys**, **Charge Inhibit Temp High – Temp Hys]**.

## 8.4  Device Functional Modes

The fuel gauge has three power modes: NORMAL, SLEEP, and HIBERNATE. In NORMAL mode, the fuel gauge is fully powered and can execute any allowable task. In SLEEP mode the fuel gauge exists in a reduced-power state, periodically taking measurements and performing calculations. Finally, in HIBERNATE mode, the fuel gauge is in a very low power state, but can be awoken by communication or certain I/O activity.

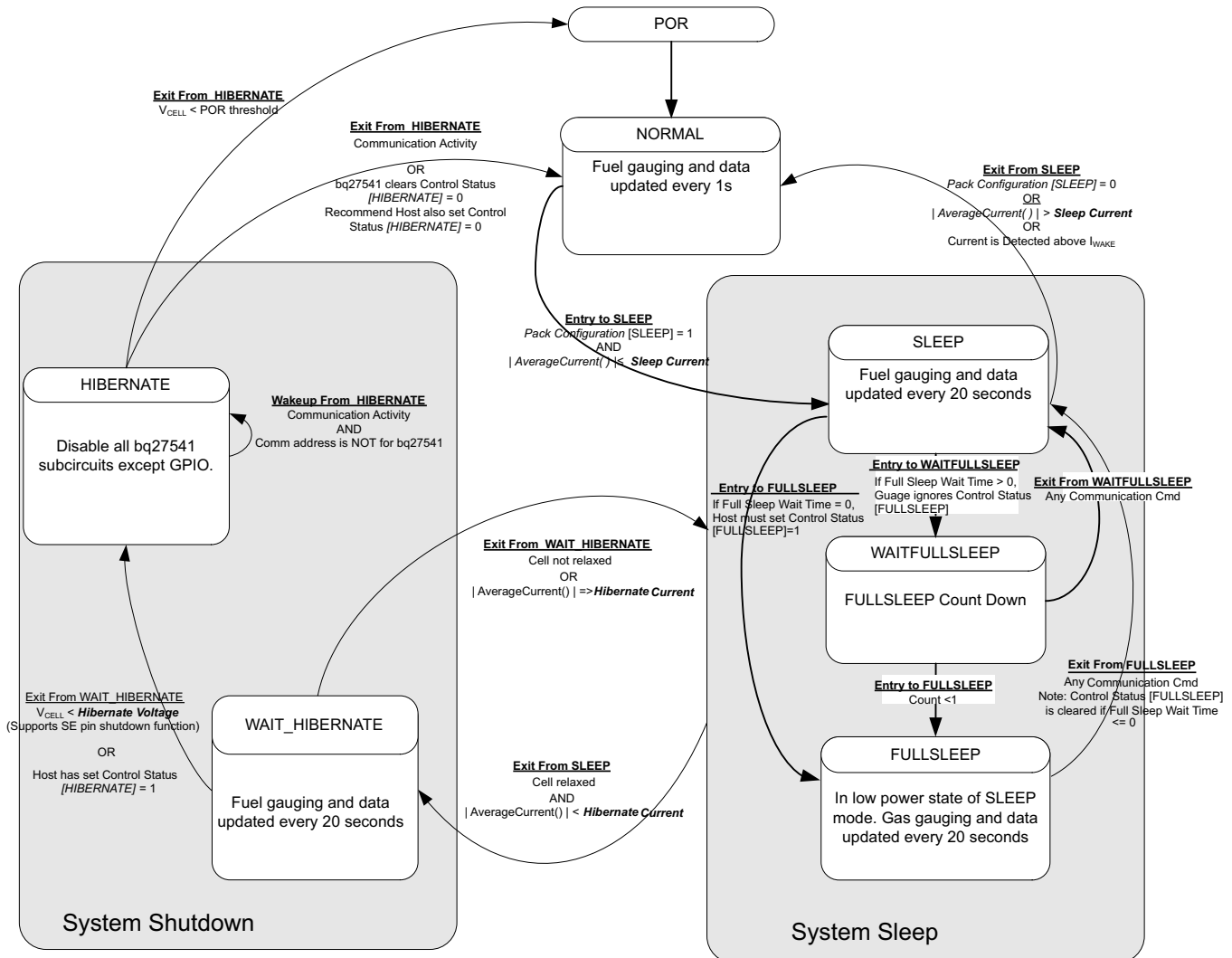The relationship between these modes is shown in Figure 3. Details are described in the sections that follow.



**Figure 3.  Power Mode Diagram**

## Device Functional Modes (continued)

### 8.4.1  NORMAL Mode

The fuel gauge is in NORMAL Mode when not in any other power mode. During this mode, *AverageCurrent( )*, *Voltage( )*, and *Temperature( )* measurements are taken, and the interface data set is updated. Decisions to change states are also made. This mode is exited by activating a different power mode.

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

### 8.4.2  SLEEP Mode

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

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

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

The fuel gauge exits SLEEP if any entry condition is broken, specifically when: (1) *AverageCurrent( )* rises above **Sleep Current**, or (2) a current in excess of $I_{WAKE}$ through $R_{SENSE}$ is detected when the $I_{WAKE}$ comparator is enabled.

### 8.4.3  FULLSLEEP Mode

FULLSLEEP mode is entered automatically when the bq27541-G1 is in SLEEP mode and the timer counts down to 0 (*Full Sleep Wait Time* > 0). FULLSLEEP mode is entered immediately after entry to SLEEP if **Full Sleep Wait Time** is set to 0 and the host sets the *[FULLSLEEP]* bit in the CONTROL_STATUS register using the SET_FULLSLEEP subcommand.

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

The gauge exits the FULLSLEEP mode when there is any communication activity. The [*FULLSLEEP*] bit can remain set (*Full Sleep Wait Time* > 0) or be cleared (*Full Sleep Wait Time* ≤ 0) after exit of FULLSLEEP mode. Therefore, EVSW communication activity might cause the gauge to exit FULLSLEEP MODE and display the [*FULLSLEEP*] bit as clear. The execution of SET_FULLSLEEP to set [*FULLSLEEP*] bit is required when *Full Sleep Wait Time* ≤ 0 in order to re-enter FULLSLEEP mode.

While in FULLSLEEP mode, the fuel gauge can suspend serial communications as much as 4 ms by holding the comm line(s) low. This delay is necessary to correctly process host communication, since the fuel gauge processor is mostly halted in SLEEP mode.

The fuel gauge exits FULLSLEEP if any entry condition is broken, specifically when: (1) *AverageCurrent( )* rises above **Sleep Current**, or (2) a current in excess of $I_{WAKE}$ through $R_{SENSE}$ is detected when the $I_{WAKE}$ comparator is enabled.

### 8.4.4  HIBERNATE Mode

HIBERNATE mode should be used for long-term pack storage or when the host system needs to enter a low-power state, and minimal gauge power consumption is required. This mode is ideal when the host is set to its own HIBERNATE, SHUTDOWN, or OFF modes. The gauge waits to enter HIBERNATE mode until it has taken a valid OCV measurement (cell relaxed) and the magnitude of the average cell current has fallen below Hibernate Current. When the conditions are met, the fuel gauge can enter HIBERNATE due to either low cell voltage or by having the *[HIBERNATE]* bit of the CONTROL_STATUS register set. The gauge will remain in HIBERNATE mode until any communication activity appears on the communication lines and the address is for bq27541. In addition, the SE pin shutdown mode function is supported only when the fuel gauge enters HIBERNATE due to low cell voltage.

When the gauge wakes up from HIBERNATE mode, the *[HIBERNATE]* bit of the CONTROL_STATUS register is cleared. The host is required to set the bit to allow the gauge to re-enter HIBERNATE mode if desired.

## Device Functional Modes (continued)

Because the fuel gauge is dormant in HIBERNATE mode, the battery should not be charged or discharged in this mode, because any changes in battery charge status will not be measured. If necessary, the host equipment can draw a small current (generally infrequent and less than 1 mA, for purposes of low-level monitoring and updating); however, the corresponding charge drawn from the battery will not be logged by the gauge. Once the gauge exits to NORMAL mode, the IT algorithm will take about 3 seconds to re-establish the correct battery capacity and measurements, regardless of the total charge drawn in HIBERNATE mode. During this period of re-establishment, the gauge reports values previously calculated prior to entering HIBERNATE mode. The host can identify exit from HIBERNATE mode by checking if *Voltage()* < Hibernate Voltage or *[HIBERNATE]* bit is cleared by the gauge.

If a charger is attached, the host should immediately take the fuel gauge out of HIBERNATE mode before beginning to charge the battery. Charging the battery in HIBERNATE mode will result in a notable gauging error that will take several hours to correct. It is also recommended to minimize discharge current during exit from HIBERNATE.

Note: The HIBERNATE mode is only available in I²C mode and is disabled when HDQ mode is used.

### 8.4.5 Power Control

#### 8.4.5.1 Reset Functions

When the fuel gauge detects a software reset by sending *[RESET] Control( )* subcommand, 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.

#### 8.4.5.2 Wake-Up Comparator

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

**Table 13. I$_{WAKE}$ Threshold Settings[1]**

| IWAKE | RSNS1 | RSNS0 | Vth(SRP-SRN) |
|:---:|:---:|:---:|:---:|
| 0 | 0 | 0 | Disabled |
| 1 | 0 | 0 | Disabled |
| 0 | 0 | 1 | 1.0 mV or −1.0 mV |
| 1 | 0 | 1 | 2.2 mV or −2.2 mV |
| 0 | 1 | 0 | 2.2 mV or −2.2 mV |
| 1 | 1 | 0 | 4.6 mV or −4.6 mV |
| 0 | 1 | 1 | 4.6 mV or −4.6 mV |
| 1 | 1 | 1 | 9.8 mV or −9.8 mV |

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

#### 8.4.5.3 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 V$_{CC}$ voltage does not fall below its minimum of 2.4 V during Flash write operations.

### 8.4.6 Autocalibration

The fuel gauge 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.

Autocalibration of the ADC begins on entry to SLEEP mode, except if *Temperature( )* is ≤ 5°C or *Temperature( )* ≥ 45°C.

The fuel gauge also performs a single offset calibration when: (1) the condition of *AverageCurrent( )* ≤ 100 mA and (2) {voltage change since last offset calibration ≥ 256 mV} or {temperature change since last offset calibration is greater than 8°C for ≥ 60 seconds}.

Capacity and current measurements will continue at the last measured rate during the offset calibration when these measurements cannot be performed. If the battery voltage drops more than 32 mV during the offset calibration, the load current has likely increased considerably; hence, the offset calibration will be aborted.

## 8.5 Communications

### 8.5.1 Authentication

The fuel gauge can act as a SHA-1/HMAC authentication slave by using its internal engine. Sending a 160-bit SHA-1 challenge message to the fuel gauge will cause the gauge to return a 160-bit digest, based upon the challenge message and a hidden, 128-bit plain-text authentication key. If this digest matches an identical one generated by a host or dedicated authentication master, and when operating on the same challenge message and using the same plain text keys, the authentication process is successful.

### 8.5.2 Key Programming (Data Flash Key)

By default, the fuel gauge contains a default plain-text authentication key of 0x0123456789ABCDEFFEDCBA9876543210. This default key is intended for development purposes. It should be changed to a secret key and the part immediately sealed, before putting a pack into operation. Once written, a new plain-text key cannot be read again from the fuel gauge while in SEALED mode.

Once the fuel gauge is UNSEALED, the authentication key can be changed from its default value by writing to the *Authenticate( )* Extended Data Command locations. A 0x00 is written to *BlockDataControl( )* to enable the authentication data commands. The *DataFlashClass()* is issued 112 (0x70) to set the Security class. Up to 32 bytes of data can be read directly from the *BlockData*() (0x40 through 0x5F) and the authentication key is located at 0x48 (0x40 + 0x08 offset) to 0x57 (0x40 + 0x17 offset). The new authentication key can be written to the corresponding locations (0x48 to 0x57) using the *BlockData*() command. The data is transferred to the data flash when the correct checksum for the whole block (0x40 to 0x5F) is written to *BlockDataChecksum*() (0x60). The checksum is (255 – x) where x is the 8-bit summation of the *BlockData*() (0x40 to 0x5F) on a byte-by-byte basis. Once the authentication key is written, the gauge can then be SEALED again.

### 8.5.3 Key Programming (Secure Memory Key)

As the name suggests, the secure-memory authentication key is stored in the secure memory of the fuel gauge. If a secure-memory key has been established, only this key can be used for authentication challenges (the programmable data flash key is not available). The selected key can only be established or programmed by special arrangements with TI, using the TI *Secure B-to-B Protocol*. The secure-memory key can never be changed or read from the fuel gauge.

### 8.5.4 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( )*, instead.

Next, the host writes a 20-byte authentication challenge to the *Authenticate( )* address locations (0x40 through 0x53). After a valid checksum for the challenge is written to *AuthenticateChecksum( )*, the bq27541-G1 uses the challenge to perform the SHA-1/HMAC computation, in conjunction with the programmed key. The bq27541-G1 completes the SHA-1/HMAC computation and write the resulting digest to *Authenticate()*, overwriting the pre-existing challenge. The host should wait at least 45 ms to read the resulting digest. The host may then read this response and compare it against the result created by its own parallel computation.

## Communications (continued)

### 8.5.5 HDQ Single-pin Serial Interface

The HDQ interface is an asynchronous return-to-one protocol where a processor sends the command code to the fuel gauge. 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 pullup resistor. The 8-bit command code consists of two fields: the 7-bit HDQ command code (bits 0:6) and the 1-bit RW field (MSB bit 7). The RW field directs the fuel gauge either to:

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

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

HDQ serial communication is normally initiated by the host processor sending a break command to the fuel gauge. 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 fuel gauge is now ready to receive information from the host processor.

The fuel gauge is shipped in the $I^2C$ mode. TI provides tools to enable the HDQ peripheral. The SLUA408 application report provides details of HDQ communication basics.

### 8.5.6 HDQ Host Interruption Feature

The default fuel gauge behaves as an HDQ slave only device when HDQ mode is enabled. If the HDQ interrupt function is enabled, the fuel gauge is capable of mastering and also communicating to a HDQ device. There is no mechanism for negotiating who is to function as the HDQ master and care must be taken to avoid message collisions. The interrupt is signaled to the host processor with the fuel gauge mastering an HDQ message. This message is a fixed message that will be used to signal the interrupt condition. The message itself is 0x80 (slave write to register 0x00) with no data byte being sent as the command is not intended to convey any status of the interrupt condition. The HDQ interrupt function is disabled by default and needs to be enabled by command.

When the SET_HDQINTEN subcommand is received, the fuel gauge will detect any of the interrupt conditions and assert the interrupt at 1-second intervals until the CLEAR_HDQINTEN command is received or the count of HDQHostIntrTries has lapsed.

The number of tries for interrupting the host is determined by the data flash parameter named **HDQHostIntrTries**.

#### 8.5.6.1 Low Battery Capacity

This feature will work identically to SOC1. It will use the same data flash entries as SOC1 and will trigger interrupts as long as SOC1 = 1 and *HDQIntEN* = 1.

#### 8.5.6.2 Temperature

This feature will trigger an interrupt based on the OTC (Over-Temperature in Charge) or OTD (Over-Temperature in Discharge) condition being met. It uses the same data flash entries as OTC or OTD and will trigger interrupts as long as either the OTD or OTC condition is met and *HDQIntEN* = 1.

## Communications (continued)

### 8.5.7 I2C Interface

The fuel gauge supports the standard I2C 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.
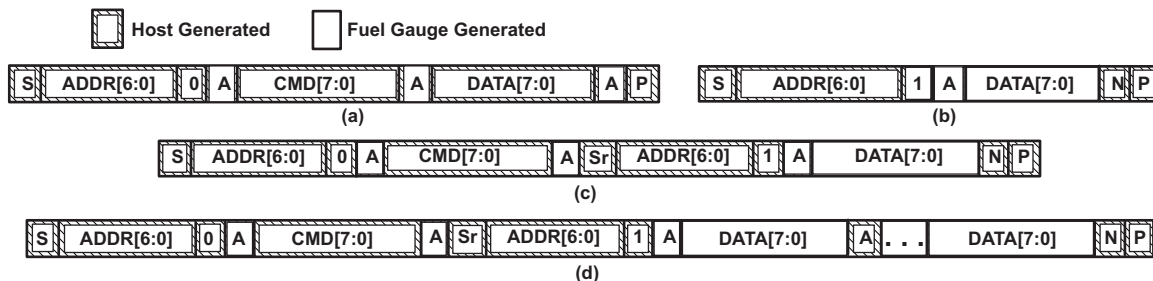


**Figure 4. Supported I2C Formats**

(a) 1-byte write

(b) Quick read

(c) 1 byte-read

(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 I2C communication engine, increments whenever data is acknowledged by the fuel gauge or the I2C master. The quick writes function in the same manner and are a convenient means of sending multiple bytes to consecutive command locations (such as two-byte commands that require two bytes of data).

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



Attempt to read an address above 0x7F (NACK command):



Attempt at incremental writes (NACK all extra data bytes sent):



Incremental read at the maximum allowed read address:
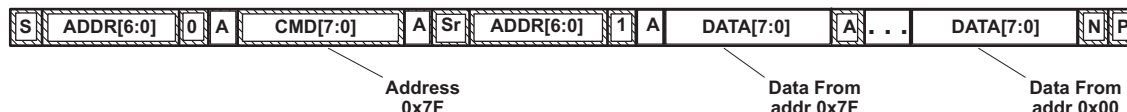


The I2C engine releases both SDA and SCL if the I2C bus is held low for t(BUSERR). If the fuel gauge was holding the lines, releasing them frees the master to drive the lines. If an external condition is holding either of the lines low, the I2C engine enters the low-power sleep mode.

#### 8.5.7.1 I2C Time Out

The I2C engine will release both SDA and SCL if the I2C bus is held low for about 2 seconds. If the fuel gauge was holding the lines, releasing them will free for the master to drive the lines.

## Communications (continued)

### 8.5.7.2 $I^2C$ Command Waiting Time

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

| S | ADDR[6:0] | 0 | A | CMD[7:0] | A | DATA [7:0] | A | DATA [7:0] | A | P | 66μs |

| 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 | 66μs |

Waiting time between control subcommand and reading results

| S | ADDR[6:0] | 0 | A | CMD[7:0] | A | Sr | ADDR[6:0] | 1 | A | DATA [7:0] | A | DATA [7:0] | A |

| DATA [7:0] | A | DATA [7:0] | N | P | 66μs |
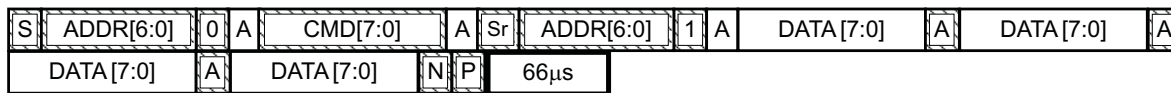
Waiting time between continuous reading results

### 8.5.7.3 $I^2C$ Clock Stretching

$I^2C$ clock stretches can occur during all modes of fuel gauge operation. In the SLEEP and HIBERNATE modes, a short clock stretch will occur on all $I^2C$ traffic as the device must wake-up to process the packet. In NORMAL and SLEEP+ modes, clock stretching will only occur for packets addressed for the fuel gauge. The timing of stretches will vary as interactions between the communicating host and the gauge are asynchronous. The $I^2C$ clock stretches may occur after start bits, the ACK/NAK bit and first data bit transmit on a host read cycle. The majority of clock stretch periods are small (≤ 4 ms) as the $I^2C$ interface peripheral and CPU firmware perform normal data flow control. However, less frequent but more significant clock stretch periods may occur when data flash (DF) is being written by the CPU to update the resistance (Ra) tables and other DF parameters such as Qmax. Due to the organization of DF, updates need to be written in data blocks consisting of multiple data bytes.

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

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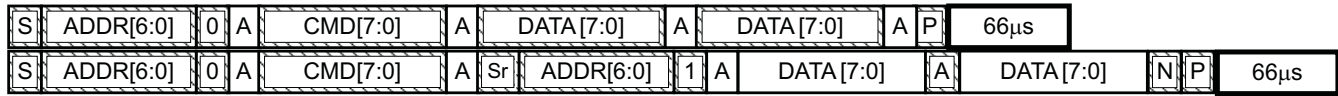o erase the copy buffer page, 6 ms to write the data into the copy buffer and the program progress indicator (2 ms for each individual write). The second part of the update is writing to the DF and requires 44 ms for DF block update. This includes a 20-ms each page erase for two pages and 2-ms each row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

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

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

## 8.6 Programming

### 8.6.1 Standard Data Commands

The fuel gauge uses a series of 2-byte standard commands to enable system reading and writing of battery information. Each standard command has an associated command-code pair, as indicated in Table 14. Each protocol has specific means to access the data at each Command Code. DataRAM is updated and read by the gauge only once per second. Standard commands are accessible in NORMAL operation mode.

**Table 14. Standard Commands**

| NAME | | COMMAND CODE | UNIT | SEALED ACCESS |
|---|---|---|---|---|
| Control( ) | CNTL | 0x00 and 0x01 | NA | RW |
| AtRate( ) | AR | 0x02 and 0x03 | mA | RW |
| UnfilteredSOC() | UFSOC | 0x04 and 0x05 | % | R |
| Temperature( ) | TEMP | 0x06 and 0x07 | 0.1°K | R |
| Voltage( ) | VOLT | 0x08 and 0x09 | mV | R |
| Flags( ) | FLAGS | 0x0A and 0x0B | NA | R |
| NomAvailableCapacity( ) | NAC | 0x0C and 0x0D | mAh | R |
| FullAvailableCapacity( ) | FAC | 0x0E and 0x0F | mAh | R |
| RemainingCapacity( ) | RM | 0x10 and 0x11 | mAh | R |
| FullChargeCapacity( ) | FCC | 0x12 and 0x13 | mAh | R |
| AverageCurrent( ) | AI | 0x14 and 0x15 | mA | R |
| TimeToEmpty( ) | TTE | 0x16 and 0x17 | Minutes | R |
| FilteredFCC() | FFCC | 0x18 and 0x19 | mAh | R |
| StandbyCurrent( ) | SI | 0x1A and 0x1B | mA | R |
| UnfilteredFCC() | UFFCC | 0x1C and 0x1D | mAh | R |
| MaxLoadCurrent( ) | MLI | 0x1E and 0x1F | mA | R |
| UnfilteredRM() | UFRM | 0x20 and 0x21 | mAh | R |
| FilteredRM() | FRM | 0x22 and 0x23 | mAh | R |
| AveragePower( ) | AP | 0x24 and 0x25 | mW or cW | R |
| InternalTemperature( ) | INTTEMP | 0x28 and 0x29 | 0.1°K | R |
| CycleCount( ) | CC | 0x2A and 0x2B | Counts | R |
| StateOfCharge( ) | SOC | 0x2C and 0x2D | % | R |
| StateOfHealth( ) | SOH | 0x2E and 0x2F | % / num | R |
| PassedCharge( ) | PCHG | 0x34 and 0x35 | mAh | R |
| DOD0( ) | DOD0 | 0x36 and 0x37 | Hex | R |
| SelfDischargeCurrent() | SDSG | 0x38 and 0x39 | mA | R |

### 8.6.1.1  Control( ): 0x00 and 0x01

Issuing a *Control( )* command requires a subsequent 2-byte subcommand. These additional bytes specify the particular control function desired. The *Control( )* command allows the system to control specific features of the fuel gauge during normal operation and additional features when the fuel gauge is in different access modes, as described in Table 15.

**Table 15.  *Control( )* Subcommands**

| CNTL FUNCTION | CNTL DATA | SEALED ACCESS | DESCRIPTION |
|---|---|---|---|
| CONTROL_STATUS | 0x0000 | Yes | Reports the status of DF Checksum, Hibernate, IT, etc. |
| DEVICE_TYPE | 0x0001 | Yes | Reports the device type of 0x0541 (indicating bq27541-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 |
| Reserved | 0x0004 | No | Not to be used |
| RESET_DATA | 0x0005 | Yes | Returns reset data |
| Reserved | 0x0006 | No | Not to be used |
| PREV_MACWRITE | 0x0007 | Yes | Returns previous *Control()* subcommand code |
| CHEM_ID | 0x0008 | Yes | Reports the chemical identifier of the Impedance Track™ configuration |
| BOARD_OFFSET | 0x0009 | No | Forces the device to measure and store the board offset |
| CC_OFFSET | 0x000A | No | Forces the device to measure internal CC offset |
| CC_OFFSET_SAVE | 0x000B | No | 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 Control Status register to 1 |
| SET_HIBERNATE | 0x0011 | Yes | Forces CONTROL_STATUS *[HIBERNATE]* to 1 |
| CLEAR_HIBERNATE | 0x0012 | Yes | Forces CONTROL_STATUS *[HIBERNATE]* to 0 |
| SET_SHUTDOWN | 0x0013 | Yes | Enables the SE pin to change state |
| CLEAR_SHUTDOWN | 0x0014 | Yes | Disables the SE pin from changing state |
| SET_HDQINTEN | 0x0015 | Yes | Forces CONTROL_STATUS *[HDQIntEn]* to 1 |
| CLEAR_HDQINTEN | 0x0016 | Yes | Forces CONTROL_STATUS *[HDQIntEn]* to 0 |
| STATIC_CHEM_CHKSUM | 0x0017 | Yes | Calculates chemistry checksum |
| SEALED | 0x0020 | No | Places the bq27541-G1 in SEALED access mode |
| IT_ENABLE | 0x0021 | No | Enables the Impedance Track™ algorithm |
| CAL_ENABLE | 0x002D | No | Toggle bq27541-G1 calibration mode |
| RESET | 0x0041 | No | Forces a full reset of the bq27541-G1 |
| EXIT_CAL | 0x0080 | No | Exit bq27541-G1 calibration mode |
| ENTER_CAL | 0x0081 | No | Enter bq27541-G1 calibration mode |
| OFFSET_CAL | 0x0082 | No | Reports internal CC offset in calibration mode |

#### 8.6.1.1.1 CONTROL_STATUS: 0x0000

Instructs the fuel gauge to return status information to *Control( )* addresses 0x00 and 0x01. The status word includes the following information.

**Table 16. CONTROL_STATUS Flags**

|              | bit7     | bit6      | bit5      | bit4    | bit3 | bit2    | bit1 | bit0      |
|--------------|----------|-----------|-----------|---------|------|---------|------|-----------|
| **High Byte** | SE       | FAS       | SS        | CALMODE | CCA  | BCA     | RSVD | HDQHOSTIN |
| **Low Byte**  | SHUTDWN  | HIBERNATE | FULLSLEEP | SLEEP   | LDMD | RUP_DIS | VOK  | QEN       |

|              |   |
|-------------:|---|
| SE = | Status bit indicating the SE pin is active. True when set. Default is 0. |
| FAS = | Status bit indicating the bq27541-G1 is in FULL ACCESS SEALED state. Active when set. |
| SS = | Status bit indicating the bq27541-G1 is in the SEALED State. Active when set. |
| CALMODE = | Status bit indicating the calibration function is active. True when set. Default is 0. |
| CCA = | Status bit indicating the bq27541-G1 Coulomb Counter Calibration routine is active. The CCA routine will take place approximately 1 minute after the initialization and periodically as gauging conditions change. Active when set. |
| BCA = | Status bit indicating the bq27541-G1 Board Calibration routine is active. Active when set. |
| RSVD = | Reserved. |
| HDQHOSTIN = | Status bit indicating the HDQ interrupt function is active. True when set. Default is 0. |
| SHUTDWN = | Control bit indicating that the SET_SHUTDOWN command has been sent and the state of the SE pin can change to signal an external shutdown of the fuel gauge when conditions permit. See the Shutdown Mode section. |
| HIBERNATE = | Status bit indicating a request for entry into HIBERNATE from SLEEP mode has been issued. True when set. Default is 0. |
| FULLSLEEP = | Status bit indicating the bq27541-G1 is in FULLSLEEP mode. True when set. The state can be detected by monitoring the power used by the bq27541-G1 because any communication will automatically clear it. |
| SLEEP = | Status bit indicating the bq27541-G1 is in SLEEP mode. True when set. |
| LDMD = | Status bit indicating the bq27541-G1 Impedance Track™ algorithm is using *constant-power* mode. True when set. Default is 0 (*constant-current* mode). |
| RUP_DIS = | Status bit indicating the bq27541-G1 Ra table updates are disabled. True when set. |
| VOK = | Status bit indicating cell voltages are OK for Qmax updates. True when set. |
| QEN = | Status bit indicating the bq27541-G1 Qmax updates are enabled. True when set. |

#### 8.6.1.1.2 DEVICE_TYPE: 0x0001

Instructs the fuel gauge to return the device type to addresses 0x00 and 0x01. The bq27541-G1 device type returns 0x0541.

#### 8.6.1.1.3 FW_VERSION: 0x0002

Instructs the fuel gauge to return the firmware version to addresses 0x00 and 0x01. The bq27541-G1 firmware version returns 0x0224.

#### 8.6.1.1.4 HW_VERSION: 0x0003

Instructs the fuel gauge to return the hardware version to addresses 0x00 and 0x01. For bq27541-G1, 0x0000 or 0x0060 is returned.

#### 8.6.1.1.5 RESET_DATA: 0x0005

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

#### 8.6.1.1.6 PREV_MACWRITE: 0x0007

Instructs the fuel gauge to return the previous *Control()* subcommand written to addresses 0x00 and 0x01. The value returned is limited to less than 0x0020.

#### 8.6.1.1.7 CHEM_ID: 0x0008

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

#### 8.6.1.1.8 BOARD_OFFSET: 0x0009

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

#### 8.6.1.1.9 CC_OFFSET: 0x000A

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

#### 8.6.1.1.10 CC_OFFSET_SAVE: 0x000B

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

#### 8.6.1.1.11 DF_VERSION: 0x000C

Instructs the gas gauge to return the data flash version stored in **DF Config Version** to addresses 0x00 and 0x01.

#### 8.6.1.1.12 SET_FULLSLEEP: 0x0010

Instructs the gas gauge to set the *[FULLSLEEP]* bit in Control Status register to 1. This will allow the gauge to enter the FULLSLEEP power mode after the transition to SLEEP power state is detected. In FULLSLEEP 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$^2$C communications the first I$^2$C message will incur a 6-to 8-ms clock stretch while the oscillator is started and stabilized. A communication to the device in FULLSLEEP will force the part back to the SLEEP mode.

#### 8.6.1.1.13 SET_HIBERNATE: 0x0011

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

**NOTE**
Note: The HIBERNATE mode is only available in I$^2$C mode and is disabled when HDQ mode is used.

#### 8.6.1.1.14 CLEAR_HIBERNATE: 0x0012

Instructs the fuel gauge to force the CONTROL_STATUS *[HIBERNATE]* bit to 0. This will prevent the gauge from entering the HIBERNATE power mode after the transition to SLEEP power state is detected unless *Voltage()* is less than **Hibernate V**. It can also be used to force the gauge out of HIBERNATE mode.

#### 8.6.1.1.15 SET_SHUTDOWN: 0x0013

Sets the CONTROL_STATUS *[SHUTDWN]* bit to 1, thereby enabling the SE pin to change state. The Impedance Track™ algorithm controls the setting of the SE pin, depending on whether the conditions are met for fuel gauge shutdown or not.

#### 8.6.1.1.16 CLEAR_SHUTDOWN: 0x0014

Disables the SE pin from changing state. The SE pin is left in a high-impedance state.

#### 8.6.1.1.17 SET_HDQINTEN: 0x0015

Instructs the fuel gauge to set the CONTROL_STATUS *[HDQIntEn]* bit to 1. This enables the HDQ Interrupt function. When this subcommand is received, the device will detect any of the interrupt conditions and assert the interrupt at one second intervals until the CLEAR_HDQINTEN command is received or the count of HDQHostIntrTries has lapsed (default 3).

#### 8.6.1.1.18 CLEAR_HDQINTEN: 0x0016

Instructs the fuel gauge to set the CONTROL_STATUS *[HDQIntEn]* bit to 0. This disables the HDQ Interrupt function.

#### 8.6.1.1.19 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 value stored in the data flash **Static Chem DF Checksum**. If the value matches, the MSB is cleared to indicate pass. If it does not match, the MSB is set to indicate failure. The checksum can be used to verify the integrity of the chemistry data stored internally.

#### 8.6.1.1.20 SEALED: 0x0020

Instructs the gas gauge to transition from UNSEALED state to SEALED state. The gas gauge should always be set to SEALED state for use in customer's end equipment as it prevents spurious writes to most Standard Commands and blocks access to most data flash.

#### 8.6.1.1.21 IT ENABLE: 0x0021

This command 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 system test is completed.

#### 8.6.1.1.22 RESET: 0x0041

This command instructs the gas gauge to perform a full reset. This command is only available when the gas gauge is UNSEALED.

#### 8.6.1.1.23 EXIT_CAL: 0x0080

This command instructs the gas gauge to exit calibration mode.

#### 8.6.1.1.24 ENTER_CAL: 0x0081

This command instructs the gas gauge to enter calibration mode.

#### 8.6.1.1.25 OFFSET_CAL: 0x0082

This command instructs the gas gauge to perform offset calibration.

### 8.6.1.2  AtRate( ): 0x02 and 0x03

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

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

### 8.6.1.3  UnfilteredSOC( ): 0x04 and 0x05

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

### 8.6.1.4  Temperature( ): 0x06 and 0x07

This read-only function returns an unsigned integer value of the battery temperature in units of 0.1°K measured by the fuel gauge and is used for fuel gauging algorithm. It reports either the *InternalTemperature()* or the external thermistor temperature depending on the setting of **[TEMPS]** bit in **Pack Configuration**.

### 8.6.1.5  Voltage( ): 0x08 and 0x09

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

### 8.6.1.6  Flags( ): 0x0A and 0x0B

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

**Table 17. Flags Bit Definitions**

|  | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|---|---|
| **High Byte** | OTC | OTD | BATHI | BATLOW | CHG_INH | RSVD | FC | CHG |
| **Low Byte** | OCVTAKEN | ISD | TDD | HW1 | HW0 | SOC1 | SOCF | DSG |

**High Byte**

OTC = Over-Temperature in Charge condition is detected. True when set. Refer to the Data Flash Safety Subclass parameters for threshold settings.

OTD = Over-Temperature in Discharge condition is detected. True when set. Refer to the Data Flash Safety Subclass parameters for threshold settings.

BATHI = Battery High bit indicating a high battery voltage condition. Refer to the Data Flash **BATTERY HIGH** parameters for threshold settings.

BATLOW = Battery Low bit indicating a low battery voltage condition. Refer to the Data Flash **BATTERY LOW** parameters for threshold settings.

CHG_INH = Charge Inhibit indicates the temperature is outside the range **[Charge Inhibit Temp Low, Charge Inhibit Temp High]**. True when set.

RSVD = Reserved.

FC = Full-charged is detected. FC is set when charge termination is reached and **FC Set%** = -1 (see the Charging and Charge Termination Indication section for details) or State of Charge is larger than **FC Set%** and **FC Set%** is not −1. True when set.

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

**Low Byte**

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

ISD = Internal Short is detected. True when set.

TDD = Tab Disconnect is detected. True when set.

HW[1:0] = Device Identification. Default is 01

SOC1 = State-of-Charge-Threshold 1 is detected. SOC1 bit is set when State of Charge reaches below **SOC1 Set Threshold**. SOC1 bit clears when State-of-Charge gets above **SOC1 Clear Threshold**. True when set.

SOCF = State-of-Charge-Threshold Final is detected. SOCF bit is set when State of Charge reaches below **SOCF Set Threshold**. SOCF bit clears when State-of-Charge gets above **SOCF Clear Threshold**. True when set.

DSG = Discharging detected. True when set.

### 8.6.1.7   NominalAvailableCapacity( ): 0x0C and 0x0D

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

### 8.6.1.8   FullAvailableCapacity( ): 0x0E and 0x0F

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

### 8.6.1.9   RemainingCapacity( ): 0x10 and 0x11

This read-only command pair returns the compensated battery capacity remaining (*UnfilteredRM()*) when the **[SmoothEn]** bit in **Operating Configuration C** is cleared or filtered compensated battery capacity remaining (*FilteredRM()*) when **[SmoothEn]** is set. Units are mAh.

### 8.6.1.10   FullChargeCapacity( ): 0x12 and 0x13

This read-only command pair returns the compensated capacity of fully charged battery (*UnfilteredFCC()*) when the **[SmoothEn]** bit in **Operating Configuration C** is cleared or filtered compensated capacity of fully charged battery (*FilteredFCC()*) when **[SmoothEn]** is set. Units are mAh. *FullChargeCapacity()* is updated at regular intervals, as specified by the IT algorithm.

### 8.6.1.11   AverageCurrent( ): 0x14 and 0x15

This read-only command pair returns a signed integer value that is the average current flow through the sense resistor. It is updated every 1 second. Units are mA.

### 8.6.1.12   TimeToEmpty( ): 0x16 and 0x17

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

### 8.6.1.13   FilteredFCC( ): 0x18 and 0x19

This read-only command pair returns the filtered compensated capacity of the battery when fully charged when the **[SmoothEn]** bit in **Operating Configuration C** is set. Units are mAh. *FilteredFCC()* is updated at regular intervals, as specified by the IT algorithm.

### 8.6.1.14   StandbyCurrent( ): 0x1A and 0x1B

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

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

### 8.6.1.15   UnfilteredFCC( ): 0x1C and 0x1D

This read-only command pair returns the compensated capacity of the battery when fully charged. Units are mAh. *UnFilteredFCC()* is updated at regular intervals, as specified by the IT algorithm.

### 8.6.1.16  MaxLoadCurrent( ): 0x1E and 0x1F

This read-only function returns a signed integer value, in units of mA, of the maximum load conditions of the system. The *MaxLoadCurrent( )* is an adaptive measurement which is initially reported as the maximum load current programmed in **Initial Max Load Current**. If the measured current is ever greater than **Initial Max Load Current**, then *MaxLoadCurrent( )* updates to the new current. *MaxLoadCurrent( )* is reduced to the average of the previous value and **Initial Max Load Current** whenever the battery is charged to full after a previous discharge to an SOC less than 50%. This prevents the reported value from maintaining an unusually high value.

### 8.6.1.17  UnfilteredRM( ): 0x20 and 0x21

This read-only command pair returns the compensated battery capacity remaining. Units are mAh.

### 8.6.1.18  FilteredRM( ): 0x22 and 0x23

This read-only command pair returns the filtered compensated battery capacity remaining when *[SmoothEn]* bit in **Operating Configuration C** is set. Units are mAh.

### 8.6.1.19  AveragePower( ): 0x24 and 0x25

This read-word function returns an unsigned integer value of the average power of the current discharge. It is negative during discharge and positive during charge. A value of 0 indicates that the battery is not being discharged. The value is reported in units of mW (**Design Energy Scale** = 1) or cW (**Design Energy Scale** = 10).

### 8.6.1.20  InternalTemperature( ): 0x28 and 0x29

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

### 8.6.1.21  CycleCount( ): 0x2A and 0x2B

This read-only function returns an unsigned integer value of the number of cycles the battery has experienced with a range of 0 to 65,535. One cycle occurs when accumulated discharge ≥ **CC Threshold**.

### 8.6.1.22  StateOfCharge( ): 0x2C and 0x2D

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

### 8.6.1.23  StateOfHealth( ): 0x2E and 0x2F

0x2E SOH percentage: this read-only function returns an unsigned integer value, expressed as a percentage of the ratio of predicted *FCC(25°C, SOH Load I)* over the *DesignCapacity()*. The *FCC(25°C, SOH Load I)* is the calculated full charge capacity at 25°C and the SOH current rate which is specified by *SOH Load I*. The range of the returned SOH percentage is 0x00 to 0x64, indicating 0 to 100% correspondingly.

### 8.6.1.24  PassedCharge( ): 0x34 and 0x35

This signed integer indicates the amount of charge passed through the sense resistor since the last IT simulation in mAh.

### 8.6.1.25  DOD0( ): 0x36 and 0x37

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

### 8.6.1.26  SelfDischargeCurrent( ): 0x38 and 0x39

This read-only command pair returns the signed integer value that estimates the battery self-discharge current.

### 8.6.2   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 18. For details on the SEALED and UNSEALED states, see Access Modes.

**Table 18. Extended Commands**

| NAME | | COMMAND CODE | UNIT | SEALED ACCESS[1] [2] | UNSEALED ACCESS[1] [2] |
|---|---|---|---|---|---|
| Reserved | RSVD | 0x38 and 0x39 | NA | R | R |
| PackConfig( ) | PCR | 0x3A and 0x3B | Hex# | R | R |
| *DesignCapacity( )* | DCAP | 0x3C and 0x3D | mAh | R | R |
| *DataFlashClass( )* [2] | DFCLS | 0x3E | NA | NA | RW |
| *DataFlashBlock( )* [2] | DFBLK | 0x3F | NA | RW | RW |
| *BlockData( ) / Authenticate( )* [3] | A/DF | 0x40 through 0x53 | NA | RW | RW |
| *BlockData( ) / AuthenticateCheckSum( )* [3] | ACKS/DFD | 0x54 | NA | RW | RW |
| *BlockData( )* | DFD | 0x55 through 0x5F | NA | R | RW |
| *BlockDataCheckSum( )* | DFDCKS | 0x60 | NA | RW | RW |
| *BlockDataControl( )* | DFDCNTL | 0x61 | NA | NA | RW |
| *DeviceNameLength( )* | DNAMELEN | 0x62 | NA | R | R |
| *DeviceName( )* | DNAME | 0x63 through 0x6C | NA | R | R |
| Reserved | RSVD | 0x6D through 0x7F | NA | R | R |

(1)   SEALED and UNSEALED states are entered via commands to *Control( )* 0x00 and 0x01.
(2)   In SEALED mode, data flash cannot be accessed through commands 0x3E and 0x3F.
(3)   The *BlockData( )* command area shares functionality for accessing general data flash and for using Authentication. See Authentication for more details.

#### 8.6.2.1   PackConfig( ): 0x3A and 0x3B

SEALED and UNSEALED Access: This command returns the value stored in **Pack Configuration** and is expressed in hex value.

#### 8.6.2.2   DesignCapacity( ): 0x3C and 0x3D

SEALED and UNSEALED Access: This command returns the value stored in **Design Capacity** and is expressed in mAh. This is intended to be the theoretical or nominal capacity of a new pack, but has no bearing on the operation of the fuel gauge functionality.

#### 8.6.2.3   DataFlashClass( ): 0x3E

This command sets the data flash class to be accessed. The subclass ID to be accessed should be entered in hexadecimal.

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

#### 8.6.2.4   DataFlashBlock( ): 0x3F

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

SEALED Access: This command directs which data flash block is accessed by the *BlockData( )* command. Writing a 0x00 to *DataFlashBlock( )* specifies the *BlockData( )* command transfers authentication data. Issuing a 0x01 or 0x02 instructs the *BlockData( )* command to transfer **Manufacturer Info Block A or B**, respectively.

### 8.6.2.5 BlockData( ): 0x40 Through 0x5F

This command range is used to transfer data for data flash class access. This command range is the 32-byte data block used to access **Manufacturer Info Block A or B**. **Manufacturer Info Block A** is read only for the sealed access. UNSEALED access is read/write.

### 8.6.2.6 BlockDataChecksum( ): 0x60

The host system should write this value to inform the device that new data is ready for programming into the specified data flash class and block.

UNSEALED Access: This byte contains the checksum on the 32 bytes of block data read or written to data flash. The least-significant byte of the sum of the data bytes written must be complemented ( [255 – x] , for x the 8-bit summation of the *BlockData()* (0x40 to 0x5F) on a byte-by-byte basis.) before being written to 0x60.

SEALED Access: This byte contains the checksum for the 32 bytes of block data written to **Manufacturer Info Block A**. The least-significant byte of the sum of the data bytes written must be complemented ( [255 – x] , for x the 8-bit summation of the *BlockData()* (0x40 to 0x5F) on a byte-by-byte basis.) before being written to 0x60.

### 8.6.2.7 BlockDataControl( ): 0x61

UNSEALED Access: This command is used to control data flash access mode. The value determines the data flash to be accessed. Writing 0x00 to this command enables *BlockData( )* to access general data flash.

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

### 8.6.2.8 DeviceNameLength( ): 0x62

UNSEALED and SEALED Access: This byte contains the length of the **Device Name**.

### 8.6.2.9 DeviceName( ): 0x63 Through 0x6C

UNSEALED and SEALED Access: This block contains the device name that is programmed in **Device Name**.

### 8.6.2.10 Reserved – 0x6A Through 0x7F

# 9 Data Flash Interface

## 9.1 Accessing the Data Flash

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

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

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

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

Occasionally, a data flash Class will be larger than the 32-byte block size. In this case, the *DataFlashBlock( )* command is used to designate which 32-byte block the desired locations reside in. 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 67, it must reside in the third 32-byte block. Hence, *DataFlashBlock( )* is issued 0x02 to set the block offset, and the offset used to index into the *BlockData( )* memory area is 0x40 + 67 *modulo* 32 = 0x40 + 16 = 0x40 + 0x03 = 0x43.

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

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

## 9.2 Manufacturer Information Blocks

The fuel gauge contains 64 bytes of user programmable data flash storage: **Manufacturer Info Block A** and **Manufacturer Info Block B**. The method for accessing these memory locations is slightly different, depending on whether the device is in UNSEALED or SEALED modes.

When in UNSEALED mode and when 0x00 has been written to *BlockDataControl( )*, accessing the Manufacturer Info Blocks 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 B** is defined as having a Subclass = 58 and an Offset = 32 through 63 (32-byte block). The specification of Class = System Data is not needed to address **Manufacturer Info Block B**, but is used instead for grouping purposes when viewing data flash info in the 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 or 0x02 with this command causes the corresponding information block (A or B respectively) to be transferred to the command space 0x40 through 0x5F for editing or reading by the system. Upon successful writing of checksum information to *BlockDataChecksum( )*, the modified block is returned to data flash.

**Manufacturer Information Blocks (continued)**

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

## 9.3 Access Modes

The bq27541-G1 provides three security modes (FULL ACCESS, UNSEALED, and SEALED) that control data flash access permissions according to Table 19. *Data Flash* refers to those data flash locations, Table 20 through Table 26, that are accessible to the user. *Manufacture Information* refers to the two 32-byte blocks.

**Table 19. Data Flash Access**

| SECURITY MODE | DATA FLASH | MANUFACTURER INFORMATION |
|---|---|---|
| FULL ACCESS | RW | RW |
| UNSEALED | RW | RW |
| SEALED | None | R (A); RW (B) |

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

## 9.4 SEALING or UNSEALING Data Flash

The fuel gauge implements a key-access scheme to transition between SEALED, UNSEALED, and FULL-ACCESS modes. Each transition requires that a unique set of two keys be sent to the fuel gauge via the *Control( )* control command. The keys must be sent consecutively, with no other data being written to the *Control( )* register in between. Note that to avoid conflict, the keys must be different from the codes presented in the *CNTL DATA* column of Table 15 subcommands.

When in SEALED mode the *[SS]* bit of CONTROL_STATUS is set, but when the UNSEAL keys are correctly received by the fuel gauge, the *[SS]* bit is cleared. When the full-access keys are correctly received the CONTROL_STATUS *[FAS]* bit is cleared.

Both **Unseal Key** and **Full-Access Key** have two words and are stored in data flash. The first word is Key 0 and the second word is Key 1. The order of the keys sent to fuel gauge are Key 1 followed by Key 0. The order of the bytes for each key entered through the *Control( )* command is the reverse of what is read from the part. For an example, if the Unseal Key is 0x56781234, key 1 is 0x1234 and key 0 is 0x5678. Then *Control( )* should supply 0x3412 and 0x7856 to unseal the part. The **Unseal Key** and the **Full-Access Key** can only be updated when in FULL-ACCESS mode.

## 9.5 Data Flash Summary

Table 20 through Table 26 summarize the data flash locations available to the user, including their default, minimum, and maximum values.

**Table 20. Data Flash Summary—Configuration Class**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | | | MIN | MAX | DEFAULT | |
| 2 | Safety | 0 | OT Chg | I2 | 0 | 1200 | 550 | 0.1°C |
| | | 2 | OT Chg Time | U1 | 0 | 60 | 2 | s |
| | | 3 | OT Chg Recovery | I2 | 0 | 1200 | 500 | 0.1°C |
| | | 5 | OT Dsg | I2 | 0 | 1200 | 600 | 0.1°C |
| | | 7 | OT Dsg Time | U1 | 0 | 60 | 2 | s |
| | | 8 | OT Dsg Recovery | I2 | 0 | 1200 | 550 | 0.1°C |
| 32 | Charge Inhibit Cfg | 0 | Chg Inhibit Temp Low | I2 | –400 | 1200 | 0 | 0.1°C |
| | | 2 | Chg Inhibit Temp High | I2 | –400 | 1200 | 450 | 0.1°C |
| | | 4 | Temp Hys | I2 | 0 | 100 | 50 | 0.1°C |
| 34 | Charge | 0 | Charging Voltage | I2 | 0 | 4600 | 4200 | mV |
| 36 | Charge Termination | 0 | Taper Current | I2 | 0 | 1000 | 100 | mA |
| | | 2 | Min Taper Capacity | I2 | 0 | 1000 | 25 | mAh |
| | | 4 | Taper Voltage | I2 | 0 | 1000 | 100 | mV |
| | | 6 | Current Taper Window | U1 | 0 | 60 | 40 | s |
| | | 7 | TCA Set % | I1 | –1 | 100 | 99 | % |
| | | 8 | TCA Clear % | I1 | –1 | 100 | 95 | % |
| | | 9 | FC Set % | I1 | –1 | 100 | –1 | % |
| | | 10 | FC Clear % | I1 | –1 | 100 | 98 | % |
| | | 11 | DODatEOC Delta T | I2 | 0 | 1000 | 50 | 0.1°C |
| 48 | Data | 0 | Rem Cap Alarm | I2 | 0 | 700 | 100 | mA |
| | | 8 | Initial Standby | I1 | –256 | 0 | –10 | mA |
| | | 9 | Initial MaxLoad | I2 | –32767 | 0 | –500 | mA |
| | | 17 | Cycle Count | U2 | 0 | 65535 | 0 | |
| | | 19 | CC Threshold | I2 | 100 | 32767 | 900 | mAh |
| | | 23 | Design Capacity | I2 | 0 | 32767 | 1000 | mAh |
| | | 25 | Design Energy | I2 | 0 | 32767 | 5400 | mWh |
| | | 27 | SOH Load I | I2 | –32767 | 0 | –400 | mA |
| | | 29 | TDD SOH Percent | I1 | 0 | 100 | 80 | % |
| | | 40 | ISD Current | I2 | 0 | 32767 | 10 | Hour Rate |
| | | 42 | ISD I Filter | U1 | 0 | 255 | 127 | |
| | | 43 | Min ISD Time | U1 | 0 | 255 | 7 | Hour |
| | | 44 | Design Energy Scale | U1 | 0 | 255 | 1 | |
| | | 45 | Device Name | S11 | x | x | bq2754X-G1 | - |

## Data Flash Summary (continued)

**Table 20. Data Flash Summary—Configuration Class (continued)**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
|---|---|---|---|---|---|---|---|---|
| | | | | | MIN | MAX | DEFAULT | |
| 49 | Discharge | 0 | SOC1 Set Threshold | U2 | 0 | 65535 | 150 | mAh |
| | | 2 | SOC1 Clear Threshold | U2 | 0 | 65535 | 175 | mAh |
| | | 4 | SOCF Set Threshold | U2 | 0 | 65535 | 75 | mAh |
| | | 6 | SOCF Clear Threshold | U2 | 0 | 65535 | 100 | mAh |
| | | 9 | BL Set Volt Threshold | I2 | 0 | 16800 | 2500 | mV |
| | | 11 | BL Set Volt Time | U1 | 0 | 60 | 2 | s |
| | | 12 | BL Clear Volt Threshold | I2 | 0000 | 16800 | 2600 | mV |
| | | 14 | BH Set Volt Threshold | I2 | 0 | 16800 | 4500 | mV |
| | | 16 | BH Volt Time | U1 | 0 | 60 | 2 | s |
| | | 17 | BH Clear Volt Threshold | I2 | 0000 | 16800 | 4400 | mV |
| 56 | Manufacturer Data | 0 | Pack Lot Code | H2 | 0x0 | 0xFFFF | 0x0 | - |
| | | 2 | PCB Lot Code | H2 | 0x0 | 0xFFFF | 0x0 | - |
| | | 4 | Firmware Version | H2 | 0x0 | 0xFFFF | 0x0 | - |
| | | 6 | Hardware Revision | H2 | 0x0 | 0xFFFF | 0x0 | - |
| | | 8 | Cell Revision | H2 | 0x0 | 0xFFFF | 0x0 | - |
| | | 10 | DF Config Version | H2 | 0x0 | 0xFFFF | 0x0 | - |
| 57 | Integrity Data | 6 | Static Chem DF Checksum | H2 | 0x0 | 0x7FFF | 0x0 | |
| 59 | Lifetime Data | 0 | Lifetime Max Temp | I2 | 0 | 1400 | 0 | 0.1°C |
| | | 2 | Lifetime Min Temp | I2 | −600 | 1400 | 500 | 0.1°C |
| | | 4 | Lifetime Max Pack Voltage | I2 | 0 | 32767 | 2800 | mV |
| | | 6 | Lifetime Min Pack Voltage | I2 | 0 | 32767 | 4200 | mV |
| | | 8 | Lifetime Max Chg Current | I2 | −32767 | 32767 | 0 | mA |
| | | 10 | Lifetime Max Dsg Current | I2 | −32767 | 32767 | 0 | mA |
| 60 | Lifetime Temp Samples | 0 | LT Flash Cnt | U2 | 0 | 65535 | 0 | |
| 64 | Registers | 0 | Pack Configuration | H2 | 0x0 | 0xFFFF | 0x1177 | |
| | | 2 | Pack Configuration B | H1 | 0x0 | 0xFF | 0xA7 | |
| | | 3 | Pack Configuration C | H1 | 0x0 | 0xFF | 0x18 | |
| 66 | Lifetime Resolution | 0 | LT Temp Res | U1 | 0 | 255 | 10 | Num |
| | | 1 | LT V Res | U1 | 0 | 255 | 25 | Num |
| | | 2 | LT Cur Res | U1 | 0 | 255 | 100 | Num |
| | | 3 | LT Update Time | U2 | 0 | 65535 | 60 | Num |
| 68 | Power | 0 | Flash Update OK Voltage | I2 | 0 | 4200 | 2800 | mV |
| | | 2 | Sleep Current | I2 | 0 | 100 | 10 | mA |
| | | 11 | Hibernate I | U2 | 0 | 700 | 8 | mA |
| | | 13 | Hibernate V | U2 | 2400 | 3000 | 2550 | mV |
| | | 15 | FS Wait | U1 | 0 | 255 | 0 | s |

**Table 21. Data Flash Summary—System Data Class**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
|---|---|---|---|---|---|---|---|---|
| | | | | | MIN | MAX | DEFAULT | |
| 58 | Manufacturer Info | 0 through 31 | Block A 0 through 31 | H1 | 0x0 | 0xFF | 0x0 | - |
| | | 32 through 63 | Block B 0 through 31 | H1 | 0x0 | 0xFF | 0x0 | - |

**Table 22. Data Flash Summary—Gas (Fuel) Gauging Class**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
|---|---|---|---|---|---|---|---|---|
| | | | | | MIN | MAX | DEFAULT | |
| 80 | IT Cfg | 0 | Load Select | U1 | 0 | 255 | 1 | |
| | | 1 | Load Mode | U1 | 0 | 255 | 0 | |
| | | 21 | Max Res Factor | U1 | 0 | 255 | 15 | |
| | | 22 | Min Res Factor | U1 | 0 | 255 | 5 | |
| | | 25 | Ra Filter | U2 | 0 | 1000 | 800 | |
| | | 67 | Terminate Voltage | I2 | 2800 | 3700 | 3000 | mV |
| | | 69 | Term V Delta | I2 | 0 | 4200 | 200 | mV |
| | | 72 | ResRelax Time | U2 | 0 | 65534 | 500 | s |
| | | 76 | User Rate-mA | I2 | 2000 | 9000 | 0 | mA |
| | | 78 | User Rate-Pwr | I2 | 3000 | 14000 | 0 | mW or cW |
| | | 80 | Reserve Cap-mAh | I2 | 0 | 9000 | 0 | mA |
| | | 82 | Reserve Energy | I2 | 0 | 14000 | 0 | mWh or cWh |
| | | 86 | Max Scale Back Grid | U1 | 0 | 15 | 4 | |
| | | 87 | Max DeltaV | U2 | 0 | 65535 | 200 | mV |
| | | 89 | Min DeltaV | U2 | 0 | 65535 | 0 | mV |
| | | 91 | Max Sim Rate | U1 | 0 | 255 | 1 | C/rate |
| | | 92 | Min Sim Rate | U1 | 0 | 255 | 20 | C/rate |
| | | 93 | Ra Max Delta | U2 | 0 | 65535 | 43 | mΩ |
| | | 95 | Qmax Max Delta % | U1 | 0 | 100 | 5 | mAh |
| | | 96 | DeltaV Max Delta | U2 | 0 | 65535 | 10 | mV |
| | | 102 | Fast Scale Start SOC | U1 | 0 | 100 | 10 | % |
| | | 103 | Charge Hys V Shift | I2 | 0 | 2000 | 40 | mV |
| 81 | Current Thresholds | 0 | Dsg Current Threshold | I2 | 0 | 2000 | 60 | mA |
| | | 2 | Chg Current Threshold | I2 | 0 | 2000 | 75 | mA |
| | | 4 | Quit Current | I2 | 0 | 1000 | 40 | mA |
| | | 6 | Dsg Relax Time | U2 | 0 | 8191 | 60 | s |
| | | 8 | Chg Relax Time | U1 | 0 | 255 | 60 | s |
| | | 9 | Quit Relax Time | U1 | 0 | 63 | 1 | s |
| | | 10 | Max IR Correct | U2 | 0 | 1000 | 400 | mV |

**Table 22. Data Flash Summary—Gas (Fuel) Gauging Class (continued)**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
|---|---|---|---|---|---|---|---|---|
| | | | | | MIN | MAX | DEFAULT | |
| 82 | State | 0 | Qmax Cell 0 | I2 | 0 | 32767 | 1000 | mAh |
| | | 2 | Cycle Count | U2 | 0 | 65535 | 0 | |
| | | 4 | Update Status | H1 | 0x0 | 0x6 | 0x0 | |
| | | 5 | V at Chg Term | I2 | 0 | 5000 | 4200 | mV |
| | | 7 | Avg I Last Run | I2 | −32768 | 32767 | −299 | mA |
| | | 9 | Avg P Last Run | I2 | −32768 | 32767 | −1131 | mW |
| | | 11 | Delta Voltage | I2 | −32768 | 32767 | 2 | mV |
| | | 15 | T Rise | I2 | 0 | 32767 | 20 | Num |
| | | 17 | T Time Constant | I2 | 0 | 32767 | 1000 | Num |

**Table 23. Data Flash Summary—OCV Table Class**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
|---|---|---|---|---|---|---|---|---|
| | | | | | MIN | MAX | DEFAULT | |
| 83 | OCV Table | 0 | Chem ID | H2 | 0x0 | 0xFFFF | 0x0128 | num |

**Table 24. Data Flash Summary—Ra Table Class**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
|---|---|---|---|---|---|---|---|---|
| | | | | | MIN | MAX | DEFAULT | |
| 88 | R_a0 | 0 | Cell0 R_a flag | H2 | 0x0 | 0x0 | 0xFF55 | - |
| | | 2 through 31 | Cell0 R_a 0 through 14 | I2 | 183 | 183 | 407 | $2^{-10}\Omega$ |
| 89 | R_a0x | 0 | xCell0 R_a flag | H2 | 0xFFFF | 0xFFFF | 0xFFFF | - |
| | | 2 through 31 | xCell0 R_a 0 through 14 | I2 | 183 | 183 | 407 | $2^{-10}\Omega$ |

**Table 25. Data Flash Summary—Calibration Class**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
|---|---|---|---|---|---|---|---|---|
| | | | | | MIN | MAX | DEFAULT | |
| 104 | Data | 0 | CC Gain | F4 | 1.0e−1 | 4.0e+1 | 0.4768 | |
| | | 4 | CC Delta | F4 | 2.9826e+4 | 1.193046e+6 | 567744.56 | |
| | | 8 | CC Offset | I2 | −32768 | 32767 | −1200 | mA |
| | | 10 | Board Offset | I1 | −128 | 127 | 0 | µA |
| | | 11 | Int Temp Offset | I1 | −128 | 127 | 0 | |
| | | 12 | Ext Temp Offset | I1 | −128 | 127 | 0 | |
| | | 13 | Pack V Offset | I1 | −128 | 127 | 0 | |
| 107 | Current | 1 | Deadband | U1 | 0 | 255 | 5 | mA |

**Table 26. Data Flash Summary—Security Class**

| SUBCLASS ID | SUBCLASS | OFFSET | NAME | DATA TYPE | VALUE | | | UNIT (EVSW UNIT)* |
|---|---|---|---|---|---|---|---|---|
| | | | | | MIN | MAX | DEFAULT | |
| 112 | Codes | 0 | Sealed to Unsealed | H4 | 0x0 | 0xFFFF FFFF | 0x3672 0414 | - |
| | | 4 | Unsealed to Full | H4 | 0x0 | 0xFFFF FFFF | 0xFFFF FFFF | - |
| | | 8 | Authen Key3 | H4 | 0x0 | 0xFFFF FFFF | 0x0123 4567 | - |
| | | 12 | Authen Key2 | H4 | 0x0 | 0xFFFF FFFF | 0x89AB CDEF | - |
| | | 16 | Authen Key1 | H4 | 0x0 | 0xFFFF FFFF | 0xFEDC BA98 | - |
| | | 20 | Authen Key0 | H4 | 0x0 | 0xFFFF FFFF | 0x7654 3210 | - |

**Table 27. Data Flash to EVSW Conversion**

| Class | SubClass ID | SubClass | Offset | Name | Data Type | Data Flash Default | Data Flash Unit | EVSW Default | EVSW Unit | Data Flash (DF) to EVSW Conversion |
|---|---|---|---|---|---|---|---|---|---|---|
| Gas Gauging | 80 | IT Cfg | 78 | User Rate-Pwr | I2 | 0 | cW or 10W | 0 | mW or cW | DF × 10 |
| Gas Gauging | 80 | IT Cfg | 82 | Reserve Energy | I2 | 0 | cWh or 10cWh | 0 | mWh or cWh | DF × 10 |
| Calibration | 104 | Data | 0 | CC Gain | F4 | 0.47095 | Num | 10.124 | mΩ | 4.768 / DF |
| Calibration | 104 | Data | 4 | CC Delta | F4 | 5.595e5 | Num | 10.147 | mΩ | 5677445 / DF |
| Calibration | 104 | Data | 8 | CC Offset | I2 | −1200 | Num | −0.576 | mV | DF × 0.0048 |
| Calibration | 104 | Data | 10 | Board Offset | I1 | 0 | Num | 0 | µV | DF × 0.0075 |

# 10 Application and Implementation

## 10.1 Typical Applications
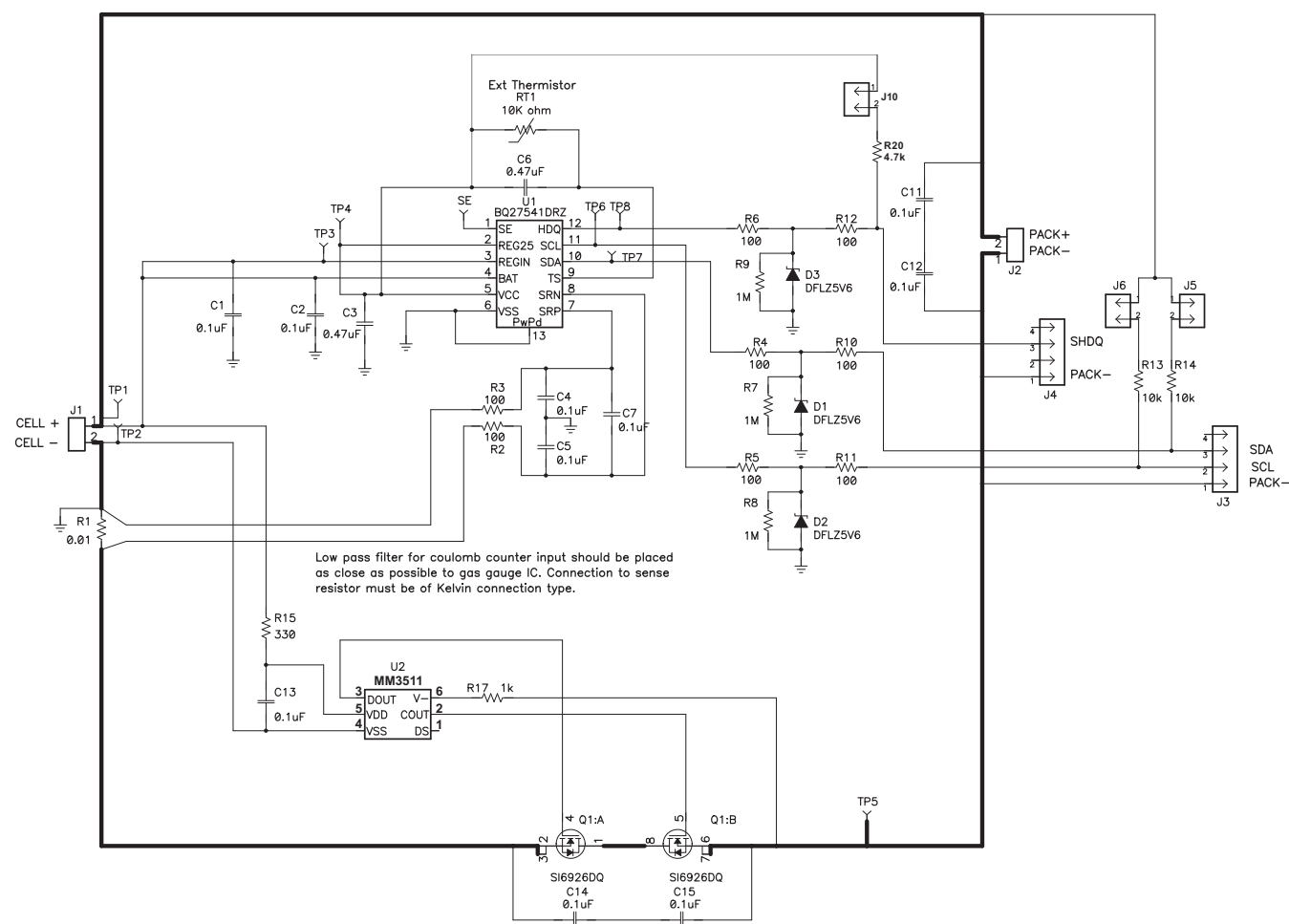


R7, R8, and R9 are optional pulldown resistors if pullup resistors are applied.

# 11 Device and Documentation Support

## 11.1 Documentation Support

### 11.1.1 Related Documentation from Texas Instruments

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

1. *bq27541 EVM: Single Cell Impedance Track™ Technology* User's Guide (SLUU273)

## 11.2 Trademarks

Impedance Track is a trademark of Texas Instruments.

## 11.3 Electrostatic Discharge Caution

These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

## 11.4 Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms and definitions.

# 12 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

# 重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改， 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内，且 TI 认为 有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定，否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应 用相关的风险，客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予 的直接或隐含权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息，不能构成从 TI 获得使用这些产品或服 务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可，或是 TI 的专利权或其它 知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分，仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况 下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时，如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分，则会失去相关 TI 组件 或服务的所有明示或暗示授权，且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意，尽管任何应用相关信息或支持仍可能由 TI 提供，但他们将独力负责满足与其产品及在其应用中使用 TI 产品 相关的所有法律、法规和安全相关要求。客户声明并同意，他们具备制定与实施安全措施所需的全部专业技术和知识，可预见 故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因 在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理商造成的任何损失。

在某些场合中，为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特 有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此，此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III（或类似的生命攸关医疗设备）的授权许可，除非各方授权官员已经达成了专门管控此类使 用的特别协议。

只有那些 TI 特别注明属于军用等级或"增强型塑料"的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同 意，对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用，其风险由客户单独承担，并且由客户独 力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品，这些产品主要用于汽车。在任何情况下，因使用非指定产品而无法达到 ISO/TS16949 要 求，TI不承担任何责任。

| 产品 | | 应用 | |
|------|------|------|------|
| 数字音频 | www.ti.com.cn/audio | 通信与电信 | www.ti.com.cn/telecom |
| 放大器和线性器件 | www.ti.com.cn/amplifiers | 计算机及周边 | www.ti.com.cn/computer |
| 数据转换器 | www.ti.com.cn/dataconverters | 消费电子 | www.ti.com/consumer-apps |
| DLP® 产品 | www.dlp.com | 能源 | www.ti.com/energy |
| DSP - 数字信号处理器 | www.ti.com.cn/dsp | 工业应用 | www.ti.com.cn/industrial |
| 时钟和计时器 | www.ti.com.cn/clockandtimers | 医疗电子 | www.ti.com.cn/medical |
| 接口 | www.ti.com.cn/interface | 安防应用 | www.ti.com.cn/security |
| 逻辑 | www.ti.com.cn/logic | 汽车电子 | www.ti.com.cn/automotive |
| 电源管理 | www.ti.com.cn/power | 视频和影像 | www.ti.com.cn/video |
| 微控制器 (MCU) | www.ti.com.cn/microcontrollers | | |
| RFID 系统 | www.ti.com.cn/rfidsys | | |
| OMAP应用处理器 | www.ti.com/omap | | |
| 无线连通性 | www.ti.com.cn/wirelessconnectivity | 德州仪器在线技术支持社区 | www.deyisupport.com |

## PACKAGING INFORMATION

| Orderable Device | Status (1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan (2) | Lead finish/ Ball material (6) | MSL Peak Temp (3) | Op Temp (°C) | Device Marking (4/5) | Samples |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BQ27541DRZR-G1 | ACTIVE | SON | DRZ | 12 | 3000 | RoHS & Green | NIPDAU | Level-2-260C-1 YEAR | -40 to 85 | BQ 7541 | Samples |
| BQ27541DRZT-G1 | ACTIVE | SON | DRZ | 12 | 250 | RoHS & Green | NIPDAU | Level-2-260C-1 YEAR | -40 to 85 | BQ 7541 | Samples |

**(1)** The marketing status values are defined as follows:

**ACTIVE:** Product device recommended for new designs.

**LIFEBUY:** TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

**NRND:** Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

**PREVIEW:** Device has been announced but is not in production. Samples may or may not be available.

**OBSOLETE:** TI has discontinued the production of the device.

**(2)** **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".

**RoHS Exempt:** TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.

**Green:** TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

**(3)** MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

**(4)** There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

**(5)** Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

**(6)** Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

**Important Information and Disclaimer:** The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

# PACKAGE MATERIALS INFORMATION

## TAPE AND REEL INFORMATION

**REEL DIMENSIONS**

**TAPE DIMENSIONS**

| A0 | Dimension designed to accommodate the component width |
|----|----|
| B0 | Dimension designed to accommodate the component length |
| K0 | Dimension designed to accommodate the component thickness |
| W | Overall width of the carrier tape |
| P1 | Pitch between successive cavity centers |

**QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE**
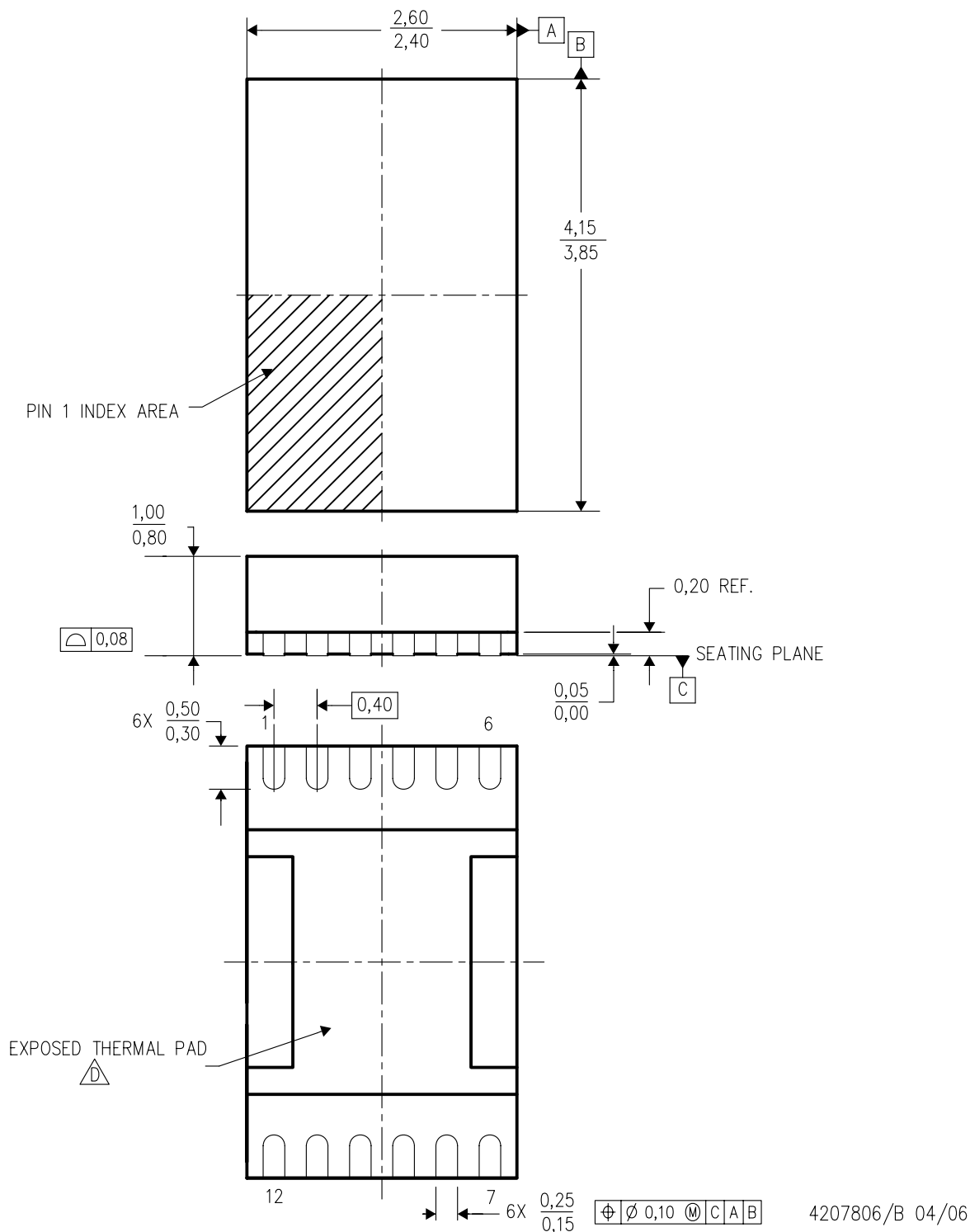
*All dimensions are nominal

| Device | Package Type | Package Drawing | Pins | SPQ | Reel Diameter (mm) | Reel Width W1 (mm) | A0 (mm) | B0 (mm) | K0 (mm) | P1 (mm) | W (mm) | Pin1 Quadrant |
|--------|--------------|------------------|------|------|---------------------|---------------------|---------|---------|---------|---------|--------|----------------|
| BQ27541DRZR-G1 | SON | DRZ | 12 | 3000 | 330.0 | 12.4 | 2.8 | 4.3 | 1.2 | 4.0 | 12.0 | Q2 |
| BQ27541DRZT-G1 | SON | DRZ | 12 | 250 | 180.0 | 12.4 | 2.8 | 4.3 | 1.2 | 4.0 | 12.0 | Q2 |

# PACKAGE MATERIALS INFORMATION

**TAPE AND REEL BOX DIMENSIONS**



*All dimensions are nominal

| Device | Package Type | Package Drawing | Pins | SPQ | Length (mm) | Width (mm) | Height (mm) |
|---|---|---|---|---|---|---|---|
| BQ27541DRZR-G1 | SON | DRZ | 12 | 3000 | 552.0 | 367.0 | 36.0 |
| BQ27541DRZT-G1 | SON | DRZ | 12 | 250 | 552.0 | 185.0 | 36.0 |

## DRZ (S−PDSO−N12)                    PLASTIC SMALL OUTLINE

2,60
2,40

A B

4,15
3,85

PIN 1 INDEX AREA

1,00
0,80

0,20 REF.

0,08

SEATING PLANE

0,05
0,00

C

6X 0,50
0,30

0,40

1                    6

EXPOSED THERMAL PAD
D

12                    7    6X 0,25
0,15

⌖ Ø 0,10 M C A B        4207806/B 04/06

NOTES:    A.    All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M−1994.
          B.    This drawing is subject to change without notice.
          C.    Small Outline No−Lead (SON) package configuration.
          D.    The package thermal pad must be soldered to the board for thermal and mechanical performance.
                See the Product Data Sheet for details regarding the exposed thermal pad dimensions.
          E.    This package is lead−free.
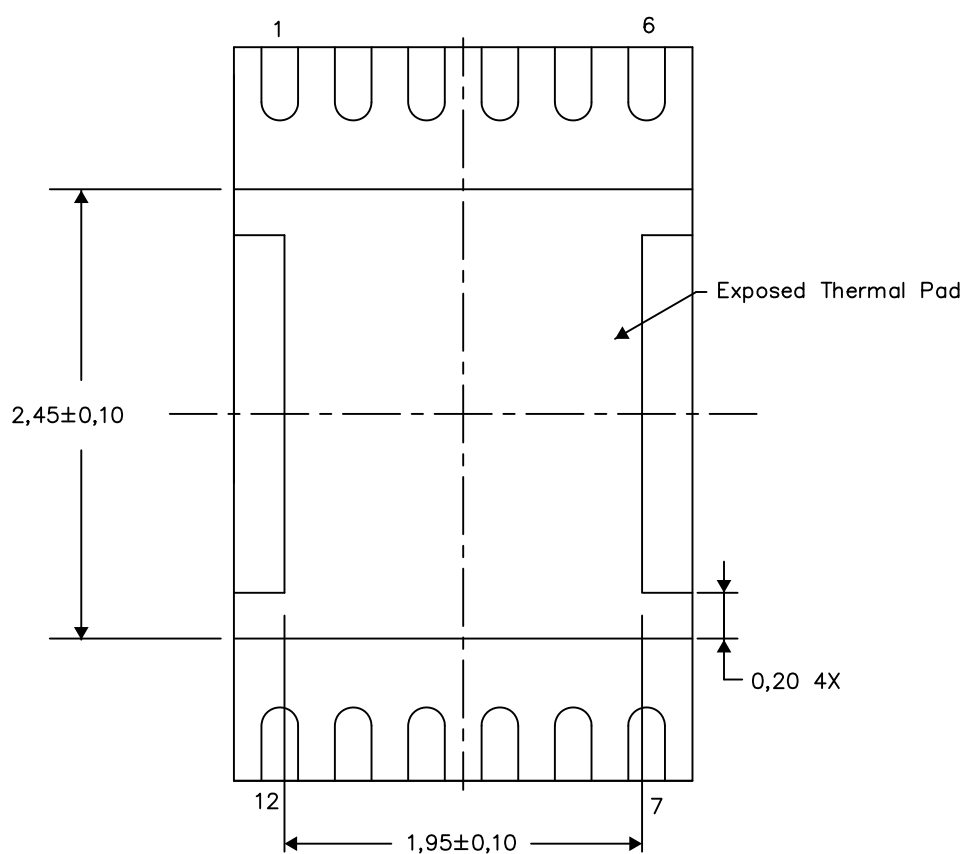
DRZ (R−PDSO−N12)                    PLASTIC SMALL OUTLINE NO−LEAD

THERMAL INFORMATION

This package incorporates an exposed thermal pad that is designed to be attached directly to an external heatsink. The thermal pad must be soldered directly to the printed circuit board (PCB). After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For information on the Quad Flatpack No−Lead (QFN) package and its advantages, refer to Application Report, QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271. This document is available at www.ti.com.

The exposed thermal pad dimensions for this package are shown in the following illustration.
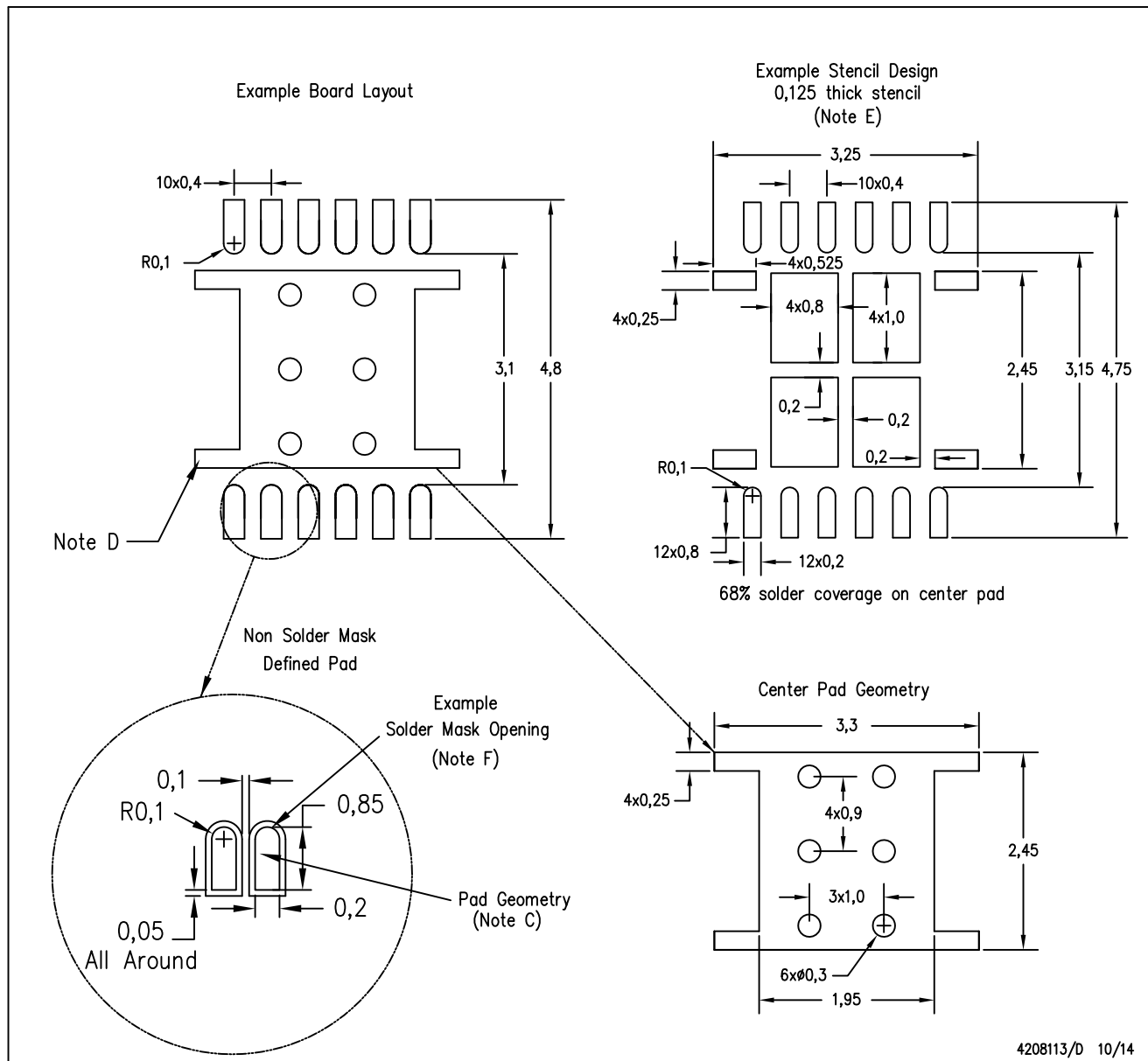


Bottom View
Exposed Thermal Pad Dimensions

4208114/F  10/14

NOTE:   All linear dimensions are in millimeters

## DRZ (S−PDSO−N12)      PLASTIC SMALL OUTLINE NO−LEAD

Example Board Layout

Example Stencil Design
0,125 thick stencil
(Note E)

10x0,4

3,25

10x0,4

R0,1

4x0,525

4x0,25

4x0,8

4x1,0

3,1   4,8

2,45   3,15   4,75

0,2

0,2

Note D

R0,1

0,2

12x0,8

12x0,2

68% solder coverage on center pad

Non Solder Mask
Defined Pad

Example
Solder Mask Opening
(Note F)

Center Pad Geometry

3,3

0,1

R0,1

0,85

4x0,25

4x0,9

2,45

0,2

Pad Geometry
(Note C)

3x1,0

0,05

All Around

6xØ0,3

1,95

4208113/D   10/14

NOTES:   A.   All linear dimensions are in millimeters.
           B.   This drawing is subject to change without notice.
           C.   Publication IPC−7351 is recommended for alternate designs.
           D.   This package is designed to be soldered to a thermal pad on the board.  Refer to Application Note, Quad Flat−Pack
               Packages, Texas Instruments Literature No. SCBA017, SLUA271, and also the Product Data Sheets
               for specific thermal information, via requirements, and recommended board layout.  These documents are available at
               www.ti.com <http://www.ti.com>.
           E.   Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release.  Customers should
               contact their board assembly site for stencil design recommendations.  Refer to IPC 7525 for stencil design considerations.
           F.   Customers should contact their board fabrication site for minimum solder mask web tolerances between signal pads.

**TEXAS INSTRUMENTS**
www.ti.com