Application Report
SLUA875–August 2018

Advanced Charge Algorithm Application Note

Jasper Yi

ABSTRACT

This document details the various features and options available in bq40z50-R2 that can be used to control a Li-ion charger voltage and current settings. Various advanced charging algorithms can be implemented by configuring the data flash values in the fuel gauge accordingly.

Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The bq40z5x family of battery gauge devices provides Advanced Charge Algorithm options which allow the user to implement flexible custom charging profiles based on the *Temperature( )*, *Voltage( )*, *RSoC( )*, *CycleCount( )* and *StateOfHealth( )*information. It can minimize charging time and prolong battery lifetime by using the recommended values reported in the *ChargingVoltage( )* and *ChargingCurrent( )* gauge registers. This application note details how to configure the values in Data Flash to obtain optimized *ChargingVoltage( )* and *ChargingCurrent( )* by showing a few examples.

2 Dataflash Configuration Overview

Lithium-ion cells can become damaged and possibly dangerous if they are overcharged, especially at extreme temperatures. To improve the safety of charging lithium-ion batteries, some vendors have their own charging profiles based on a segmented temperature range. Figure 1 shows an example for recommended charge current and charge voltage over temperature adhere the following conditions.

- T1 ≤ T2 ≤ T5 ≤ T6 ≤ T3 ≤ T4

In Figure 1, T5 = T6 since both have the same charging voltage and current:

- *T1 ~ T4* are located in *Temperature Ranges: T1 Temp... T4 Temp*

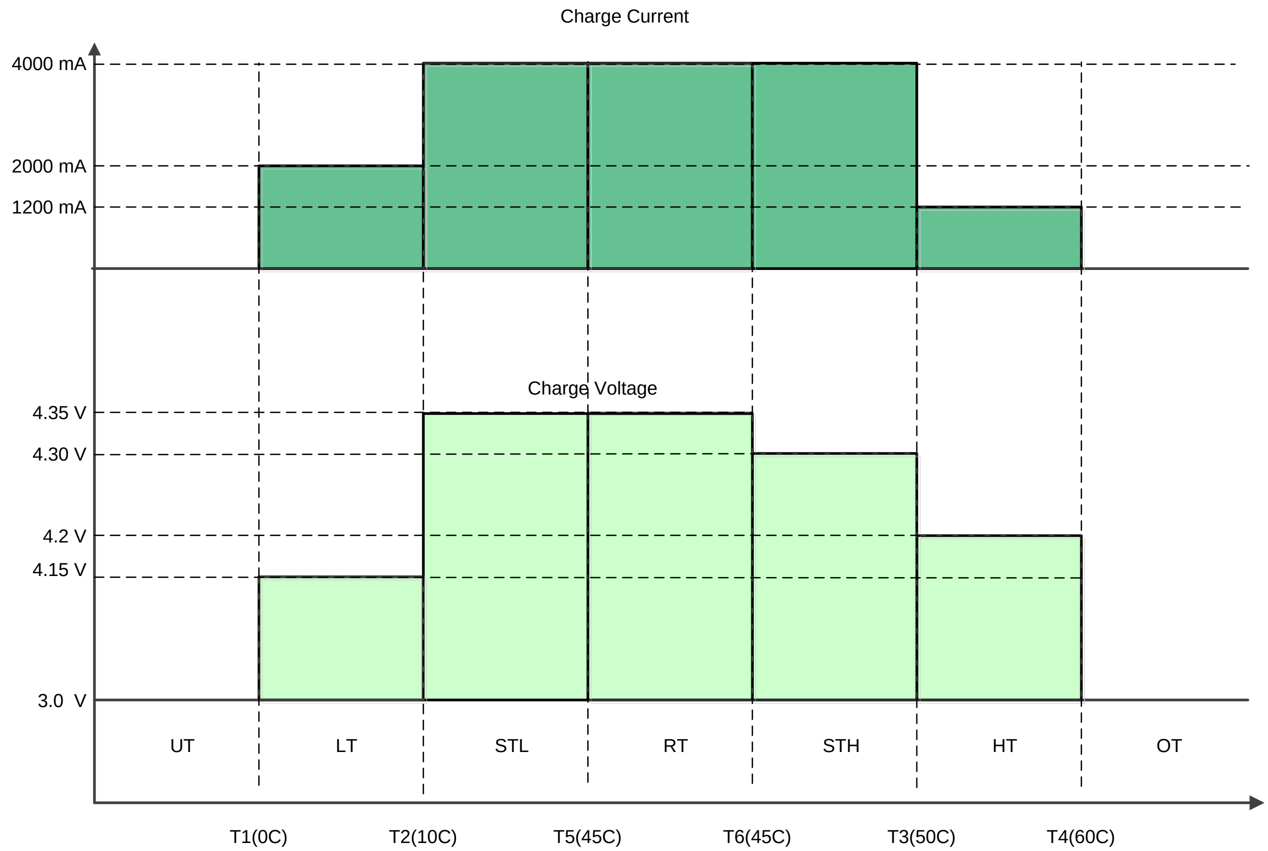

Figure 1. Charging profile example

2.1 *Temperature Configuration*

Determine *Temperature Ranges: Tx Temp* based on the charging profile.

Set the temperature values as follows:

- *T1 Temp* = 0
- *T2 Temp* = 10
- *T5 Temp* = 45
- *T6 Temp* = 45
- *T3 Temp* = 50
- *T4 Temp* = 60

The Low Temp Charging range is between *T1 Temp* and *T2 Temp*.

Standard Temp Low Charging range is between *T2 Temp* and *T5 Temp*.

Rec Temp Charging range is between*T5 Temp* and *T6 Temp*

Standard Temp High Charging range is between *T6 Temp* and *T3 Temp*.

High Temp Charging range is between *T3 Temp* and *T4 Temp*.

Over *T4 Temp* is Over Temp Charging.

See Table 1 for the DF parameters:

Table 1. DF Parameters

Class	Subclass	Name	Type	Min	Max	Value	Unit
Advanced Charging Algorithm	Temperature Ranges	T1 Temp	I1	–128	127	0	°C
Advanced Charging Algorithm	Temperature Ranges	T2 Temp	I1	–128	127	10	°C
Advanced Charging Algorithm	Temperature Ranges	T5 Temp	I1	–128	127	45	°C
Advanced Charging Algorithm	Temperature Ranges	T6 Temp	I1	–128	127	45	°C
Advanced Charging Algorithm	Temperature Ranges	T3 Temp	I1	–128	127	50	°C
Advanced Charging Algorithm	Temperature Ranges	T4 Temp	I1	–128	127	60	°C

2.2 Charge Voltage Configuration

Determine *Temp Charging: Voltage*

The *ChargingVotage( )* is determined based on *Temperature Ranges*.

Set the temperature values as follows:

- *Low Temp Charging: Voltage* = 4150 mV
- *Standard Temp Low Charging: Voltage* = 4350 mV
- *Rec Temp Charging: Voltage* = 4350 mV
- *Standard Temp High Charging: Voltage* = 4300 mV
- *High Temp Charging: Voltage* = 4200 mV

Table 2. Voltage

Class	Subclass	Name	Type	Min	Max	Value	Unit
Advanced Charging Algorithm	Low Temp Charging	Voltage	I2	0	32767	4150	mV
Advanced Charging Algorithm	Standard Temp Low Charging	Voltage	I2	0	32767	4350	mV
Advanced Charging Algorithm	Rec Temp Charging	Voltage	I2	0	32767	4350	mV
Advanced Charging Algorithm	Standard Temp High Charging	Voltage	I2	0	32767	4300	mV
Advanced Charging Algorithm	High Temp Charging	Voltage	I2	0	32767	4200	mV

The *ChargingVotage( )* changes depending on the detected *Temperature( )* and Cell numbers configured in *DA Configuration[CC1:CC0] bits combination*.

For example, if [CC1:CC0] is set to 1,0 which is 3 cells configuration,*ChargingVotage( )* = X *Temp charging:Voltage* x 3.

2.3 Charge Current Configuration

The *ChargingCurrent( )* is determined based on *Voltage Range* and *Temperature Ranges*. So, *ChargingCurrent( )* is set according to *Voltage Range* at each Temperature Range.

The *ChargingCurrent( )* is set based on temperature only regardless of battery voltage level.

If *ChargeCurrent( )* is set regardless of battery voltage level, *Voltage Range* is not important except for *Precharge Start Voltage* and *Charging Voltage Low* in normal charging.

Use the following conditions to set Voltage Range:

- CVL ≤ CVM ≤ CVH

Precharge Start Voltage is minimum cell voltage to enter Precharge mode from Fast Charge mode. The gauge enters Precharge mode if Min Cell Voltage 1..4 < a *Precharge Start Voltage*. *Charging Voltage Low* is maximum Precharge Voltage range before entering Fast charge mode from Precharge mode. The gauge enters Fast Charge mode if Max Cell Voltage 1..4 > *Charging Voltage Low*.

Determine *Temp Charging: Voltage* and X *Temp Charging: Current Low, Current Med and Current High* at each *Temperature Ranges*.

2.4 Voltage Range Configuration

Determine the Voltage Range:

- *Precharge Start Voltage* = 2500 mV
- *Charging Voltage Low* = 3000 mV
- *Charging Voltage Med* = 3600 mV
- *Charging Voltage High* = 4000 mV

Table 3. Voltage Range

Class	Subclass	Name	Type	Min	Max	Value	Unit
Advanced Charging Algorithm	Voltage Range	Precharge Start Voltage	I2	0	32767	2500	mV
Advanced Charging Algorithm	Voltage Range	Charging Voltage Low	I2	0	32767	3000	mV
Advanced Charging Algorithm	Voltage Range	Charging Voltage Med	I2	0	32767	3600	mV
Advanced Charging Algorithm	Voltage Range	Charging Voltage High	I2	0	32767	4000	mV

2.5 Set X Temp Charging

Current Low, Med and High at each *Temp Charging* as followings:

- *Low Temp Charging: Current Low* = 2000 mA,
- *Low Temp Charging: Current Med* = 2000 mA,
- *Low Temp Charging: Current High* = 2000 mA
- *Standard Temp Low Charging: Current Low* = 4000 mA,
- *Standard Temp Low Charging: Current Med* = 4000 mA
- *Standard Temp Low Charging: Current High* = 4000 mA
- *Rec Temp Charging: Current Low* = 4000 mA
- *Rec Temp Charging: Current Med* = 4000 mA
- *Rec Temp Charging: Current High* = 4000 mA
- *Standard Temp High Charging: Current Low* = 4000 mA,
- *Standard Temp High Charging: Current Med* = 4000 mA
- *Standard Temp High Charging: Current High* = 4000 mA
- *High Temp Charging: Current Low* = 1200 mA,

- *High Temp Charging: Current Med* = 1200 mA
- *High Temp Charging: Current High* = 1200 mA

Table 4. Temp Charging

Class	Subclass	Name	Type	Min	Max	Value	Unit
Advanced Charging Algorithm	Low Temp Charging	Current Low	I2	0	32767	2000	mA
Advanced Charging Algorithm	Low Temp Charging	Current Med	I2	0	32767	2000	mA
Advanced Charging Algorithm	Low Temp Charging	Current High	I2	0	32767	2000	mA
Advanced Charging Algorithm	Standard Temp Low Charging	Current Low	I2	0	32767	2000	mA
Advanced Charging Algorithm	Standard Temp Low Charging	Current Med	I2	0	32767	4000	mA
Advanced Charging Algorithm	Standard Temp Low Charging	Current High	I2	0	32767	4000	mA
Advanced Charging Algorithm	Rec Temp Charging	Current Low	I2	0	32767	4000	mA
Advanced Charging Algorithm	Rec Temp Charging	Current Med	I2	0	32767	4000	mA
Advanced Charging Algorithm	Rec Temp Charging	Current High	I2	0	32767	4000	mA
Advanced Charging Algorithm	Standard Temp High Charging	Current Low	I2	0	32767	4000	mA
Advanced Charging Algorithm	Standard Temp High Charging	Current Med	I2	0	32767	4000	mA
Advanced Charging Algorithm	Standard Temp High Charging	Current High	I2	0	32767	4000	mA
Advanced Charging Algorithm	High Temp Charging	Current Low	I2	0	32767	1200	mA
Advanced Charging Algorithm	High Temp Charging	Current Med	I2	0	32767	1200	mA
Advanced Charging Algorithm	High Temp Charging	Current High	I2	0	32767	1200	mA

2.6 *SoC Based* ChargingCurrent( ) *Configuration*

When using *specific SoC ranges* instead of *voltages to determine the charging current*, then the *Charging Configuration[SOC_CHARGE]* bit should be set to "1". The voltage thresholds is replaced by SoC thresholds.

The *voltage range* is still important since transitions happen if both the voltage range and *Charging SoC : Mid and High* conditions are met.

The transition happens at the following condition:

a. [LV] state and *RelativeStateOfCharge( ) > Charging SoC Mid*; move to [MV].
b. [MV] state and *RelativeStateOfCharge( ) > Charging SoC High*; move to [HV].

Table 5. SOC Range

Class	Subclass	Name	Type	Min Value	Max Value	Default Value	Unit
Advanced Charge Algorithm	SOC Range	Charging SOC Mid	U1	0	100	50	%
Advanced Charge Algorithm	SOC Range	Charging SOC High	U1	0	100	75	%
Advanced Charge Algorithm	SOC Range	Charging SOC Hysteresis	U1	0	100	1	%

2.7 Charging Current

If the *ChargingCurrent( )* requires to be segmented based on *Voltage Range at Section 2.4*, it can be set by X *Temp Charging: Current Low, Current Med* and *Current High* value each *Temperature Ranges*. Here is the example to segment the value at Low Temp range in Figure 2. This method is the same for other temperature ranges.

Table 6. Charging Current

Class	Subclass	Name	Type	Min	Max	Value	Unit
Advanced Charging Algorithm	PCHG	Current	I2	0	32767	200	mA
Advanced Charging Algorithm	Low Temp Charging	Current Low	I2	0	32767	1000	mA
Advanced Charging Algorithm	Low Temp Charging	Current Med	I2	0	32767	1500	mA
Advanced Charging Algorithm	Low Temp Charging	Current High	I2	0	32767	2000	mA

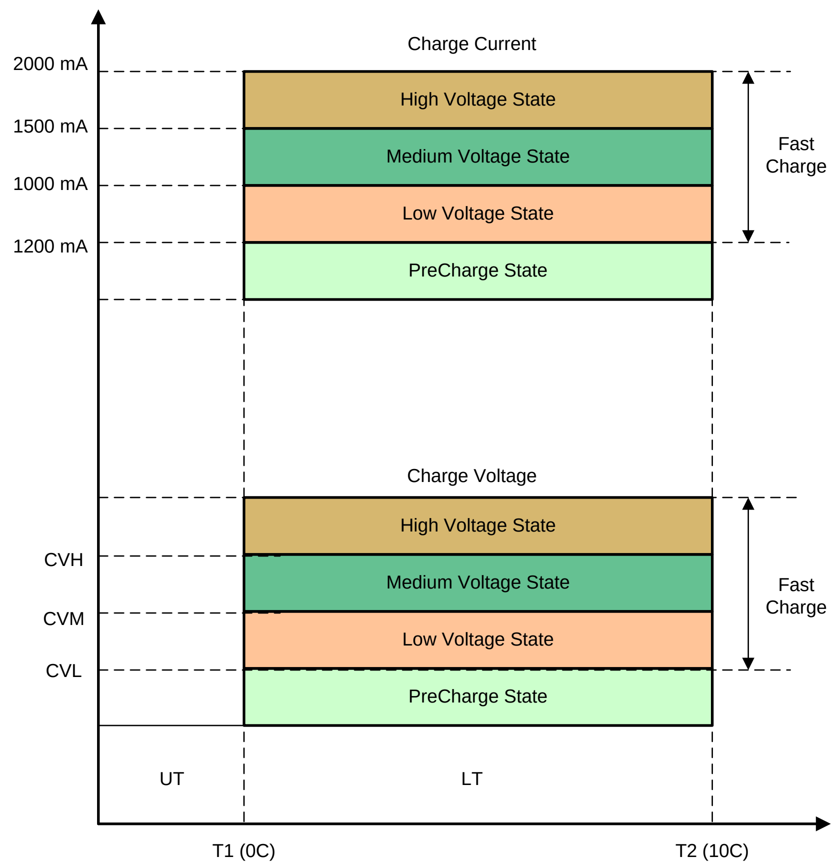

Figure 2. Charge Current vs Voltage Range

3 Charging Loss compensation

To speed up charging cycle, it is ideal to stay in the Constant Current mode as long as possible before transitioning to Constant Voltage. In the pack, many elements like the sense resistor, CHG/DSG FETs, as well as the layout contribute to a sometimes significant IR drop between Pack and Cell. This can force the charger to move to Constant Voltage mode early which extends the charging time since the charger transitions to CV mode based on its output regulation point (Charger output Voltage). Therefore if the gauge can provide an IR compensated *ChargingVoltage( )* to the system, it would shorten charge time.

The gauge can read the cell voltage very accurately by measuring the cell voltages directly. So, the gauge can compensate for IR losses caused by the elements mentioned above and minimize charge time by providing IR compensated *ChargingVoltage( )* if the *Configuration[CCC]* bit is set to "1".

When *Voltage( )* < Charging Algorithm Voltage and *Current( )* > *CCC Current Threshold*. *ChargingVoltage( )* is increased with accounting for IR drop between Pack Voltage and Cell Voltage. So, *ChargingVoltage( )* reports Original *ChargingVoltage( )* + *(PackVoltage( ) - Voltage( ))*. The system can increase charger output voltage based on compensated *ChargingVoltage( )* information.

ChargingVoltage( ) is clamped to the Charging Algorithm + *CCC Voltage Threshold* level for the maximum safety not to overcharge the cell.

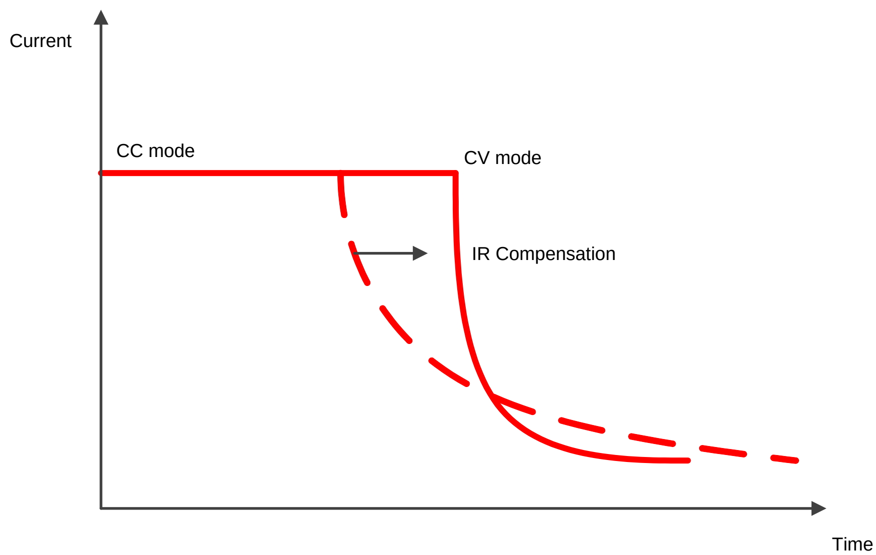

Figure 3. IR Compensation

4 Battery Degradation

Lithium-ion battery capacity is degraded and impedance is increased as battery cell ages. If Charge Voltage and Charge Current are adjusted as battery ages, it helps to increase battery longevity. According to the Study, the battery voltage is the most important factor to speed up the degradation. The gauge provides the reduced *ChargingVoltage( )* and *ChargingCurrent( )* information based on either *CycleCountt( )* or *SOH( )* . Only either *CycleCount( )* or *SOH( )* can be used at one time.

Degraded *ChargingVoltage( )* is enabled when *ChargingConfiguration[CYCLE_DEGRADE]* or *ChargingConfiguration[SOH_DEGRADE]* bit is set. Degraded *ChargingCurrent( )* is enabled when *ChargingConfiguration[DEGRADE_CC]* bit is set.

Cycle Count or SOH Threshold are configured at *Degrade Mode 1,2,3*. Charge Voltage Degradation level and Charge Current Degradation level are also configured at *Degrade Mode 1,2,3*.

For example, if it requires to reduce the *ChargingVoltage( )* by 50 mV after 200 cycles, set:

- *Degrade Mode 1 : Cycle Threshold* = 200
- *Degrade Mode 1: Voltage Degradation* = 50
- *Degrade Mode 1: Current Degradation* = 0

After that, if it is required to reduce the *ChargingVoltage( )* by 100 mV and *ChargingCurrent( )* by 10 % after 300 cycles, set:

- *Degrade Mode 2: Cycle Threshold* = 300
- *Degrade Mode 1: Voltage Degradation* = 50
- *Degrade Mode 2: Current Degradation* = 10 since *Degrade Mode 2: Voltage Degradation*
- *Degrade Mode 2: Current Degradation* are accumulated value from *Degrade Mode 1: Voltage Degradation*
- *Degrade Mode 1 : Current Degradation*

Table 7. Degrade Mode

Class	Subclass	Name	Type	Min	Max	Value	Unit
Advanced Charging Algorithm	Degrade Mode 1	Cycle Threshold	U2	0	65535	200	—
Advanced Charging Algorithm	Degrade Mode 1	Voltage Degradation	I2	0	32767	50	mV
Advanced Charging Algorithm	Degrade Mode 1	Current Degradation	I2	0	100	0	%
Advanced Charging Algorithm	Degrade Mode 2	Cycle Threshold	U2	0	65535	300	—
Advanced Charging Algorithm	Degrade Mode 2	Voltage Degradation	I2	0	32767	50	mV
Advanced Charging Algorithm	Degrade Mode 2	Current Degradation	I2	0	100	10	%

5 SMBus Broadcast

The gauge supports SMBus Broadcast mode which makes the gauge as a master on the bus and broadcasts the information to the System Host or Smart Battery Charger as a master. It gives the benefit that the host doesn't need polling the registers to read *ChargingVoltage( )* and *ChargingCurrent( )* since the gauge sends the information periodically. An SMBus master can only start a packet if the SMBus has been idle for more than 50 µs. Once this requirement has been met, the master immediately takes control of the bus by sending a start bit. If another master controls the bus, the firmware-controlled bus does not detect that the bus is no longer busy, which causes arbitration to be lost. Make sure that CPU does not have a problem with multi-master SMBus to use Broadcast function.

When *BatteryMode[CHGM]* bit is "0", it enables *ChargingVoltage( )* and *ChargingCurrent( )* to host an smart battery charger. When *BatteryMode[AM]* bit is "0", it enables *AlarmWarning( )* broadcasts to host an smart battery charger. If *SBS configuration[HPE]* bit is set, Master mode broadcasts to the System host address are PEC enabled. If the *SBS configuration[CPE]* bit is set, Master mode broadcasts to the System Charger(0x12). If the *SBS configuration[BCAST]* bit is set, Master mode is enabled and the gauge broadcasts *ChargingVoltage( )*, *ChargingCurrent( )* and *AlarmWarning( )* to the Smart Charger (0x12) and the host.(0x10).

There are two main advantages in using the SMBus Broadcast function:

1. It provides the battery with all power it can handle without overcharging – Maximum Safe Charge
2. It correctly charges batteries with different chemistries and voltages.

The *AlarmWarning( )* broadcast can be used to warn the System Host or Smart Battery Charger of potentially dangerous situations that are verify by reading the status registers.

bq40z50 supports PEC (Packet Error Checking) to enhance data integrity during SMBus communication. PEC is an extra byte of data added to the end of the communication packet that is derived from a simple CRC-8 checksum. In read operations, the bq40z50 is responsible for sending the PEC packet to the host which determines if the PEC is valid. In write operations, the host is responsible for sending the PEC to the slave which determines if the PEC is valid

Make sure CPU does not have a problem with multi-master SMBus before enabling the SMBus Broadcast feature. Many CPUs do not behave properly if another SMBus master is on the same bus.

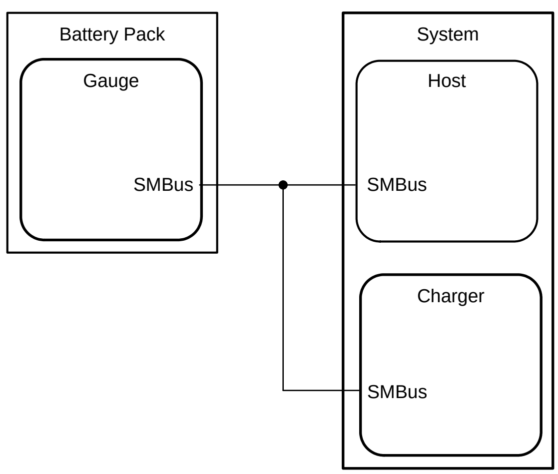

Figure 4. SMBus Connection

6 Acronym

UT: Under Temperature

LT: Low Temperature

STL: Standard Low Temperature

RT: Recommended Temperature

STH: Standard High Temperature

HT: High Temperature

OT: Over Temperature

CVL: Charging Voltage Low

CVM: Charging Voltage Med

CVH: Charging Voltage High

CCL: Charging Current Low

CCM: Charging Current Med

CCH: Charging Current High

FCC: Full Charge Capacity

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