Qualcomm Technologies, Inc.

Understanding PM8150B Fuel Guage, ADC, and Battery Interface Modules

Application Note

80-PD996-14 Rev. A

July 30, 2018

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	July 2018	Initial release

Contents

Figures

Tables

1 Introduction

1.1 Fuel Gauge, ADC, and BIM Features

- Fuel gauge module:
  - Hardware autonomous with minimal software interaction required
  - State of charge (SoC) estimation is done by means of a mixed-mode algorithm using voltage and current information provided by ADC module
  - Dedicated coulomb counter
  - In-system battery resistance measurements
    - Used to calibrate comprehensive impedance model
    - Aids in accurate SoC tracking over temperature and battery age
  - Supports multiple battery profiles in software
  - Supports current sensing via internal battery FET
  - Configurable 100% and 0% SoC points, based on current and voltage, respectively
- ADC module:
  - 16-bit dedicated current ADC
    - Trimmed in ATE
    - Periodically calibrated in hardware with no software assistance
  - Uses ADC module's 16-bit dedicated voltage ADC
    - Trimmed in ATE
    - Periodically calibrated in hardware with no software assistance
- Battery interface module (BIM)
  - Battery missing detection on BATT_THERM and BATT_ID pins

1.2 General description

The purpose of this document is to provide detailed information on how the fuel gauge, ADC, and BIM modules work together to provide a robust battery monitoring solution, with a primary focus on fuel gauging. The fuel gauge offers a hardware-based algorithm that can accurately estimate the SoC by mixing both current and voltage monitoring techniques. This ensures excellent short-term linearity and long-term accuracy. Furthermore, zero current load conditions are not required to maintain accuracy due to the in-system equivalent series resistance (ESR) measurements and impedance modeling.

Using precise measurements of voltage, current, temperature provided by the ADC module, an accurate SoC is delivered over a broad range of operating conditions. Long term accuracy is achieved via a complex compensation algorithm that uses battery temperature, modeled and measured battery impedance, and a capacity learning algorithm.

Note: New to PM8150B, the fuel gauge, ADC's, and battery missing detection functionality have been broken into separate modules. This was done to simplify internal designs and make it easier to mix and match IP between PMIC's. This will have little to no impact on our OEM's but will affect how things are documented and explained.

1.3 Acronyms

Table 1-1 Acronyms

Acronyms	Description
ADC	Analog-to-digital converter
BMD	Battery missing detection
CV	Constant voltage
EOC	End of charge
ESR	Equivalent series resistance
FCC	Fast charge current
FG	Fuel gauge
FV	Float voltage – chargers constant voltage charging set point
IMA	Interleaved memory access
OCV	Open circuit voltage
SoC	State of charge
DMA	Direct memory access
BIM	Battery Interface Module
AKA	"Also Known As"

1.4 Fuel gauge, ADC, and BIM pin descriptions

See *PM8150B Power Management Device Specification* (80-PD996-1) for pin descriptions.

1.5 Generation 4 fuel gauge (PM8150B) vs. previous generation 3 fuel gauge (PMI8998)

Table 1-2 Comparison of fuel gauge, generations 3 and 4

Feature	Generation 4 FG, ADC, and BIM module (PM8150B)	Generation 3 Fuel gauge module (PMI8998)
Hardware differences		
Current sensing	▪ Internal current sensing POR ▪ Current sensing up to ~10 A	▪ Internal current sensing POR ▪ External current sensing **not supported** ▪ Internal current sensing POR for parallel charging ▪ Current sensing up to 8.5 A
Update Cycle	▪ 1 s	▪ 1.47 s
Fuel gauge memory	▪ DMA used for fuel gauge memory access	▪ IMA used for fuel gauge memory access
ADCs	▪ ADC's shared between all modules	▪ Dedicated VADC for fuel gauging and BCL ▪ Dedicated IADC for fuel gauging and BCL ▪ Round-robin ADC for housekeeping on PMI
Housekeeping functions	▪ VADC now used instead of dedicated RRADC.	▪ Round robin ADC used for housekeeping (RRADC)
Battery current limiting	▪ Functionality still exists within PM8150B, but module separated from FG and no longer covered in this document ▪ See *PM8150B Design Guidelines/Training Slides* (80-PD996-5) for more information.	▪ Voltage and current measurement timing depends on two power levels ▪ Algorithm primarily controlled by hardware ▪ Mitigation done autonomously in hardware by limits management
Algorithm		
ESR measurement timing		
Discharge – pulse timing	▪ Every ~96 s if qualifying transient does not occur	▪ Every ~145.5 s if qualifying transient does not occur
Charging – pulse timing	▪ Every ~96 s if qualifying transient does not occur	▪ Every ~28 s if qualifying transient does not occur
ESR Modeling	▪ ESR modeling added to this generation of FG ▪ Model calibrated periodically during charging ▪ Improves ESR tracking and reliability, especially at lower temperatures ▪ ESR modeled by six 3rd order polynomials which models ESR over SoC and temperature. ▫ 3 for charge and 3 for discharge.	▪ No ESR modeling done on PMI8998 or any previous generation of QC hardware Fuel Gauges
Rslow mapping	▪ Three 2nd order polynomials for Rslow modeling at low SoC that captures impedance trend change ▪ Piecewise linear modeling of Rslow over SoC and temperature which captures all bumps and changes in Rslow ▪ Improved Rslow modeling for low temperatures and for entire SoC range	▪ One 2nd order polynomial for Rslow modeling at low SoC ▪ Two 2nd order polynomials for modelling Rslow over temperature; for charge and discharge separately

Feature	Generation 4 FG, ADC, and BIM module (PM8150B)	Generation 3 Fuel gauge module (PMI8998)
OCV Modeling	▪ Three 3rd order polynomial models for OCV vs. SoC for cold, room, and warm temperatures; for both charge and discharge separately ▪ Improved OCV accuracy over temperature	▪ Two 3rd order polynomial models for OCV vs. SoC for room temperature; both charge and discharge separately ▪ Entire OCV curve shifted based on 3rd order OCV to temp model; for both charge and discharge separately
Temperature Modeling	▪ 5th order polynomial temperature model ▪ Improved temperature measurement accuracy, especially at low temperatures	▪ 3rd order polynomial model
New software features		
TBD	TBD	TBD

1.6 Tasks and configurations needed for fuel gauge commercialization

The fuel gauge is a powerful and accurate tool for determining state of charge while still being simple to use, and only requiring a small number of configurations. This section is a guide for customers who are designing for and using the fuel gauge and supplies all of the necessary configurations to ensure optimal fuel gauge performance.

Note: Please note this a quick and simple guide. Please see the sections following this guide for more in depth information on these topics.

1.6.1 Pre-board design stage

1.6.1.1 Battery pack design

Thermistor selection and placement

- Ensure battery vendor does not place any capacitance internal to the cell on the NTC or RID nets.
- Supported NTC resistance range: 10 k–100 k.
  - This is the resistance of the thermistor at 25°C.
- Supported NTC beta value range: 3200–4400.
  - When choosing a thermistor, QTI recommends reviewing Figure 2-2 and Figure 2-3 for simulated temperature measurement accuracy vs. beta value.
    - Note: This table is not a specification, and only an example based on simulation, but should track closely to real life accuracy.
    - See *PM8150B Power Management Device Specification* (80-PD996-1), for more information about temperature accuracy related specifications.
- It is strongly recommended that the NTC be placed on the PCM of the battery pack.
  - Avoid placing the NTC on the board, even if it is near the cell as this will impact temperature tracking and SoC tracking performance.

    - Ensure that the NTC is not placed near, on top, or below, the IBATT current-carrying trace of the PCM.
        - Not following this recommendation will cause poor measurement accuracy during large transient conditions where the trace heats up faster than the cell, resulting in poor SoC tracking behavior.
            - Most commonly seen during beginning of charging or a large discharge.

Thermistor measurement delay configurations

- Some customers knowingly, or many times unknowingly, place small amounts of capacitance on the thermistor net, typically within the PCM of the pack.
    - Placing capacitance on the BATT_THERM net is NOT supported by QTI, and not validated internally to work with the provided thermistor measurement delay.
        - This is only available as a last resort for OEM’s that fail to follow our guidelines.
        - The delay must be set to ensure that the sampling on BATT_THERM is settled before the ADC samples, or inaccurate (larger than real) temperature will be measured
    - Currently NO delay is supported on PM8150B ES parts.
- For hardware information about this setting, see Section 2.1.3.2.6.
- For information about how to configure this in software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

Beta Coefficients

- The fuel gauge models battery temperature using a 5th order interpolation of Steinhart-Hart.
- To align the hardware model with the physical model the 5th order polynomial coefficients of the thermistor must be programmed.
    - Failing to do this will result in poor temperature accuracy.
    - These coefficients are provided within the battery profile as a part of characterization process but can be requested via Salesforce if needed.
- For more information about how to configure this in software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

RID selection

- Ensure battery vendor does not place any capacitance internal to the cell on the NTC or RID nets.
- Supported readable RID range: 1 k–450 k.
    - An RID value must be chosen within this range.
    - It is recommended that separate RID values be chosen for cells from different battery vendors, so individual profiles can be stored in software that are custom specifically to that cell.

- Fake battery detection happens 6.75 kohm–8.25 kohm, so a production ID must be outside of this range.
  - This software feature prevents charging from taking place when a ~7.5 kohm resistor is detected.
  - This range is programmable; for more information about programming this range, see *SM8150 PMIC Fuel Gauge Software User Guide* (80-PF777-74).

Table 1-3 Battery pack recommendations

Min. overcurrent protection threshold for charge and discharge (OCP)	Min. OCP delay time for charge and discharge	Max. undervoltage detection threshold for discharge (UVD)	Min. UVD delay time for discharge
8 A	10 ms	2.4 V	10 ms

NOTE: For proper battery current limiting (BCL) functionality, the minimum OCP and minimum UVD delay timing specifications must be met.

1.6.2 Board design stage

1.6.2.1 Layout

- Layout is critical for both basic functionality and optimal performance.
- Serious issues can arise when the layout guidelines are not followed; pay special attention to Fuel Gauge and ADC layout guidelines.
- For the layout guidelines, see the last section of *PM8150B Design Guidelines* (80-PD996-5).

1.6.2.2 Battery profile

- A custom battery profile is required when using the fuel gauge for SoC tracking.
  - Generic profiles are not supported.
- For more information about obtaining a battery profile, see Section 2.2.3.
- It is recommended that each cell vendor have its own RID value and custom profile.
- Allow a minimum of four weeks to receive the battery profile from the date of delivery of the battery to QTI.

1.6.3 Post-board design – required fuel gauge configurations

1.6.3.1 100% and 0% SoC endpoint matching related configurations

VBATT full

- This is one of the two configurable inputs to the SoC full estimate loop, which affects the 100% SoC prediction algorithm.

- This must be set to 10 mV less than the float voltage setting of the charger.
  - For example, for a 4.4 V battery, this setting should be 4.39 V.
  - If this is not set correctly, 100% SoC may not be reached.
- For more hardware information about this setting, see Section 2.2.2.3.
- For information about how to configure this in software, see *SM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

IBATT full

- This is one of the two configurable inputs to the SoC full loop, which affects the 100% SoC prediction algorithm.
- IBATT full must be larger in magnitude (more negative) than the charger's termination current so that IBATT full is reached before the charger's termination.
  - For example, if the charger will terminate charge at -100 mA, IBATT full must be set to < -100 mA, preferably by more than 50mA.
  - In a common example, charger termination current = -150 mA; IBATT full = -200 mA.
- Keep in mind that when the battery current is negative, it denotes current flowing into the battery.
- This must be configured with a negative sign within the board file in software.
- This is also referred to as the system termination current.
- For hardware information about this setting, see Section 2.2.2.3.
- For information on how to configure this in software, see PM8150B *PMIC Fuel Gauge Software User Guide* (80-PF777-74).

VBATT cutoff

- This is the only configurable input for the SoC empty loop, which affects the 0% SoC prediction algorithm.
- This provides a voltage-based target for the SoC empty loop.
- This setting depends on the customer's specific design, requirements, and should be validated extensively with BCL to ensure expected operation.
- This setting is also referred to as the cutoff voltage threshold.
- For more hardware information about this setting, see Section 2.2.2.3.
- For information about how to configure this in software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

Automatic recharge

- Two types of recharge schemes are offered:
  - SoC based recharge
  - Battery voltage based recharge
- This depends on the customer's requirements and is free to choose either.

- For information about how to configure this in software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

Monotonic slope limiting configuration

- In hardware, the fuel gauge can limit the magnitude of change of the monotonic SoC between every fuel gauge update cycle (1 s).
- For hardware information about this setting, see Section 2.2.2.4.1.
- For information about how to configure this in software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

Capacity learning and cycle counting

- In software, the fuel gauge driver can perform capacity learning and cycle counting.
  - QTI recommends that all customers leave this feature enabled.
  - Learned capacity is fed back into Fuel Gauge hardware to be used by the algorithm.
- For information about how to configure this in software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

2 Detailed module information

2.1 Fuel gauge, ADC, BIM hardware functions

2.1.1 ADC current sensing

The battery current is read every FG cycle (1 s) by a 16 bit dedicated delta-sigma current ADC (IADC). The IADC measures current using the internal battery FET (BATFET) as the sense element. Current is reported to the Fuel Gauge and used for SoC tracking purposes, as well as reported to any other module which may request it. When read via the FG memory the current reads as positive when discharging and as negative when charging. The battery current and voltage are read synchronously by the ADC module so the FG module can use these readings to calculate battery impedance. During sampling, the ADC conversion time will take 160 ms to complete.

NOTE: Current is reported as positive when leaving the battery, and negative when entering the battery,

2.1.2 ADC voltage sensing

The battery voltage is read every FG cycle (1 s) by a 16 bit dedicated delta-sigma current ADC (VADC). The VADC measures voltage using the dedicated VBATT_SNS_P and VBATT_SNS_N differential sense pins. Voltage is reported to the Fuel Gauge and used for SoC tracking purposes, as well as reported to any other module which may request it. The battery current and voltage are read synchronously by the ADC module so the FG module can use these readings to calculate battery impedance. During sampling, the ADC conversion time will take 160 ms to complete.

2.1.3 PM8150B house keeping

2.1.3.1 Overview

The voltage and current ADC's within the ADC module are used for housekeeping purposes. The list of ADC channels, divide by ratios, and accuracy specifications, are provided within the *PM8150B Power Management Device Specification* (80-PD996-1) and will not be covered in this application note.

2.1.3.2 Battery temperature sensing and modeling

2.1.3.2.1 Overview

The ADC module measures the battery's temperature via the BATT_THERM pin using the VADC. The voltage information is provided to the Fuel Gauge module and used for temperature modeling and provides an output temperature. This information is used by the Fuel Gauge by its many different temperature models.

2.1.3.2.2 Thermistor selection

PM8150B supports 10 k–100 k, 3200–4400 beta thermistors. When selecting a thermistor it is recommended to review Figure 2-2 and Figure 2-3. This table provides example simulated worst case accuracy data of what can be expected when selecting a thermistor.

NOTE: Figure 2-1, Figure 2-2, and Figure 2-3 are NOT official specifications, and only to be used as reference examples based on simulation data. For temperature measurement specifications, see *PM8150B Power Management Device Specification* (80-PD996-1).

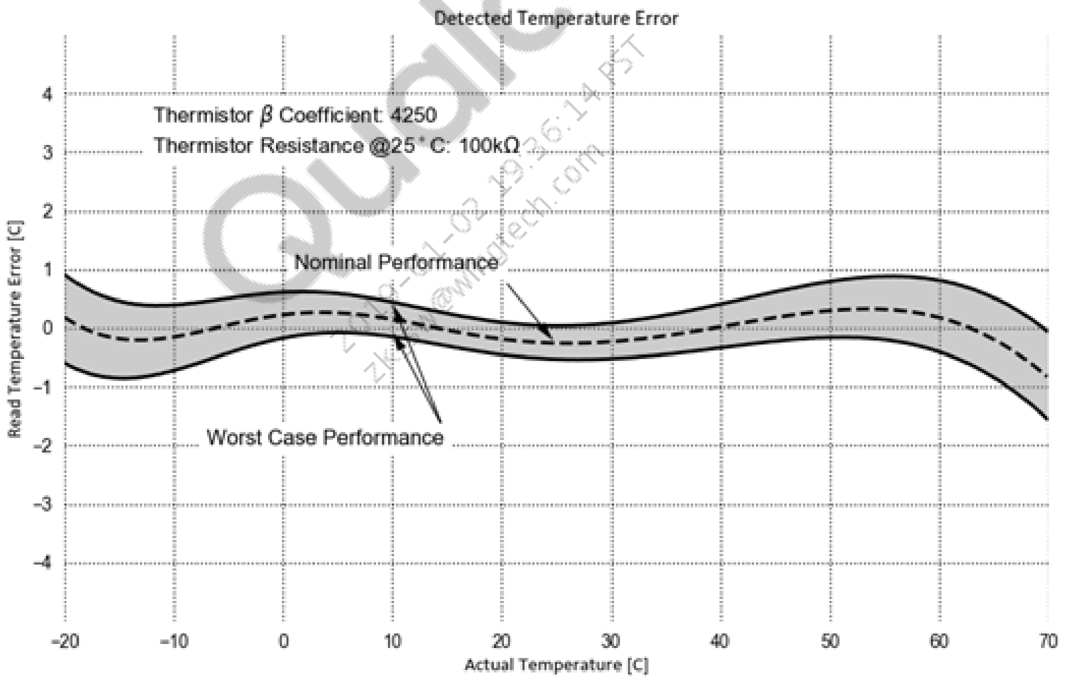
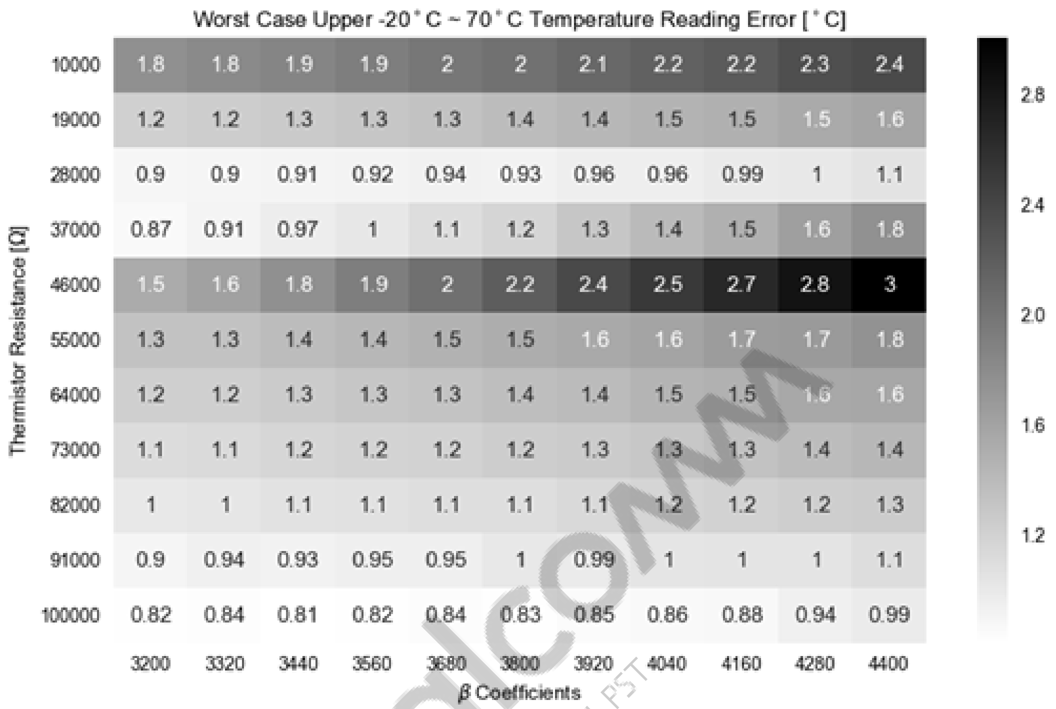
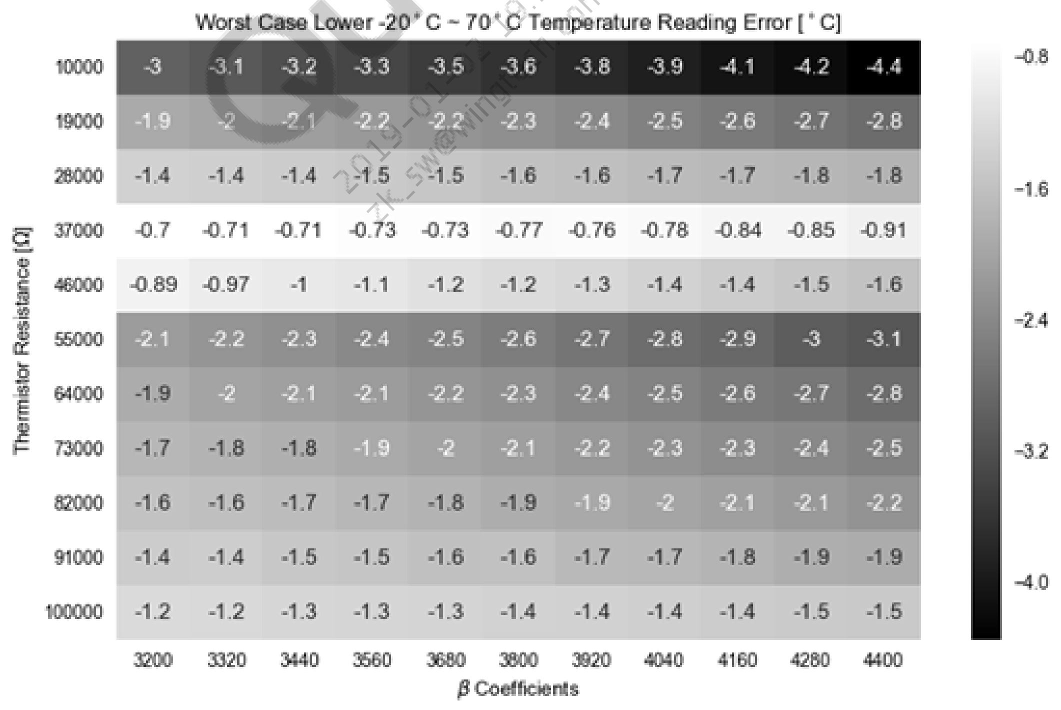


Figure 2-1 Default 100 kohm 4250 beta thermistor



Thermistor Resistance [Ω]	3200	3320	3440	3560	3680	3800	3920	4040	4160	4280	4400
10000	1.8	1.8	1.9	1.9	2	2	2.1	2.2	2.2	2.3	2.4
19000	1.2	1.2	1.3	1.3	1.3	1.4	1.4	1.5	1.5	1.5	1.6
28000	0.9	0.9	0.91	0.92	0.94	0.93	0.96	0.96	0.99	1	1.1
37000	0.87	0.91	0.97	1	1.1	1.2	1.3	1.4	1.5	1.6	1.8
46000	1.5	1.6	1.8	1.9	2	2.2	2.4	2.5	2.7	2.8	3
55000	1.3	1.3	1.4	1.4	1.5	1.5	1.6	1.6	1.7	1.7	1.8
64000	1.2	1.2	1.3	1.3	1.3	1.4	1.4	1.5	1.5	1.6	1.6
73000	1.1	1.1	1.2	1.2	1.2	1.2	1.3	1.3	1.3	1.4	1.4
82000	1	1	1.1	1.1	1.1	1.1	1.1	1.2	1.2	1.2	1.3
91000	0.9	0.94	0.93	0.95	0.95	1	0.99	1	1	1	1.1
100000	0.82	0.84	0.81	0.82	0.84	0.83	0.85	0.86	0.88	0.94	0.99

Figure 2-2 Worst case positive temperature error



Thermistor Resistance [Ω]	3200	3320	3440	3560	3680	3800	3920	4040	4160	4280	4400
10000	-3	-3.1	-3.2	-3.3	-3.5	-3.6	-3.8	-3.9	-4.1	-4.2	-4.4
19000	-1.9	-2	-2.1	-2.2	-2.2	-2.3	-2.4	-2.5	-2.6	-2.7	-2.8
28000	-1.4	-1.4	-1.4	-1.5	-1.5	-1.6	-1.6	-1.7	-1.7	-1.8	-1.8
37000	-0.7	-0.71	-0.71	-0.73	-0.73	-0.77	-0.76	-0.78	-0.84	-0.85	-0.91
46000	-0.89	-0.97	-1	-1.1	-1.2	-1.2	-1.3	-1.4	-1.4	-1.5	-1.6
55000	-2.1	-2.2	-2.3	-2.4	-2.5	-2.6	-2.7	-2.8	-2.9	-3	-3.1
64000	-1.9	-2	-2.1	-2.1	-2.2	-2.3	-2.4	-2.5	-2.6	-2.7	-2.8
73000	-1.7	-1.8	-1.8	-1.9	-2	-2.1	-2.2	-2.3	-2.3	-2.4	-2.5
82000	-1.6	-1.6	-1.7	-1.7	-1.8	-1.9	-1.9	-2	-2.1	-2.1	-2.2
91000	-1.4	-1.4	-1.5	-1.5	-1.6	-1.6	-1.7	-1.7	-1.8	-1.9	-1.9
100000	-1.2	-1.2	-1.3	-1.3	-1.3	-1.4	-1.4	-1.4	-1.4	-1.5	-1.5

Figure 2-3 Worst case negative temperature error

2.1.3.2.3 Thermistor placement and protection circuit

It is strongly recommended the NTC be placed internal to the battery pack. If this is not an option, it is possible to use an external thermistor placed on the board near the battery, but keep in mind that doing so is an added risk, as this may affect the temperature tracking performance of the cell and negatively affect the SoC tracking accuracy.

CAUTION: Ensure battery protection circuit vendor does not place the NTC next to, above, the battery current carrying trace or this will likely cause heat to couple onto the thermistor much faster than into the cell, resulting in inaccuate temperature measurements during high current transient conditions. Battery characterization does not use the thermistor of the cell, and only relies on the temperature of a controlled chamber, so models generated during characterization will not be able to compensate for poor protection circuit layout.

2.1.3.2.4 Thermistor modeling

The fuel gauge uses a 5th order polynomial fit of the Steinhart-Hart thermistor model. To align this model with the NTC chosen “beta coefficients” are programmed into the fuel gauge hardware. These coefficients are the 5th order polynomial fit coefficients and will be provided as a part of the characterization process within the battery profile. These coefficients are optimized for the NTC information provided, and should not be modified by hand.

This is an improvement over previous generations that used a 3rd order polynomial, which had an innate deficiency below 0C.

2.1.3.2.5 Pull up and Timing

The ADC uses a ratiometric conversion when measuring BATT_THERM’s voltage. The advantage of this measurement scheme is that the pull-up that is used to bias the thermistor is the same regulator that supplies the ADC, which allows for a more robust measurement scheme. The ADC has 3 configurable pull up resistors for all channels: 30 k, 100 k, or 400 k. The pull up for BATT_THERM is selected automatically by software using a pre-programmed configuration included within the battery profile. The pull up is chosen during profiling such that the accuracy is optimized for the thermistor within the battery pack. The measurement timing is controlled automatically by hardware. When measured, by default, BATT_THERM is pulled up to 1.875 V and measured every TBD. We do not recommend changing the thermistor measurement timing. The pull up voltage cannot be changed.

The pull up selected is based on the following thresholds:

- 30 k pull up if thermistor is: $R_{@25C} \leqslant$ 54.78 kohm
- 100 k pull up if thermistor is: 54.79 kohm < $R_{@25C} \leqslant$ 200 kohm.
- 200 kohm pull up if thermistor is: $R_{@25C}$ > 200 kohm

2.1.3.2.6 Capacitance on BATT_THERM

Placing any capacitance on the BATT_THERM net, whether on the board, or within the battery pack, is NOT supported and STRONGLY discouraged. We do not verify functionality of BATT_THERM measurements or accuracy with any capacitance on this net, and failure to follow our guidelines may lead to inaccurate temperature measurements (typically a higher offset than real) that do not meet our specifications. The added capacitance slews the application of the pull up on the BATT_THERM net such that it may not have sufficient time to settle to its final value before the ADC conversion begins, resulting in averaging unsettled measurements, lowering the overall measured voltage.

CAUTION: Check with the battery vendor, or within the battery datasheet, to ensure no capacitance has been placed in parallel with the NTC as some vendors may do this without informing their customer.

2.1.3.2.7 Defeaturing BATT_THERM

Qualcomm® cannot support the use of the fuel gauge or charger module if BATT_THERM is not routed to an NTC in or near the pack. The fuel gauge requires temperature measurements for essential operation, and the charger module requires temperature measurements for thermal balancing and JEITA safety limiting.

2.1.3.3 Battery identification

2.1.3.3.1 Overview

Battery identification is done by the ADC directly after battery insertion. During boot software will read and use the RID information to load the corresponding profile.

2.1.3.4 RID selection

- An RID value must be selected within the supported detectable range: 1 kohm to 450 kΩ.
- A 10% tolerance difference is recommended between RID values for multiple cells.
- 7.5 kohm is the nominal resistance for fake battery detection.
  - Fake battery detection is handled in software.
  - Default software range for fake battery detection is 6.75 kohm–8.25 kohm.
  - This value prevents charging for use of debug boards and power supplies.
  - For more information about software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

2.1.3.4.1 Battery identification routine

Detection is done on battery insertion, and subsequently whenever software forces a measurement. Once the detection is resolved, an interrupt is fired (BT_ID, 0x4210) to inform the system about the result. See Table 3-1 for the interrupts available.

Detection is done with the following algorithm:

1. Battery is inserted, or software forces RID read.
2. 400 k pull up is applied, conversion takes place, and voltage information is stored within BATT_ID_HI_BIAS_LSB 0x4266 and BATT_ID_HI_BIAS_MSB 0x4267.

3. 100 k pull up is applied, conversion takes place, and voltage information is stored within BATT_ID_MED_BIAS_LSB 0x426E and BATT_ID_MED_BIAS_MSB 0x426F.
4. 30 k pull up is applied, conversion takes place, and voltage information is stored within BATT_ID_LO_BIAS_LSB 0x4276 and BATT_ID_LO_BIAS_MSB 0x4277.
5. ADC module fires BT_ID 0x4210 interrupt and software choose the voltage value closest to 1.875 V / 2 = 0.9375 V.

2.1.4 Charger interaction (PMI8998 vs. PM8150B)

2.1.4.1 Charge termination – no longer fuel gauge

The Fuel Gauge is no longer involved with charge termination. The charger module intercepts battery current information from the ADC module to make this decision.

2.1.4.2 JEITA – no longer fuel gauge

The Fuel Gauge is no longer involved with JEITA. The charger module intercepts thermistor voltage information from the ADC module to make this decision.

2.1.4.3 Auto recharge – voltage based no longer fuel gauge

The fuel gauge is no longer used as the auto recharge source for voltage, this information is intercepted from the ADC module by the charger module.

SoC based recharge is still available and is controlled by software. For more information about how to configure auto recharge in software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

2.1.5 Battery Interface Module (BIM)

2.1.5.1 Overview

A dedicated module has been created for detecting and standardizing the method of detecting battery removal. Critical features of the BMD module are not only to detect when the battery has been removed, but to do so with low current draw, to deglitch the output value, to store the battery removal in a latched register, and to have the selectable ability to gate-off operation when the VADC is making a measurement. The low current draw is important in sleep/standby and off-mode detection, where each uA of current consumption is critical. The deglitching of the output avoids false detection during events that could potentially break the connection between cell and device; i.e. a fall. Whenever the selected input fails to pull the voltage below the comparator threshold due to the battery being removed and an open condition being present, the NO_DEB_BAT_GONE interrupt (0x01A10 bit 2) immediately fires. When the comparator is set for sufficient deglitch time, the DEB_BAT_GONE interrupt will be set (0x01A10 bit 1).

2.1.5.2 Module inputs

The BMD peripheral selects BATT_ID OR BATT_THERM for monitoring battery presence. BAT_ID is the default input setting. If a battery does not have an RID, BATT_THERM can be selected instead to be used as the BMD source.

2.1.5.3 Measurement timing

The timing of BMD checks depends on the operating state of the PMIC. For normal operation in the active state, BMD is always applied. While in sleep it is measured every 1ms, and while the PMIC is off, it is measured every 500 ms. These timers are configurable, but not recommended to be changed.

2.2 Fuel gauge algorithm

2.2.1 Battery models

2.2.1.1 Overview

The battery resistance is a key parameter in determining a battery's usable charge and health. To model a cells impedance, the battery model shown in Figure 2-4 is used. The total battery resistance can be broken down into two parts:

- Equivalent series resistance (ESR)
  - ESR is the resistance measured immediately when a load is applied, with a large contributor being the metal present in the circuit.
  - This is represented as R1 in the battery model in Figure 2-4.
  - ESR varies over temperature and state of charge.
  - This parameter is modeled during characterization and the model is provided to the FG hardware via the battery profile.
    - This model is periodically calibrated by measuring ESR in real time, in system, to account for cell aging and cell to cell variation.
- Rslow (Also known as Ionic Resistance)
  - Rslow is the resistance to current flow due to various electrochemical factors such as electrolyte conductivity, ion mobility, and electrode surface area.
  - This is represented by R2 and C2 in the battery model in Figure 2-4.
  - This parameter is modeled based on its relationship to ESR over temperature and SoC.

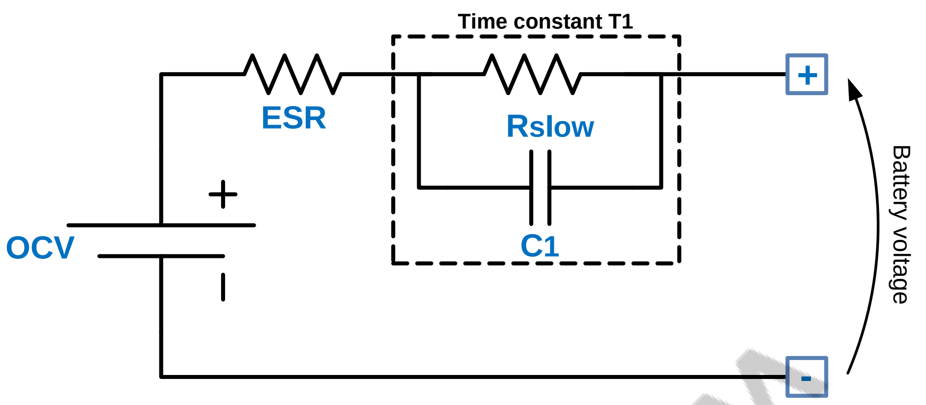


Figure 2-4 Battery model

2.2.1.2 ESR modeling

ESR is modeled in hardware using two-third order polynomial models for ESR vs. SoC; 1 for charge and 1 for discharge. ESR is also modeled and adjusted vs. temperature using two 3rd order models; again 1 for charge and 1 for discharge. This creates a comprehensive model in hardware that the fuel gauge uses to reference ESR based over all use cases. These models are all generated during battery characterization and optimized based on characterization data. These models cannot or tuned by our OEM's, so they will only be covered at a high level. See Figure 2-5 below for a 3-dimensional example of this model.

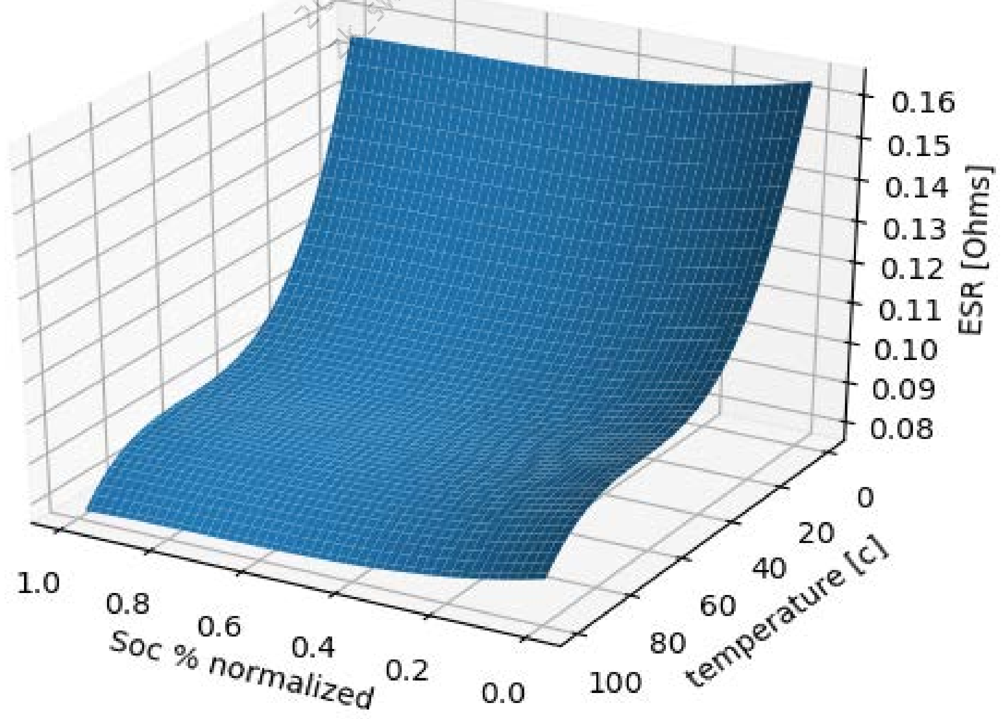


Figure 2-5 ESR Modeling

2.2.1.3 ESR model calibration

The ESR model needs to be calibrated to maintain accuracy due to battery aging, or cell to cell variation. Calibration is done in the 2 following ways:

- **ESR fast calibration:** ESR fast calibration is only used when an initial or very fast alignment of the ESR model is required. This will take place during boot with a newly connected battery, or after a dVdd or deeper reset takes place. This calibration can take place over all temperature and SoC ranges, including discharge or charging. Fast calibration consists of 10 ESR pulses firing 10 seconds apart for a total of 100 seconds.
- **Normal periodic calibration**: During normal operation ESR model calibration will only take place during charging and only within a preprogrammed SoC and temperature window. By default the temperature window is 40°C–10°C, and 20% SoC to 80% SoC. See Figure 2-6 below for an example.

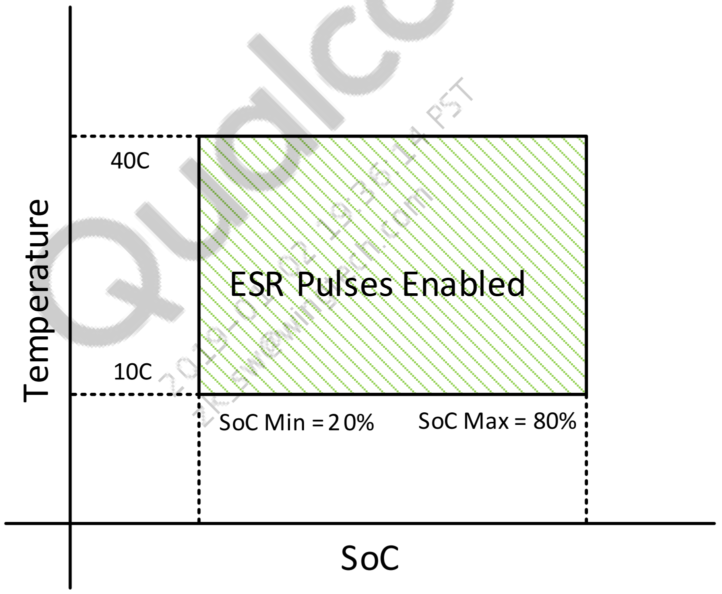


Figure 2-6 ESR calibration window

2.2.1.4 ESR pulse details - Single-path charging:

During single path charging the Fast Charge Current (FCC) of the charger is decremented to create the voltage and current transients necessary to measure ESR. The decrement magnitude for single path charging depends on the table below. All FCC decrements must exceed the qualification threshold of 120mA to qualify for an update to the model. ESR pulses will take place every TBD seconds. If the measurement fails, hardware will automatically retry every TBD seconds until a pulse is successfully qualified. During an ESR pulse the ADC will sample battery voltage and current synchronously for the duration of the pulse, which is roughly ~160ms. This short decrement allows the fuel gauge to obtain the information necessary to estimate ESR with a minimal effect on charge time.

2.2.1.5 ESR pulse details – Parallel charging:

During parallel charging the ESR pulse magnitude is different than during single path charging. The timing remains the same, but instead of using a value from Table 2-1 for the magnitude, the fuel gauge uses a fixed 300 mA FCC setting.

NOTE: This is **NOT a decrement** of 300 mA from the set FCC, but instead the charger **FCC is set to 300 mA.**

Table 2-1 ESR pulses during single-path charging

FCC code used by the charger	Transition point based on the battery current
No ESR pulses	x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	TBD > x > TBD
TBD	x < TBD

2.2.1.6 Rslow modeling

Rslow is modeled by a complex combination models that adjust over temperature and SoC. Too much detail on these models provides no benefit for OEM's as it is not supported or recommended OEM's attempt to modify these models by hand. These models are generated and optimized during battery characterization.

Two improvements to Rslow modeling over previous generation fuel gauge's are worth mentioning:

1. Addition of Rslow scaling model that models changing impedance breakaway knee at cold temperatures and low SoC. This model will help improve SoC performance and reduce SoC bulging at low temperatures and low SoC's. See Figure 2-7 for an example of these improvements.
2. Addition of Rslow shape array, which is adjusted over temperature and SoC. This is a combination of piecewise linear equations that model all dips and bumps in Rslow. This will improve SoC tracking over the entire SoC range, and reduce bulging behavior seen at cold temperatures. See Figure 2-7 for an example of these improvements.

High level breakdown of Rslow is modeling:

- RslowToESR ratio (**Base Scaling Factor**) – ESR to Rslow ratio found during characterization. If no other model was in effect, then Rslow = ESR * RslowtoESR.
- Rslow_Temp(T) – Consists of multiple models that adjust Rslow over temperature. This captures the overall trend of Rslow, as well as how Rslow changes at low and high SoC's, including the changing knee shape at lower temperatures. See Figure 2-7 below for an example.
- Rslow_SoC(SoC,T) – Consists of multiple models that effect Rslow over SoC. This includes a shape array, which is a combination of multiple piecewise linear equations which model all of the bumps and changes in Rslow over SoC and temperature. See Figure 2-7 below for an example.

Each set of models will calculate a scaling factor to adjust the ESR * RslowToESR. This is a simplified form of how Rslow is calculated:

- Rslow = ESR * RslowToESR * Rslow_temp(T) * Rslow_soc(SOC,T)

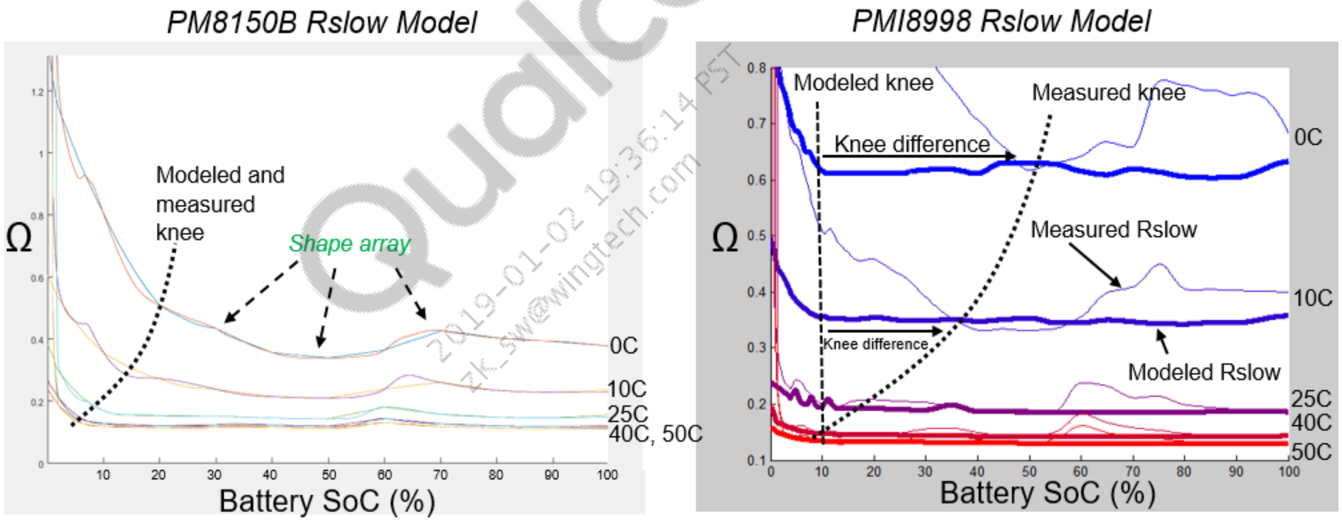


Figure 2-7 Rslow example and difference between PMI8998 and PM8150B

2.2.1.7 OCV modeling

In addition to impedance modeling, the fuel gauge also contains comprehensive models of OCV vs. SoC vs. temperature. This information is used by the voltage mode algorithm which will be covered later. OCV is modeled by using three curves for charge and discharge separately. One curve at 0°C, one at 25°C, and another at 50°C. Any point in between is interpolated. Each curve consists of three conjoined 3rd order polynomials which are generated during characterization and are a part of the battery profile.

This modeling is an improvement over previous hardware fuel gauge generations. Previous generations modeled OCV with one curve for charge and discharge separately, and this curve was shifted upward or downward depending on a separate second order polynomial model. The old model could not consider shape changes of the OCV curve over temperature. See Figure 2-8 for an example.

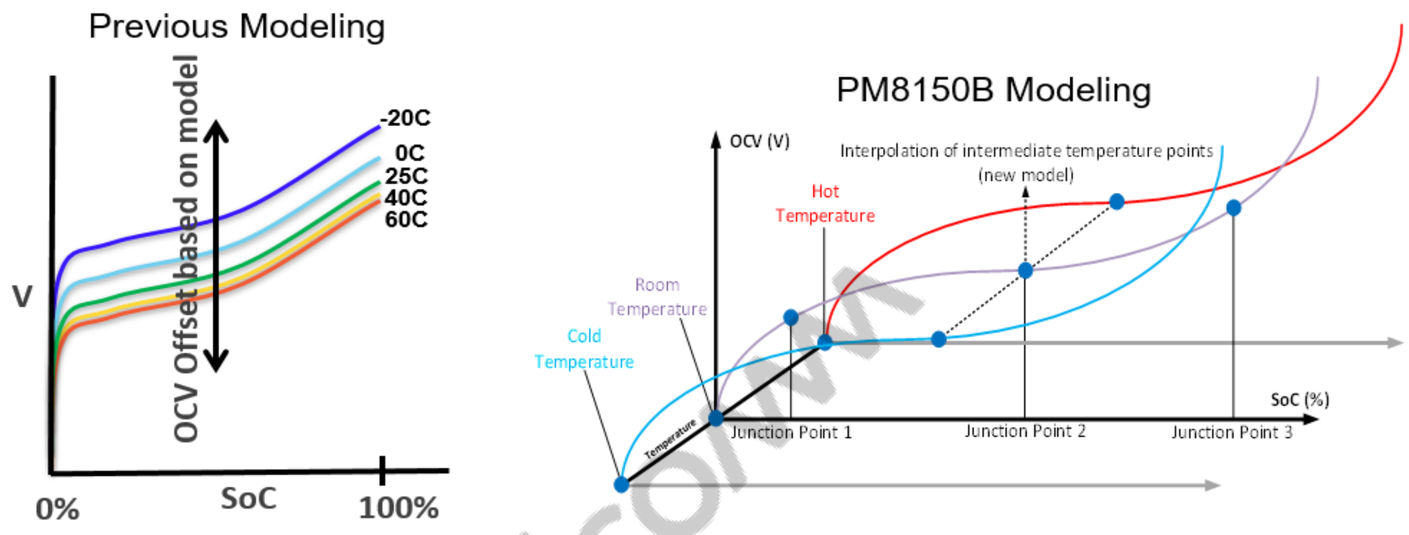


Figure 2-8 Example of OCV modeling and improvement over previous generation fuel gauges

2.2.2 SoC Overview

The fuel gauge reports SoC in four different layers:

- Coulomb Counted SoC (CC SoC): This is the coulomb-counted SoC represented as a percentage of the current available battery capacity.
- Battery SoC: This is the combination of the coulomb counted SoC and the voltage mode correction. This layer adds in the voltage mode correction using the battery model and profile.
- System SoC: This is a filtered version of the battery SoC and is responsible for the endpoint matching of 0% SoC to the programmed cutoff voltage, and 100% to the programmed system termination current.
- Monotonic SoC: This is the final layer of SoC that is displayed to the end user. This layer provides control of the slope and monotonicity of the state of charge.

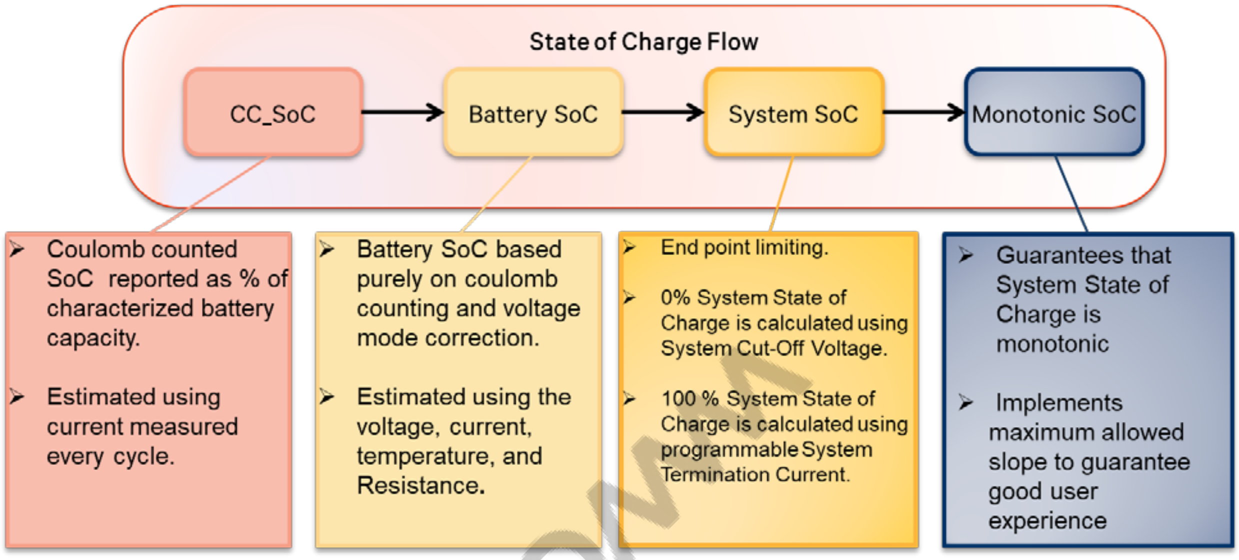

State of Charge Flow
CC_SoC
Battery SoC
System SoC
Monotonic SoC
Coulomb counted SoC reported as % of characterized battery capacity.
Estimated using current measured every cycle.
Battery SoC based purely on coulomb counting and voltage mode correction.
Estimated using the voltage, current, temperature, and Resistance.
End point limiting.
0% System State of Charge is calculated using System Cut-Off Voltage.
100 % System State of Charge is calculated using programmable System Termination Current.
Guarantees that System State of Charge is monotonic
Implements maximum allowed slope to guarantee good user experience

Figure 2-9 SoC flow diagram

2.2.2.1 Coulomb Counted SoC (CC SoC)

CC SoC is the first layer of the fuel gauge SoC algorithm. This SoC is reported as a percentage of the available battery capacity which changes as the cell ages and capacity learning takes place.

NOTE: CC SoC can be negative (< 0%) or greater than 100%, and this is expected behavior. This is expected because the CC SoC will accumulate IADC error over time. The algorithm only cares about the coulomb count delta, and not magnitude of the coulomb count percentage. Once the coulomb count or voltage mode correction reach their maximum or minimum possible values (saturate), hardware will add or subtract the accumulated voltage mode correction percentage into the CC SoC percentage, and reset the voltage mode correction to 0%.

2.2.2.2 Battery SoC

Battery SoC is a combination of the coulomb counted SoC and the voltage mode correction algorithm. The voltage mode algorithm uses the coulomb counted SoC as its base, and corrects on top of it using measured voltage, current, temperature, and modeled impedance and OCV. This feedback loop helps correct the SoC over temperature and compensates error due to coulomb counting or loss of capacity due to cell aging. This creates a more robust and accurate SoC over multiple use cases vs. just a purely coulomb counted solution. Figure 2-10 gives an example of how the mixing is performed.

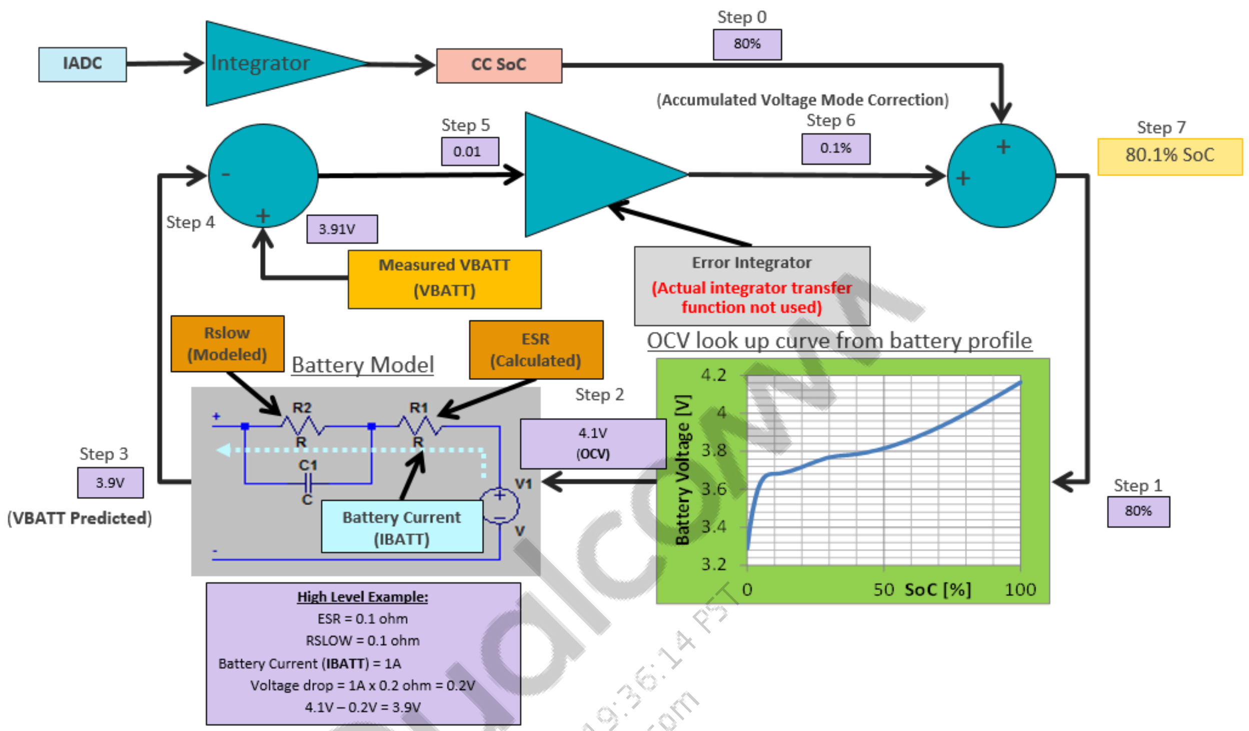


Figure 2-10 Simplified Battery SoC algorithm example

NOTE: This is a high level simplified example, but it useful in understanding how the battery SoC behaves.

2.2.2.3 System SoC

System SoC is an intermediate layer between the battery and the monotonic SoC's, and is responsible for matching 100% to the programmed system termination current (AKA IBATT Full), and 0% to the cutoff voltage (AKA VBATT Cutoff). See Figure 2-11 for a example. System SoC is calculated in real time as follows:

$$\text{System SoC} = \frac{100\%}{\text{SoC Full} - \text{SoC Empty}} * (\text{Battery Soc} - \text{SoC Empty})$$

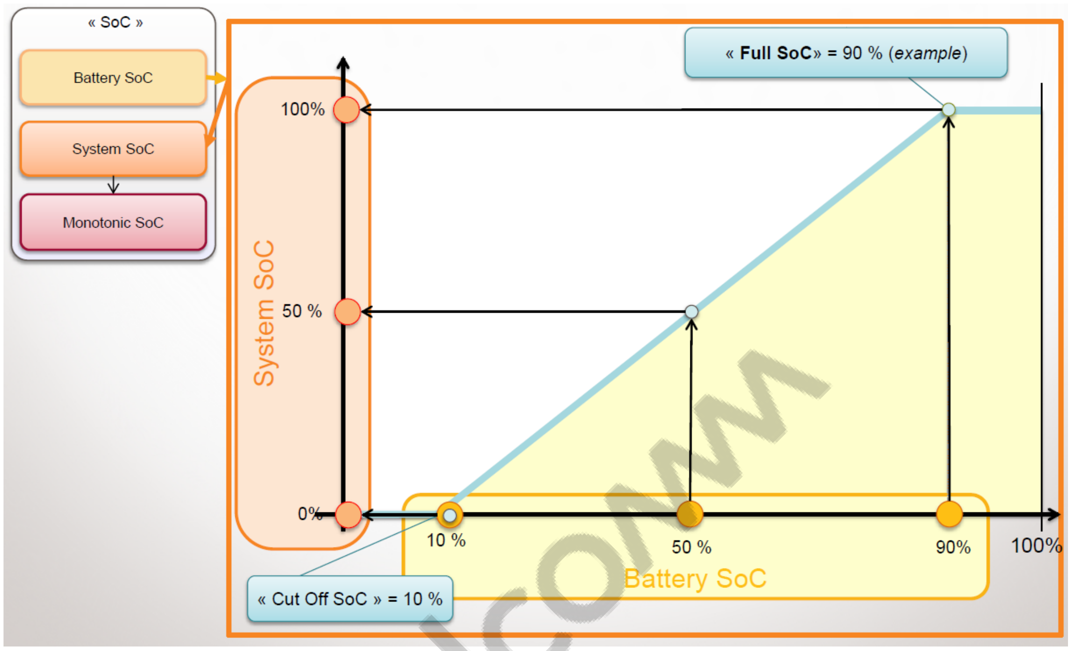


Figure 2-11 System SoC endpoint matching

2.2.2.3.1 Endpoint matching hardware loops

SoC Full is the name of the feedback loop in hardware responsible for matching the 100% SoC point to the IBATT full current setting. The configurable inputs to these loops are IBATT full and VBATT full, both of which need to be set with the guidance provided here to achieve optimal accuracy.

Programmable Inputs:

- **IBATT Full** is the battery current based setting that is used to match 100% SoC to battery current. This should be set such that it is reached before the charge termination current set within charger software, or 100% SoC may not be reached. It is recommended this be set so it is reached 50 mA before charge termination.

NOTE: Battery current is reported as negative to denote current entering the batter, so this must be set less than (more negative) than the charge termination current so 100% is reached before charge termination.

- **VBATT full** is a voltage based setting used in the feedback loop responsible for matching the 100% SoC point to the programmed IBATT full termination current. This must be set 10 mV less than the charger's set float voltage (FV). This 10 mV buffer is to compensate for the charger modules FV inaccuracy. If this is too close to the prorammed FV setting of the charger it is possible 100% may never be reached.

WARNING: System termination accuracy is optimal when VBATT Full is set to the FV of the charger. Since this is not possible due to the charger modules FV accuracy it is common for 100% to be reported before reaching the system termination current. In some instances OEM's have set this closer to FV to achieve improved accuracy, but this must be validated to be OK and be done at the OEM's own risk.

Measured and Modeled Inputs:

- **ESR and Rslow** are both inputs to the SoC Full loop and effect the voltage prediction feedback loop of SoC Full.

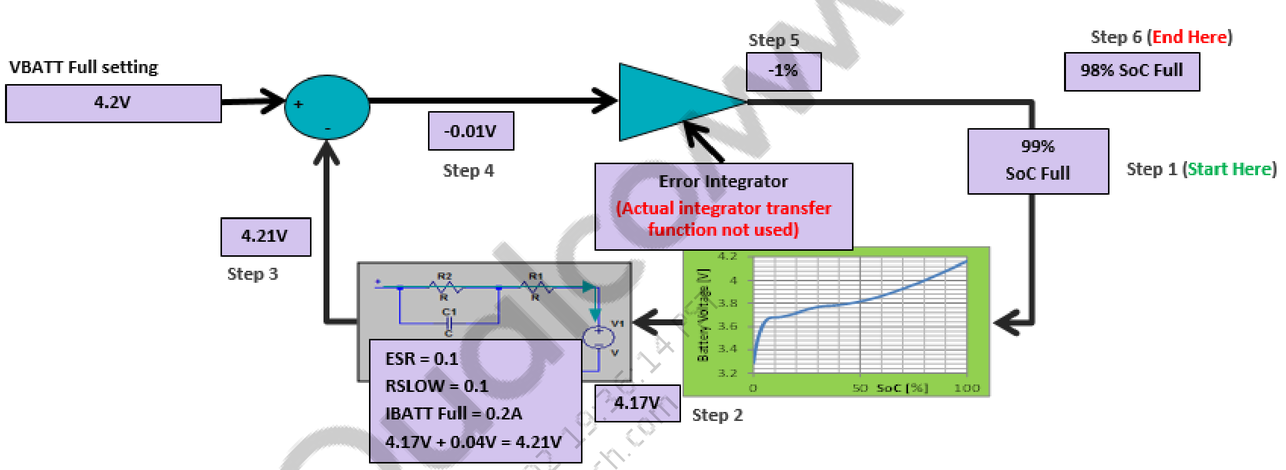


Figure 2-12 High level example of SoC Full loop

SoC Empty is the feedback loop responsible for matching the 0% SoC point to the cutoff voltage. The configurable inputs to this loop is are VBATT Cutoff and IBATT Cutoff.

Programmable Inputs:

- **VBATT Cutoff** is the battery based voltage setting used in the feedback loop to report 0% SoC. 3.2 V is set internally on test platforms for validation. Anything below 3.2 V should be validated by OEM's to ensure stability at low battery voltages over temperature.
- **IBATT Cutoff** is a battery current based setting with intention of providing a battery voltage based buffer during any battery condition where a shutdown could take place before reaching 0% SoC. The SoC Empty loop will use the maximum of either battery current or IBATT Cutoff in its feedback loop. The purpose is this be set such that it mimics typical current draw when exiting sleep and turning the backlight on of the device, so an end user can read the SoC.

Measured and Modeled Inputs:

- **Battery current** is used when greater than the IBATT Cutoff setting.
- **ESR and Rslow** are both inputs to the SoC Empty loop and effect the voltage prediction feedback loop of SoC Empty.

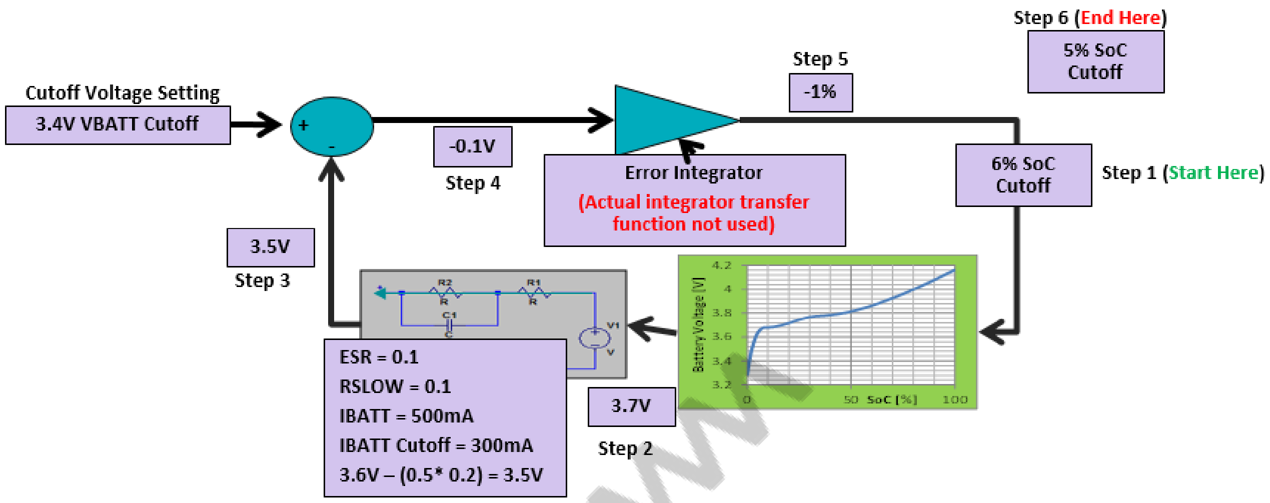


Figure 2-13 High level example of SoC Empty loop

2.2.2.4 Monotonic SoC

The last filtered version of SoC is referred to as the Monotonic SoC. This is because it is responsible for ensuring monotonicity during discharge. It is possible during low current conditions, or during temperature ramps, the battery SoC could increase, even while discharging. Since this is undesirable behavior, the monotonic filter prevents the SoC from increasing unless a charger is present.

The monotonic SoC also has a configurable slope limiter. This can be implemented by the customer depending on their requirements and will limit how much the SoC changes percentage wise for every 1 s (fuel gauge cycle).

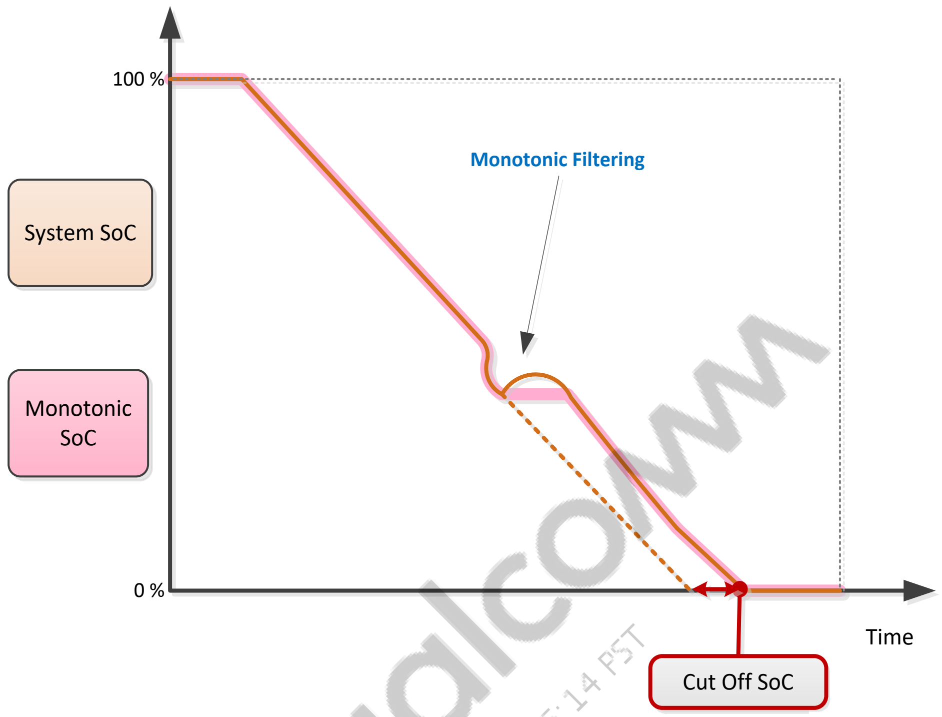


Figure 2-14 Monotonic filtering example

2.2.2.4.1 Slope limiting

- There are four distinct slopes that can be configured in software that depend on evert combination of charging or discharge and high or low temperature, which the mid point temperature threshold being configurable. This setting prevents the SoC from changing by the programmed percentage for every 1 fuel gauge cycle (1 s). For information about how to configure this in software, see *PM8150B PMIC Fuel Gauge Software User Guide* (80-PF777-74).

2.2.3 Battery profiling

2.2.3.1 Requesting characterization

Customers must submit a battery characterization request to obtain a battery profile from Qualcomm. This is done via the “Battery Characterization” case type.

WARNING: The Battery Characterization case type is NOT a Wireless Device Support case type.

Customers may characterize their batteries themselves using the software tool QBCSW found on CreatePoint, but cannot generate the final profile. This final process, which is referred to as “Post Processing”, is required to generate all of the necessary models using the data collected by QBCSW. This software is not available to customers for two reasons:

1. The models generated after post processing under go a strict review process by our battery characterization expert. This is done in house to ensure quality of the profile provided to OEM's.
2. The post processing software contains proprietary code QTI is licensed to use, but not to provide to its OEM's.

Profile generation is broken down into two steps:

1. Characterization: Characterization is the process through which the battery impedance, capacity, and OCV relationships are all measured over temperature and state of charge; for both charge and discharge. Only raw data is generated and cannot be used by the fuel gauge until the post processing step.

Post-processing: After characterization, high iteration curve fitting is done on the raw data to generate the necessary polynomial coefficients for the fuel gauge models. These polynomials model how ESR, Rslow, and OCV vary over temperature and SoC, and are necessary for fuel gauge operation.

3 Register information

3.1 Fuel gauge memory overview

The fuel gauge digital contains its own memory space which is typically referred to as “SRAM”. This memory is separate from the SPMI peripheral, which is addressed by our Hardware Register Description documentation or the register tool on CreatePoint. SRAM is accessed via the MEM_IF peripheral in SPMI using Direct Memory Access (DMA).

3.2 Fuel gauge interrupts

Table 3-1 Fuel gauge interrupts

Peripheral	Interrupt	Description	Register
FG_BATT_SOC	MSOC_FULL	Monotonic SoC = 100%	0x4010
	MSOC_HIGH	Monotonic SoC ≥ high threshold	0x4010
	MSOC_EMPTY	Monotonic SoC = 0% or VBATT < Volt_Empty	0x4010
	MSOC_LOW	Monotonic SoC ≤ Low threshold	0x4010
	MSOC_DELTA	Monotonic SoC change exceeds programmed delta	0x4010
	BSOC_DELTA	Battery SoC change exceeds programmed delta	0x4010
	SOC_READY	Fuel gauge completes a first estimate	0x4010
	SOC_UPDT	Triggers when SoC information is updated (every cycle)	0x4010
FG_BATT_INFO	WDOG_EXP	FG watchdog has expired	0x4110
	ESR_DELTA	Battery resistance has changed by more (increment or decrement) than the specified threshold	0x4110
	ESR_PULSE_PRE	In next cycle, ESR pulse is to be requested	0x4110
	VBT_PRD_DELTA	Predicted battery voltage is exceeding (lower or higher) the defined threshold	0x4110
	VBT_LOW	Battery Voltage < VBATT_LOW threshold	0x4110
FG_ADC_RR	BT_TMPR_COLD	Battery temperature cold	0x4210
	BT_TMPR_HOT	Battery temperature hot	0x4210
	BT_TMPR_DELTA	Battery temperature change exceeds the delta threshold	0x4210
	BT_ID	Battery ID conversion data available from ADC module	0x4210
	BT_MISS	Battery missing reported by BIM	0x4210
FG_MEM_IF	MEM_GNT	Memory granted for access; algorithm cannot access memory and is paused	0x4310
	DMA_XCP	Direct memory access (DMA) exception	0x4311
	IMA_XCP	Interleave memory access (IMA) exception	0x4312

Peripheral	Interrupt	Description	Register
	IMA_RDY	Behaves as Ready/End of Transaction depending on configuration	0x4313

3.3 Fuel gauge RAM register map

Table 3-2 Register map

Customer Register Name	Parsed by Customer Parser	Category	Address (Decimal)	Bit Offset	Bit Size	Signed	Enum Offset	LSB	Unit	Description
ESR Qual Threshold	TRUE	fixed8	12	8	8	TRUE	0	0.015625	A	Current delta between previous and current cycle threshold that needs to be exceeded to qualify an ESR calculation
Discharge ESR Timer Max	TRUE	fixed8	17	0	8	FALSE	0	1	FG Cycles	Discharge ESR timer maximum count expiration value. Once timer count reaches this value an ESR measurement takes place.
Discharge ESR Timer Re-attemp Start	TRUE	fixed8	17	8	8	FALSE	0	1	FG Cycles	Discharge ESR timer reattempt count start value. New start point for timer when a ESR measurement has failed.
Charge ESR Timer Max	TRUE	fixed8	18	0	8	FALSE	0	1	FG Cycles	Charge ESR timer maximum count expiration value. Once timer count reaches this value an ESR measurement takes place.
Charge ESR Timer Re-attemp Start	TRUE	fixed8	18	8	8	FALSE	0	1	FG Cycles	Charge ESR timer reattempt count start value. New start point for timer when a ESR measurement has failed.
IBATT Cutoff	TRUE	fixed16	19	0	16	TRUE	0	0.000488	A	Minimum load current used in Cutoff SoC loop.
VBATT Cutoff	TRUE	fixed16	20	0	16	FALSE	0	0.000244	V	Cutoff voltage.
VBATT Cutoff Ki	TRUE	fixed8	21	0	8	FALSE	0	0.000061	V	Integration gain in voltage loop.
IBATT Full	TRUE	fixed16	22	0	16	TRUE	0	0.000488	A	Charge current for 100% reporting (not charge termination current).
VBATT Full	TRUE	fixed16	23	0	16	FALSE	0	0.000244	V	Float voltage target for 100% reporting. Should set less than target FV due to charger FV tolerance.

Customer Register Name	Parsed by Customer Parser	Category	Address (Decimal)	Bit Offset	Bit Size	Signed	Enum Offset	LSB	Unit	Description
VM Ki Disch Low	TRUE	fixed8	25	8	8	FALSE	0	0.000061	N/A	Low current integration gain in voltage mode loop for discharge.
VM Ki Disch Med	TRUE	fixed8	26	0	8	FALSE	0	0.000061	N/A	Medium current integration gain in voltage mode loop for discharge.
VM Ki Discharge High	TRUE	fixed8	26	8	8	FALSE	0	0.000061	N/A	High current integration gain in voltage mode loop for discharge.
VM Ki Discharge Low to Med Th	TRUE	fixed8	27	0	8	FALSE	0	0.015625	A	Discharge low to medium current threshold for voltage mode loop.
VM Ki Discharge Med to High Th	TRUE	fixed8	27	8	8	FALSE	0	0.015625	A	Discharge medium to high current threshold for voltage mode loop.
VM Ki Charge Low	TRUE	fixed8	28	0	8	FALSE	0	0.000061	N/A	Low current integration gain in voltage mode loop for charge.
VM Ki Charge Med	TRUE	fixed8	28	8	8	FALSE	0	0.000061	N/A	Medium current integration gain in voltage mode loop for charge.
VM Ki Charge High	TRUE	fixed8	29	0	8	FALSE	0	0.000061	N/A	High current integration gain in voltage mode loop for charge.
VM Ki Charge Low to Med Th	TRUE	fixed8	29	8	8	FALSE	0	0.015625	A	Charge low to medium current threshold for voltage mode loop.
VM Ki Charge Med to High Th	TRUE	fixed8	30	0	8	FALSE	0	0.015625	A	Charge medium to high current threshold for voltage mode loop.
Bat SoC Delta Th	TRUE	fixed8	30	8	8	FALSE	0	0.000488	%	Battery SoC change threshold to trigger interrupt
Mon Slope Limiter Config	TRUE	fixed8	32	0	8	FALSE	0	0.000122	%	Monotonic SoC slope limiter setting in % Monotonic SoC per 1 second update cycle.

Customer Register Name	Parsed by Customer Parser	Category	Address (Decimal)	Bit Offset	Bit Size	Signed	Enum Offset	LSB	Unit	Description
Mon SoC Delta Th	TRUE	fixed8	32	8	8	FALSE	0	0.000488	%	Monotonic SoC change threshold to trigger interrupt
Mon SoC Low Th	TRUE	fixed8	33	0	8	FALSE	0	0.003906	%	Low Monotonic SoC interrupt threshold
Mon SoC Empty Th	TRUE	fixed8	33	8	8	FALSE	0	0.003906	%	Empty Monotonic SoC interrupt threshold. Triggered when Monotonic SoC reaches 0%.
Mon SoC High Th	TRUE	fixed8	34	0	8	FALSE	0	0.003906	%	High Monotonic SoC interrupt threshold
Mon SoC Full Th	TRUE	fixed8	34	8	8	FALSE	0	0.003906	%	Full Monotonic SoC interrupt threshold. Triggered when Monotonic SoC reaches 100%.
VBATT Empty	TRUE	fixed8	35	8	8	FALSE	2	0.015625	V	Programmable threshold when reached Fuel Gauge is shutdown by software. Add 2.0V base to obtain threshold.
Saturation Auto Clear En	TRUE	bool	60	10	1	N/A	0	T/F	N/A	CC SoC or Voltage mode correction saturation auto clear enable.
Mon SoC Mon Filter En	TRUE	bool	60	13	1	N/A	0	T/F	N/A	Monotonic SoC control which prevents SoC increase with discharge current.
Mon Slope Limiter En	TRUE	bool	60	14	1	N/A	0	T/F	N/A	Monotonic SoC slope limit control.
Nominal Capacity	TRUE	fixed16	271	0	16	FALSE	0	1	mAh	Capacity found from characterization in mAH
Rconn	TRUE	fixed16	275	0	16	TRUE	0	0.000122	Ohms	Software writes here the programmed Rconn
Learned Capacity	TRUE	fixed16	285	0	16	FALSE	0	1	mAh	Learned battery capacity in mAH
Backup Learned Capacity	TRUE	fixed16	290	0	16	FALSE	0	1	mAh	Backup learned battery capacity in mAH
Cycle Count 1	TRUE	fixed16	291	0	16	FALSE	0	1	N/A	Cycle counter bucket 1
Cycle Count 2	TRUE	fixed16	292	0	16	FALSE	0	1	N/A	Cycle counter bucket 2

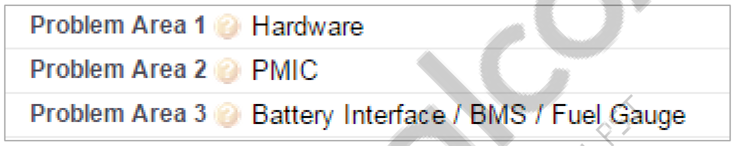
Customer Register Name	Parsed by Customer Parser	Category	Address (Decimal)	Bit Offset	Bit Size	Signed	Enum Offset	LSB	Unit	Description
Cycle Count 3	TRUE	fixed16	293	0	16	FALSE	0	1	N/A	Cycle counter bucket 3
Cycle Count 4	TRUE	fixed16	294	0	16	FALSE	0	1	N/A	Cycle counter bucket 4
Cycle Count 5	TRUE	fixed16	295	0	16	FALSE	0	1	N/A	Cycle counter bucket 5
Cycle Count 6	TRUE	fixed16	296	0	16	FALSE	0	1	N/A	Cycle counter bucket 6
Cycle Count 7	TRUE	fixed16	297	0	16	FALSE	0	1	N/A	Cycle counter bucket 7
Cycle Count 8	TRUE	fixed16	298	0	16	FALSE	0	1	N/A	Cycle counter bucket 8
Profile Integrity bit	TRUE	bool	299	0	1	N/A	0	T/F	N/A	0: profile is of default trim 1: profile loaded by software
UEFI Load Sts	TRUE	bool	299	1	1	N/A	0	T/F	N/A	0: profile source is not uefi 1: profile source is uefi
UEFI FG Restart STs	TRUE	bool	299	2	1	N/A	0	T/F	N/A	0: restart not initiated by uefi 1: restart initiated by uefi
HLOS Load Sts	TRUE	bool	299	3	1	N/A	0	T/F	N/A	0: profile source is not hlos 1: profile source is hlos
IBATT Old	TRUE	fixed16	317	0	16	TRUE	0	0.000488	A	Battery current measured from t -1 cycle
VBATT Old	TRUE	fixed16	318	0	16	FALSE	0	0.000244	V	Battery voltage measured from t - 1 cycle
IBATT	TRUE	fixed16	320	0	16	TRUE	0	0.000488	A	Battery current measured durring last cycle t
VBATT	TRUE	fixed16	321	0	16	FALSE	0	0.000244	V	Battery voltage measured durring last cycle t
IBATT Low Pass	TRUE	fixed24	322	0	24	TRUE	0	0.000002	I	Battery current passed through low pass filter
VBATT Low Pass	TRUE	fixed24	326	0	24	TRUE	0	0.000002	V	Battery voltage passed through low pass filter
Battery Temp	TRUE	fixed16	328	0	16	TRUE	0	0.25	C	Calculated temp in [C]
ESR Modeled	TRUE	fixed16	342	0	16	FALSE	0	0.000244	Ohms	Modeled ESR used by Fuel Gauge

Customer Register Name	Parsed by Customer Parser	Category	Address (Decimal)	Bit Offset	Bit Size	Signed	Enum Offset	LSB	Unit	Description
ESR Timer Sts	TRUE	fixed8	346	0	8	FALSE	0	1	FG Cycles	ESR pulse timer similar to PMI8998
Rslow	TRUE	fixed16	368	0	16	FALSE	0	0.000244	Ohms	Rslow
OCV Calculated	TRUE	fixed16	417	0	16	FALSE	0	0.000244	V	Calculated OCV
Predicted Battery Voltage	TRUE	fixed24	432	0	24	FALSE	0	0.000244	V	Predicted battery voltage for battery SoC VM loop
Voltage Mode Correction	TRUE	fixed32	447	0	32	TRUE	0	9.31E-10	%	Battery SoC voltage mode error correction in %
Battery SoC	TRUE	fixed32	449	0	32	FALSE	0	2.33E-10	%	Battery SoC
Cutoff SoC	TRUE	fixed16	453	0	16	FALSE	0	0.000015	%	Cutoff SoC
Full SoC	TRUE	fixed16	455	0	16	FALSE	0	0.000015	%	Full SoC
CC SoC	TRUE	fixed32	458	0	32	TRUE	0	9.31E-10	%	Coulomb count used by software for capacity learning (reset by software)
CC SoC software	TRUE	fixed32	460	0	32	TRUE	0	9.31E-10	%	Coulomb Count SoC used by FG algorithm (reset when saturated)
System SoC	TRUE	fixed16	462	0	16	FALSE	0	0.000015	%	System SoC
Monotonic SoC	TRUE	fixed16	463	0	16	FALSE	0	0.000015	%	Monotonic SoC

4 Troubleshooting

Additional troubleshooting information will be provided in a future revision of this documentation.

If an issue is suspected to be fuel gauge related and the FAQ section of this document did not help, it is recommended creating a fuel gauge case via Salesforce. The case will be routed automatically to someone who can assist in debugging.



Problem Area 1	Hardware
Problem Area 2	PMIC
Problem Area 3	Battery Interface / BMS / Fuel Gauge

Figure 4-1 Example problem area for a fuel gauge case in Salesforce

Provide the following information when submitting a fuel gauge case:

- Detailed description and steps used to reproduce issue.
- SRAM logs:
  - SRAM logs are the most powerful tool for debugging fuel gauge issues. They will be requested for every issue. Customers must generate them and submit them with the case. These logs should be provided in the proper file format with no SPMI fuel gauge peripheral or charger peripheral logs. Failure to do so might require QTI to request these logs to be recaptured.

These are not the usual Linux Android logs. For more information about the process for collecting these logs, see *SM8150 PMIC Fuel Gauge Software User Guide* (80-PF777-74).

5 Frequently asked questions

5.1 Algorithm

5.1.1 How do I test the fuel gauge performance?

QTI does not provide any tools to customers for SoC accuracy testing. QTI recommends customers develop their own knowledge and test methods for SoC accuracy testing. QTI tests the fuel gauge extensively and has SoC accuracy data to provide upon customer request.

If there are any concerns with fuel gauge performance it is recommend that customers create a case, and QTI can assist in debugging.

5.1.2 Can the fuel gauge cycle of 1s be changed?

The 1s fuel gauge update cycle is fixed in hardware and cannot be modified. Most concerns with this measurement interval arise because the fuel gauge may miss transients that do not happen during the 160 ms conversion window. QTI's internal testing has shown that missing fast transients that do not last longer than 1s has negligible effect on SoC tracking accuracy. This is due to the fuel gauge having a mixed mode algorithm of coulomb counting and voltage mode correction. The voltage mode aids in correcting any missing coulombs, or accumulating coulomb count error.

5.1.3 Can the ESR measurements pulsing be disabled or their frequency changed?

If the fuel gauge is used to report SoC, the ESR pulses **cannot** be disabled.

If the fuel gauge is not used in a design to report SoC, the ESR pulses can be disabled, but it is strongly recommended that QTI be contacted before making the decision to not use QTI's fuel gauge to verify there will not be any unforeseen issues.

NOTE: Typically if a customer decides to use a 3rd party fuel gauge, the decision is based on incorrect, vague, or misleading information. Contact QTI if there are any concerns with specific features or specifications of the fuel gauge to verify concerns are valid. It is beneficial to both parties when OEM's use our FG instead of attempting to design in a 3rd party solution, which QC will **not** support.

Changing the calibration interval although possible, it is recommended against. If this is changed SoC accuracy cannot be guaranteed as this timing will differ from what has been validated internally.

5.1.4 IBATT full termination current is inaccurate; what could be wrong?

The feedback loop internal to the fuel gauge responsible for matching the 100% point to the system termination current uses the following parameters: IBATT full, VBATT full, ESR and Rslow. Check the following:

1. VBATT full must be set to 10 mV less than the programmed float voltage to compensate for the charger's FV inaccuracy or 100% SoC may not be reached. Ensure the chargers float voltage is within specification. The VBATT Full can be increased, which will improve IBATT Full 100% SoC matching, but it must be at a customer's own risk. It is possible some devices' FV may not meet or exceed the VBATT Full setting due to FV inaccuracy, which could potentially lead to 100% not being reached.
2. Ensure IBATT Full is set correctly, and occurs before charger termination by a recommended margin of 50 mA.
3. If IBATT Full and VBATT Full are both set correctly, it is possible there is an impedance measurement/modeling issue. It is recommended a case be created in Salesforce, and Qualcomm help investigate this further.

5.2 Battery thermistor

5.2.1 The battery temperature measurements are out of specification; what could be wrong?

5.2.1.1 Capacitance on BATT_THERM

This is a common issue that is attributed to added capacitance on the BATT_THERM net. Additional capacitance on this net is not recommended and may cause temperature measurement accuracy issues. QTI does not do internal validation with any capacitance on the BATT_THERM net. When checking for added capacitance, the battery pack's internal protection circuit is commonly overlooked, so be sure to check this as well.

5.2.1.2 Beta coefficients not programmed correctly

These coefficients are provided as a part of the battery characterization process. If these coefficients were not provided, or their accuracy is in questions, it is recommended a Salesforce case be created.

NOTE: A profile containing incorrect beta coefficients is extremely unlikely as these are generated automatically by software during the characterization process based on the battery thermistor information provided.

5.2.1.3 ADC Pull up not chosen correctly

The ADC chooses one of three pull ups to use for ADC measurements on BATT_THERM. 30 k, 100 k, and 400 k. This parameter will be included in the battery profile and will be chosen by software during the characterization process. If this parameter is not within the battery profile, please contact QTI via Salesforce to receive this pull up software parameter.

5.3 Battery identification

5.3.1 The battery RID accuracy is out of specification; what could be wrong?

This is a common issue that is attributed to added capacitance on the BATT_ID net. Additional capacitance on this net is **not supported** and causes RID accuracy measurement issues. When checking for added capacitance, the battery pack's internal protection circuit is commonly overlooked.