*Application Note*

# *Theory and Implementation of Impedance Track™ Battery Fuel-Gauging Algorithm in bq2750x Family*

*Ming Yu, Yevgen Barsukov, and Michael Vega* *Battery Management*

**ABSTRACT**

This application report outlines the theory of Impedance Track™ [1] (IT) technology used in the bq2750x series of fuel gauge ICs for single-cell Li-ion application (e.g., smart phones, media players, and PDAs). The implementation of the IT algorithm in the bq2750x family is reviewed, and the setting of data flash constants associated with the fuel-gauging algorithm is described in detail.

[1] Impedance Track algorithm is protected by US Patents US6832171, US6789026, and US6892148.

## Table of Contents

## List of Figures

## Trademarks

## 1 Summary of the Algorithm Operation

The gas gauge algorithm uses three types of information to calculate remaining capacity (*DataRAM.Remaining Capacity( )*) and full-charge capacity (*DataRAM.Full Charge Capacity( )*).

1. Chemical: depth of discharge (DOD) and total chemical capacity $Q_{max}$
2. Electrical: internal battery resistance dependence on DOD
3. External: load and temperature

*DataRAM.Full Charge Capacity( )* is defined as the amount of charge passed from a fully charged state until the voltage defined in *DF.Terminate Voltage* flash constant is reached at a given rate of discharge, after subtracting the reserve capacity *(DF.Reserve Capacity)*.

Note that *DataRAM.Full Charge Capacity( )* depends on the rate of discharge and is lower at higher rates and low temperatures because the cell I*R drop causes the Terminate Voltage threshold to be reached earlier.

## 2 Parameters Updated by the Gas Gauge in More Detail

### 2.1 Modes of Algorithm Operation

The algorithm differentiates between *charge*, *discharge*, and *relaxation* modes of operation. During *charge* mode, the *DataRAM.Flags( ) [DSG]* bit is cleared, and during *discharge* and *relaxation* mode, it is set. Entry and exit of each mode is controlled by Data Flash (DF) parameters in the subclass Gas Gauging: Current Thresholds section as illustrated in Figure 2-1. Charge mode is exited, and relaxation mode is entered when *DataRAM.Average Current( )* goes below *DF.Quit Current* after *DF.Chg Relax Time* period. Discharge mode is entered when *DataRAM.Average Current( )* goes below *DF.Dsg Current Threshold* and after *DF. Quit Relax Time* period. Discharge mode is exited, and relaxation mode is entered when *DataRAM.Average Current( )* goes above negative *DF.Quit Current* threshold and after *DF.Dsg Relax Time* period. Charge mode is entered when *DataRAM.Average Current( )* goes above *DF.Chg Current Threshold* and after *DF.Quit Relax Time* period.

### 2.2 Update of Chemical Depth of Discharge (DOD)

The gas gauge updates information on chemical depth of discharge ($DOD_0$) based on open-circuit voltage (OCV) readings when in a *relaxed* state. DOD is found by correlating DOD with OCV using a predefined table DOD(OCV,T) stored as reserved data flash parameters. The table is specific for a particular chemistry such as $LiCoO_2$/carbon, $LiMn_2O_4$/carbon etc., and can be identified by reading the chemistry ID through sending ChemID( ) command 0x0008, then reading ChemID(). The gas gauge can be set up for a particular chemistry by using a specific firmware file (*.senc) that can be downloaded from bq2750x production folder on power.ti.com. The chemistry profile also can be programmed into bq2750x using the bqEASY wizard.

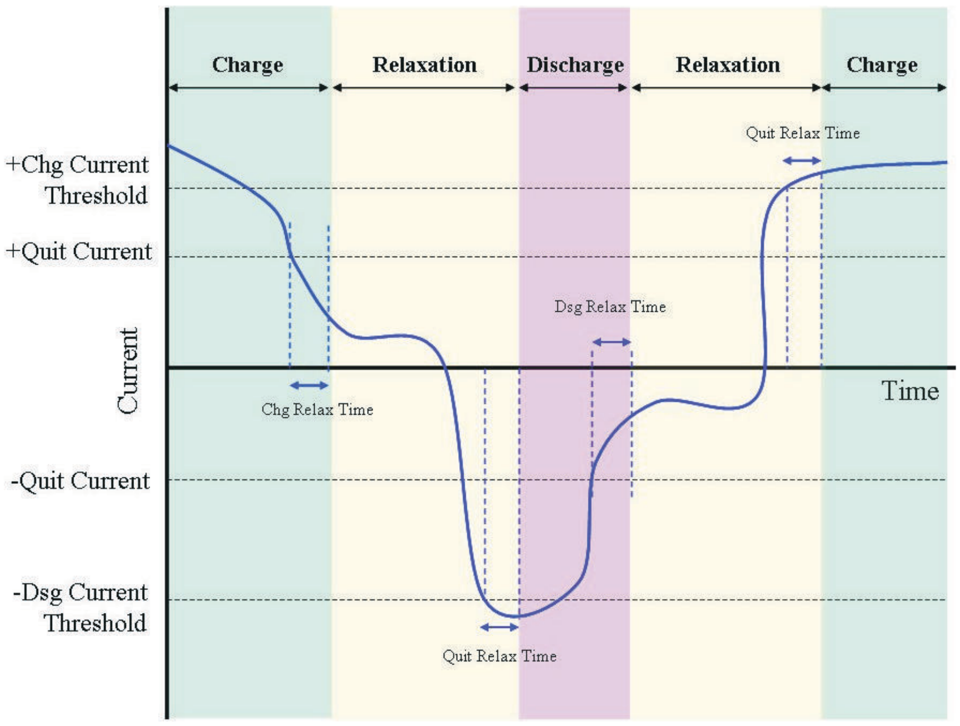


**Figure 2-1. Example of Algorithm Operation Mode Changes With Varying *DataRAM.Average Current( )***

Figure 2-2 shows the timing of parameter updates during relaxation mode. After a 30-minute relaxation period is passed, the dV/dt <4 µV/s condition is checked. Once it is satisfied, OCV readings are taken. After that, OCV readings continue to be taken every 100 seconds. DOD is calculated based on each measured OCV reading using linear interpolation DOD = f(OCV,T). Integrated PassedCharge is set to zero at each $DOD_0$ update.

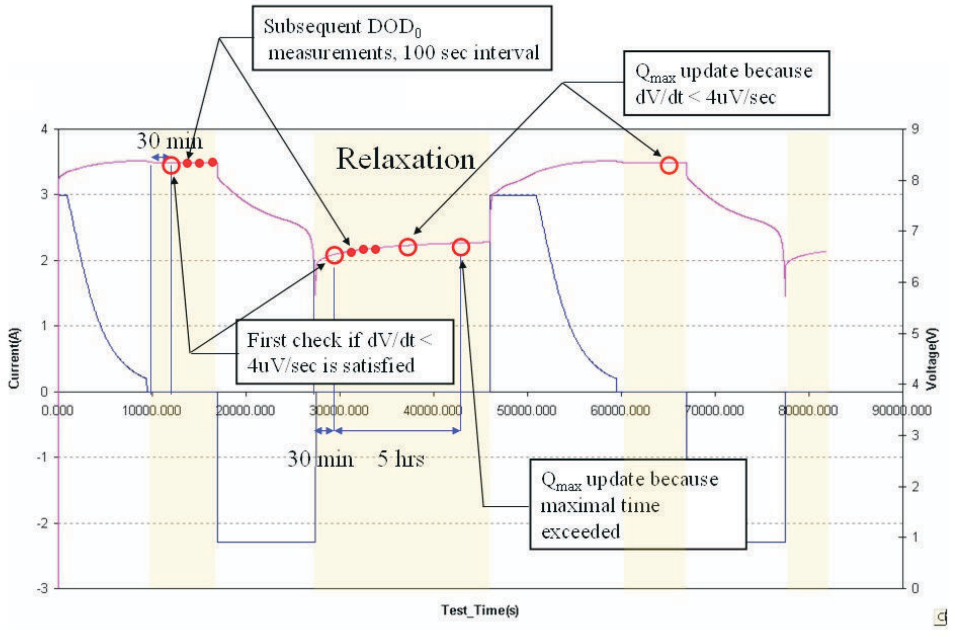


**Figure 2-2. Timing of $DOD_0$ and $Q_{max}$ Updates During Relaxation Mode**

If the current during the OCV reading is non-zero, then an IR correction is done. The first iteration of DOD is found from the uncorrected OCV reading; then the resistance value is found from the R(DOD) table and used to correct the OCV value as OCV`=OCV-I*R. Then, the corrected DOD is found from OCV`. This method achieves the best accuracy if the current during relaxation mode is below a C/20 rate. This is why it is recommended that the *DF.Quit Current* not exceed C/20.

If no $DOD_0$ has been measured until the relaxation state is exited, the previous $DOD_0$ is used along with the PassedCharge integrated since the last DOD reading.

During charge and discharge modes, the *present* DOD is recalculated every second as DOD = $DOD_0$ + PassedCharge/$Q_{max}$. DOD is used for determining when a resistance update needs to occur, as well as the starting point for a Remaining Capacity (and FCC) calculation. Remaining capacity calculations occur immediately after discharge onset, at every resistance update, and after entering relaxation mode

## 2.3 Update of $Q_{max}$

Maximal chemical capacity $Q_{max}$ for a battery cell is stored in Data Flash as *DF.Qmax Cell x*, where x = 0,1, the cell number. The GG updates $Q_{max}$ based on two DOD readings made before and after charge, or discharge. For example, $DOD_1$ is taken during relaxation, then a discharge mode starts, and PassedCharge is integrated. Following this, another relaxation mode is entered, and $DOD_2$ is taken;

$DOD_1$ and $DOD_2$ are calculated from OCV readings in a well-relaxed state, as exemplified in Figure 2-2 for subsequent $DOD_0$ measurement. A well-relaxed state is detected if dV/dt<4 μV/s or maximal waiting time of 5 hours is exceeded. The first condition is satisfied in typical batteries after approximately 1 hour if DOD is between 0 and 80%, and 3–4 hours if DOD is above 80%. At low temperature relaxation takes a longer time.

In order to ensure high accuracy of the DOD measurement, the $Q_{max}$ calculation does not occur if the temperature is above 40°C, or below 10°C. It also does not occur if at least one of voltage measurements for $DOD_1$ or $DOD_2$ was taken in the cell voltage range between 3737 mV and 3800 mV because of very flat OCV(DOD) dependence in this range. These limits are chemistry dependent and are specified for different chemistries separately.

$Q_{max}$ is calculated as $Q_{max}$ = PassedCharge / ($DOD_2$ – $DOD_1$)

The data flash constant *DF.Update Status* increments by 1 when the first $Q_{max}$ update takes place (e.g., from 0 to 1 if no resistance update were made or from 1 to 2 if a resistance update was made).

PassedCharge has to be more than 37% of *DF.Design Capacity* for an update to occur. For the first cycle (with *DF.Update Status* = 0), 90% of *DF.Design Capacity* is required because this cycle takes place in the factory settings and $Q_{max}$ is learned for the first time.

In order to prevent $Q_{max}$ fluctuations, a first-order smoothing filter is applied to all $Q_{max}$ readings except in the first cycle. Readings with lower PassedCharge are assigned lower weights in the smoothing.

## 2.4 Update of Resistance

Resistance is updated during discharge, as summarized in Figure 2-3. The first resistance update happens after a certain waiting time to prevent distortion from transients after a load onset. The waiting time is 500 seconds by default, but later the update waiting time decreases if the maximal discharge duration (*DF.Max Dsg Duration*) is less than 500 seconds. The waiting time is defined as *DF.Max Dsg Duration* –200. Waiting time can decrease to as short as 100 seconds.

The calculation is performed by comparing the measured voltage with the OCV value at the same DOD, that is taken from the OCV(DOD,T) table:

$$dV = V - OCV(DOD,T)$$

$$R(DOD) = dV/I$$

Resistance measurements are taken continuously and stored in RAM.

Resistance is updated in the Data Flash (in *DF.Ra Table*) after each 11.1% of DOD charge is exceeded (DOD charge is PassedCharge/$Q_{max}$). When DOD reaches 77.7%, resistance is updated after each 3.3%. The final resistance update is made after discharge is terminated. .

The constant *DF.Update Status* increments by 1 when the first grid-point resistance update takes place (e.g., from 0 to 1 if no $Q_{max}$ update were made before, or from 1 to 2 if a $Q_{max}$ update was made before).

Before storage to Data Flash, the resistance values are normalized to 0°C as Ra[dod] = R[dod]/exp(Rb[dod]*T) where R is the measured resistance value at a given DOD. Rb[DOD] is the value of temperature coefficient of impedance change at a given DOD stored as a reserved data-flash table, and T is temperature in °C. Note that values of resistance normalized to 0°C are somewhat larger than values at room temperature and so cannot be directly compared with R=dV/I values.

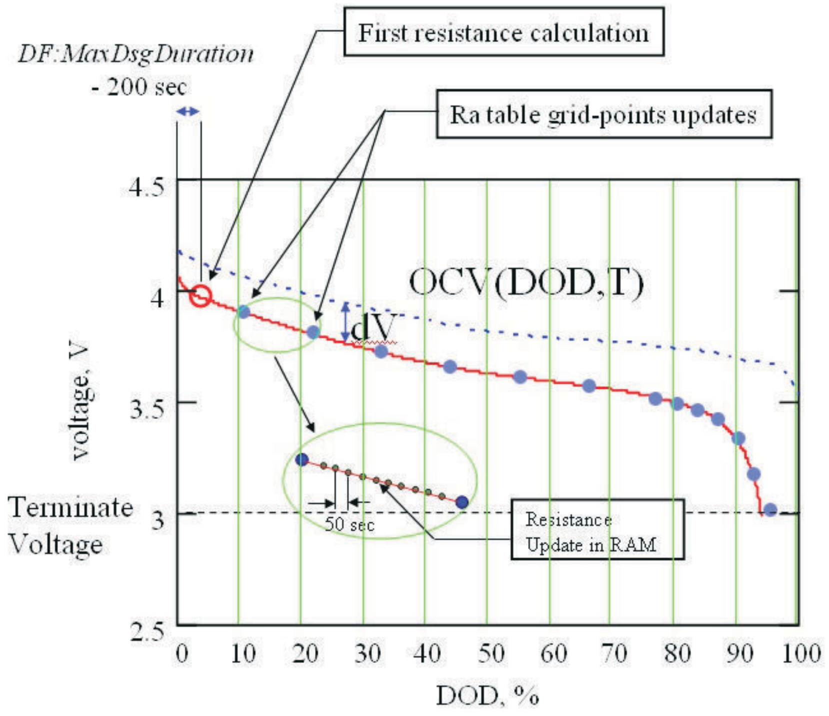


**Figure 2-3. Impedance Updates**

Resistance values for the grid points with a higher DOD than presently updated are scaled by the same factor as the present grid-point change, e.g., by factor Ra_new/Ra_old. In this way, faster convergence of the resistance profile is achieved.

Values in *DF.Ra Table* are stored in milliohm units, in the format *DF.PackX Ra N* where X is pack number from 0 to 1, and N is grid-point number from 0 to 14 that corresponds to 11.1% increments of DOD until 77.7%, and then 3.3% increments of DOD. In bq27500, there are two Ra tables (Ra and Rax) for each of the supported two battery packs. To save data-flash space, tables are compressed. Two additional parameters (Base R and Gain) are used in compression. The *decompression* formula for I=1..14 is
R[i] = Base R + sum(R_compressed[k], k = 1 .. I)×$2^{Gain}$, and R[0]=Base R.

The *DF.PackX Ra flag* and *DF.PackX Rax flag* are used for interchanging data-flash column usage for reducing the number of DF writes. A flag value of 55 indicates the presently used data column, whereas FF indicates the presently unused data column.

Before *DF.Update Status* is set to 2, if during resistance update DOD exceeds 100%, or resistance appears negative, which are both indications of a too-small DF.$Q_{max}$ Cellx initial guess, *DF.Qmax Cell x* will increment by 11.1%, and all resistances will be recalculated. This is normal behavior during the first *learning* cycle. However, if the initial guess of *DF.Qmax Cell x* was too far from the correct value, a second cycle might be needed to achieve full resistance accuracy. To avoid this, set *DF.Qmax Cell x* to a value specified by the cell manufacturer, multiplied by number of parallel cells.

After *DF.Update Status* is set to 2, resistance change is limited to below 5 times and above 0.2 times of its original value, and to positive values.

## 2.5 Update of Temperature Model

Because temperature changes significantly during the course of a discharge, the algorithm needs to be able to predict the future temperature. This is needed for temperature correction of battery impedance (R = Ra x exp(Rb x T)) during voltage simulation near the end of discharge. To achieve this, the algorithm collects T(t) dependence data during discharge. It is used to update parameters of a simple thermal model including a heat exchange coefficient and a thermal time constant. These parameters are updated at the same time as resistances. The algorithm also records the outside temperature (T_out) during relaxation periods. These parameters are used to define a function T(t, T_start) that calculates a temperature profile starting from present temperature, T_start, and continuing until the end of discharge.

## 2.6 Update of DataRAM.Remaining Capacity (RM) and DataRAM.Full Charge Capacity (FCC)

Update of RM and FCC takes place after each resistance grid-point update, at the end of discharge, and at the exit of relaxation mode.

FCC consists of three parts:

*DataRAM.Full Charge Capacity( )* = $Q_{start}$+ PassedCharge + RM

Components of FCC are indicated in an example in Figure 2-4.

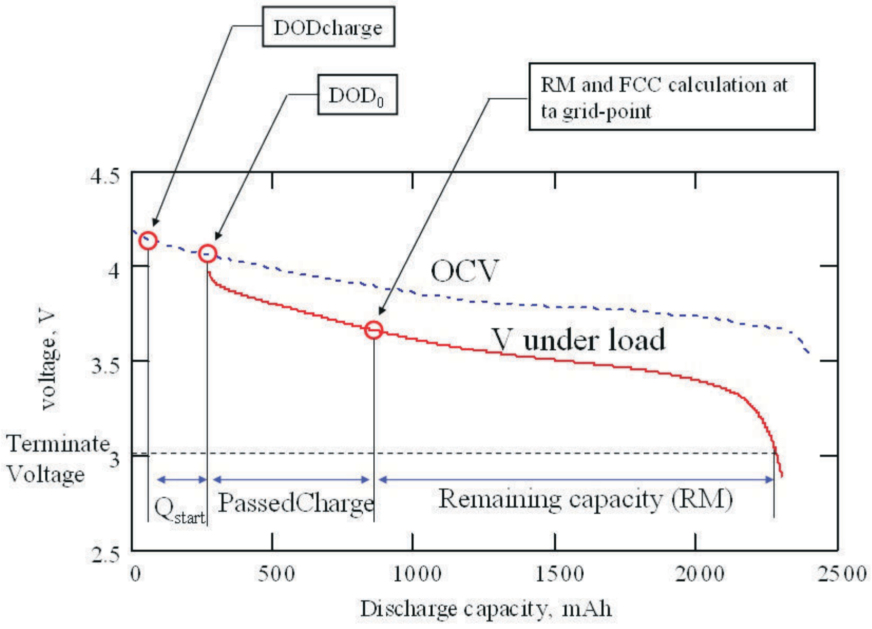


**Figure 2-4. Components of *DataRAM.Full Charge Capacity( )* Value**

1. $Q_{start}$ is the charge that would have passed to make DOD = $DOD_0$ from a fully charged state (DODcharge). For a fully charged battery $Q_{start}$ = 0. $Q_{start}$ is recalculated at the exit of relaxation mode. In the case of constant current, it is simply $Q_{start} = Q_{max} \times (DOD_0 – \text{DODcharge})$, but for the constant power case a voltage simulation is run. DODcharge is assigned equal to $DOD_0$ at first $DOD_0$ update after charge termination by taper current. Note that DODcharge is somewhat higher than 0 because chargers typically do not charge a battery to full.
2. PassedCharge is the coulomb count integrated during the present discharge or charge and set to zero at every $DOD_0$ update.
3. Remaining capacity is calculated after each resistance grid-point update and at the end of discharge

*DataRAM.Remaining Capacity( )* (RM) is calculated using a voltage simulation. The GG starts a simulation at the present DODstart = $DOD_0$ + PassedCharge/$Q_{max}$ and continues calculating voltage V(DODx,T) = OCV(DODx,T)+ I×R(DODx,T) by incrementing DOD with dDOD increment of 4%. DOD[i]=DODstart + dDOD×I. This incrementing is continued until the simulated voltage V(DOD[i]) becomes less than *DF.Terminate Voltage*. Once that happens, the final DOD is revealed. *DataRAM.Remaining Capacity( )* = (DODfin-DODstart)×$Q_{max}$.

Current that is used in the simulation is the average current during the present discharge (several types of averaging can be selected using *DF.Load Select* data flash constant). A simulation can run in constant current mode (*DF.Load Mode* = 0) or constant power mode (*DF.Load Mode* = 1).

## 2.7 Update of DataRAM.Remaining Capacity( ) and DataRAM.State Of Charge( ) Values

Whereas *DataRAM.Full Charge Capacity( )* is only updated at a few points during a discharge as previously described, the *DataRAM.Remaining Capacity( )* is updated continuously (every 1 second) based on the integrated charge. *DataRAM.Remaining Capacity( )* = RM – Q_integrated where Q_integrated is charge passed since the last RM calculation. The value of *DataRAM.Remaining Capacity( )* is also used to update relative *DataRAM.State Of Charge( )* every second as relative *DataRAM.State Of Charge( )* = *DataRAM.Remaining Capacity( )* × 100/ *DataRAM.Full Charge Capacity( )*.

The same value is used to calculate run time to empty as *DataRAM.Time To Empty( )* = *DataRAM.Remaining Capacity( )* / *DataRAM.Average Current( )*.

Note that even if a simulation of RM is run in constant power mode (*DF.Load Mode* = 1), the reporting of *DataRAM.Remaining Capacity( )* and *DataRAM.Time To Empty( )* can be done in both mAh or in 10-mWh values. These values are always reported in mAh or are derived from mAh values:

- *DataRAM.Remaining Capacity( )*
- *DataRAM.Full Charge Capacity( )*
- *DataRAM.Time To Empty( )*
- *DataRAM.Nominal Avail. Capacity( )*
- *DataRAM.Full Available Capacity( )*
- *DataRAM.State of Charge( )*

These values are always reported in mWh or derived from mWh values:

- *DataRAM.Available Energy( )*
- *DataRAM.Average Power( )*
- *DataRAM.TimeToEmpty Const. Power( )*

In case of constant power mode (*DF.LoadMode* = 1), the run time to empty is calculated as *DataRAM.TimeToEmpty( )* = *DataRAM.AvailableEnergy( )* / *DataRAM.AveragePower( )* and is generally more accurate for most devices because of increased power consumption at low voltages.

## 3 Real Application Example

### 3.1 GSM Smart Phone Application

Unlike notebook battery applications with a relatively constant load profile, cell phone/smart phone or PDA, depending on the actual communication protocol (GSM, CDMA, 3G GSM or 3G CDMA), has a pulsating load profile. Questions for this type of application are always related to how the Impedance Track™ can accurately predict the battery remaining capacity. What is the impact of the pulsating current? In this example, a GSM smart phone is used to check out the Impedance Track™ accuracy.

As previously mentioned, both the $Q_{max}$ and Resistance can be updated only when certain criteria are met. Under a pulsating current, an accurate OCV reading may be difficult to achieve. A discussion of careful examination of these criteria follows.

First of all, it has to be made clear that zero current is NOT needed for an OCV reading. Only a LOW current (i.e., less than C/20 rate) is needed. This is common in cell phone/smart phone application when the phone is in standby mode. During the relaxation, a short spike current pulse will not wake-up the gauge from relaxation mode because the *DF.Quit Relax Time = 1* by default. Current has to stay high longer than 1 second to exit the relaxation mode.

Secondly, if a single voltage read point was taken exactly at the moment of the spike, it is ignored because the IT checks if the current exceeded *DF.Dsg Current Threshold*. If it did, then the previous OCV reading is used. As shown in Figure 2-3, these reading are stored in RAM.

The GSM phone used in this testing is a registered phone. During testing, actual phone calls along with applications such as playing games, checking emails, as well as standby mode, are used. The battery voltage and the discharge current are logged with time stamp. The reported RSOC by IT is also logged.

As shown in Figure 3-2, the current reported by IT is being averaged by both a analog-to-digital converter (ADC) in the device as well as by IT algorithm. Although the GSM phone is running on GSM protocol* but the reported peak current is around 200 mA to 300 mA when the phone's LCD screen is on and only around 80 mA when the screen is off.

*GSM waveform: 1A for 0.48ms and 72mA for 4.76ms

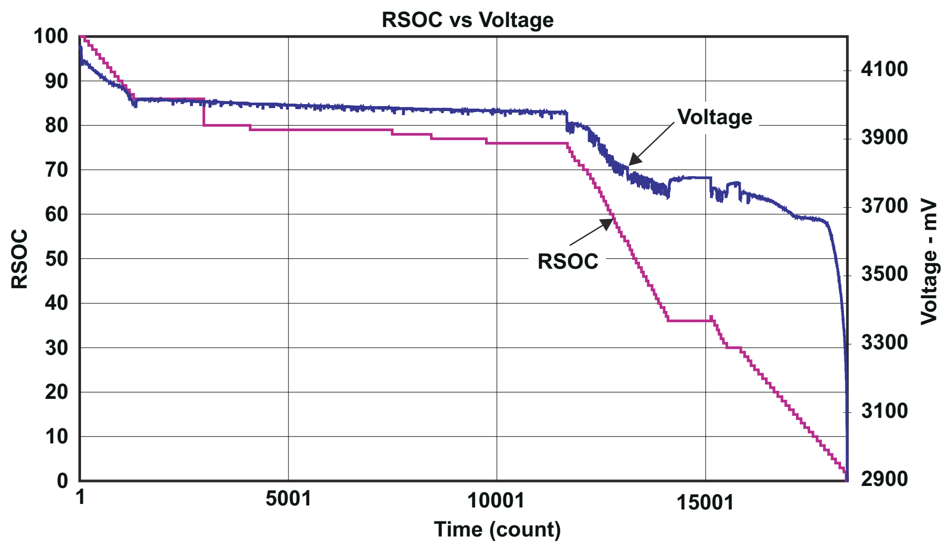


**Figure 3-1. GSM Smart Phone: Relative State of Charge (RSOC) vs Battery Voltage**

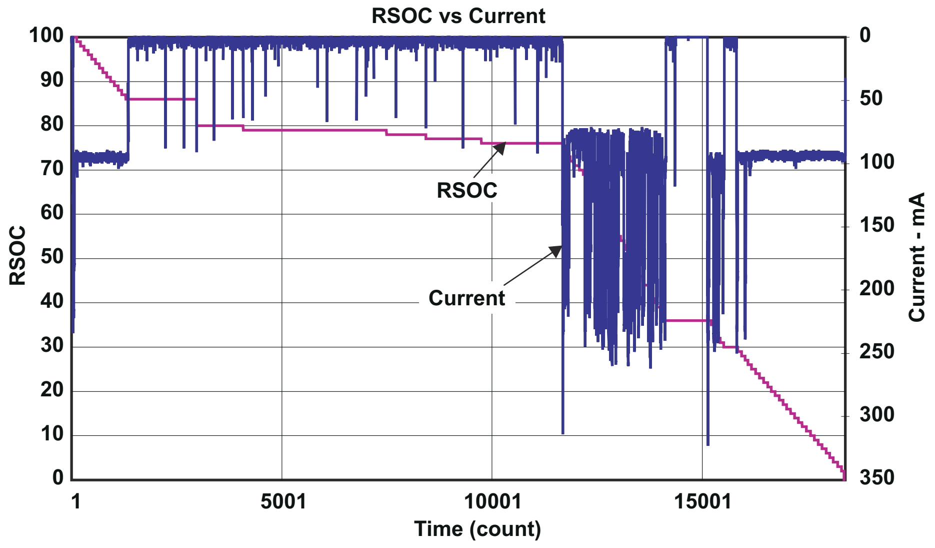


**Figure 3-2. GSM Smart Phone: Relative State of Charge (RSOC) vs Discharge Current**

The RSOC accuracy for this particular test then is calculated from the data log file and Figure 3-3 shows the RSOC accuracy during the whole discharge cycle. As the discharge starts, the error is around 4% but as the phone goes into standby mode, which is around RSOC = 85%, the accurate OCV reading is taken by IT regardless of the high-current spikes (with delay). From RSOC = 80% down to RSOC = 0%, the accuracy is within 2%.

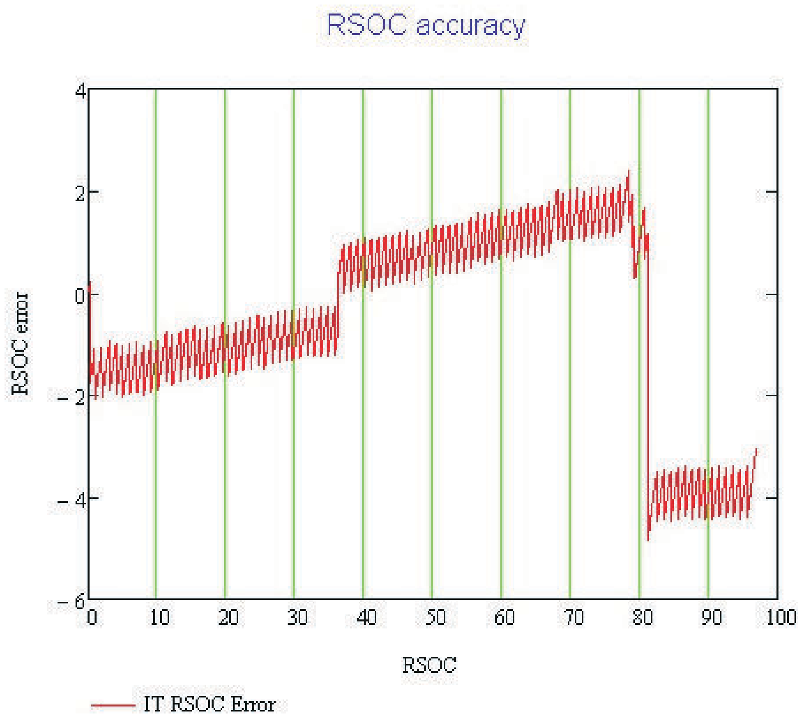
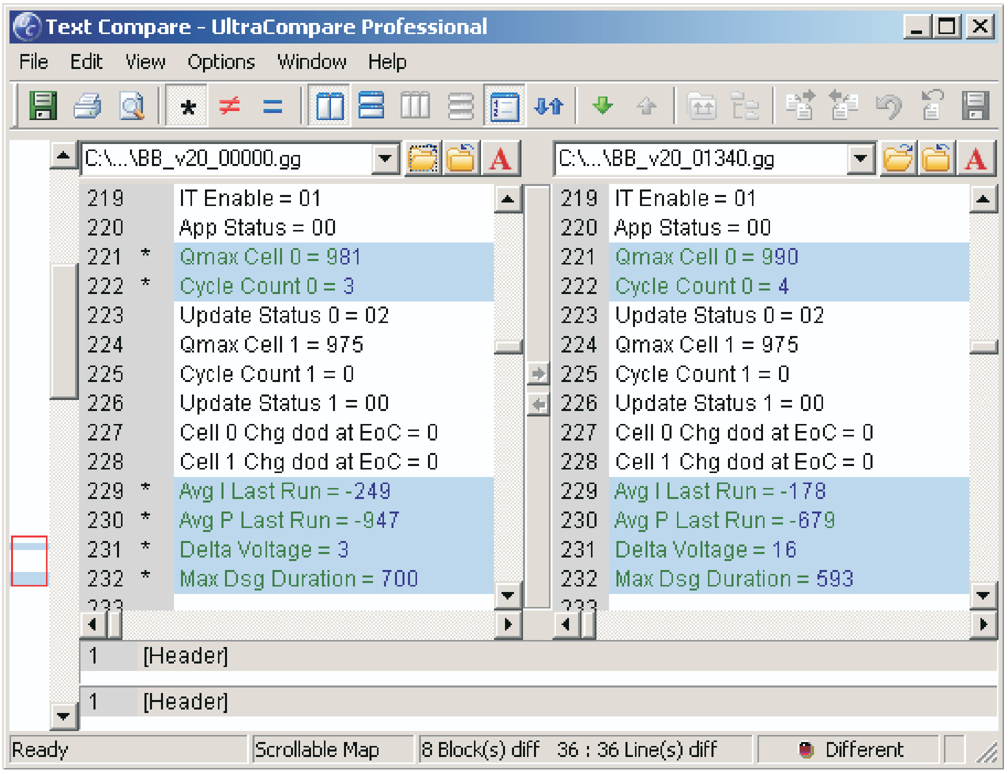


**Figure 3-3. GSM Smart Phone: RSOC Accuracy**

This test has proved that the IT algorithm works as expected even for a pulsating load profile. The high-current spike does not affect the IT accuracy.

After comparing the original data flash contents against the final data flash contents (see Figure 3-4), it is clearly shown that $Q_{max}$ has been updated during the real application test. It has proven that the gauge can meet the $Q_{max}$ update conditions to complete $Q_{max}$ update as well as RSOC update.


Text Compare - UltraCompare Professional
File Edit View Options Window Help
C:\...\BB_v20_00000.gg
C:\...\BB_v20_01340.gg

```
219    IT Enable = 01                 219  IT Enable = 01
220    App Status = 00                220  App Status = 00
221 *  Qmax Cell 0 = 981              221  Qmax Cell 0 = 990
222 *  Cycle Count 0 = 3              222  Cycle Count 0 = 4
223    Update Status 0 = 02           223  Update Status 0 = 02
224    Qmax Cell 1 = 975              224  Qmax Cell 1 = 975
225    Cycle Count 1 = 0              225  Cycle Count 1 = 0
226    Update Status 1 = 00           226  Update Status 1 = 00
227    Cell 0 Chg dod at EoC = 0      227  Cell 0 Chg dod at EoC = 0
228    Cell 1 Chg dod at EoC = 0      228  Cell 1 Chg dod at EoC = 0
229 *  Avg I Last Run = -249          229  Avg I Last Run = -178
230 *  Avg P Last Run = -947          230  Avg P Last Run = -679
231 *  Delta Voltage = 3              231  Delta Voltage = 16
232 *  Max Dsg Duration = 700         232  Max Dsg Duration = 593

1    [Header]
1    [Header]

Ready | Scrollable Map | 8 Block(s) diff   36 : 36 Line(s) diff | Different
```

**Figure 3-4. Flash Data Log Shown $Q_{max}$ Has Been Updated**

## 4 Revision History

| Changes from Revision * (January 2008) to Revision A (November 2022) | Page |
|---|---|
| • Updated the numbering format for tables, figures, and cross-references throughout the document.................. | 1 |