Application Note AN2294

Li-Ion/Li-Polymer Battery Charger with Fuel Gauge Function

Author: Oleksandr Karpin

Associated Project: Yes
Associated Part Family: CY8C24794, CY8C27xxx
PSoC Designer Version: 4.2 SP2
Associated Application Notes: AN2107, AN2258, AN2267, AN2314

Abstract
This Application Note describes fuel gauge technology inside a battery charger. The proposed fuel gauge application is independent of battery chemistry and the quantity of battery cells. Calculation techniques for the various battery operating parameters are described.

Introduction

Rechargeable batteries are used in many applications to power a variety of devices. Accurate fuel gauge information allows consumers to make informed decisions regarding the use of portable power. When the capacity of the battery cell falls below the predetermined threshold, the user is notified of the limited amount of operational time remaining. This protects the device from unexpected shutdown and loss of important data. The main calculation parameters for the fuel gauge are:

- Absolute Capacity (ACR)
- State of Charge (SOC)
- Runtime Remaining in Active Mode
- Runtime Remaining in Suspend Mode
- Full-Charge Time Remaining
- Rapid-Charge Time Remaining

There are two methods to monitor and determine the remaining battery capacity:

1. Voltage measurement-based fuel gauge.
2. Coulomb counter-based fuel gauge.

The first method relies on the correlation between battery voltage and the remaining capacity. The second is based on the integration of all battery charge and discharge currents to determine how much energy is left. For both methods, there are a variety of factors that influence the calculation of remaining capacity.

Such factors include:

- Temperature
- Discharge Rate
- Battery Age
- Self-Discharge
- Battery Chemistry
- State of Charge

Failure to take these factors into account can result in significantly inaccurate readings.

This Application Note describes implementation of an accurate fuel gauge. The unique architecture of the PSoC™ device integrates a hardware solution of a battery charger and the flexible µC-based fuel gauge algorithm with minimal external components at a very affordable price. The PSoC device's non-volatile memory satisfies the fuel gauge memory requirements using various correction lookup tables and storing calculations. All application functionality is realized on one microprocessor, which allows designers to build very small portable devices and to fit the electronics into a small battery package. When the designer wants to update the algorithm with the latest fuel gauge technology, to add the calculation of a new fuel gauge parameter, or to include new correction factors, only the application firmware needs to be modified. The PSoC device's in-circuit and self-programming capabilities make these operations simple.

The technical specifications for a Li-Ion/Li-Polymer battery charger with fuel gauge functionality are listed in Table 1.

The terms and abbreviations used in this Application Note are listed in Table 2.

Table 1. Specifications for Li-Ion/Li-Polymer Battery Charger with Fuel Gauge Functionality

Fuel Gauge Parameters	
Item	**Specification**
Fuel Gauge Remaining Battery Capacity Method	Coulomb Counter-Based Fuel Gauge
Fuel Gauge Calculation Parameters	Absolute Capacity (ACR) State of Charge (SOC) Runtime Remaining in Active Mode Runtime Remaining in Suspend Mode Full-Charge Time Remaining Rapid-Charge Time Remaining
Fuel Gauge Correction	Temperature Discharge Rate Battery Age Rate Self–Discharge
Fuel Gauge State of Charge (SOC) Measurement Error	1%
Battery Charger Parameters	
Item	**Specification**
Built-In Battery Charger Type	Single-Cell Li-Ion/Li-Polymer Battery Charger
Power Supply Voltage	5…5.25V
Power Consumption	20 mA
Battery Temp. Measurement Range (Re-Configurable in Software)	-20…60°C
Battery Current Measurement Error (Not Calibrated)	5%
Battery Voltage Measurement Error (After Calibration)	0,5%
Temperature Measurement Absolute Error	1°C
User Interface	2 Buttons and 3 State LEDs
PC Communication Interface	RS232
PC Communication Speed	115200

Table 2. Terms and Abbreviations

Item	Description
Absolute Capacity (ACR)	The actual battery charge value (fuel gauge Coulomb counter value or Accumulated Current Register ACR), in coulombs.
State of Charge (SOC)	The status of the battery charge (ACR) between empty and full points in percent.
Discharge Rate	The amount of current taken from the battery over a period of time.
Active Mode	The battery is discharged at a fast rate (large load is applied). For example, when a mobile phone transmits a signal.
Suspend Mode	The battery is discharged at a slow rate. Discharge current in suspend mode is much smaller then in active mode. For example, when a mobile phone displays data on the LCD.
Runtime Remaining in Active Mode[1]	The amount of time that remains for the device to work in active mode. Note that during the charge phase, calculation of this parameter is based on a predefined constant average value and during the discharge phase the calculation is based on the actual active discharge current.
Runtime Remaining in Suspend Mode	The amount of time that remains for the device to work in suspend mode. Note that in suspend mode, battery energy is conserved to minimize battery drain and the processor stays in sleep mode. Therefore, calculation of this parameter is based on the predefined constant average value.
Full-Charge Time Remaining	The amount of time that remains between the battery being fully charged and the actual battery charge state.
Rapid-Charge Time Remaining	The amount of time that remains from current battery charge state to the completion of rapid charge state.
Open Circuit Voltage (OCV)	The voltage on the battery with no load applied.
Full Charge	Fully charged battery capacity (ACR value when battery is fully charged).
Active Empty	Fully discharged battery residual capacitance by using active mode rate (ACR value when battery is fully discharged).
Suspend Empty	Fully discharged battery residual capacitance by using suspend mode rate (ACR value when battery is fully discharged).
Breakpoint	Some point that is used for interpolation of the ACR curve during the charge phase. In this project, it is measured from breakpoint to full charge.
Fuel Gauge Learning Charge Cycle	The full battery charge process that immediately follows a full discharge. Note that after completion of the fuel gauge learning charge cycle, the ACR value is equal to the total battery capacity at the present temperature.

[1] In this Application Note, a load model with constant current supply is selected. If the applied load has constant power dissipation (the active current is increased when the cell voltage drops) then the remaining time is not directly related to the remaining charge. It is related to the remaining energy, which can easily be calculated by reading battery voltage.

Battery Capacity Monitoring Methods

Voltage-Based Fuel Gauge

This is the earliest known method to monitor battery capacity. It is based on the correlation between battery voltage and remaining capacity. Thus, if the no load voltage (open circuit voltage, OCV) has dropped to a certain level, the capacity will drop by a corresponding amount. This is only true if no load is applied to the battery during measurement. When a load is applied, the battery voltage is distorted by the voltage drop due to the battery's internal impedance. This is a typical situation when trying to determine remaining capacity. Equation (1) shows the voltage drop value on the battery's internal resistance:

$$V_r = I \cdot r \qquad \textbf{(1)}$$

V_r is the voltage drop on the battery's internal resistance. I is the current through the battery. r is the internal resistance.

Moreover, even when the load is removed, the relaxation processes inside the battery continue to change the voltage for a period lasting from 30 minutes to several hours. To compensate for internal resistance, the following must be considered:

1. The internal resistance depends on the SOC. This value is more than 10 times higher in the discharged state than in the charged state. Therefore, the average value of internal resistance cannot be used during the calculation.
2. The internal resistance depends on temperature. It increases about 1.5 times with every 10°C decrease. Therefore, a multi-dimensional lookup table of internal resistance compensation must be used. The value of internal resistance depends on temperature, state of charge, and discharge rate.
3. There may be a variation between impedance in different cells of up to 15%. This variation produces a significant difference in voltage correction at high loads.
4. Internal impedance changes with age. A typical Li-Ion/Li-Polymer battery doubles its internal impedance in 70-100 cycles, while its no-load capacity decreases during the same time period by only 2-3%.

   If this effect is not taken into account, the voltage-based fuel gauge algorithm will give values that are 30-50% inaccurate after approximately 20-25% (500 cycles) of its estimated lifespan.

Furthermore, the problem with Li-Ion/Li-Polymer batteries also appears between OCV and SOC correlation. Li-Ion/Li-Polymer batteries have a relatively flat discharge rate. The voltage remains essentially constant until the battery is nearly fully discharged. Therefore, it is difficult to provide precise measurement and correlation of the open circuit voltage with the state of charge.

Coulomb Counter-Based Fuel Gauge

A more accurate method to monitor battery capacity involves the use of a Coulomb counter-based fuel gauge. This fuel gauge is based on the integration of cumulative battery charge and discharge currents to determine how much energy is left. The cumulative battery currents are measured in coulomb (current per unit of time). The quantity of coulombs can be calculated as:

$$Q = I \cdot t \qquad \textbf{(2)}$$

The fuel gauge periodically monitors the current at known intervals. There is an assumption that the current does not change or changes very little during the sample period. The current going out of the pack is subtracted from a pack capacity value and the current going into the pack is added. This value is reset after each full charge or full discharge. The reset value depends on many factors including temperature, self-discharge, age, discharge rate, etc.

Using the Coulomb counter-based method presents the following problems:

1. It is difficult to obtain precise measurement of the low standby current.
2. Even when a portable device is completely off, the rechargeable battery continues to discharge a small amount of current over time due to the internal shunt resistance always present in the battery (self-discharge).
3. Total battery capacity decreases with age. The value of the total capacity is updated only if a full charge occurs immediately after a full discharge. But the average portable device user does not ever fully discharge his battery, so actual battery capacity may be considerably decreased before the fuel gauge updates its internal value.

4. The battery state of charge (SOC) is dependent upon temperature and discharge rate.

These problems can easily be solved and, for example, in certain applications this technique can yield accuracy of 1% or better. Therefore, in this application, the fuel gauge is based on the Coulomb counter-based method.

Fuel Gauge Fundamentals

The following section describes the fundamentals of the Coulomb counter-based fuel gauge method used in this application. Figure 1 depicts the fuel gauge technique flowchart.

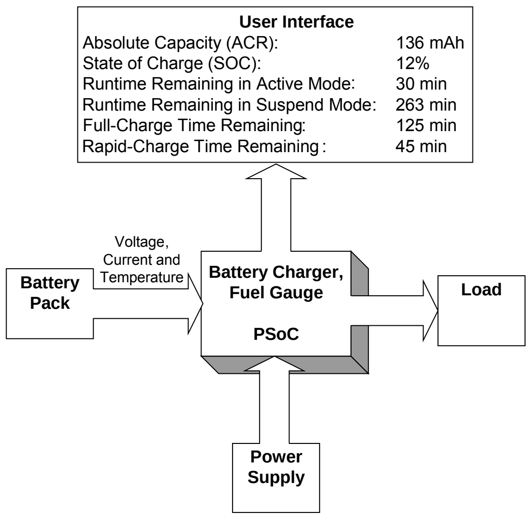

User Interface
Absolute Capacity (ACR): 136 mAh
State of Charge (SOC): 12%
Runtime Remaining in Active Mode: 30 min
Runtime Remaining in Suspend Mode: 263 min
Full-Charge Time Remaining: 125 min
Rapid-Charge Time Remaining: 45 min
Voltage, Current and Temperature
Battery Pack
Battery Charger, Fuel Gauge
PSoC
Load
Power Supply

Figure 1. Fuel Gauge Technique Flowchart

The fuel gauge periodically receives the battery voltage, current, and temperature values. Based on these values the fuel gauge produces:

- Absolute Capacity (ACR)
- State of Charge (SOC)
- Runtime Remaining in Active Mode
- Runtime Remaining in Suspend Mode
- Full-Charge Time Remaining
- Rapid-Charge Time Remaining

To achieve the desired accuracy, compensation factors are continually applied to account for internal battery characteristics and environmental conditions. Figure 2 shows the discharge parameters for operation of the fuel gauge.

The accumulated current register (ACR) gathers current flow through the battery and shows the total current flowing in and out of the battery pack during the entire life of the battery. The “gathering” operation is performed once per second. For improved accuracy, the ACR uses the average current value during this sample period (IchAvrg, Figure 2).

Accurate calculation of fuel gauge parameters requires consideration of internal battery characteristics and environmental conditions. Factors affecting battery capacity and state of charge include:

- Self-Discharge
- Temperature
- Discharge Rate
- Cell Aging

Therefore, correcting the influence of all of these factors must be implemented in the fuel gauge algorithms.

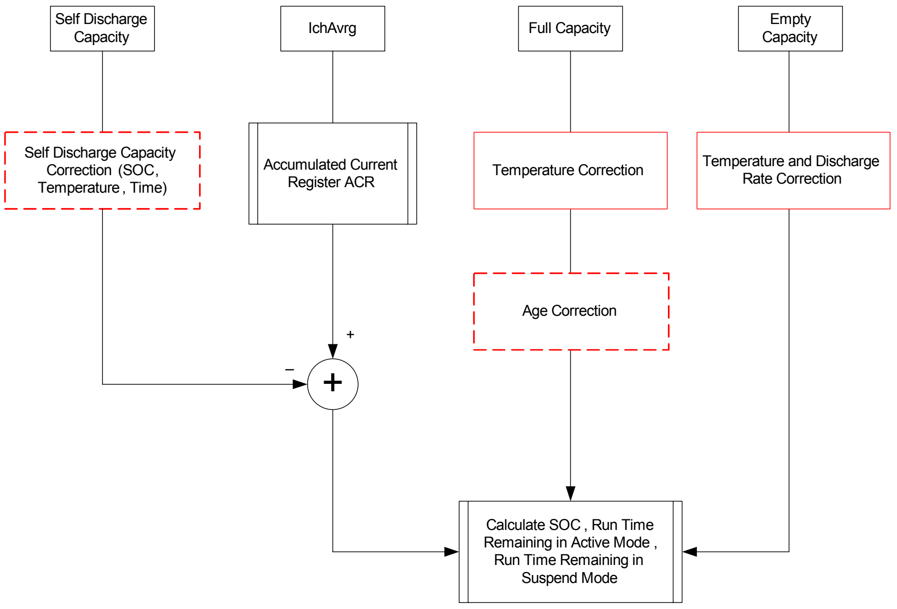

Self Discharge Capacity
IchAvrg
Full Capacity
Empty Capacity
Self Discharge Capacity Correction (SOC, Temperature, Time)
Accumulated Current Register ACR
Temperature Correction
Temperature and Discharge Rate Correction
Age Correction
+
–
Calculate SOC, Run Time Remaining in Active Mode, Run Time Remaining in Suspend Mode

Figure 2. Discharge Parameters for Fuel Gauge Operation

Self-Discharge

Self-discharge is caused by parasitic leakage. Its value depends upon time, temperature, SOC, and uses a non-linear transfer function (see Figure 3). Therefore, model data relating to self-discharge should be stored in a multi-dimensional lookup table. While a portable device is running, the self-discharge will be subtracted at a rate determined from the self-discharge table. Self-discharge must always be calculated and subtracted regardless of battery activity (regardless of whether the battery is discharging or idle), because a self-discharge internal shunt resistance is constantly present in the battery.

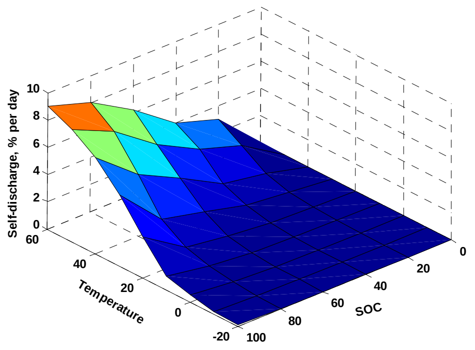


Figure 3. Self-Discharge Lookup Table Example

Self-discharge for some Li-Ion/Li-Polymer batteries is extremely low (near 3% per month) and difficult to measure.

The impact of this self-discharge on the ACR result is less than the accuracy of the charge measurement. Therefore, it can be completely ignored in the fuel gauge equations.

To obtain self-discharge empirical data:

1. Establish fuel gauge battery capacity.
2. Set battery to fixed SOC.
3. Store at set temperature for the fixed period (for example, 1 day).
4. Discharge battery and determine retained capacity.
5. Repeat steps 2-4 at other temperatures.
6. Repeat steps 2-5 for other SOCs.

Temperature and Discharge Rate

Battery capacity varies greatly, depending on temperature and discharge rate. Figure 4 shows a temperature and discharge rate lookup table example.

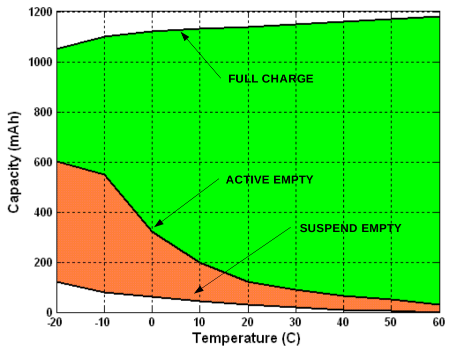


Figure 4. Temperature and Discharge Rate Lookup Table Example

In Figure 4, the FULL CHARGE line represents the relationship between the battery's total capacity and temperature. The ACTIVE EMPTY and SUSPEND EMPTY lines represent the fully discharged residual capacitance versus temperature using active and suspend discharge rates, respectively. Note that for applications where the battery can be discharged by more than two discharge rates, this lookup table must be expanded.

If the battery is fully charged at 60°C and then fully discharged at 0°C at the active current rate, the amount of the charge that can be removed from the battery would be equal to:

$$Q = FULL\left[60^{0}C\right] - EMPTY_{ACTIVE}\left[0^{0}C\right] \quad \textbf{(3)}$$

But this fully discharged battery at 0°C would not be fully discharged at the higher temperature. For example, the battery can then be discharged at 40°C on an amount of charge equal to:

$$Q = EMPTY_{ACTIVE}\left[0^{0}C\right] - EMPTY_{ACTIVE}\left[40^{0}C\right] \quad \textbf{(4)}$$

If the battery is partly discharged at temperature T_1 and I_1 and then discharged completely at temperature T_2 and rate I_2, $EMPTY_{I2}[T_2]$ will be defined as the battery discharged state. Similarly, a battery that is fully discharged at a high current rate can be further discharged at a low current rate by an amount equal to the difference between the ACTIVE EMPTY and SUSPEND EMPTY points.

Therefore, in order to determine the remaining battery capacity, it is only necessary to know the present temperature and discharge rate.

To conserve memory, the temperature-increment is set to 10°C and linear interpolation between the data points is made. The parameter value *P* at temperature *T* is calculated by Equation (5):

$$P_T = P\left[Lo\right] - (T - Lo)\cdot\frac{P\left[Hi\right] - P\left[Lo\right]}{10} \quad \textbf{(5)}$$

Lo and Hi are the incremental values closest to temperature *T* ($Lo = 10 \cdot N$, $Hi = 10\cdot(N+1)$ and N is the signed integer value).

Procedure for obtaining temperature and discharge rate empirical data:

1. Discharge the battery to the fully discharged state at the highest temperature with the minimum possible discharge rate. Reset ACR. (This is the lowest possible battery charge empty point and should be set equal to zero.)
2. Set the fixed environment temperature (start with highest temperature).
3. Charge battery to full state and record ACR value to FULL CHARGE at this temperature.
4. Discharge battery at fixed rate (start with the maximum rate).
5. Record ACR value as the empty point for this temperature and discharge rate.
6. Repeat steps 4 and 5 for other discharge rates (continue to discharge for all required discharge rates in decreasing order).
7. Repeat steps 2-6 at other temperatures.

Cell Aging

With aging, the battery cell loses its ability to store a charge. For example, Li-Ion battery cells have a self-degradation rate near 30% at 500 cycles. Figure 5 shows battery capacity based on age.

To account for this, it is necessary to periodically recalculate total battery capacity (FULL CHARGE). The fuel gauge learning charge cycle is used for this purpose.

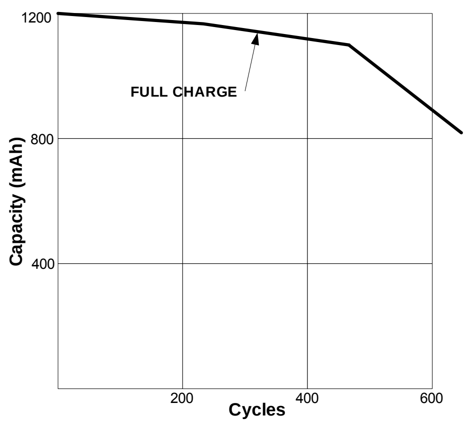


Figure 5. Cell Aging

The fuel gauge learning charge cycle is the full battery charge process that immediately follows a full discharge. After the fuel gauge learning charge cycle, the ACR value is equal to total battery capacity (correct FULL CHARGE point) at current temperature.

The FULL CHARGE point for other temperatures is proportionally recalculated using total battery capacity and current temperature. For accurate fuel gauging, the learning charge cycle must be performed under the following circumstances:

1. When the new battery is attached to the device.
2. When the ACR value, in fully charged battery state, differs from the expected (FULL CHARGE) predefined value (for example, 2%). Note that this condition must be used only with self-discharge fuel gauge correction.
3. When the charge cycle count after the last routine is greater than some predefined value (for example, 100).

If the fuel gauge detects one of the above conditions, it notifies the user of a decrease in accuracy and therefore, it recommends that the learning charge cycle routine be performed. The algorithm for the learning charge cycle is described in detail in the Fuel Gauge Algorithms section.

To provide sufficient fuel gauge accuracy it is necessary to interpolate aging between two learning cycles. For this purpose, we calculate the cell age coefficient. This shows battery capacity aging per one cycle. The equation of the cell age coefficient is:

$$CoefAGE = \frac{\dfrac{FULL_{OLD} - FULL_{NEW}}{FULL_{OLD}}}{Cycles} \quad \textbf{(6)}$$

$FULL_{NEW}$ is the new FULL CHARGE value (equal to the ACR after the learning charge cycle). $FULL_{OLD}$ is the previous FULL CHARGE value at the present battery temperature. $Cycles$ is the count of cycles between the last two fuel gauge learning charge cycle routines.

The cell age coefficient is updated and stored in non-volatile memory each time the learning charge cycle is completed. Its value can be greater than or equal to zero. Cell age correction for the FULL CHARGE parameter is performed in Equation (7):

$$FULL_{AGE} = FULL - FULL \cdot CoefAGE \cdot Cycles \quad \textbf{(7)}$$

$Cycles$ is the count of cycles after the last learning charge cycle routine.

Note that cell age correction must only be performed with self-discharge correction or when the learning charge cycle is performed immediately after battery fully discharged state (without idle state). If the fuel gauge learning charge cycle is performed frequently enough, calculation of the cell age coefficient can be completely ignored in the fuel gauge equations.

Discharge Parameter Calculation
The main discharge parameters are:

- State of Charge (SOC)
- Runtime Remaining in Active Mode
- Runtime Remaining in Suspend Mode

The SOC shows the location, in percentage, of the ACR between the empty and full points. Equation (8) shows the SOC calculation with a proposed correction:

$$SOC = \frac{ACR[S] - EMPTY[T,I]}{FULL[T,AGE] - EMPTY[T,I]} \cdot 100\% \quad \textbf{(8)}$$

[S] is the self-discharge correction. [T] is the temperature correction. [I] is the discharge rate correction. [AGE] is the age correction.

The runtime remaining in active mode is calculated by Equation (9):

$$t_{ACTIVE} = \frac{ACR[S] - EMPTY[T, I_{ACTIVE}]}{I_{ACTIVE} \cdot 60} \quad \textbf{(9)}$$

The upper fraction is measured in [Ah*sec] and the division by $I_{ACTIVE} \cdot 60$ provides, in minutes, the result of the remaining time in active mode. In a similar manner, the runtime remaining in suspend mode is calculated by Equation (10):

$$t_{SUSPEND} = \frac{ACR[S] - EMPTY[T, I_{SUSPEND}]}{I_{SUSPEND} \cdot 60} \quad \textbf{(10)}$$

Charge Parameters

Figure 6 shows the charge time parameters for fuel gauge operation. Correct calculation of the fuel gauge charge parameters requires accurate interpolation of the ACR curve during the charge phase. As a rule for this purpose, a breakpoint is selected on the ACR curve to interpolate the curve by two lines. For example, the Li-Ion/Li-Polymer charge current profile has a breakpoint between the constant current (CC) and constant voltage (CV) charge phases (see Figure 7). This can be used to plot the ACR curve interpolation (see Figure 8).

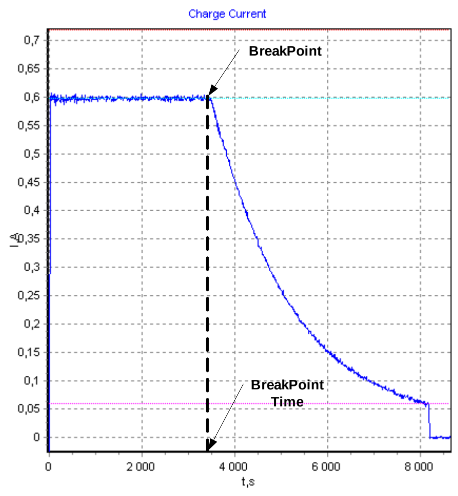


Figure 6. Charge Current Profile

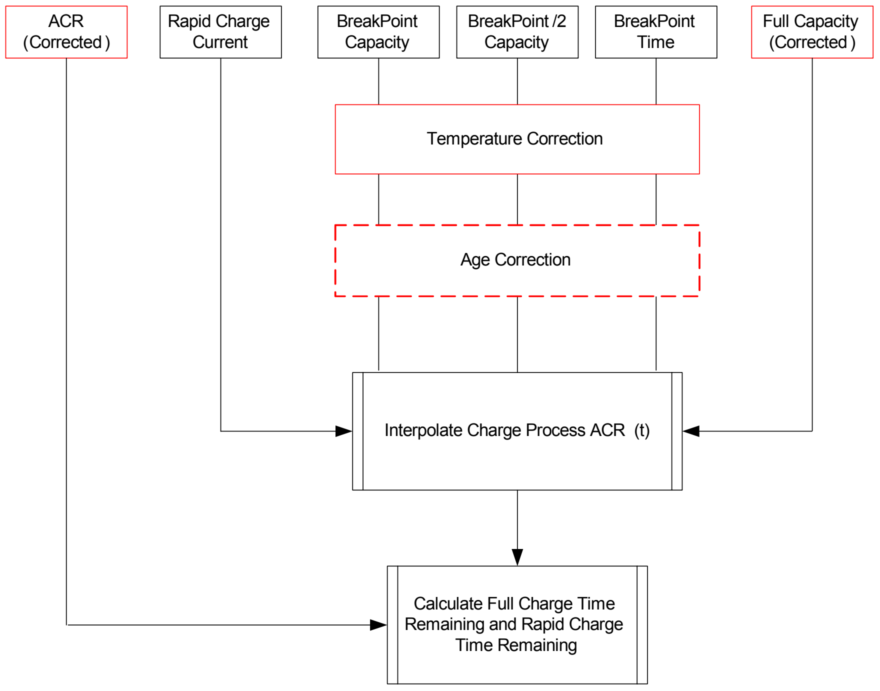

ACR (Corrected)
Rapid Charge Current
BreakPoint Capacity
BreakPoint /2 Capacity
BreakPoint Time
Full Capacity (Corrected)
Temperature Correction
Age Correction
Interpolate Charge Process ACR (t)
Calculate Full Charge Time Remaining and Rapid Charge Time Remaining

Figure 7. Charge Parameters for Fuel Gauge Operation

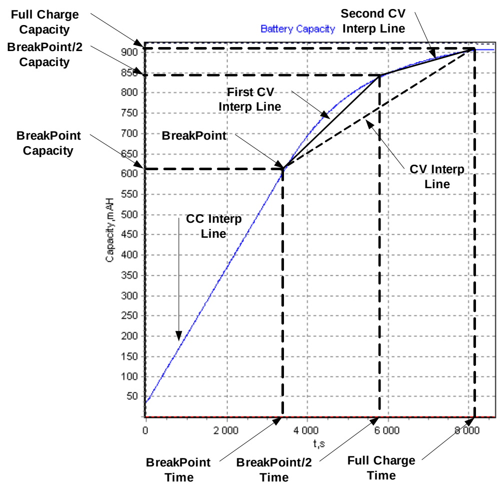


Figure 8. ACR Curve During Charge Phase

The constant current interpolation line (CC Interp Line) accurately reproduces the original ACR curve. But the second interpolation line (CV Interp Line) from the breakpoint to the end of the charge does not accurately reproduce this curve. Therefore, this ACR curve should be estimated using two interpolated lines (First and Second CV Interp Line) with equal time duration (BreakPoint/2 Time). Such a curve approximation provides a high level of accuracy during the end of the charge phase.

Procedure for obtaining the charge process interpolation empirical data:

1. Discharge battery to the fully discharged state (using any allowable discharge rate and environment temperature).
2. Set the fixed environment temperature.
3. Charge the battery to the full state.
4. Determine Breakpoint Time on the charge current profile.
5. Determine Breakpoint Capacity, Breakpoint/2 Capacity, and Breakpoint Time for this temperature on the battery's capacity (ACR) profile.
6. Repeat steps 1-5 at other temperatures.

To save time, this procedure can be combined with the procedure for obtaining temperature and discharge rate empirical data.

Note that to simplify the charge parameter calculation, the Breakpoint Capacity and Breakpoint Time should be measured from the breakpoint to the full charge point. The Breakpoint/2 Time is equal to half of the Breakpoint Time. The Breakpoint/2 Capacity corresponds to the ACR value at Breakpoint/2 Time point to the point of full charge.

Charge Parameter Calculation
The main charge parameters are:

- Full-Charge Time Remaining
- Rapid-Charge Time Remaining

The rapid-charge time remaining is calculated in Equation (11):

$$t_{RAPID} = \frac{FULL[T, AGE] - BREAK[T] - ACR[S]}{I_{RAPID} \cdot 60} \quad \textbf{(11)}$$

[S] is the self-discharge correction. [T] is the temperature correction. [AGE] is age correction.

To calculate the full-charge time remaining, the interpolation line with current ACR position is first found. Next, the charge time remaining on this line is calculated. If this is the CC interpolation line, then Equation (11) is used, otherwise Equation (12) is used:

$$t_l = BREAK\_TIME/2[T] - \frac{BREAK\_TIME/2[T] \cdot ACR_{LINE}}{LINE\_LENGTH} \quad \textbf{(12)}$$

t_l is the remaining time on the line. ACR_{LINE} is the current ACR value with an offset at the start line. $LINE\_LENGTH$ is the interpolation line length.

The value of remaining full-charge time consists of the remaining time on the interpolation line and the time after this line to the end of charge. This algorithm is shown in detail in the Fuel Gauge Algorithms section (in the FGUpdate module).

Battery Charger Hardware

Figure 9 depicts the structure of a Li-Ion/Li-Polymer battery charger with fuel gauge function. The technique for charging Li-Ion and Li-Polymer batteries is explained in detail in Application Note AN2107 "Multi-Chemistry Battery Charger." Similar battery chargers are explained in detail in Application Notes AN2267 "Single Cell Li-Ion Battery Charger" and AN2258 "Cell Balancing in a Multi-Cell Li-Ion/Li-Polymer Battery Charger." The fuel gauge technology is realized in software and is independent of battery chemistry and of the quantity of battery cells. Therefore, the hardware can be upgraded for a multi-cell battery charger (see Application Note AN2258), or the fuel gauge technology can be implemented in a multi-chemistry battery charger (see Application Note AN2107).

The following abbreviations are used in Figure 9:

RS_TX – RS232 transmitter for debug purposes (uses external level translator). It monitors temperature, voltage, current and fuel gauge statistics. RS_TX is used only during debugging and may be removed in the released product.
CPU – Central processor to implement charge and fuel gauge algorithms, and perform charge control functions.
PWM – Pulse width modulator to regulate charge current.
TIMERs – Several timers are used by the CPU in the charge and fuel gauge algorithms.
ADC – Analog-to-digital converter to digitize the analog signals.
INSAMP – Instrumentation amplifier to measure charge voltage, current and temperature.
AMUX – analog multiplexers.

In Figure 9, there is also a Li-Ion battery pack that consists of 1 cell, a linear regulator (based on Q1), a current-sense resistor, and other elements that allow the PSoC device to use/interpret battery current, voltage and temperature.

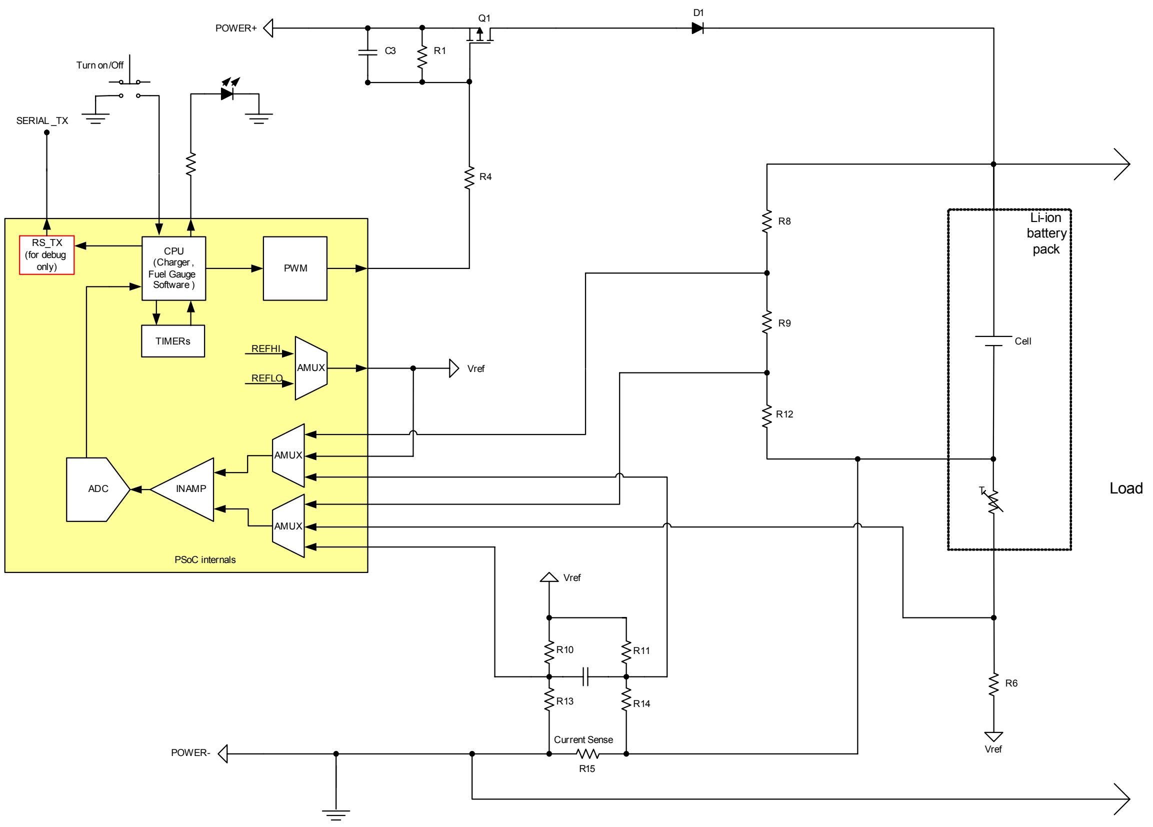


Figure 9. Structure of Li-Ion/Li-Polymer Battery Charger with Fuel Gauge Function

Device Schematic

The complete battery charger schematic is shown in Figure 10 and Figure 11.

The signal from the PWM goes to the RC-filter, which consists of 2 resistors (R1, R4) and 1 capacitor (C3). A constant voltage signal forms on the Q1 gate and is proportional to the PWM duty cycle value. Thus, the PWM with RC-filter is a PWM-DAC. The MOSFET Q1 regulates the battery charge current and is driven by the analog signal from the PWM-DAC. The PWM period is set to 2048 in order to accurately represent the charge current level, and can easily be adjusted in the firmware.

The diode D1 is used for to prevent backward current that could discharge the battery when the charger is disconnected from the supply voltage. The resistive network (R6-R14) allows for transformation of the battery current, voltage and temperature into levels that are suitable for the PSoC device's signals. The 100-mΩ resistor R15 is a current-sense resistor that is in the battery pack current path.

The charger user-interface is implemented using two buttons, SW1 and SW2, to control the process, and three LEDs to display the internal status.

In this application configuration, the green LED indicates the charge phase, the yellow LED indicates discharge phase and the red LED indicates error state. The calibration state is indicated by illuminating all three LEDs, and the idle state is indicated by dimming all three LEDs. SW1 is used to turn the device on/off and SW2 is used for test purposes.

The external voltage supply must fall within the range of 5…5.25V because diode D1 is used to prevent battery discharging. The external voltage supply is applied to the connector J3. The switch SW3 allows the device to be disconnected from the external power supply. Two diodes in the package D5 allow the processor to operate during the charge phase from the external power supply and during the discharge phase from the battery power supply.

The external load is connected to the connector J4 LOAD. The diodes D6 and D7 are used to provide an uninterruptible power supply for the LOAD connector in a manner that is similar to that of D5 for the processor. The switch on transistors Q2 and Q3 allow the power supply to be disconnected from the LOAD and protect the battery from over-discharge. The ground level is set equal to the external ground level POWER- during the charge phase or during the discharge phase to the battery pack ground that follows the current-sense resistor. In this way, only the charge battery pack current and the total battery back discharge current are passed through the current-sense resistor. These are used to supplement the battery fuel gauge functionality in the PSoC software.

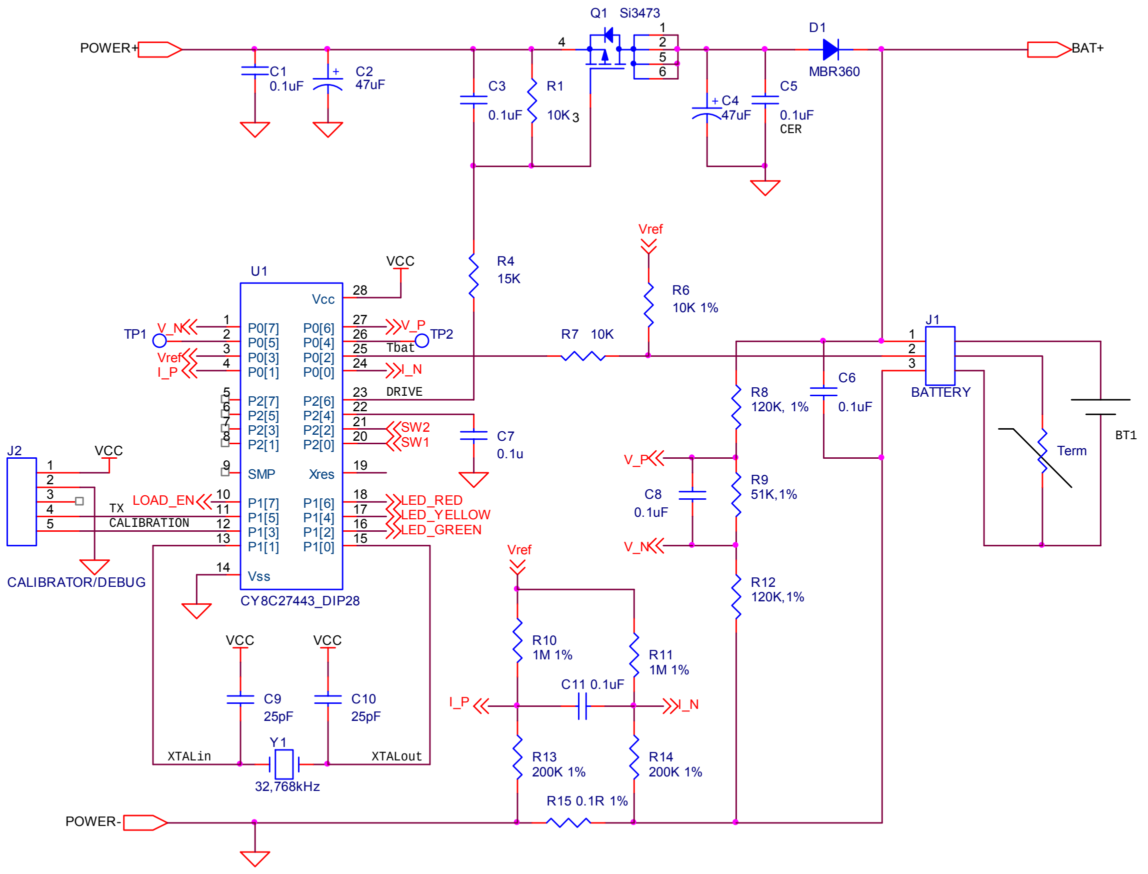


Figure 10. Battery Charger Schematic: Charger, CPU and Measuring Equipment

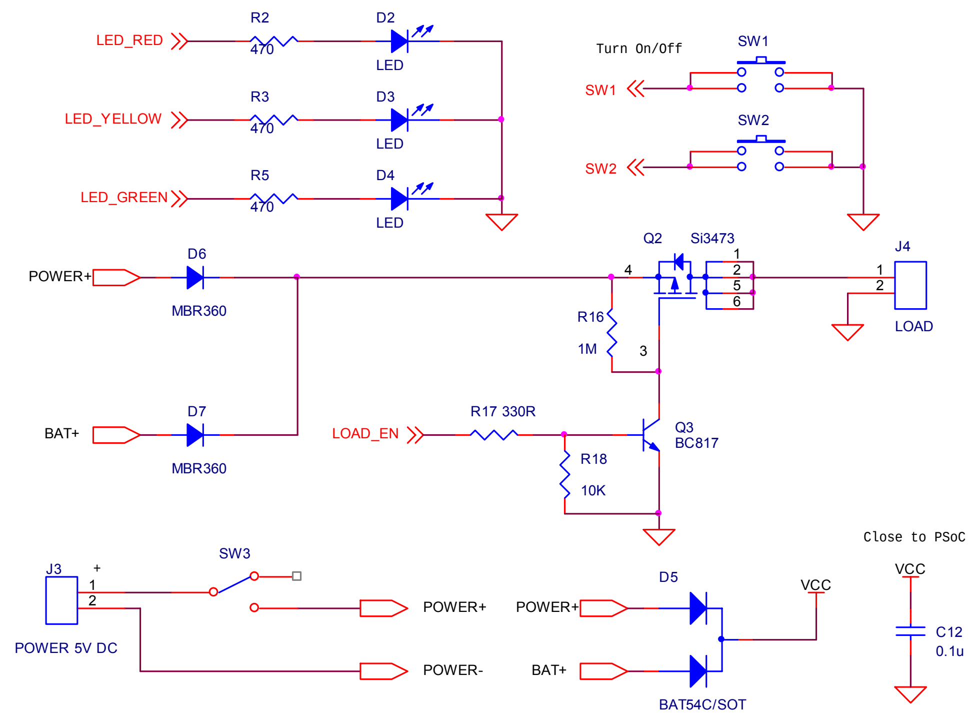


Figure 11. Battery Charger Schematic: Power Supply and User Interface

PSoC Device Internals

The internal structure of the PSoC is shown in Figure 12. The PWM has been placed on DBB00 and DBB01. This user module is configured in software as an 11-bit PWM, which provides for a sufficient number of regulation steps. The TIMER user module is based on the internal Sleep_Timer and configured to generate interrupts every 1s. This real-time clock is used to calculate other time intervals. The serial transmitter is placed on DCB13. The default exchange speed is set to 115200 baud.

In this project the three-opamp topology of the instrumental amplifier (INSAMP) is used. The INSAMP is placed on ACB00, ACB01 and ASD11. The incremental ADC is placed on the ASC10 and DCB02 blocks. The ADC resolution is set to 12 bits, and the integration time is adjusted to be precisely equal an integer number of PWM cycles.

All of the switched capacitor user modules use the same column frequency in order to eliminate aliasing problems. The analog ground bias is set to BandGap or 1.3V (RefMux is BandGap ± BandGap).

Note that if the user needs additional analog pins or requires USB support, he can import this charger to the CY8C24794 chip. The CY8C24794 PSoC device includes a full-featured, full-speed (12 Mbps) USB port and can have up to seven IO ports that connect to the global digital and analog interconnects, providing access to 4 digital blocks and 6 analog blocks. For additional information, go to http://www.cypress.com/psoc and click on CY8C24794 in the parametric table.

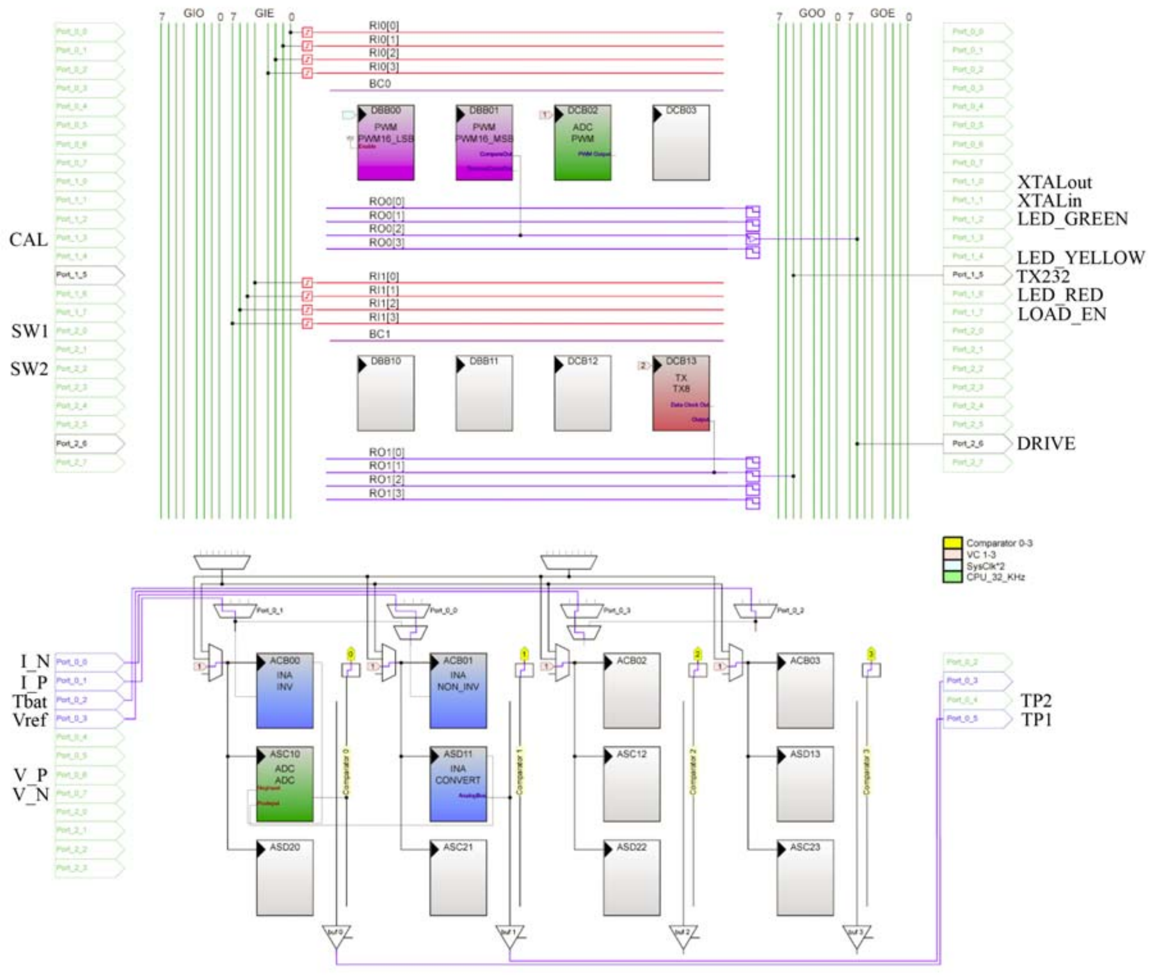


Figure 12. PSoC Internal User Module Configuration

Battery Measurements

To correctly implement the charge and fuel gauge algorithms, the charge current, battery voltage and temperature must be measured accurately.

These three parameters are measured as the voltage drops on the corresponding resistors by using the instrumental amplifier, INSAMP. The measurement is implemented as a two-stage procedure to eliminate any voltage offset from the INSAMP and ADC inputs. The INSAMP inputs are shorted together in the first stage. This state is used to measure the INSAMP and ADC offset voltage. Then the real signal is measured. At this point, the difference between the ADC code corresponding to the first and second stages is directly proportional to the battery measurement parameter without the influence of the INSAMP and ADC offset voltage.

To scale the battery current (voltage drop on the current-sense resistor) and battery voltage into levels suitable for PSoC, precise resistive dividers are used. In order to limit the current flow from the battery to the powered-down battery charger, we employ divider resistors of large nominal resistance.

To provide improved current measurement accuracy, we put a current-sense resistor in the pack current at the negative battery voltage. In this case, the voltage drop on the resistive divider (R10- R14) is independent of the battery pack voltage level. This would not be true if it were placed at the positive voltage. At the beginning of the charging process, we measure the voltage bias on the current-sense resistor and during subsequent processes we subtract it from the measured values. In this way, we partially compensate for the difference between resistor values in the resistive divider and perform current calibration as we go.

The following equation represents the current measurement scheme:

$$\Delta n = n_{\max}\frac{V_{ADC}}{V_{ref}} = n_{\max}\frac{G_{ina}\beta_I I_{bat} R_{sense}}{V_{ref}} \quad \textbf{(13)}$$

Δn is the ADC code without the influence of INSAMP and ADC offset voltage and without the voltage bias on the current-sense resistor ($\Delta n = n_{meas} - n_{offset} - n_{bias}$). $n_{\max}$ is the maximum ADC code, which is equal to 2048 for a 12-bit incremental ADC in bipolar mode. I_{bat} is the battery current. G_{ina} is INSAMP gain (4). V_{ref} is the bandgap reference voltage (1.3V). β_I is the resistive divider coefficient (0,833333):

$$\beta_I = \frac{1}{1+\frac{R_{13}}{R_{10}}} \quad \textbf{(14)}$$

Note that for a Coulomb counter-based fuel gauge, the current measurement error influences only the absolute accuracy of capacity measurement and does not affect the state of charge or remaining time. Therefore, it is not necessary to perform calibration of the current source.

The voltage measurement also is performed by the INSAMP on the corresponding resistor. The following equation depicts the voltage measurement scheme:

$$\Delta n = n_{\max}\frac{V_{ADC}}{V_{ref}} = n_{\max}\frac{G_{ina}\beta_V V_{bat}}{V_{ref}} \quad \textbf{(15)}$$

Δn is the ADC code without influence of the INSAMP and ADC offset voltage $(\Delta n = n_{meas} - n_{offset})$. $n_{\max}$ is the maximum ADC code and the value is equal to 2048 for a 12-bit incremental ADC in bipolar mode. V_{bat} is the battery voltage. G_{ina} is the INSAMP gain (1). V_{ref} is the bandgap reference voltage (1.3V). β_V is the resistive divider coefficient (0.17526):

$$\beta_V = \frac{1}{1+\frac{R_8 + R_{12}}{R_9}} \quad \textbf{(16)}$$

To provide more accurate voltage measurement, the following calibration technique is used. We store all voltage thresholds as calibrated ADC codes and then during operation, the battery voltage (its ADC code) is compared to these values. For this purpose, we use an external precision 4.2V voltage source and calibration procedure after assembly. All voltage thresholds are calibrated from this precision voltage:

$$n_{new} = n_{old}\cdot\frac{n_{4,2V\_new}}{n_{4,2V\_old}} \quad \textbf{(17)}$$

n_{new} is the new voltage threshold ADC code. n_{old} is the old voltage threshold ADC code that is calculated by Equation (15). $n_{4,2V\_new}$ is the input ADC code during the calibration procedure. $n_{4,2V\_old}$ is the old voltage threshold ADC code for 4.2V that is calculated by Equation (15). All devices must be calibrated using external reference during the manufacturing process.

For temperature measurement, we employ a reference voltage resistive divider based on a thermistor and a precision resistor (R6). We calculate thermistor resistance according to the voltage drop on the precision resistor and the value of the reference voltage. To provide the necessary temperature measurement accuracy, we first set the RefHi reference voltage and then AGND. After this, we subtract the second value of the resistor voltage drop from the first. Bias voltages RefHi (2.6V) level in the first step and AGND (1.3V) in the next are formed by using the continuous time user module, TestMux. This technique allows us to compensate for both the ADC/INSAMP offset error and the variation in the voltage drop on the current-sense resistor during the charging/discharging process. The following equations represent the temperature measurement scheme:

$$V_t^2 - V_t^1 = V_{AGND} \cdot \frac{R_{ref}}{R_{ref} + R_{term}} \quad \textbf{(18)}$$

$$n_t^2 - n_t^1 = n_{AGND} \cdot \frac{R_{ref}}{R_{ref} + R_{term}} \quad \textbf{(19)}$$

V_t^1 is the voltage level on the temperature reference resistor during application of the V_{AGND} (1.3V) reference voltage. V_t^2 is the voltage level on the temperature reference resistor during application of V_{REFHI} (2.6V) reference voltage. R_{term} is the thermistor resistance. R_{ref} is the temperature reference resistance R30 (10K). n_t^1 and n_t^2 are the ADC codes of V_t^1 and V_t^2, respectively.

n_{AGND} is the ADC code of the AGND input level and is equal to 2048 for a 12-bit incremental ADC in unipolar mode. The measurement technique is discussed in detail in Application Note AN2017.

The battery charge/discharge algorithm only needs to check for temperatures that fall within the allowed ranges: during charging, typical values are 0° to 45° degrees Celsius, and during discharging, typical values are -20° to 60° degrees Celsius. During the charge phase we add hysteresis to the lower and upper bounding temperatures. This prevents multiple triggers when the temperature is close to the preset range. If the temperature is outside the discharge range, the LOAD is turned off and the PSoC device goes into sleep mode. Therefore, a hysteresis for the discharge range is not needed. Figure 13 depicts the temperature profile.

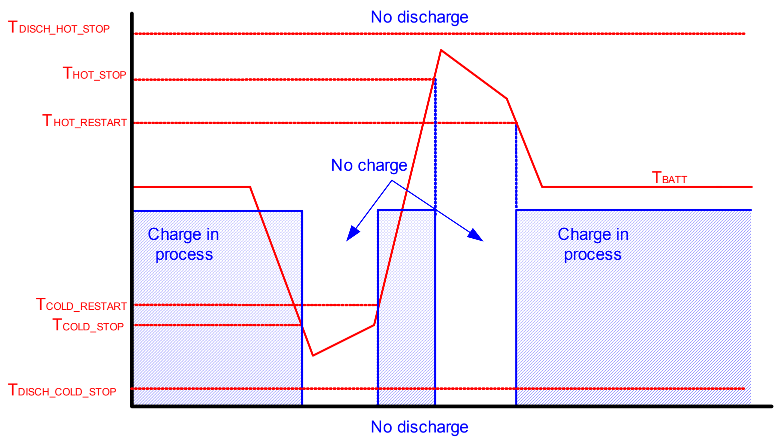


Figure 13. Temperature Profile

The fuel gauge algorithms need to know the battery (not ambient) temperature values at all times. The thermistor transfer function is non-linear. Direct calculation of the temperature value from thermistor resistance requires complex, floating-point calculations that are not well suited to an 8-bit microcontroller. However, it can be done using integer arithmetic by applying a lookup table to calculate battery temperature. To eliminate any complex mathematical calculations, this table must contain an array of the ADC code values, which correspond to the concrete temperatures. The search algorithm looks at the table value that is closest to the measured ADC code. The table index reflects a linear approximation to battery temperature between the measured points.

The charger uses an 81-point lookup table with a starting point at -20° Celsius in 1° increments up to 60°.

This accurate level of measurement is entirely sufficient for this application. For a chosen thermistor, the user need only change the temperature-resistance relationship in the 81-point table (RTC_MINUS and RTC, where T is temperature) that is located in the header file named ***globdefs.h***. This temperature measurement technique is outlined in Application Note AN2017 and explained in detail in the Application Note AN2314 “Thermistor-Based Temperature Measuring in Battery Packs.”

The Battery Charger Firmware

The battery charger firmware is separated into several modules that serve distinct functions, such as performing measurements, regulating battery charge process and timer functions, implementing charge and fuel gauge algorithms, checking the charge termination conditions, storing calibration settings into the PSoC device Flash memory, and transmitting debug data. Most of these modules are described in the Application Notes AN2107, AN2258 and AN2267. Therefore, in this section only the battery charge and fuel gauge algorithms are described.

Battery Charge Algorithm

The battery charge algorithm was implemented in the charger firmware as a state machine.

The following states were used:

- INITIALIZATION indicates charge process initialization.
- ACTIVATION depicts the battery activation charging.
- RAPID depicts the battery rapid charging.
- CHARGE COMPLETE indicates that the battery pack is charged completely.
- WAIT FOR TEMPERATURE is used to depict the idle state when the battery pack temperature is outside the allowed temperature range.
- ERROR indicates that during the charge process an error appeared. These are the following error types: over-voltage, over-current, or stage timeout exception.
- DISCHARGE indicates the battery pack discharge process and the storage device state without external power supply.
- FULL_DISCHARGED indicates that the battery pack is completely discharged and not available for further use.

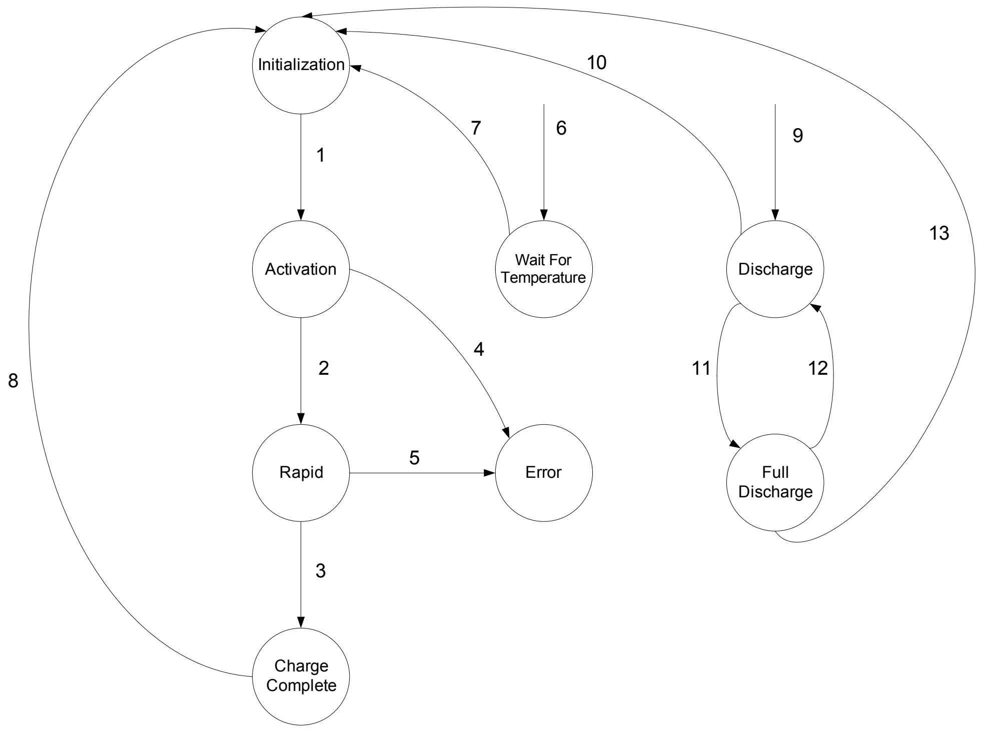


Figure 14. Battery Charger State Diagram

Figure 14 represents the battery charger state diagram. Initially, the charger is in INITIALIZATION state. After device preparation actions, the charger goes to the ACTIVATION state (1). When the battery voltage reaches the rapid start voltage, the charger leaves the ACTIVATION state and switches to the RAPID state (2). If the charge current drops below a predefined charge-terminate level (3), the charger goes to the CHARGE COMPLETE state. The charger remains in the CHARGE COMPLETE state and the charging process can be restarted if the voltage drops below some predefined level (8). The charging process can be terminated with an error if a total charge timeout or an operation charge timeout occurs, or if the battery voltage or charge current is higher than the charge termination voltage/current levels (4), (5).

The charger can jump from any state to the WAIT FOR TEMPERATURE state when the battery temperature is outside of the allowed temperature range (for ACTIVATION and RAPID states, the allowed temperature range is the charge range and for others it is the discharge range) (6). In cases of the discharge range, the PSoC device goes into sleep mode. When the user clicks on SW1, the PSoC device wakes up and the charger goes to the INITIALIZATION state (7). In cases of the charge range when temperatures fall into the defined range with some hysteresis value, the charger goes to the INITIALIZATION state (7).

Regardless of which state the charger is in, it jumps to the DISCHARGE state when the external power supply is switched off (9). If the external power supply is switched on, the charger goes to the INITIALIZATION state (10). When the battery pack discharges completely (11), the charger switches to the FULL_DISCHARGED state. In this state, the PSoC device goes into sleep mode. When the user clicks on SW1, the PSoC device wakes up and the charger goes to the INITIALIZATION state (12). Note that it would be easy to add a feature to this project that allows the charger to wake up when it is connected to the external power supply.

Sleep mode is a “virtual state.” In this state, the power consumption of the PSoC is minimal; this state is used to save battery energy. The charger goes to this state if the user clicks on SW1 or when the charger is in the FULL_DISCHARGED state. The charger wakes up when the user again clicks on SW1. In the DISCHARGE state, additional battery energy savings occur. The charger, at regular intervals of 1s as determined by the Sleep_Timer, wakes up to measure battery parameters and update the fuel gauge information and then returns to sleep mode. It is straightforward to change the transitions to this state to accommodate other energy-saving requirements. For example, at a predefined time interval, if the system senses LOAD low activity, it can be set to sleep mode.

Figure 15 and Figure 16 show the battery charger firmware flowchart that corresponds to the state diagram. Also shown are the invocation points of the fuel gauge procedures.

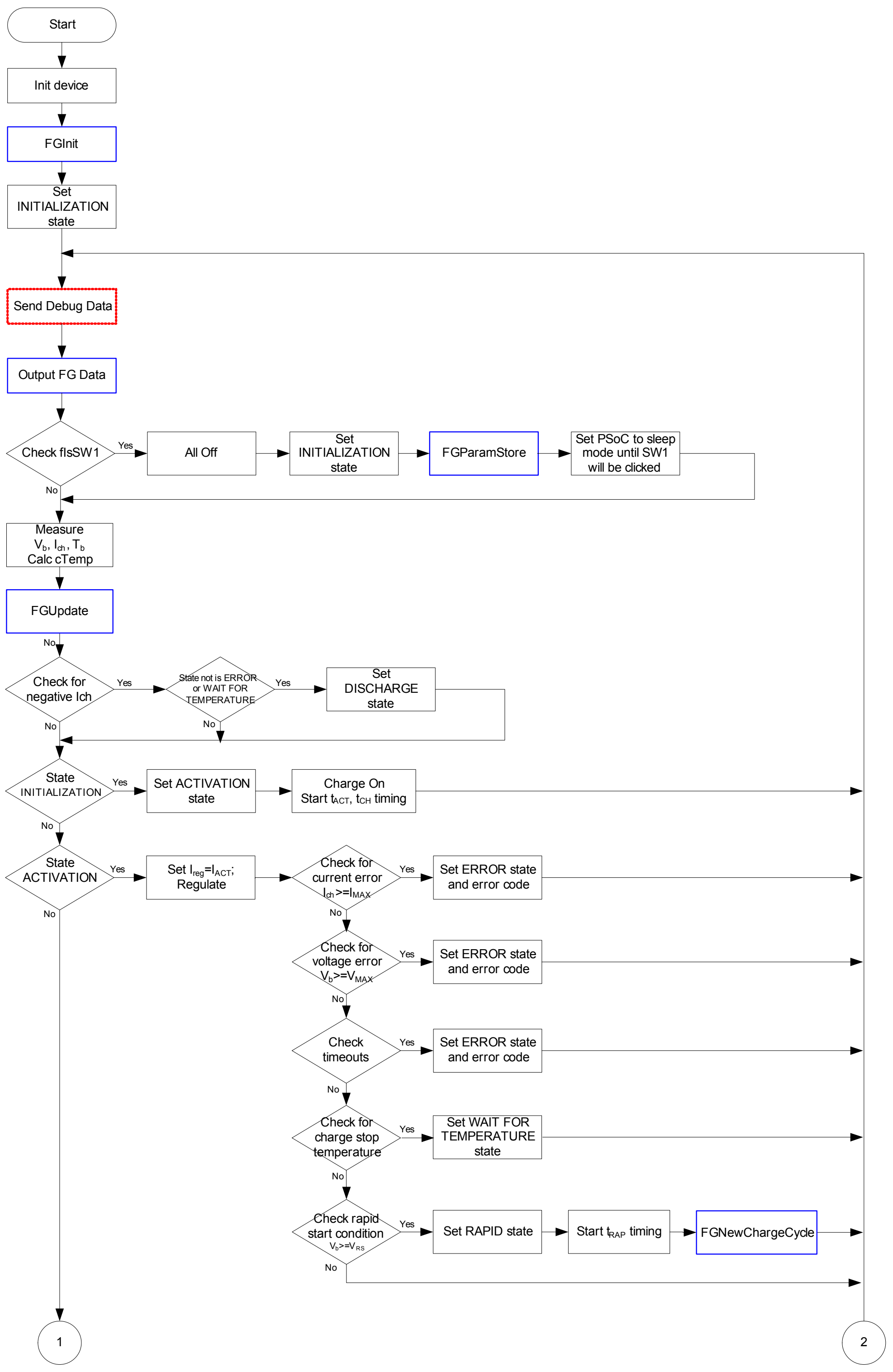

Start
Init device
FGInit
Set INITIALIZATION state
Send Debug Data
Output FG Data
Check fIsSW1
Yes
All Off
Set INITIALIZATION state
FGParamStore
Set PSoC to sleep mode until SW1 will be clicked
No
Measure V_b, I_{ch}, T_b Calc cTemp
FGUpdate
No
Check for negative Ich
Yes
State not is ERROR or WAIT FOR TEMPERATURE
Yes
Set DISCHARGE state
No
No
State INITIALIZATION
Yes
Set ACTIVATION state
Charge On Start t_{ACT}, t_{CH} timing
No
State ACTIVATION
Yes
Set I_{reg}=I_{ACT}; Regulate
Check for current error I_{ch}>=I_{MAX}
Yes
Set ERROR state and error code
No
Check for voltage error V_b>=V_{MAX}
Yes
Set ERROR state and error code
No
Check timeouts
Yes
Set ERROR state and error code
No
Check for charge stop temperature
Yes
Set WAIT FOR TEMPERATURE state
No
Check rapid start condition V_b>=V_{RS}
Yes
Set RAPID state
Start t_{RAP} timing
FGNewChargeCycle
No
No
1
2

Figure 15. Battery Charger Firmware Flowchart Part 1

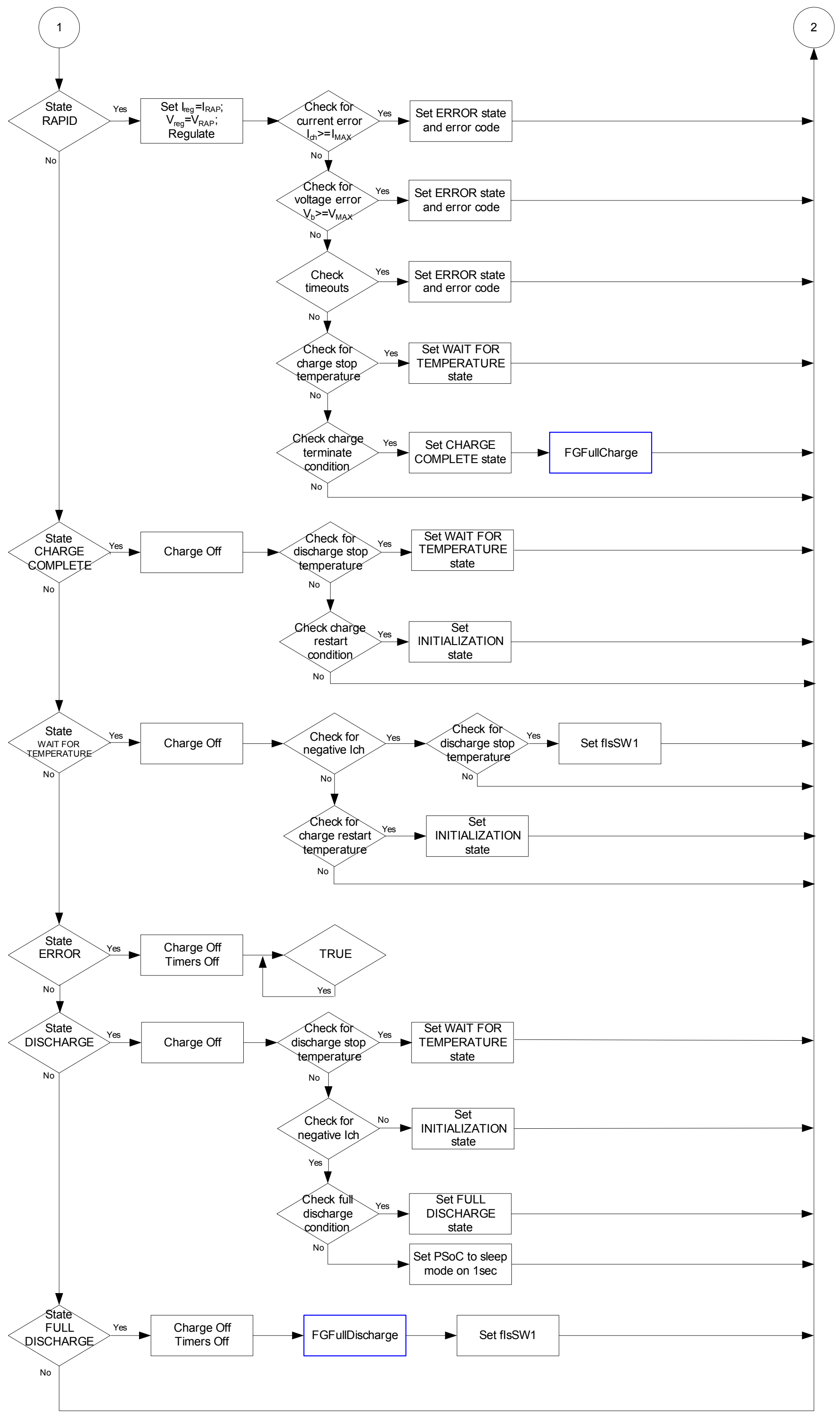

1
2
State RAPID
Yes
No
Set $I_{reg}=I_{RAP}$; $V_{reg}=V_{RAP}$; Regulate
Check for current error $I_{ch}>=I_{MAX}$
Set ERROR state and error code
Check for voltage error $V_b>=V_{MAX}$
Check timeouts
Check for charge stop temperature
Set WAIT FOR TEMPERATURE state
Check charge terminate condition
Set CHARGE COMPLETE state
FGFullCharge
State CHARGE COMPLETE
Charge Off
Check for discharge stop temperature
Check charge restart condition
Set INITIALIZATION state
State WAIT FOR TEMPERATURE
Check for negative Ich
Set flsSW1
Check for charge restart temperature
State ERROR
Charge Off Timers Off
TRUE
State DISCHARGE
Check full discharge condition
Set FULL DISCHARGE state
Set PSoC to sleep mode on 1sec
State FULL DISCHARGE
FGFullDischarge

Figure 16. Battery Charger Firmware Flowchart Part 2

Fuel Gauge Algorithms

The fuel gauge algorithms are also separated into several modules that serve distinct functions. The main program calls necessary functions at the appropriate time. Table 3 shows a brief description of each fuel gauge function:

Table 3. Fuel Gauge Function Descriptions

Item	Description
FGInit	Initializes fuel gauge variables from non-volatile memory
FGParamStore	Stores fuel gauge variables to non-volatile memory
FGUpdate	Calculates all battery fuel gauge parameters
FGNewChargeCycle	Increments total charge cycles and count of cycles after last learning charge cycle
FGLearnFullCharge	Sets battery capacity equal to fuel gauge ACR value (Fuel Gauge Learning Charge Cycle)
FGFullCharge	Sets fuel gauge ACR value to 100%
FGFullDischarge	Sets fuel gauge ACR value to 0%

Figure 17 shows fuel gauge function invocation points from the main program flowchart. The main program calls the FGInit function after device power up. The FGInit function initializes all variables that are used during fuel gauging. These variables include ACR value, expected full points over temperature, number of total charge cycles, number of charge cycles after learn, cell age coefficient, flag of fully discharged state, and charge interpolation points over temperature. When the device goes into sleep mode (for example, when the battery is fully discharged), the fuel gauge variables must be stored in non-volatile memory. This transfer is performed by the FGParamStore function.

Once the battery parameters have been measured, the FGUpdate function is periodically called. It updates all battery fuel gauge information:

- Absolute Capacity (ACR)
- State of Charge (SOC)
- Runtime Remaining in Active Mode
- Runtime Remaining in Suspend Mode
- Full-Charge Time Remaining
- Rapid-Charge Time Remaining

Figure 18 and Figure 19 show the FGUpdate fuel gauge function algorithm. The algorithm also takes into consideration the following internal battery characteristics and environmental influences:

- Self-Discharge
- Temperature
- Discharge Rate
- Cell Aging

The FGNewChargeCycle function calculates the total charge cycles and count of cycles performed since the last learning charge cycle. If the count of cycles performed since the last learning charge cycle is greater than the predefined value, then a message is sent to the user recommending that the learning charge cycle be performed. If the battery fully discharged previously then the user is notified that this charge cycle can be a learning charge cycle. Therefore, the user must pay special attention to this charge cycle and perform it completely. Figure 20 shows the FGNewChargeCycle function fuel gauge algorithm.

The FGLearnFullCharge function sets the battery capacity equal to the fuel gauge accumulated current (ACR), calculates the cell age correction coefficient, and adjusts all internal parameters that are dependent on the full charge value for the entire range of temperatures:

- Full Charge
- Breakpoint Capacity
- Breakpoint/2 Capacity
- Breakpoint Time

In such a way, this function compensates for the effects of battery aging. The FGLearnFullCharge routine is performed after every charge cycle where the battery has gone from completely empty to completely full. Figure 21 illustrates the FGLearnFullCharge function fuel gauge algorithm.

The FGFullCharge function is called by the main program when the battery reaches the fully charged state. If the ACR value at this point differs from the predefined expected value, then the user is sent a message recommending that the learning charge cycle be performed. The function sets the ACR to the expected full point that corresponds to the temperature and age. This action compensates for self-discharge and measurement error of standby current. Figure 22 shows the FGFullCharge function fuel gauge algorithm.

The FGFullDischarge function is called by the main program when the battery reaches the fully discharged state. The FGFullDischarge sets the ACR to the expected empty point that corresponds to the discharge rate and temperature. This action (as in the case of FGFullCharge) also compensates for the self-discharge and measurement error of standby current. The FGFullDischarge function also sets the flag of the fully discharged state. This flag is used after a full charge to choose between execution of FGLearnFullCharge or FGFullCharge. Figure 23 shows the FGFullDischarge function fuel gauge algorithm.

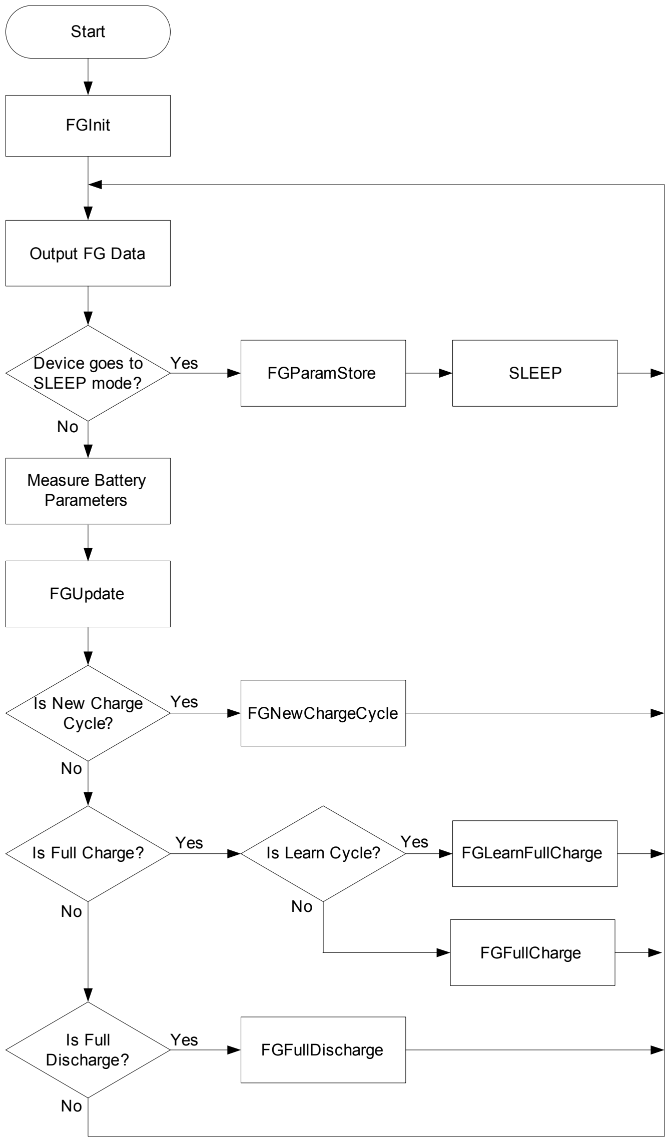


Figure 17. Fuel Gauge Function Invocation Points from the Main Program Flowchart

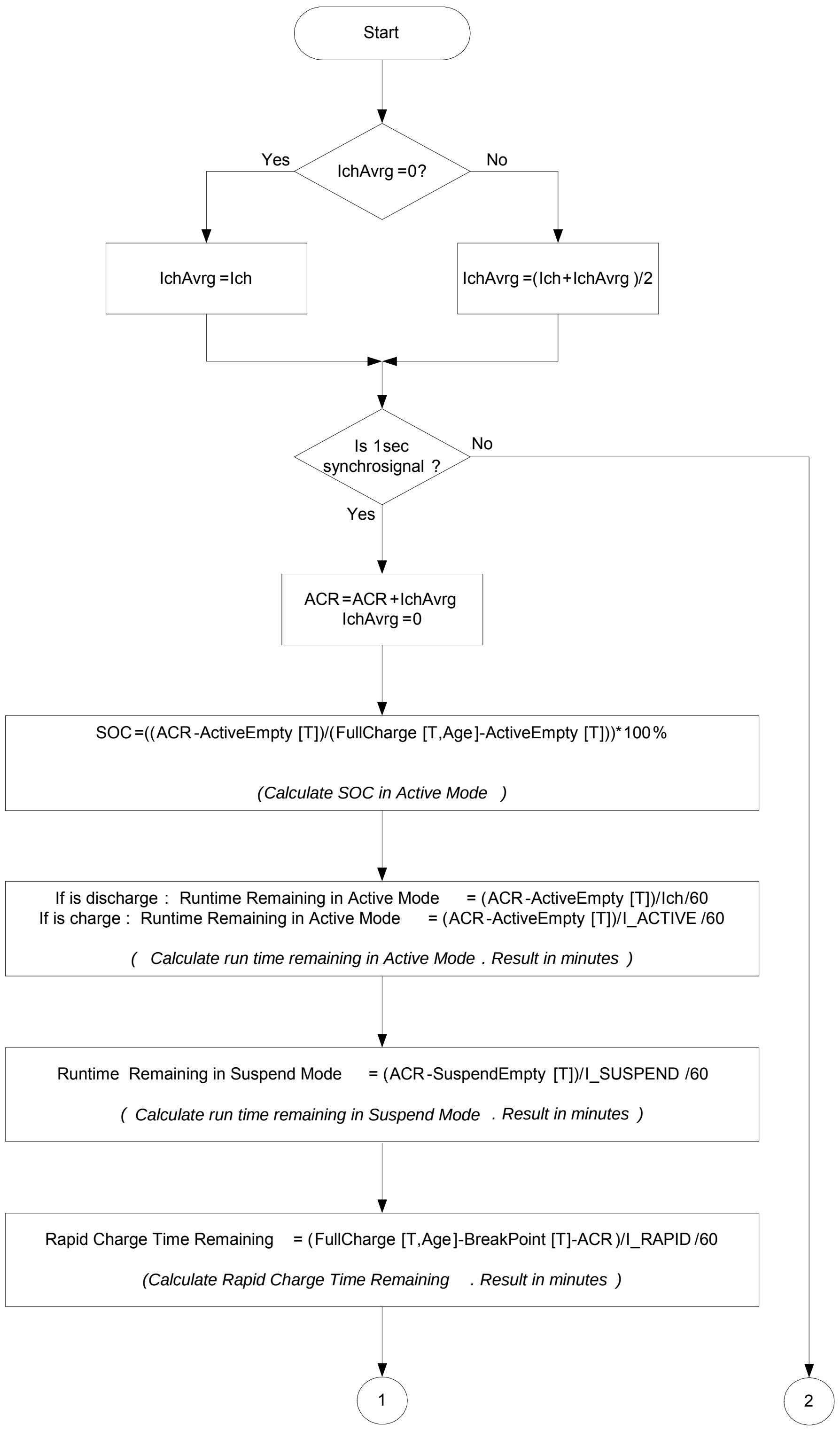

Start
IchAvrg =0?
Yes
No
IchAvrg =Ich
IchAvrg =(Ich+IchAvrg)/2
Is 1sec synchrosignal ?
No
Yes
ACR =ACR +IchAvrg
IchAvrg =0
SOC =((ACR -ActiveEmpty [T])/(FullCharge [T,Age]-ActiveEmpty [T]))*100 %
(Calculate SOC in Active Mode)
If is discharge : Runtime Remaining in Active Mode = (ACR -ActiveEmpty [T])/Ich/60
If is charge : Runtime Remaining in Active Mode = (ACR -ActiveEmpty [T])/I_ACTIVE /60
(Calculate run time remaining in Active Mode . Result in minutes)
Runtime Remaining in Suspend Mode = (ACR -SuspendEmpty [T])/I_SUSPEND /60
(Calculate run time remaining in Suspend Mode . Result in minutes)
Rapid Charge Time Remaining = (FullCharge [T,Age]-BreakPoint [T]-ACR)/I_RAPID /60
(Calculate Rapid Charge Time Remaining . Result in minutes)
1
2

Figure 18. FGUpdate Function Fuel Gauge Algorithm Part 1

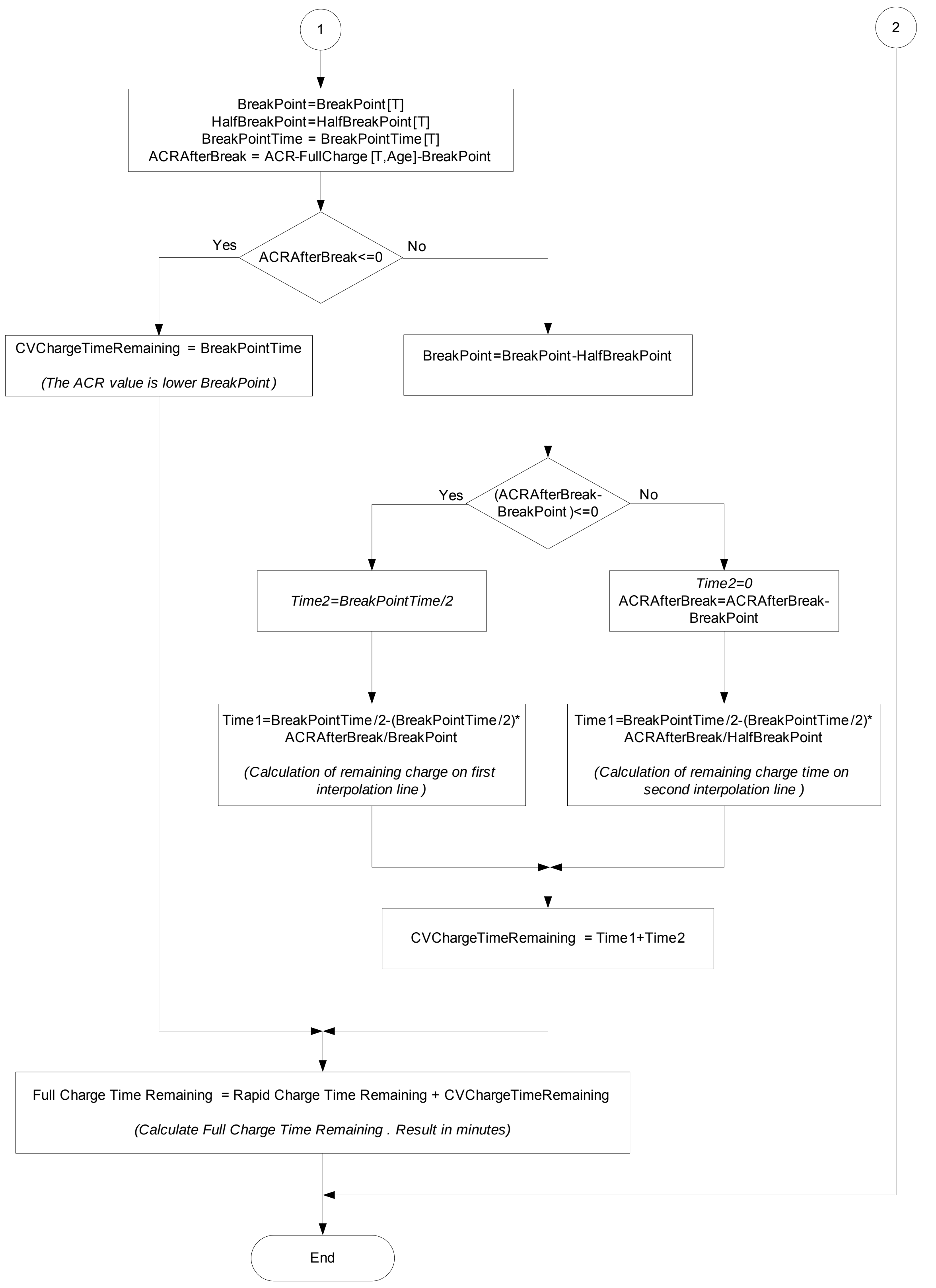

1
2
BreakPoint=BreakPoint[T]
HalfBreakPoint=HalfBreakPoint[T]
BreakPointTime = BreakPointTime[T]
ACRAfterBreak = ACR-FullCharge [T,Age]-BreakPoint
Yes
ACRAfterBreak<=0
No
CVChargeTimeRemaining = BreakPointTime
(The ACR value is lower BreakPoint)
BreakPoint=BreakPoint-HalfBreakPoint
Yes
(ACRAfterBreak-BreakPoint)<=0
No
Time2=BreakPointTime/2
Time2=0
ACRAfterBreak=ACRAfterBreak-BreakPoint
Time1=BreakPointTime/2-(BreakPointTime/2)* ACRAfterBreak/BreakPoint
(Calculation of remaining charge on first interpolation line)
Time1=BreakPointTime/2-(BreakPointTime/2)* ACRAfterBreak/HalfBreakPoint
(Calculation of remaining charge time on second interpolation line)
CVChargeTimeRemaining = Time1+Time2
Full Charge Time Remaining = Rapid Charge Time Remaining + CVChargeTimeRemaining
(Calculate Full Charge Time Remaining. Result in minutes)
End

Figure 19. FGUpdate Function Fuel Gauge Algorithm Part 2

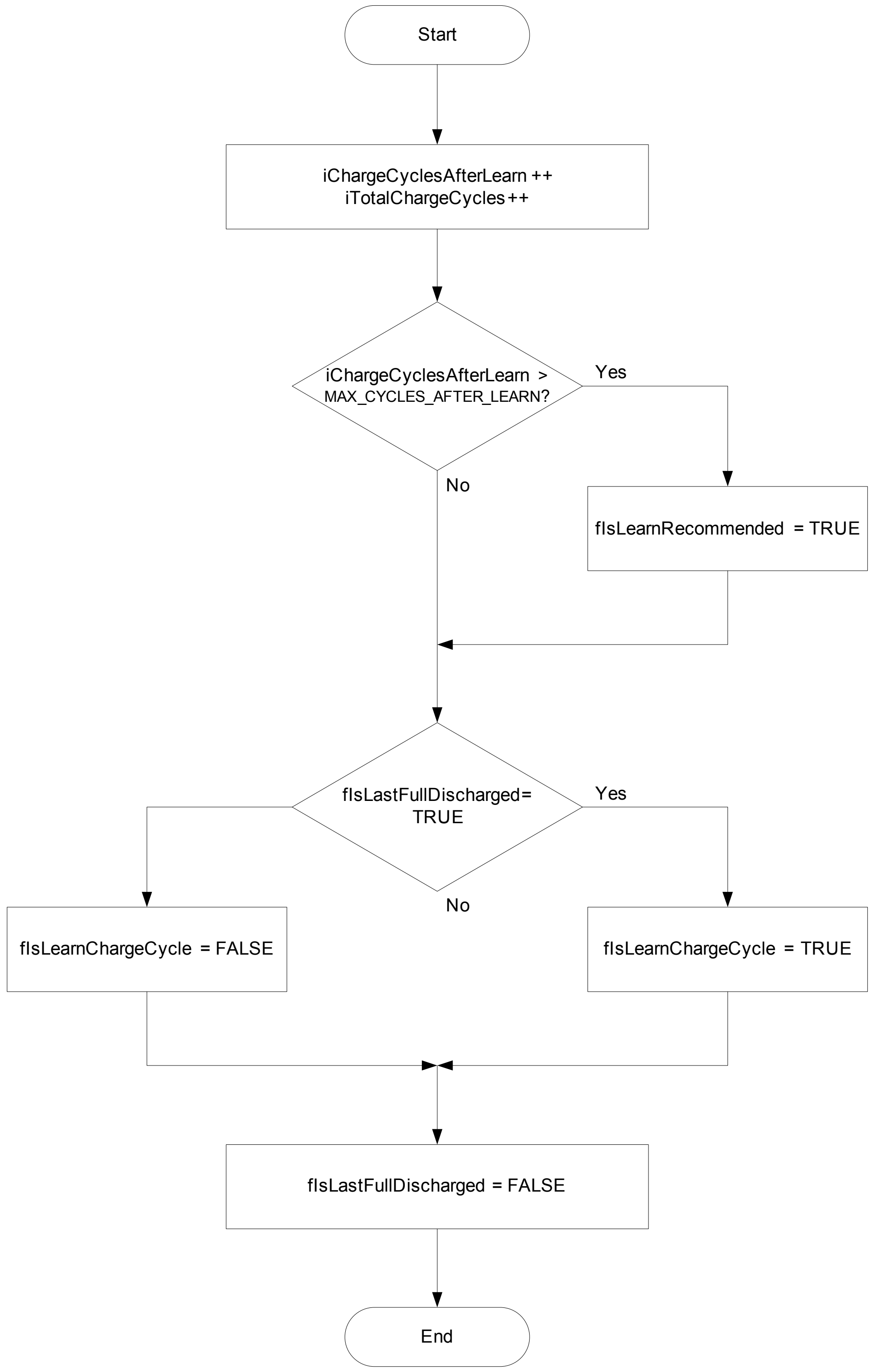

Start
iChargeCyclesAfterLearn ++
iTotalChargeCycles++
iChargeCyclesAfterLearn >
MAX_CYCLES_AFTER_LEARN?
Yes
No
fIsLearnRecommended = TRUE
fIsLastFullDischarged=
TRUE
Yes
No
fIsLearnChargeCycle = FALSE
fIsLearnChargeCycle = TRUE
fIsLastFullDischarged = FALSE
End

Figure 20. FGNewChargeCycle Function Fuel Gauge Algorithm

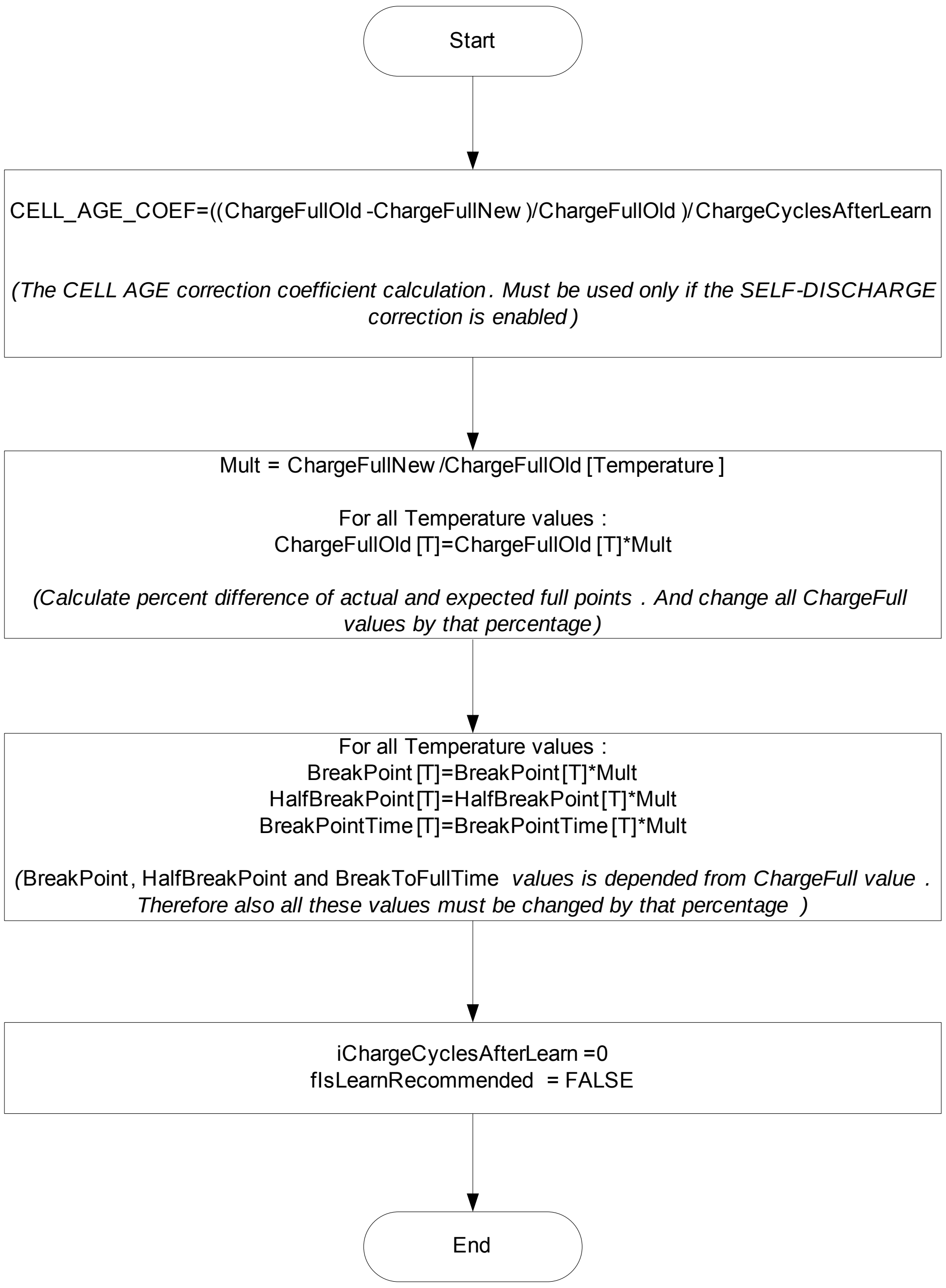

Start
CELL_AGE_COEF=((ChargeFullOld -ChargeFullNew)/ChargeFullOld)/ ChargeCyclesAfterLearn
(The CELL AGE correction coefficient calculation. Must be used only if the SELF-DISCHARGE correction is enabled)
Mult = ChargeFullNew /ChargeFullOld [Temperature]
For all Temperature values :
ChargeFullOld [T]=ChargeFullOld [T]*Mult
(Calculate percent difference of actual and expected full points . And change all ChargeFull values by that percentage)
For all Temperature values :
BreakPoint [T]=BreakPoint [T]*Mult
HalfBreakPoint [T]=HalfBreakPoint [T]*Mult
BreakPointTime [T]=BreakPointTime [T]*Mult
(BreakPoint, HalfBreakPoint and BreakToFullTime values is depended from ChargeFull value . Therefore also all these values must be changed by that percentage)
iChargeCyclesAfterLearn =0
fIsLearnRecommended = FALSE
End

Figure 21. FGLearnFullCharge Function Fuel Gauge Algorithm

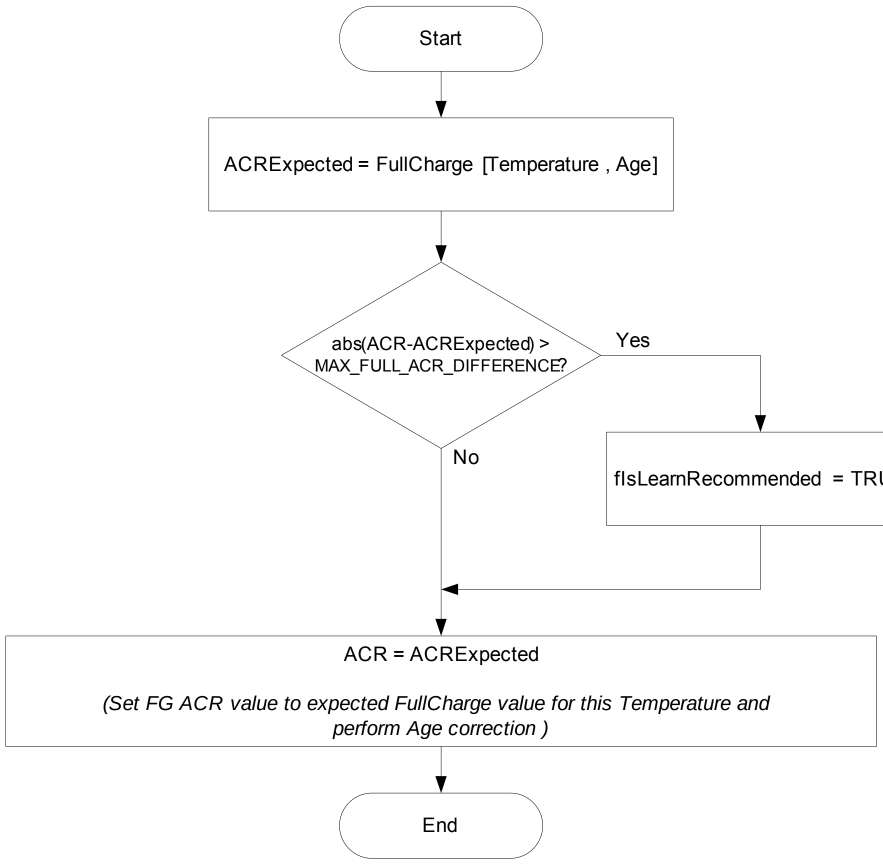

Start
ACRExpected = FullCharge [Temperature , Age]
abs(ACR-ACRExpected) > MAX_FULL_ACR_DIFFERENCE?
Yes
No
fIsLearnRecommended = TRUE
ACR = ACRExpected
(Set FG ACR value to expected FullCharge value for this Temperature and perform Age correction)
End

Figure 22. FGFullCharge Function Fuel Gauge Algorithm

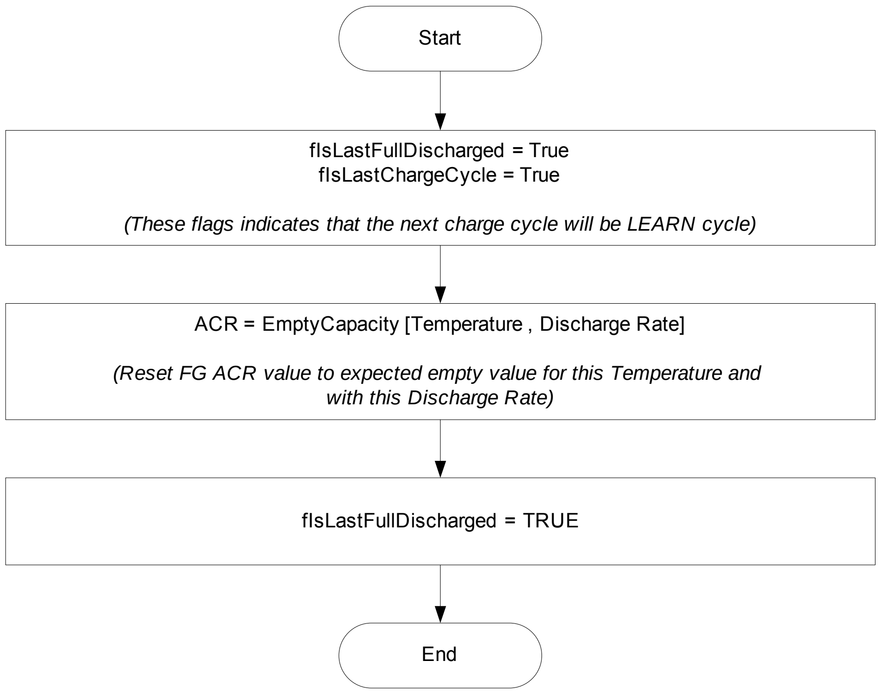

Start
fIsLastFullDischarged = True
fIsLastChargeCycle = True
(These flags indicates that the next charge cycle will be LEARN cycle)
ACR = EmptyCapacity [Temperature , Discharge Rate]
(Reset FG ACR value to expected empty value for this Temperature and with this Discharge Rate)
fIsLastFullDischarged = TRUE
End

Figure 23. FGFullDischarge Function Fuel Gauge Algorithm

Fuel Gauge and Debug Information

The Li-Ion/Li-Polymer battery charger with fuel gauge function uses serial port communication to monitor charge and fuel gauge processes. The data communication protocol is simple. Each packet consists of a start marker character, ASCII code 10 (LF), data bytes, and a stop marker character, ASCII code 13 (CR). The data bytes are encoded in hex format, so each data byte needs two bytes to be transmitted or received. Figure 24 illustrates the data packet structure.

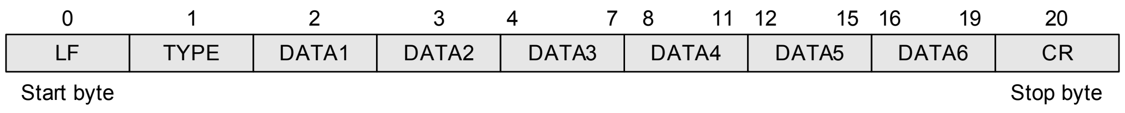


Figure 24. Data Packet Structure

The charger can send six types of packets to the PC (TYPE byte):

- 0 - Reset
- 1 - Charger measurement
- 2 - Charging voltage parameters
- 3 - Charging current parameters
- 4 - Fuel gauge discharge parameters
- 5 - Fuel gauge charge parameters

The other data bytes depend on the type of packet. Following are some examples of charger packets. (Similar charger packets are described in detail in Application Notes AN2107 and AN2267.)

Reset packet:
```
SendDebugData(DP_RESET, 0, 0, 0, 0, 0, 0);
```
Charger measurement packet:
```
SendDebugData(DP_MEASUREMENT, eState, eErrorCode, iIch, iVb, iTb, (INT)cTemperature);
```
Charging voltage parameters:
```
SendDebugData(DP_VOLTAGE_PARAM, 0, 0, V_RS, V_CRST, V_MAX, V_FULL_DISCH);
```
Charging current parameters:
```
SendDebugData(DP_CURRENT_PARAM, 0, 0, I_ACT, I_RAP, I_MAX, I_RTN);
```

Note that all of the data values in these packets are transmitted in ADC code without conversion to user-suitable formats. The charger control software converts these values and prepares the data for viewing. These parameters are sent only for debug purposes and may not be used in a production product.

However, all fuel gauge packets transmit the data in a user-suitable format. All conversions and fuel gauge calculations are performed in the PSoC device.

Therefore, the user can easily add an LCD display to the project to gauge the fuel, and change the transmit protocol or the RS232 communication protocol on the I2C, SPI, UART et al. Table 4 shows how the user can read the fuel gauge parameters (from the ***fuelgauge.c*** file in the project folder). Note that the FGUpdate function described in the previous section has two separate fucntions. It consists of the FGUpdateIchAvrg procedure (that updates the IchAvrg value) and the fuel gauge parameter calculation functions (see Table 4). After this, the result is transmitted for viewing. This modular structure allows the user to select only the necessary parameters for calculation and to easily adapt this project as appropriate to his task.

Table 4. Fuel Gauge Parameters Calls

Fuel Gauge Parameters Call	Description
lACR	Fuel gauge accumulated current variable in internal representation (ACR).
iFGACRtoCapacity(lACR)	Absolute capacity in mAh.
cFGGetActiveSOC(cTemperature)	State of charge in %.
iFGGetActiveTime(cTemperature, iIch, fIsDischarge)	Runtime remaining in active mode in minutes.
iFGGetSuspendTime(cTemperature)	Runtime remaining in suspend mode in minutes.
iChargeCyclesAfterLearn	Charge cycle count after last learning charge cycle.
iTotalChargeCycles	Total charge cycles.
iCellAgeCoef	Age correction coefficient as integer value equal to 1/CellAgeCoef.
fIsLearnChargeCycle	TRUE value indicates that if the battery has been fully charged then the fuel gauge learning charge cycle will be performed. Therefore, the user must be careful to fully complete this charge cycle. (When the battery is fully discharged, this bit is set to TRUE and if the charge process is interrupted, this bit is reset to FALSE).
fIsLearnRecommended	TRUE value indicates that performance of the fuel gauge learning charge cycle is recommended.
iFGGetRapidChargeTime(cTemperature)	Rapid charge time remaining in minutes.
iFGGetRapidChargeTime(cTemperature) + iFGGetCVChargeTime(cTemperature)	Full charge time remaining in minutes.

cTemperature is the environment temperature (in degree Celsius). Ich is the charge/discharge current (in ADC units without conversion to amperes). fIsDischarge is the flag that indicates charger discharge process (TRUE - discharge).
Note that all fuel gauge parameters are stored as global variables in the ***fuelgauge.c*** file in the project folder.

The following are examples of fuel gauge packets, which are used for debug purposes or can be used by the host controller to read fuel gauge data.

Fuel gauge charge time remaining packet:
```
iTRapid = iFGGetRapidChargeTime(cTemperature);
SendDebugData(DP_FG_CH, FG_CH_TIME, 0, (iTRapid+iFGGetCVChargeTime(cTemperature)), iTRapid, 0x55, 0x55);
```

Fuel gauge information packet:
```
SendDebugData(DP_FG, FG_INFO, LC_FALSE, iChargeCyclesAfterLearn, iTotalChargeCycles, iCellAgeCoef, 0x55);
```

Fuel gauge measurement packet:
```
SendDebugData(DP_FG, FG_MEASUREMENT, LC_TRUE, iFGACRtoCapacity(lACR),
(INT)cFGGetActiveSOC(cTemperature), iFGGetActiveTime(cTemperature, iIch, fIsDischarge), iFGGetSuspendTime(cTemperature));
```

Fuel gauge corrections parameters packet for 0 degree Celsius:
```
SendDebugData(DP_FG, FG_0C_PARAM, 0, iFGACRtoCapacity(anlChargeFull[0]), Q_ACTIVE_EMPTY_0, Q_SUSPEND_EMPTY_0, 0x55);
```

Battery Charger Parameters

All battery charger parameters are located in the header file ***globdefs.h*** in the project folder. It contains the following parameters:

Table 5. Battery Charger Parameters

Charging Parameters		
Parameter	**Unit**	**Description**
Vrs	V	Rapid charge start condition
Vrap	V	Full-charge voltage (constant charge voltage)
Vcrst	V	Re-charge voltage
Vbmax	V	Emergency shutdown voltage
Vfull_disch	V	Full-discharge voltage
Iact	A	Activation charge current
Irap	A	Rapid charge current
Ichmax	A	Emergency shutdown current
Irtn	A	Charge termination current

Timing Requirements		
Parameter	**Unit**	**Description**
TACT	second	Time limit for battery activation period
TRAPID	second	Time limit for final stage of constant charge voltage
TCHARGE	second	Time limit for total charge period
TTERM	second	Minimum time for charge complete (when Ich≤Irtn)

Thermistor Parameters		
Parameter	**Unit**	**Description**
RTC_MINUS	Ohms	Thermistor resistance at negative temperature T
RTC	Ohms	Thermistor resistance at temperature T

Thermistor Measurement Requirements		
Parameter	**Unit**	**Description**
RTERM_CH_COLD_STOP	Ohms	Thermistor resistance for cold stop battery charge
RTERM_CH_COLD_RESTART	Ohms	Thermistor resistance for cold re-start battery charge
RTERM_CH_HOT_STOP	Ohms	Thermistor resistance for hot stop battery charge
RTERM_CH_HOT_RESTART	Ohms	Thermistor resistance for hot restart battery charge
RTERM_DISCH_COLD_STOP	Ohms	Thermistor resistance for cold stop battery discharge
RTERM_DISCH_HOT_STOP	Ohms	Thermistor resistance for hot stop battery discharge

Schematic Parameters		
CURRENT_SENSE_R	Ohms	Current-sense resistor
TEMPERATURE_R_REF	Ohms	Thermistor reference resistor

Fuel Gauge Parameters

All fuel gauge parameters are located in the header file ***globdefs.h*** in the project folder. It contains the following parameters:

Table 6. Fuel Gauge Parameters

Fuel Gauge Static Parameters		
Parameter	**Unit**	**Description**
Iactive	A	Discharge current in active mode.
Isuspend	A	Discharge current in suspend mode.
Idisch_err	A	Discharge current measurement error. It is used for discharge sleep mode compensation.
MAX_CYCLES_AFTER_LEARN	quantity	Cycles since last learning charge cycle count to notify that learning charge cycle is necessary.
MAX_FULL_ACR_DIFFERENCE	mAh	Fully charged battery ACR and expected difference of FULL CHARGE values (in FGFULL function) to notify that learning charge cycle is necessary.

Initial Parameters for Discharge Phase Fuel Gauge		
Parameter	**Unit**	**Description**
QCH_FULL_T	mAh	The capacity of the fully charged cell at temperature T
QACTIVE_EMPTY_T	mAh	The capacity of the fully discharged cell in active mode (high discharge rate) at temperature T
QSUSPEND_EMPTY_T	mAh	The capacity of the fully discharged cell in suspend mode (low discharge rate) at temperature T

Initial Parameters for Charge Phase Fuel Gauge		
Parameter	**Unit**	**Description**
QBREAKPOINT _T	mAh	The capacity from Breakpoint to fully charged state at temperature T
QHALF_BREAKPOINT _T	mAh	The capacity from Breakpoint/2 to fully charged state at temperature T
TBREAKPOINT_T	minute	The charge time from Breakpoint to fully charged state at temperature T

Charger Adaptation to Selected Battery

In order to adapt the charger to the selected battery, it is necessary perform the following operations. (All configured parameters are located in the header file ***globdefs.h*** in the project folder.)

1. **Populate the thermistor temperature-resistance table** for the selected battery pack thermistor into an 81-point lookup table with starting point at -20° degrees Celsius in 1°-degree increments to 60° degrees (RTC_MINUS and RTC parameters, where T is temperature). For example, #define R0C 32650.8 shows that the thermistor resistance at 32650.8 ohms corresponds to a battery temperature of 0° Celsius.
2. **Set the allowed charge/discharge temperature range parameters** (all "Thermistor Measurement Requirement" parameters in the Battery Charger Parameters section) for the selected battery pack.
3. **Establish the battery charge profile parameters**, such as rapid charge start condition (Vrs), time limit for battery activation period (T_{ACT}), and others (all "Charging Parameters" and "Timing Requirements" in the Battery Charger Parameters section).
4. **Perform the fuel gauge temperature, discharge rate, and charge process interpolation parameter measurement procedure**.

Before starting this procedure, it is advisable to charge/discharge the battery several times:

a) Discharge the battery at the highest temperature to the fully discharged state with the lowest possible discharge rate. Reset ACR for this point. (This is the lowest battery charge empty point and it should be set equal to zero.)
b) Set the fixed environment temperature (start with the highest temperature).
c) Charge the battery to full state and record ACR value to the $Q_{CH\_FULL\_T}$ for this temperature.
d) Determine Breakpoint Time on charge current profile and write value to $T_{BREAKPOINT\_T}$.
e) Determine Breakpoint Capacity, Breakpoint/2 Capacity and Breakpoint Time for this temperature on the battery capacity (ACR) profile and record the $Q_{BREAKPOINT\_T}$ and $Q_{HALF\_BREAKPOINT\_T}$ charge process interpolation parameters.
f) Discharge the battery with active rate and record ACR value as the $Q_{ACTIVE\_EMPTY\_T}$ for this temperature.
g) Continue to discharge the battery at suspend rate and set the ACR value to $Q_{SUSPEND\_EMPTY\_T}$ for this temperature.
h) Repeat from step b for other temperatures.

Note that for Li-Ion/Li-Polymer batteries, the self-discharge rate is extremely low. Therefore, in this project this compensation is not used. Even so, all information about self-discharge correction and the collection of empirical data is described in the sections above.

After modifying of all necessary parameters it is time to build the project and program the PSoC chip. Next, the device must be calibrated from an external precision 4.2V source. The charger is now ready to work. For accurate fuel gauge readings during initial use, perform the fuel gauge learning charge cycle (as a rule, two or three times).

Future Research

The proposed fuel gauge is particularly advantageous for portable applications where a fully discharged and/or fully charged battery state rarely occurs. For example, in some hybrid electric vehicle (HEV) applications the battery remains between 20% SOC and 80% SOC and, therefore, will not typically reach the traditional calibration points of the fully charged and the fully discharged states. Accordingly, the measurement error of the fuel gauge parameters tends to grow with time of use. Future research of a fuel gauge HEV application is required.

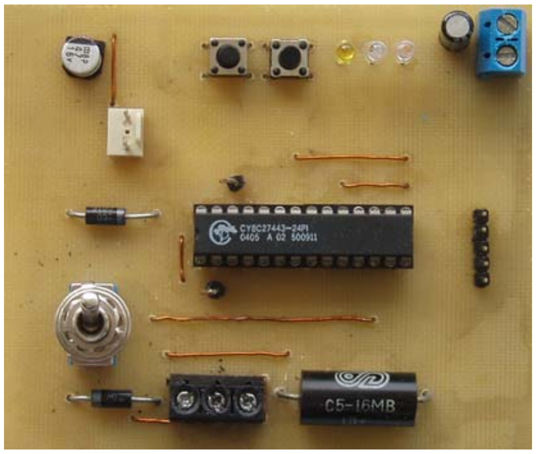


Figure 25. Battery Charger with Fuel Gauge Support Actual Size

Conclusion

A Li-Ion/Li-Polymer battery charger with fuel gauge functionality has been described. We propose a technology that is independent of battery chemistry and quantity. Therefore, the hardware can easily be expanded for a multi-cell battery charger with fuel gauge and cell-balancing functions (Application Note AN2258) or adapted for other battery types and even implemented in a multi-chemistry battery charger (Application Note AN2107). We have described techniques to calculate absolute capacity, runtime remaining in active and suspend modes, state of charge, and time remaining for rapid and full charge. We have examined in detail self-discharge, temperature, discharge rate and cell aging affects on battery capacity and state of charge. The procedures for obtaining empirical data to correct for all of these affects are described. The battery charger and fuel gauge algorithms are implemented in firmware. The proposed device can be used as a complete battery cell management system for mobile applications with an accuracy of several percent.

Appendix A: Charge/Discharge and Fuel Gauge Profile Examples

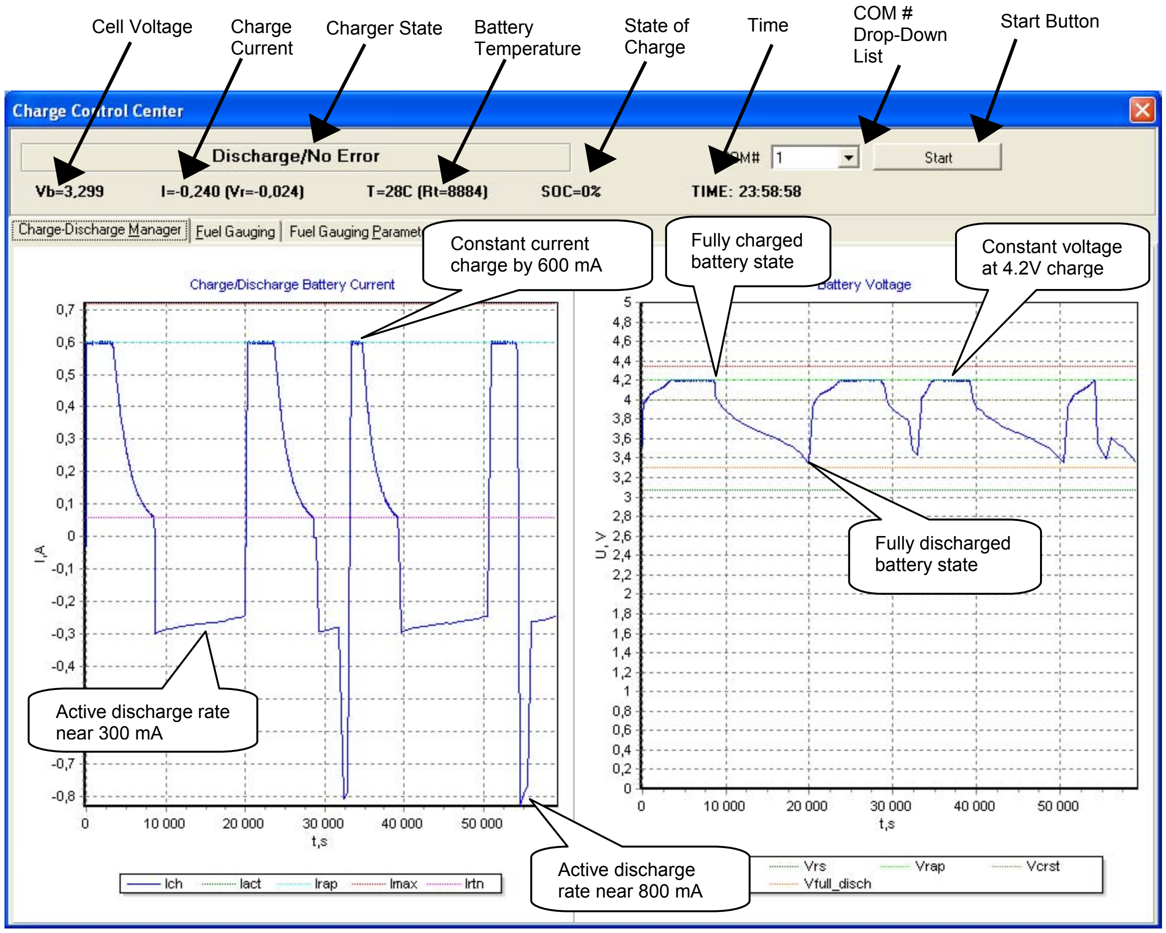
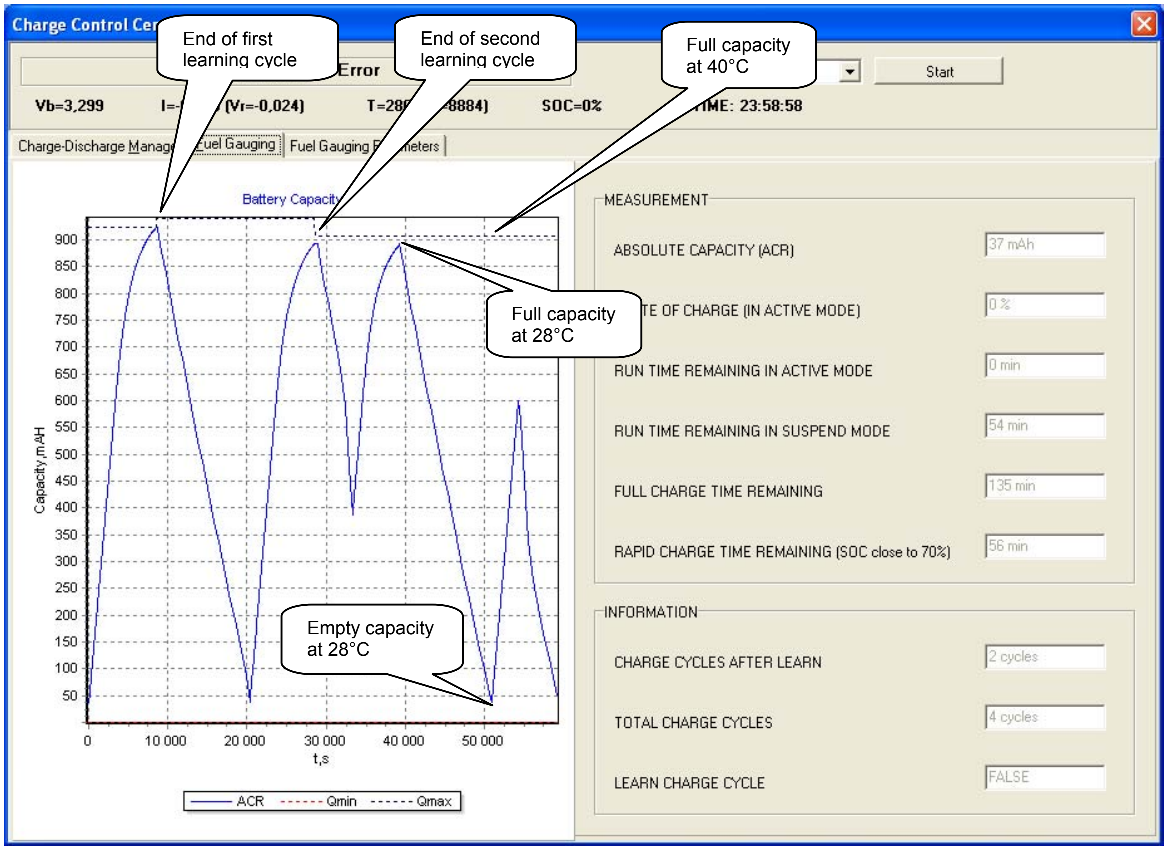


Figure 26. Charge/Discharge Manager Profile



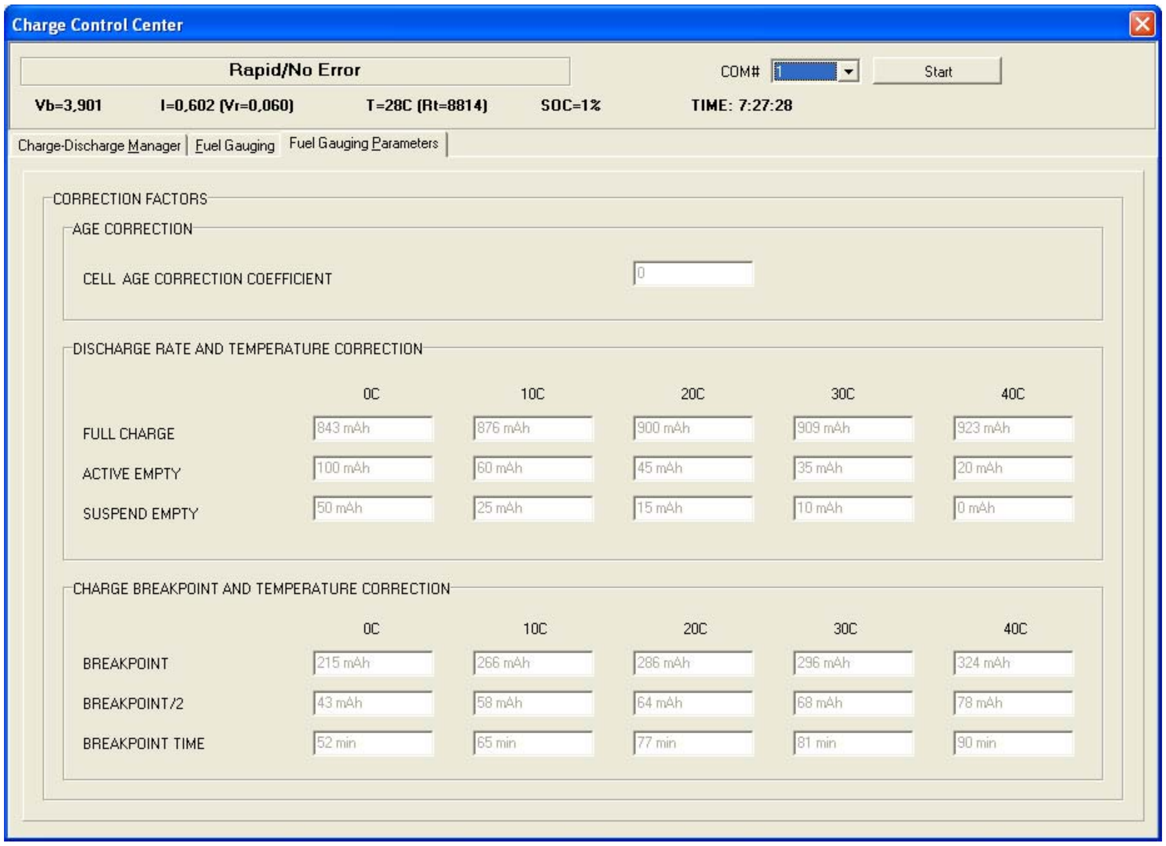
Vb=3,299 I=- (Vr=-0,024) T=28 -8884) SOC=0% TIME: 23:58:58

ABSOLUTE CAPACITY (ACR)	37 mAh
TE OF CHARGE (IN ACTIVE MODE)	0 %
RUN TIME REMAINING IN ACTIVE MODE	0 min
RUN TIME REMAINING IN SUSPEND MODE	54 min
FULL CHARGE TIME REMAINING	135 min
RAPID CHARGE TIME REMAINING (SOC close to 70%)	56 min
CHARGE CYCLES AFTER LEARN	2 cycles
TOTAL CHARGE CYCLES	4 cycles
LEARN CHARGE CYCLE	FALSE

Figure 27. Fuel Gauge Information Profile with Test

Figure 26 and Figure 27 show that the first two learning charge cycles were performed. Only after the second learning charge cycle was the required fuel gauge accuracy reached. Therefore, when a new battery is loaded into the device, it is recommended that two learning charge cycles be performed. Note that to determine correct charging parameters, a single learning charge cycle is sufficient.

After the learning charge cycles, the battery was partly discharged first at a discharge rate of 300 mA and later at a rate of 800 mA. Next, the battery was again fully charged, then fully discharged, and later partly charged. Next it was partly discharged at a discharge rate of 800 mA and then fully discharged at 300 mA. At each of these points, calculation accuracy of the fuel gauge parameters was tested. The following figures illustrate the main points in the testing process.



Rapid/No Error

Vb=3.901 I=0.602 (Vr=0.060) T=28C (Rt=8814) SOC=1% TIME: 7:27:28

CORRECTION FACTORS

AGE CORRECTION

CELL AGE CORRECTION COEFFICIENT 0

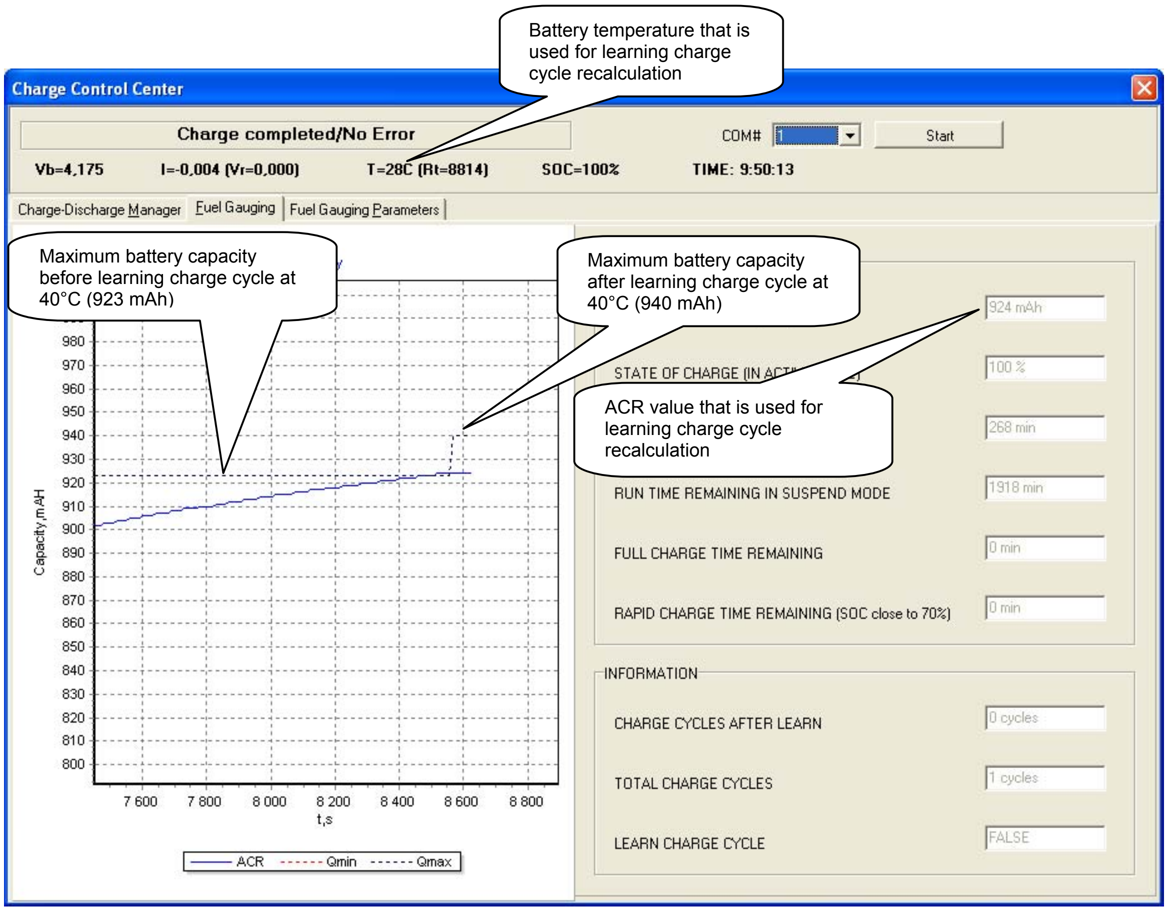
DISCHARGE RATE AND TEMPERATURE CORRECTION

	0C	10C	20C	30C	40C
FULL CHARGE	843 mAh	876 mAh	900 mAh	909 mAh	923 mAh
ACTIVE EMPTY	100 mAh	60 mAh	45 mAh	35 mAh	20 mAh
SUSPEND EMPTY	50 mAh	25 mAh	15 mAh	10 mAh	0 mAh

CHARGE BREAKPOINT AND TEMPERATURE CORRECTION

	0C	10C	20C	30C	40C
BREAKPOINT	215 mAh	266 mAh	286 mAh	296 mAh	324 mAh
BREAKPOINT/2	43 mAh	58 mAh	64 mAh	68 mAh	78 mAh
BREAKPOINT TIME	52 min	65 min	77 min	81 min	90 min

Figure 28. Fuel Gauge Correction Factors before Learning Charge Cycle


Battery temperature that is used for learning charge cycle recalculation
Maximum battery capacity before learning charge cycle at 40°C (923 mAh)
Maximum battery capacity after learning charge cycle at 40°C (940 mAh)
ACR value that is used for learning charge cycle recalculation
Charge Control Center
COM# 1
Start
Charge-Discharge Manager
Fuel Gauging
Fuel Gauging Parameters
Capacity,mAH
t,s
ACR
Qmin
Qmax

Charge completed/No Error

Vb=4.175 I=-0.004 (Vr=0.000) T=28C (Rt=8814) SOC=100% TIME: 9:50:13

	924 mAh
STATE OF CHARGE (IN ACT	100 %
	268 min
RUN TIME REMAINING IN SUSPEND MODE	1918 min
FULL CHARGE TIME REMAINING	0 min
RAPID CHARGE TIME REMAINING (SOC close to 70%)	0 min

INFORMATION

CHARGE CYCLES AFTER LEARN	0 cycles
TOTAL CHARGE CYCLES	1 cycles
LEARN CHARGE CYCLE	FALSE

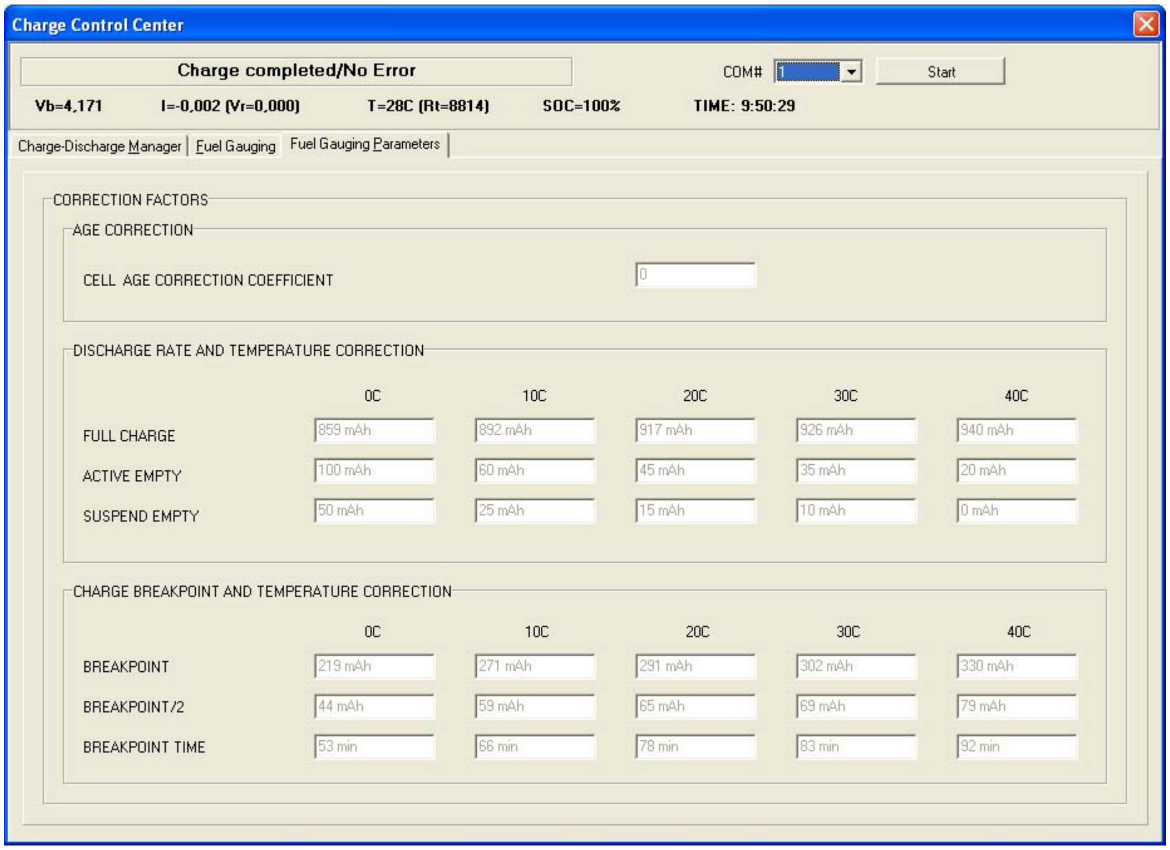
Figure 29. Fuel Gauge Profile with Zooming Learning Process

Figures 28, 29 and Figure 30 illustrate the learning process recalculation. Based on the actual total battery capacity (924 mAh) and the present temperature (28°C), the FULL CHARGE, BREAKPOINT, BREAKPOINT/2 and BREAKPOINT TIME points for all temperatures are proportionally recalculated (see Figure 21 FGLearFullCharge Function Fuel Gauge Algorithm). After the learning charge cycle, the maximum battery capacity at 40°C would be changed from 923 mAh to 940 mAh. Other changes are shown in Figures 28 and 30.

Note that the ACTIVE EMPTY and SUSPEND EMPTY points are unchanged.

Also note that the cell age correction coefficient was not enabled during testing and remained equal to zero. To enable calculation of the cell age coefficient and accommodate the change of total battery capacity over time (age correction), the user must discard the comment at *//#define FG_AGE* in the header file *globdefs.h* in the project folder.

But consideration must be made of additional conditions at the time this coefficient is calculated. For example, after the second learning charge cycle (Figure 27) a large change in total capacity occurs in a single cycle. As a result, the age coefficient will be great and with every following charge cycle, the total battery capacity (FULL CHARGE) after age correction should be corrected to a value that is lower (by an amount equal to capacity aging at 28°C per cycle. Accordingly, the measurement error of the fuel gauge parameters will greatly grow with time of use (until the next learning charge cycle). To address this problem, we could add the maximum age correction coefficient value and minimum cycles after the last learning charge cycle. If the count of the charge cycles after the learning charge cycle is smaller than the minimum value of the cycles or if the result age correction coefficient is greater than its maximum value, the result coefficient should be set to zero. This feature can be added in the next firmware release.



Charge completed/No Error

Vb=4.171 I=-0.002 (Vr=0.000) T=28C (Rt=8814) SOC=100% TIME: 9:50:29

CORRECTION FACTORS

AGE CORRECTION

CELL AGE CORRECTION COEFFICIENT: 0

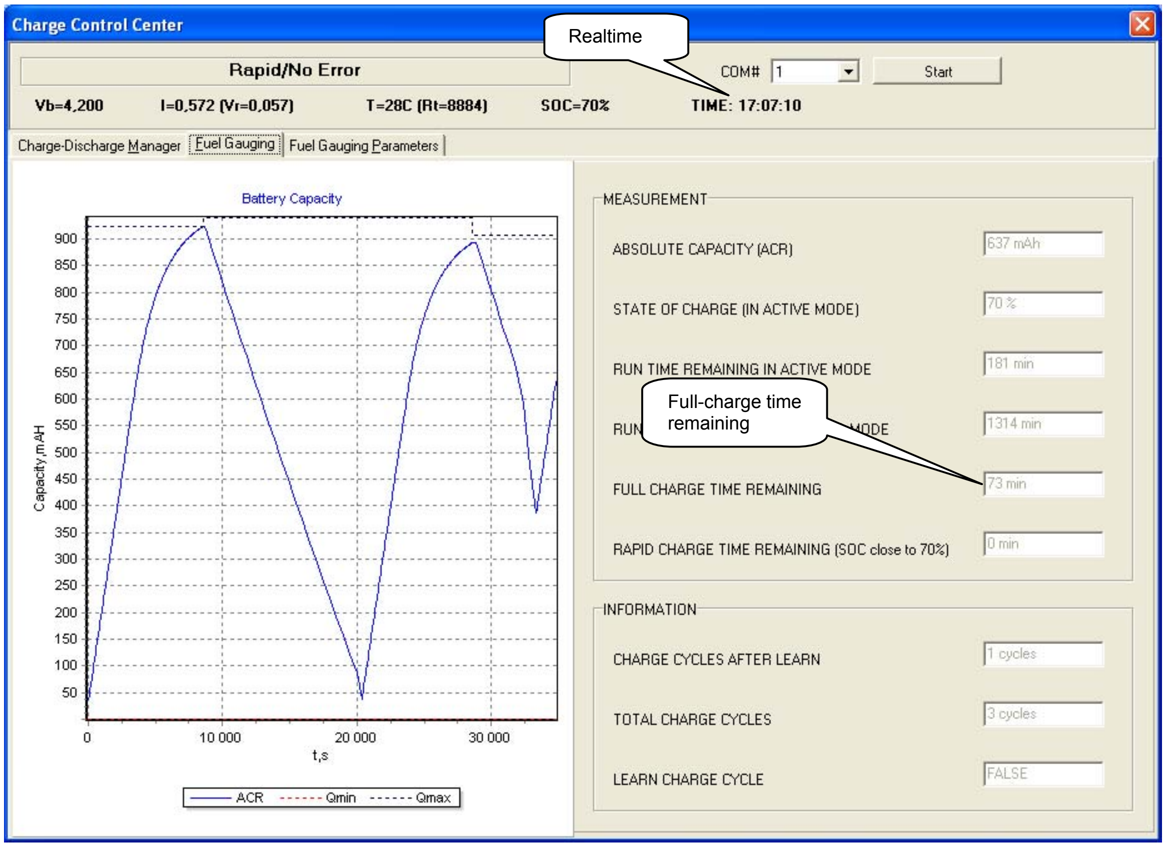
DISCHARGE RATE AND TEMPERATURE CORRECTION

	0C	10C	20C	30C	40C
FULL CHARGE	859 mAh	892 mAh	917 mAh	926 mAh	940 mAh
ACTIVE EMPTY	100 mAh	60 mAh	45 mAh	35 mAh	20 mAh
SUSPEND EMPTY	50 mAh	25 mAh	15 mAh	10 mAh	0 mAh

CHARGE BREAKPOINT AND TEMPERATURE CORRECTION

	0C	10C	20C	30C	40C
BREAKPOINT	219 mAh	271 mAh	291 mAh	302 mAh	330 mAh
BREAKPOINT/2	44 mAh	59 mAh	65 mAh	69 mAh	79 mAh
BREAKPOINT TIME	53 min	66 min	78 min	83 min	92 min

Figure 30. Fuel Gauge Correction Factors after Learning Charge Cycle

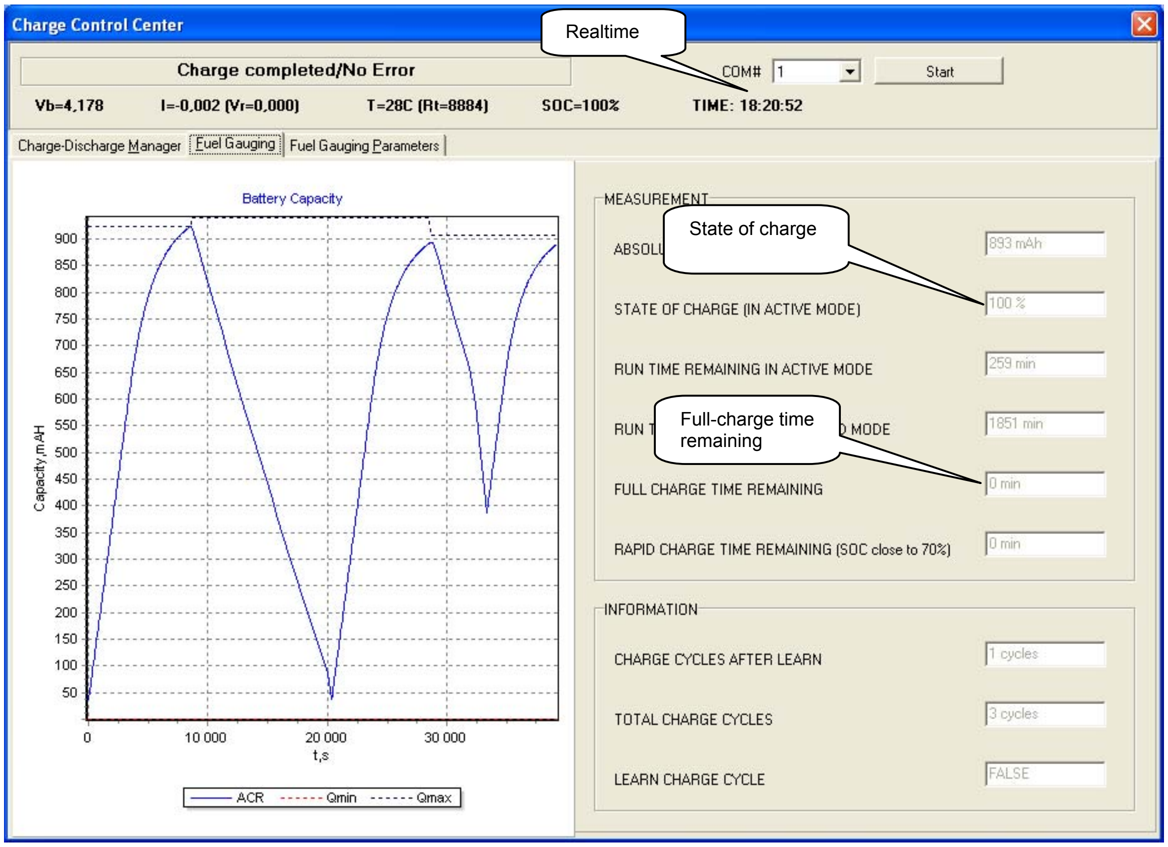


Rapid/No Error

Vb=4,200 I=0,572 (Vr=0,057) T=28C (Rt=8884) SOC=70% TIME: 17:07:10

ABSOLUTE CAPACITY (ACR)	637 mAh
STATE OF CHARGE (IN ACTIVE MODE)	70 %
RUN TIME REMAINING IN ACTIVE MODE	181 min
RUN … MODE	1314 min
FULL CHARGE TIME REMAINING	73 min
RAPID CHARGE TIME REMAINING (SOC close to 70%)	0 min
CHARGE CYCLES AFTER LEARN	1 cycles
TOTAL CHARGE CYCLES	3 cycles
LEARN CHARGE CYCLE	FALSE

Figure 31. Fuel Gauge Profile with Sample Charge State



Charge Control Center

Charge completed/No Error

COM# 1

Start

Vb=4.178 I=-0.002 (Vr=0.000) T=28C (Rt=8884) SOC=100% TIME: 18:20:52

Charge-Discharge Manager | Fuel Gauging | Fuel Gauging Parameters

MEASUREMENT

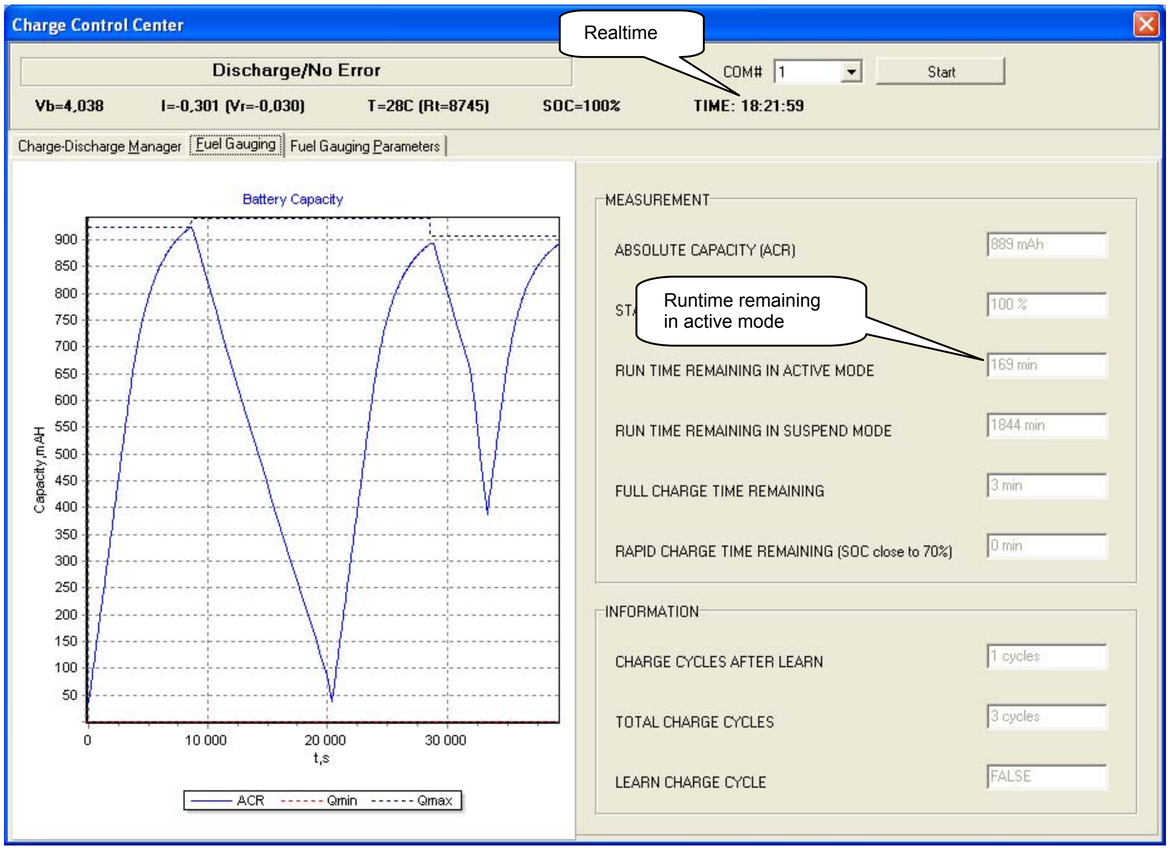
ABSOLU	893 mAh
STATE OF CHARGE (IN ACTIVE MODE)	100 %
RUN TIME REMAINING IN ACTIVE MODE	259 min
RUN T MODE	1851 min
FULL CHARGE TIME REMAINING	0 min
RAPID CHARGE TIME REMAINING (SOC close to 70%)	0 min

INFORMATION

CHARGE CYCLES AFTER LEARN	1 cycles
TOTAL CHARGE CYCLES	3 cycles
LEARN CHARGE CYCLE	FALSE

Figure 32. Fuel Gauge Profile with Full Charge End

Figure 31 shows that at 17:07, 73 minutes is remaining until the fully charged state. Figure 32 shows that a fully charged state is reached at 18:20 or 73 minutes after the sample point.



Charge Control Center

Discharge/No Error

COM# 1 Start

Vb=4,038 I=-0,301 (Vr=-0,030) T=28C (Rt=8745) SOC=100% TIME: 18:21:59

Charge-Discharge Manager | Fuel Gauging | Fuel Gauging Parameters

MEASUREMENT

ABSOLUTE CAPACITY (ACR)	889 mAh
ST	100 %
RUN TIME REMAINING IN ACTIVE MODE	169 min
RUN TIME REMAINING IN SUSPEND MODE	1844 min
FULL CHARGE TIME REMAINING	3 min
RAPID CHARGE TIME REMAINING (SOC close to 70%)	0 min

INFORMATION

CHARGE CYCLES AFTER LEARN	1 cycles
TOTAL CHARGE CYCLES	3 cycles
LEARN CHARGE CYCLE	FALSE

Figure 33. Fuel Gauge Profile with Discharging Start

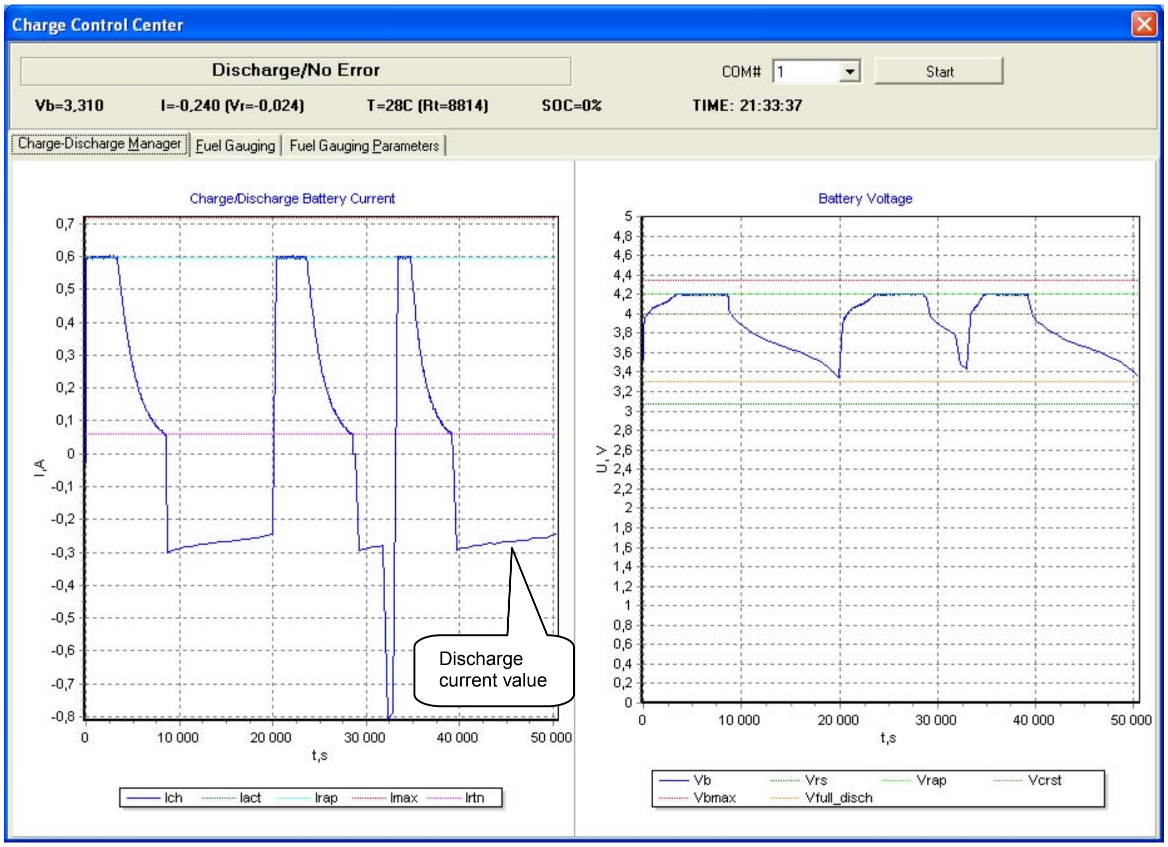
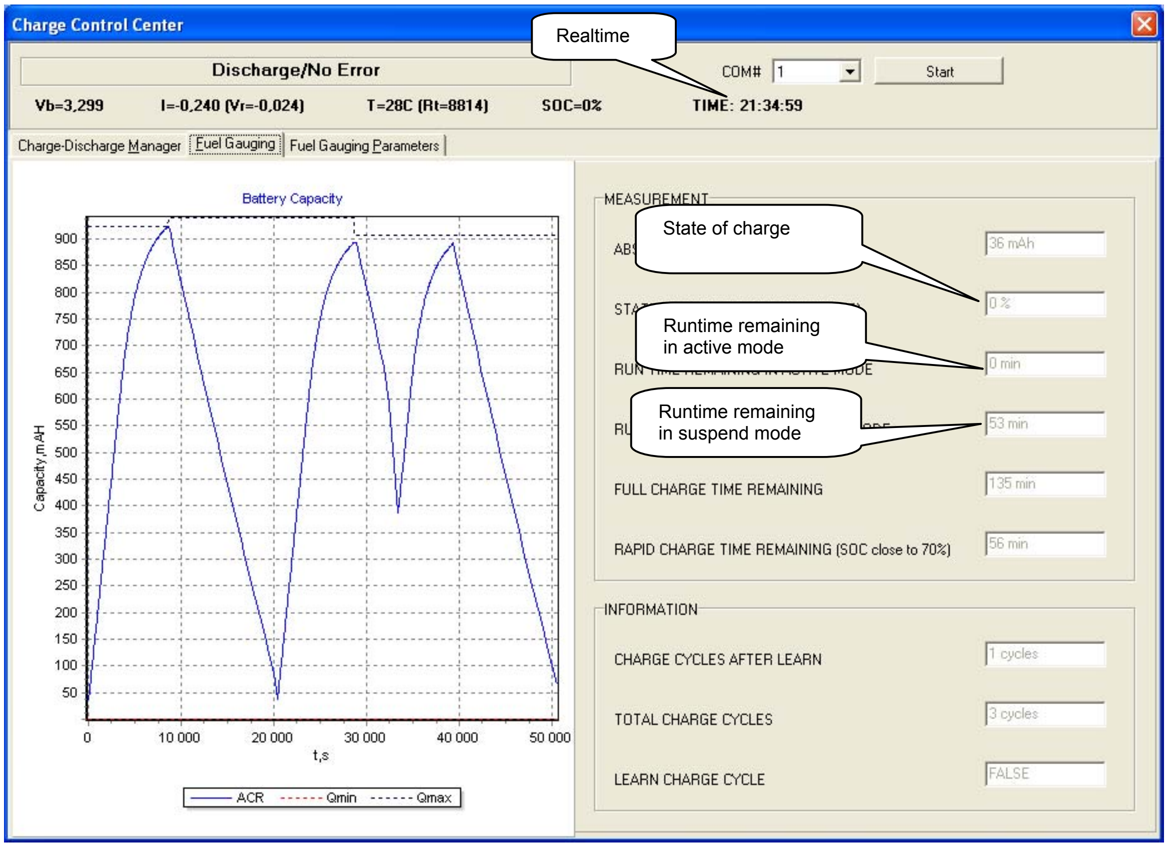


Figure 34. Charge/Discharge Manager Profile with Discharge Current Behavior


Charge Control Center

Discharge/No Error

COM# 1

Start

Vb=3,299 I=-0,240 (Vr=-0,024) T=28C (Rt=8814) SOC=0% TIME: 21:34:59

Charge-Discharge Manager | Fuel Gauging | Fuel Gauging Parameters

MEASUREMENT

ABS	36 mAh
STA	0 %
RUN TIME REMAINING IN ACTIVE MODE	0 min
RU	53 min
FULL CHARGE TIME REMAINING	135 min
RAPID CHARGE TIME REMAINING (SOC close to 70%)	56 min

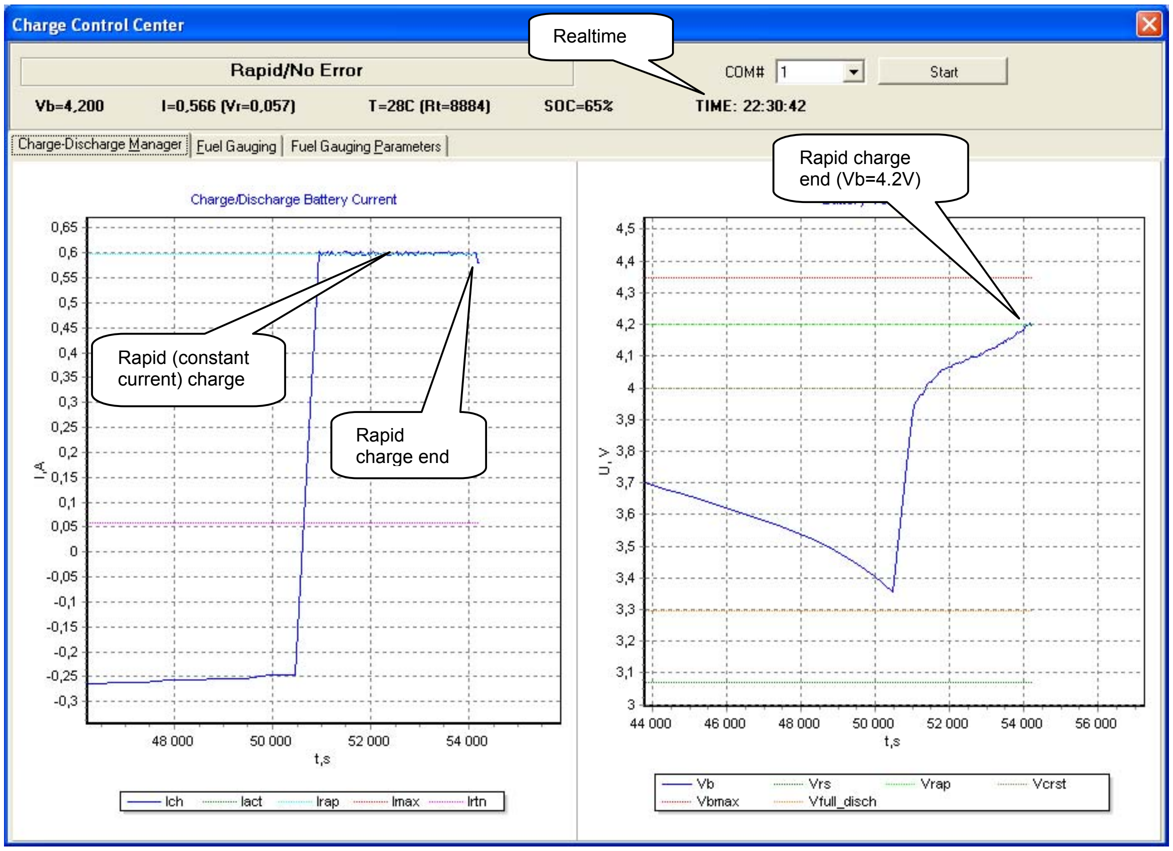
INFORMATION

CHARGE CYCLES AFTER LEARN	1 cycles
TOTAL CHARGE CYCLES	3 cycles
LEARN CHARGE CYCLE	FALSE

Figure 35. Fuel Gauge Profile with Discharging End

Figure 33 shows that at 18:21, 169 minutes remain until fully discharged state in active mode. Figure 34 illustrates the discharge current behavior. Figure 35 shows that a fully discharged state with state of charge equal to 0% is reached at 21:34 or 193 minutes after start of discharge.

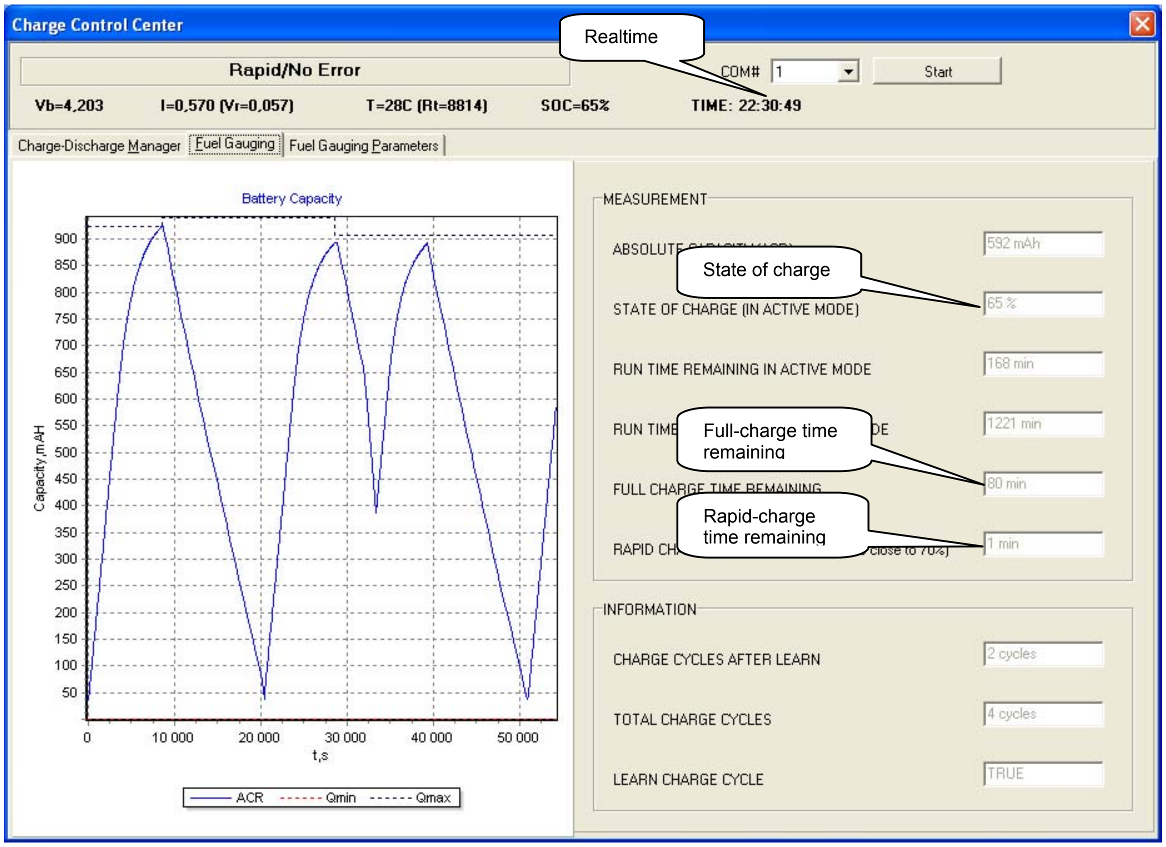
The error in time from start point to end point appears because at the start point the remaining time was calculated for constant current but that value decreased with time. Furthermore, such a battery that is fully discharged at the active discharge rate is still capable of being further discharged at the low suspend current rate (Isuspend=30 mA) for 53 minutes.


Charge Control Center

Rapid/No Error

Vb=4,200 I=0,566 (Vr=0,057) T=28C (Rt=8884) SOC=65% TIME: 22:30:42

Figure 36. Charge/Discharge Manager Profile with Zooming Rapid-Charge End



Rapid/No Error

COM# 1

Vb=4,203 I=0,570 (Vr=0,057) T=28C (Rt=8814) SOC=65% TIME: 22:30:49

MEASUREMENT

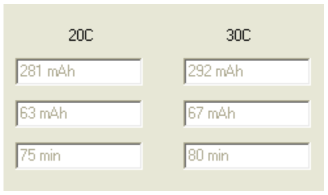
ABSOLUTE	592 mAh
STATE OF CHARGE (IN ACTIVE MODE)	65 %
RUN TIME REMAINING IN ACTIVE MODE	168 min
RUN TIME DE	1221 min
FULL CHARGE TIME REMAINING	80 min
RAPID CH close to 70%)	1 min

INFORMATION

CHARGE CYCLES AFTER LEARN	2 cycles
TOTAL CHARGE CYCLES	4 cycles
LEARN CHARGE CYCLE	TRUE

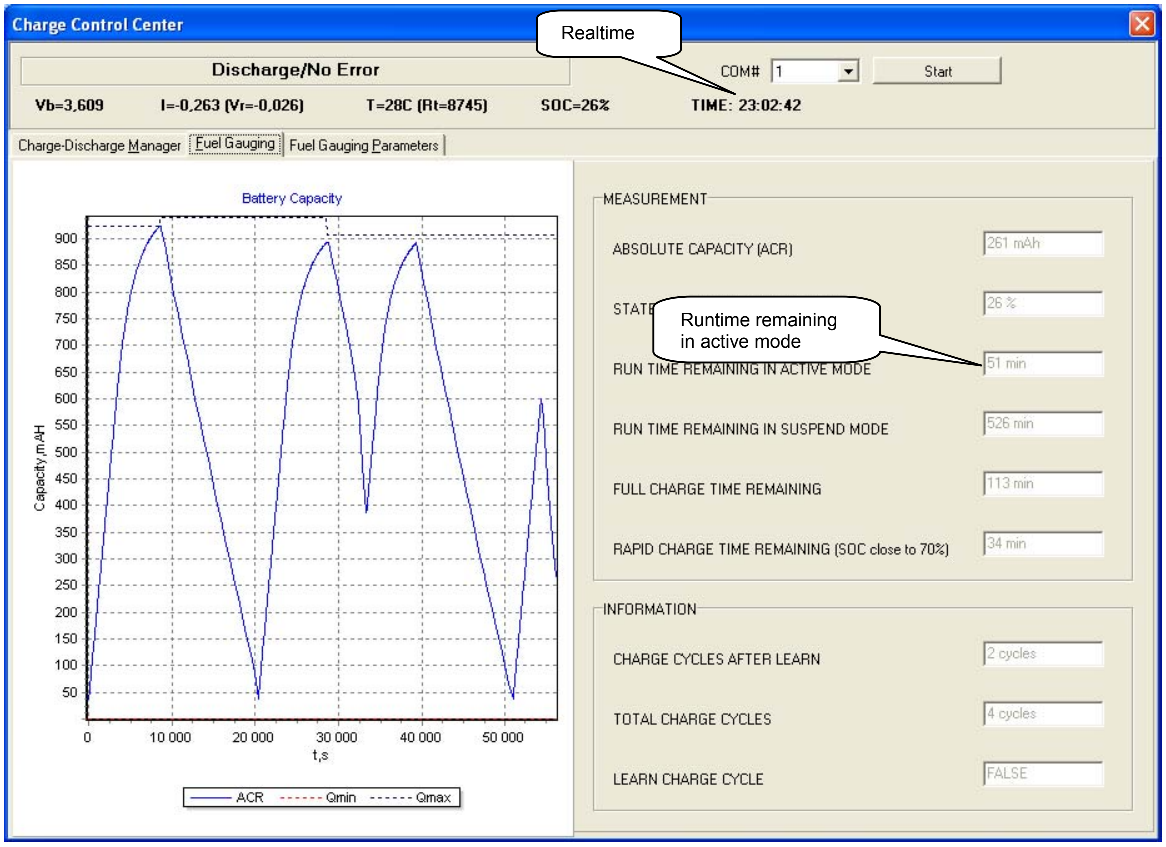
Figure 37. Fuel Gauge Profile with Rapid-Charge End

Figures 36 and 37 show the accuracy of calculation for the rapid-charge time remaining parameter. Note that the time from breakpoint to full charge point at 28°C is equal to 79 minutes (see Figure 38).



20C	30C
281 mAh	292 mAh
63 mAh	67 mAh
75 min	80 min

Figure 38. Fuel Gauge Charge Breakpoint and Temperature Correction Profile after Second Learning Charge Cycle

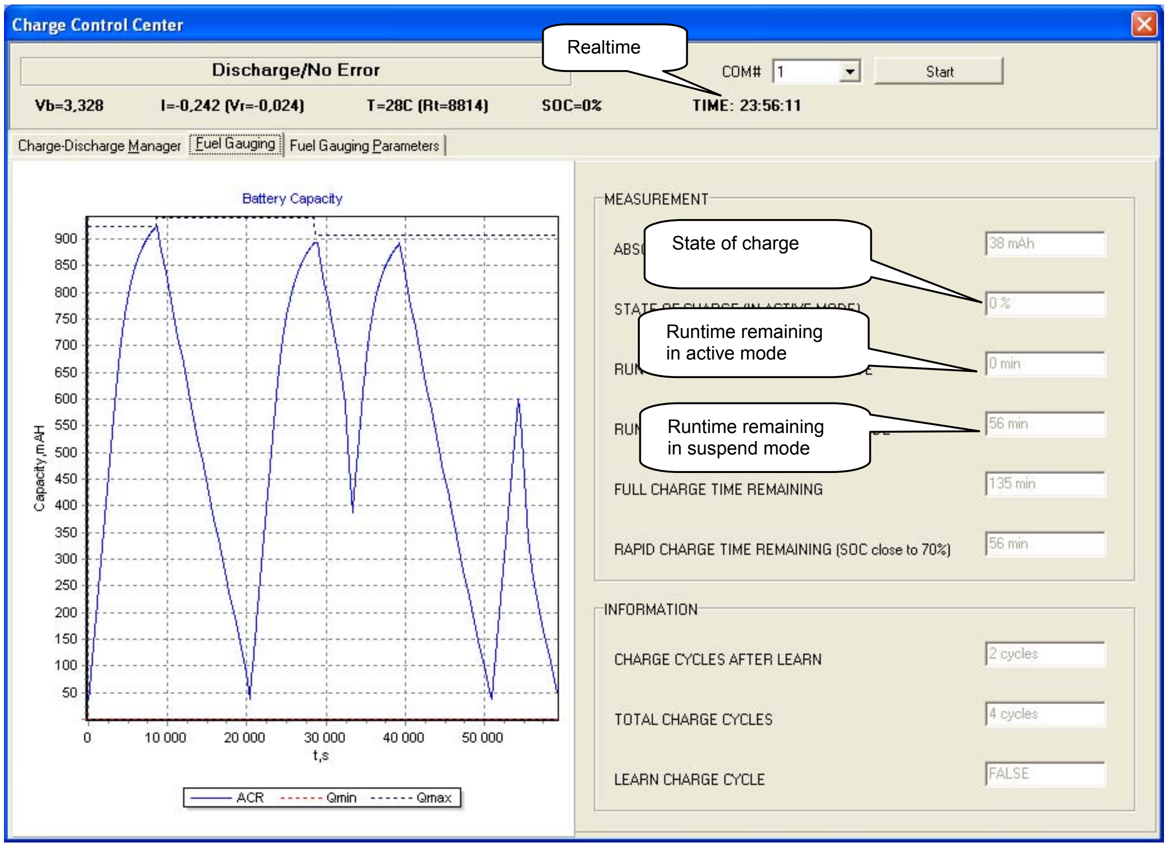


Discharge/No Error

Vb=3,609 I=-0,263 (Vr=-0,026) T=28C (Rt=8745) SOC=26% TIME: 23:02:42

ABSOLUTE CAPACITY (ACR)	261 mAh
STATE	26 %
RUN TIME REMAINING IN ACTIVE MODE	51 min
RUN TIME REMAINING IN SUSPEND MODE	526 min
FULL CHARGE TIME REMAINING	113 min
RAPID CHARGE TIME REMAINING (SOC close to 70%)	34 min
CHARGE CYCLES AFTER LEARN	2 cycles
TOTAL CHARGE CYCLES	4 cycles
LEARN CHARGE CYCLE	FALSE

Figure 39. Fuel Gauge Profile with Small Region Start Discharging where Current is Close to Constant



Discharge/No Error

Vb=3,328 I=-0,242 (Vr=-0,024) T=28C (Rt=8814) SOC=0% TIME: 23:56:11

MEASUREMENT

ABS	38 mAh
STATE OF CHARGE (IN ACTIVE MODE)	0 %
RUN	0 min
RUN	56 min
FULL CHARGE TIME REMAINING	135 min
RAPID CHARGE TIME REMAINING (SOC close to 70%)	56 min

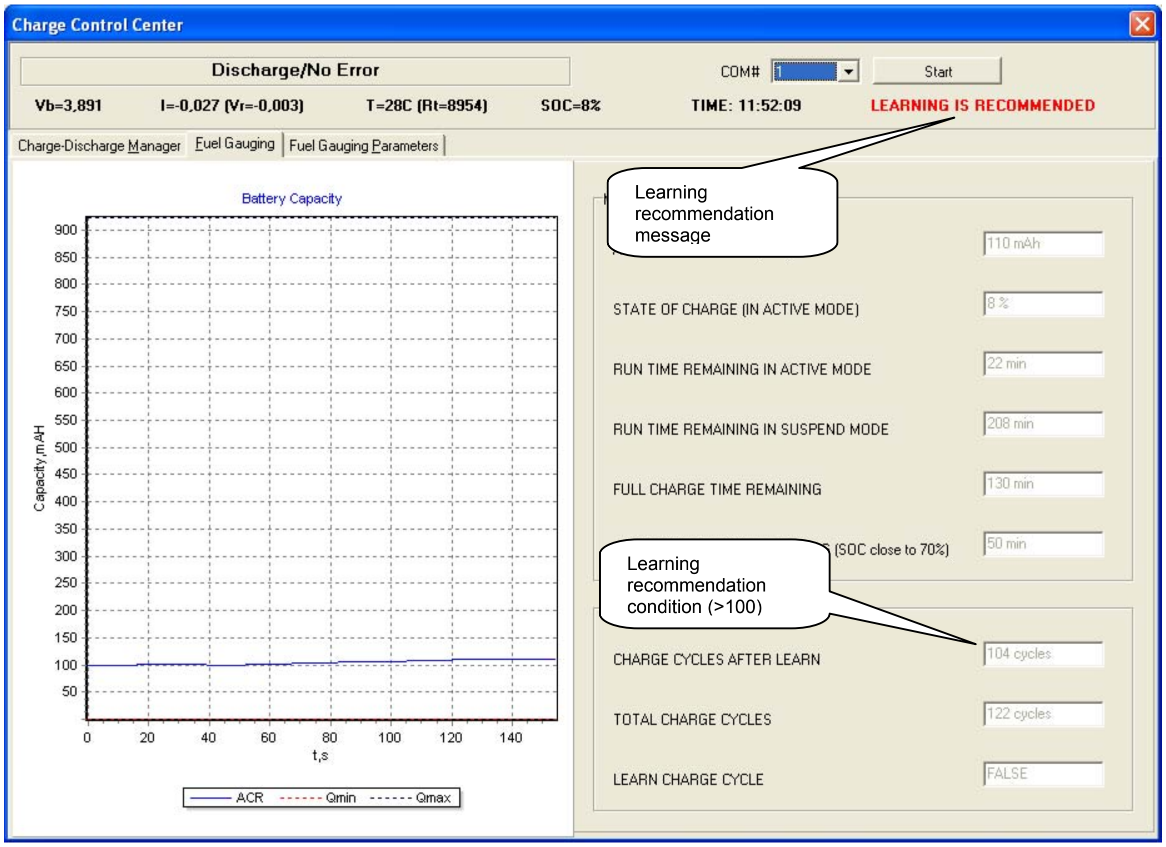
INFORMATION

CHARGE CYCLES AFTER LEARN	2 cycles
TOTAL CHARGE CYCLES	4 cycles
LEARN CHARGE CYCLE	FALSE

Figure 40. Fuel Gauge Profile with Discharging End

Figure 39 shows that at 23:02:42, 51 minutes remain until fully discharged state in active mode.

Figure 40 shows that the fully discharged state with a state of charge equal to 0% is reached at 23:56:11 or 53 minutes 29 seconds after the start of discharge. The discharge current behavior on such a small region is nearly constant (but also slightly decreased with time).


Charge Control Center

Discharge/No Error

COM# 1 Start

Vb=3.891 I=-0.027 (Vr=-0.003) T=28C (Rt=8954) SOC=8% TIME: 11:52:09 LEARNING IS RECOMMENDED

STATE OF CHARGE (IN ACTIVE MODE): 8 %

RUN TIME REMAINING IN ACTIVE MODE: 22 min

RUN TIME REMAINING IN SUSPEND MODE: 208 min

FULL CHARGE TIME REMAINING: 130 min

(SOC close to 70%): 50 min

CHARGE CYCLES AFTER LEARN: 104 cycles

TOTAL CHARGE CYCLES: 122 cycles

LEARN CHARGE CYCLE: FALSE

Figure 41. Fuel Gauge Profile with Learning Recommendation Message

Figure 41 shows an example of the learning recommendation message.

The PSoC device transmits this message if the charge cycle count after the last learning charge cycle is greater than the predefined value (100) or in a fully charged battery state if the ACR value is different from the expected value (FULL CHARGE) by more than the predefined value (20 mA).

Table 7. Summarized Result of the Proposed Fuel Gauge during Charge Parameter Calculation

Realtime	Full-Charge Time Remaining	SOC
17:07:10	73 minutes	70%
18:20:52	0 minutes	100%
∑ 73 minutes 38 seconds	∑ 73 minutes	

Table 8. Summarized Result of the Proposed Fuel Gauge during Discharge Parameters Calculation

Realtime	Runtime Remaining In Active Mode	Runtime Remaining In Suspend Mode	SOC
23:02:42	51 minutes	526 minutes	26%
23:56:11	0 minutes	56 minutes	0%
∑ 53 minutes 29 seconds	∑ 51 minutes		

Note that the error at 2 minutes 29 seconds appears because the current value decreases slightly with time (see Figure 26) and thereby increases the battery working time in active mode.

About the Author

Name:	Oleksandr Karpin
Title:	Post-Graduate Student
Background:	Oleksandr earned his computer-engineering diploma in 2001 from National University "Lvivska Polytechnika" (Lviv, Ukraine), and continues his studies there as a post-graduate student. His interests include embedded systems design and new technologies.
Contact:	karpinoo@ukr.net

Cypress Semiconductor
2700 162nd Street SW, Building D
Lynnwood, WA 98087
Phone: 800.669.0557
Fax: 425.787.4641

http://www.cypress.com/
Copyright © 2005 Cypress Semiconductor Corporation. All rights reserved.
"Programmable System-on-Chip," PSoC, PSoC Designer and PSoC Express are trademarks of Cypress Semiconductor Corp.
All other trademarks or registered trademarks referenced herein are the property of their respective owners.
The information contained herein is subject to change without notice. Made in the U.S.A.