AN4870
Application note

STC3117 system integration

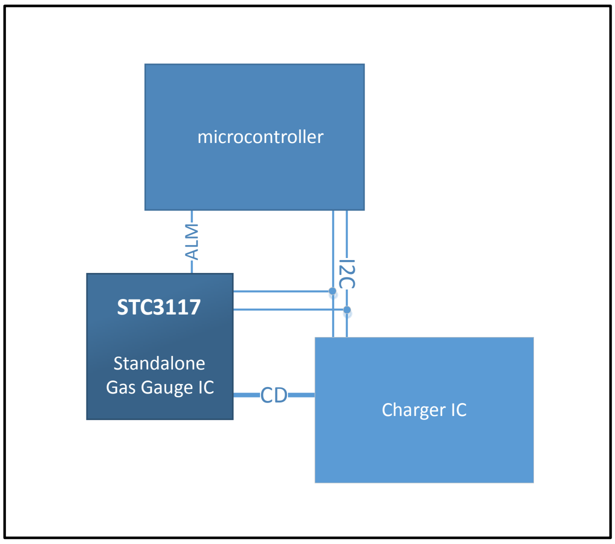
Introduction

Mobile application users demand accurate battery capacity monitoring. In particular, they are interested to know if the battery state of charge (SOC) in stressful conditions is logical and accurate. To this end, STMicroelectronics has developed a family of devices which use an internal algorithm to provide the user with accurate and reliable information about the battery's SOC. The new STC3117 chip can be used in any handheld electronic mobile phone or tablet.

The purpose of this application note is to provide the basics of how to build an accurate fuel gauge system using the STC3117. A complete fuel gauge system is a combination of hardware and software functions. The STC3117 has been designed to convert the physical nonlinear measurements of lithium-ion battery models to stable and compensated information, which can easily be reported to the final user of a mobile application. Chipset manages the real-time aspects of the battery gauging. It combines the voltage method and coulomb counter to provide the best accuracy in every application state. The application software is concerned with the reporting and calculation of application-dependent information.

The STC3117 is a system, side-battery, monitoring IC which allows batteries to be tracked from any battery manufacturer. Thanks to the standard I²C communication port, there is no need for a specific bus communication with the battery or an additional battery connector pin. Many features are embedded in the device to manage application use cases. These embedded features allow the best accuracy to be achieved in every battery connection situation. One of the most useful features of the STC3117 is its direct connection to the charger which permits the charger IC to be controlled without any additional GPIO control from the microcontroller.

Figure 1: STC3117 architecture overview



Contents

List of tables

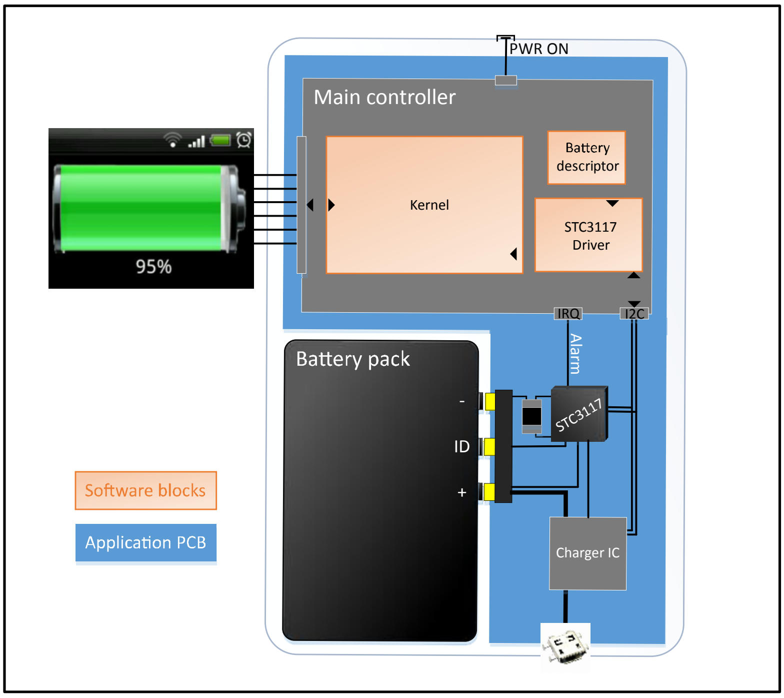
List of figures

1 Architectural overview

The system architecture has been separated into two main parts in order to facilitate the most flexible integration. From the system side, the STC3117 takes physical measurements and reports the battery state of charge (SOC) through the I²C interface while the software layer is concerned with battery parameter initialization (STC3117 initialization) and compensation using all the dedicated features of the chipset.

Depending on the targeted accuracy of the application environment, the STC3117 can be used alone without software compensation. In this configuration, the application software uses only the STC3117 register information to display the battery SOC to the user. If the temperature and aging effects need to be compensated, the software layer uses the STC3117 dedicated features.

Figure 2: System architecture overview



2 Chipset configuration parameters

This section describes how to calculate the parameters of the application.

To operate properly, the STC3117 uses typical values of the battery to track the battery state of charge (SOC) in the application. In other words, the STC3117 directly provides a battery SOC (in percent) based on physical measurements such as current, voltage, and temperature.

The parameter set is commonly called the "battery model" and can be calculated using the formulae below. This model is an image of battery behavior against the STC3117 principle. The parameters and their calculation are also described in the STC3117 datasheet.

- REG_CC_CNF: (Rsense x Nominal battery capacity) / 49.556
- REG_VM_CNF: (Internal battery impedance x Nominal battery capacity) / 977.78
- REG_ALARM_SOC has to be set according to the application needs (e.g. 10 % as a typical value)
- REG_ALARM_VOLTAGE has to be set according to the application needs (e.g. 3600 mV as a typical value)
- REG_CURRENT_THRES: (Battery capacity / 10), up to 200 mA

A lot of information is mentioned in battery datasheets and often, this information is displayed on the battery package. However, some parameters are not accurately provided by manufacturers. Such parameters can be easily measured (see *Section 2.1* and *Section 2.2*).

2.1 Battery internal impedance identification

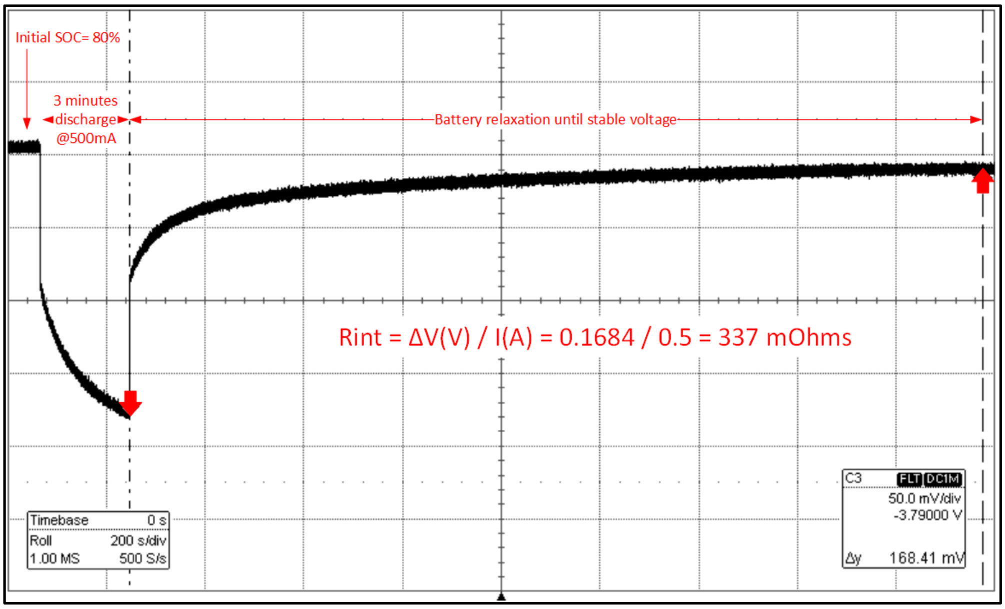
Thanks to its internal OptimGauge algorithm, the STC3117 does not depend on the variation of the internal battery impedance over the state of charge (SOC) range. In other words, only one internal battery impedance value is necessary to represent internal battery impedance behavior at a given temperature.

The internal impedance factor can be measured using several solutions which are described in the literature. One solution is to:

- measure the battery voltage at the end of several minutes discharge (with a fixed current value)
- measure the battery voltage again after 30 minutes of relaxation
- calculate the voltage difference
- use the discharge current value to find the internal battery impedance

To use this method successfully, a value of around 80 % SOC allows the VM_CNF parameter (see datasheet) to be calculated with enough accuracy to track the battery SOC correctly.

Figure 3: Internal battery impedance measurement example



2.2 Battery open circuit voltage (OCV) curve measurement and calculation

The content of the default OCV curve in the STC3117 registers at chipset startup can be easily adjusted to fit the battery OCV model. The REG_OCV_TABx and REG_SOC_TABx registers have to be filled by standard I²C write operations which reflect the real battery OCV curve.

The battery OCV curve can be characterized and evaluated by measuring the relaxed voltage of the battery at a given state of charge (SOC). Each data point of the table below has to be filled by its measured value. For example, the "Adapted OCV (mV)" cells in column 7 should be measured using the relaxed voltages of the corresponding "Default SOC (%)" values in column 2. These measured points can then be used to define the REG_OCV_TAB (in 0.55 mV unit) in column 8.

Table 1: How to calculate OCV values

Table entry	Default SOC (%)	Default OCV			Adapted OCV		
		Address	OCV (mV)	Hex (0.55 mV)	Address	OCV (mV)	Hex (0.55 mV)
0	0	0x30	3300	1770	0x30		
1	3	0x32	3541	1926	0x32		
2	6	0x34	3618	19B2	0x34		
3	10	0x36	3658	19FB	0x36		
4	15	0x38	3695	1A3E	0x38		
5	20	0x3A	3721	1A6D	0x3A		
6	25	0x3C	3747	1A9D	0x3C		
7	30	0x3E	3761	1AB6	0x3E		

Table entry	Default SOC (%)	Default OCV			Adapted OCV		
		Address	OCV (mV)	Hex (0.55 mV)	Address	OCV (mV)	Hex (0.55 mV)
8	40	0x40	3778	1AD5	0x40		
9	50	0x42	3802	1B01	0x42		
10	60	0x44	3863	1B70	0x44		
11	65	0x46	3899	1BB1	0x46		
12	70	0x48	3929	1BE8	0x48		
13	80	0x4A	3991	1C58	0x4A		
14	90	0x4C	4076	1CF3	0x4C		
15	100	0x4E	4176	1DA9	0x4E		

3 Schematic guidelines

The STC3117 is designed to limit the external component amount versus accuracy of the reported information. The device provides standard application signal interfaces (i.e. BatD, CD) that are used to give the best accuracy in specific application-use cases. Also, depending on the chipset configuration and accuracy target, an external sense resistor should be used which helps provide the best accuracy versus flexibility in mixed mode.

3.1 Sense resistor selection

To use the STC3117 enhanced mixed mode, a sense resistor must be placed between the battery minus pin and the application ground. The sense resistor should be capable of measuring current accurately so, it is imperative not to connect any signal on the battery minus pin except the sense resistor.

The external sense resistor measures the current which is used internally by the device and integrated over time to provide a coulomb counter feature which tracks the battery state of charge (SOC). The choice of sense resistor is extremely important and has a direct impact on battery monitoring accuracy. The following points concerning the sense resistor value and reference should be considered:

1. Sense resistor impedance value
2. Sense resistor impedance accuracy
3. Sense resistor power dissipation capability
4. Sense resistor deviation over temperature range (TCR)

Note that a sense resistor selection proposal is available in *Table 2: "Proposed sense resistor references"*).

3.1.1 Sense resistor impedance value

The sense resistor impedance depends on the application maximum current and the STC3117 Vin_gg input voltage range (see datasheet).

Vin_gg = Rcg x I_{Max}

Where Rcg = sense resistor value in mΩ and I_{Max} = application peak current in A

Vin_gg must be below 40 mV for I_{Max} charging and above -40 mV for I_{Max} discharging.

Example: a sense resistor of 20 mΩ is chosen for a maximum application current of 1.7 A.

3.1.2 Sense resistor impedance accuracy

The precision of the sense resistor depends on the targeted accuracy of the application. An accuracy of 1 % allows the STC3117 to attain its best accuracy. An accuracy of 2 % allows the cost to be decreased and allows the voltage mode to correct the column counting drift error during application low-power mode.

3.1.3 Sense resistor power dissipation capability

To limit self-heating of the sense resistor due to current flow into it, the power dissipation capability of the chosen sense reference has to be in line with the application current consumption. Assuming a voltage drop of 40 mV on the sense resistor and in the case of a 10 mΩ resistor, the sense resistor power dissipation capability has to be at least 160 mW.

3.1.4 Sense resistor deviation over temperature range (TCR)

The sense resistor impedance value depends on ambient temperature. The generated error from temperature deviation has to be limited in order to ensure that the impact of temperature on current measurement accuracy is minimized. A 0.3 % impedance value deviation at maximum temperature deviation is acceptable as it does not impact the current measurement accuracy over the temperature range too much.

Example: a 75 ppm TCR for a sense resistor of 10 mΩ, with a 40 °C maximum temperature deviation from typical temperature:

MaxError: ((TCR x Rsense x d) / Rsense) x 100 = ((0.000075 x 0.01 x 40) / 0.01) x 100 = 0.3 %

Where d = temperature in ˚C

3.1.5 Sense resistor selection proposal

Table 2: Proposed sense resistor references

Product reference	Manufacturer	Impedance	Accuracy	Power dissipation	Sense resistor deviation over temperature range (TCR)	Package
ERJM03NF10MV	PANASONIC	10 mΩ	±1 %	250 mW	±100 ppm/°C	0603
WSL0603R0100FEA	VISHAY DALE		±1 %	100 mW	±75 ppm/°C	

3.2 BatD pin system integration

The BatD pin detects battery removal events: either the third battery pin is an ID pin or a thermal information pin. The following three situations describe several application cases where the battery is removed and how the STC3117 detects such an event:

1. **Battery is removed during a low consumption stage:** At this time, the application consumption is low and the Vbat voltage can take time to decrease after battery removal (application capacitor discharging period). At the same time, the third battery pin is disconnected and the BatD pin voltage is fixed by the pull-up voltage value. Battery removal is detected rapidly by the BatD feature.
2. **Battery is removed during a high consumption stage:** At this time the application consumption is high and the Vbat voltage decreases rapidly. The STC3117 detects battery removal and in this case the BatD pin is not used.
3. **Battery is removed if the charger is already connected:** At this time, Vbat voltage maintains the same value or increases slightly. The BatD pin is used to detect battery removal as the battery voltage information cannot be used.

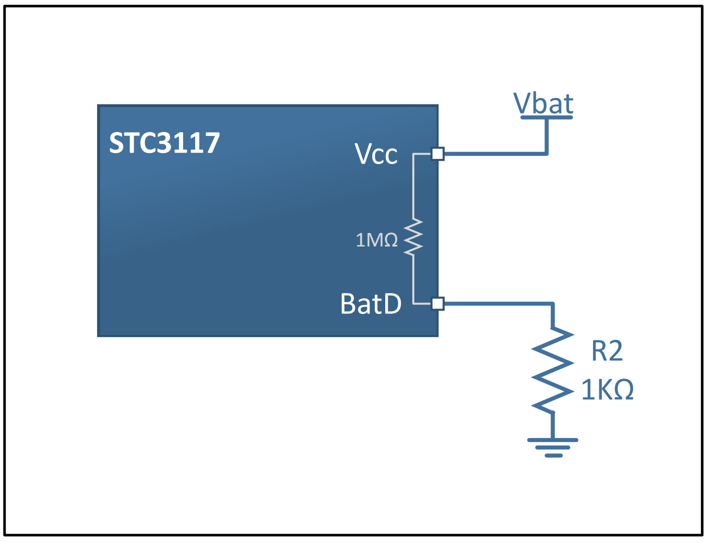
During battery removal, the BatD pull-up voltage should be ON and higher than 1.6 V internal threshold.

For most applications, the pull-up voltage should always be available and above the threshold. The best voltage to use is the Vbat voltage itself.

The following configurations can be used. In each one, the R2 resistor is mandatory.

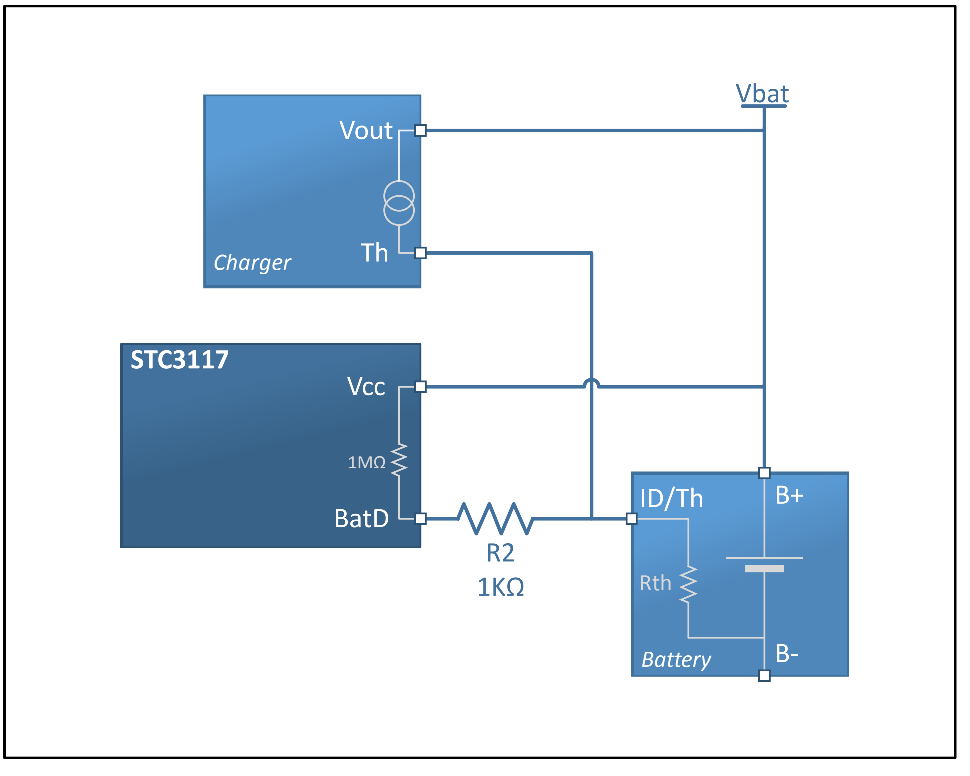
Configuration 1: If the STC3117 BatD pin is pulled down to ground, the BatD feature is disabled. Battery removal and swap events are only detected using the battery voltage reference.

Figure 4: STC3117 application integration, BatD pin is not used



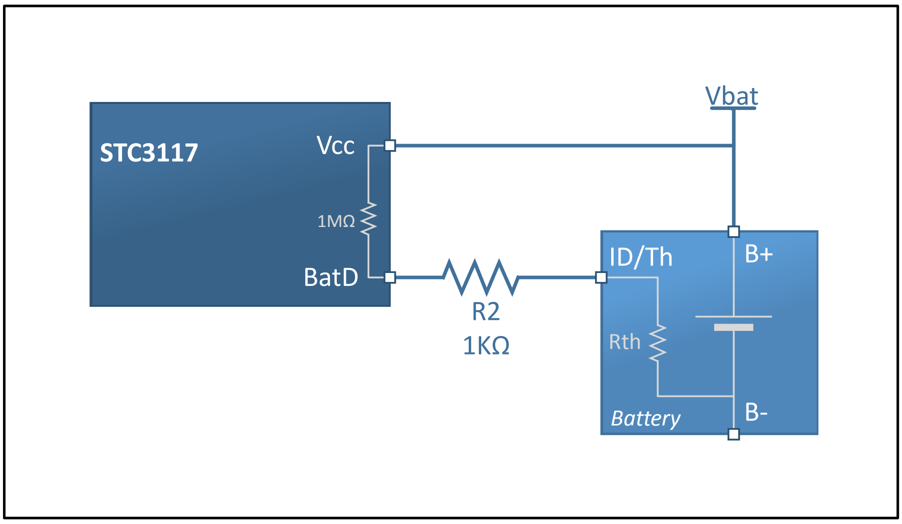
Configuration 2: If the third battery pin is used by the charger thermal protection signal, the STC3117 BatD pin has to be connected to this signal and to the charger. This does not modify the charger thermal security because the BatD pin is input only.

Figure 5: STC3117 application integration if charger provides an internal biasing current



Configuration 3: The third battery pin can be connected to the STC3117 BatD pin even if it has no charger or if the application charger does not manage the battery security pin. In this configuration, the BatD feature improves battery presence detection and provides useful information about battery presence to the application.

Figure 6: STC3117 application integration, battery security pin connected to BatD pin only



3.3 CD pin system integration

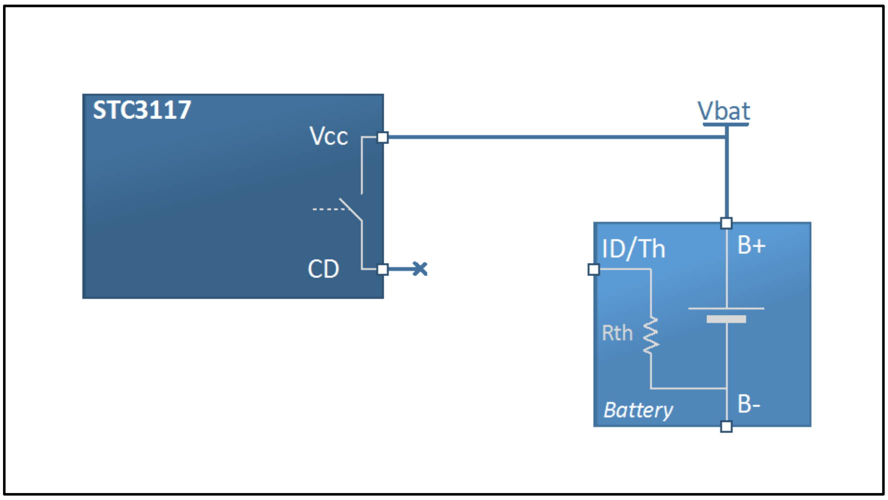
The STC3117 CD feature provides a control signal for a charger disable pin. This pin, the CD pin, is designed to maintain the charger OFF during the first battery open circuit voltage (OCV) measurement just after battery insertion. It can also disable the charger during normal application operations when it is controlled by software through the I^2C interface.

The STC3117 CD feature maintains the accuracy of the first OCV measurement in all application start-up sequences. This feature prevents the first OCV measurement from a non-null charging current. The CD signal is forced high during the first OCV measurement sequence to disable the charger and to guarantee measurement accuracy.

The CD pin can be directly controlled by software with the FORCE_CD bit of the REG_MODE register.

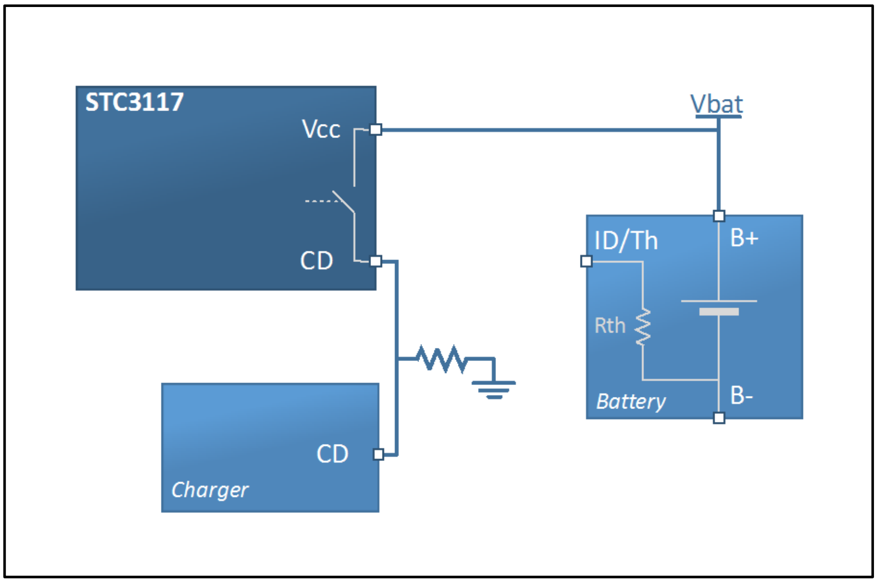
Configuration 1: The CD pin is not used and remains "unconnected".

Figure 7: STC3117 application integration, CD pin not used



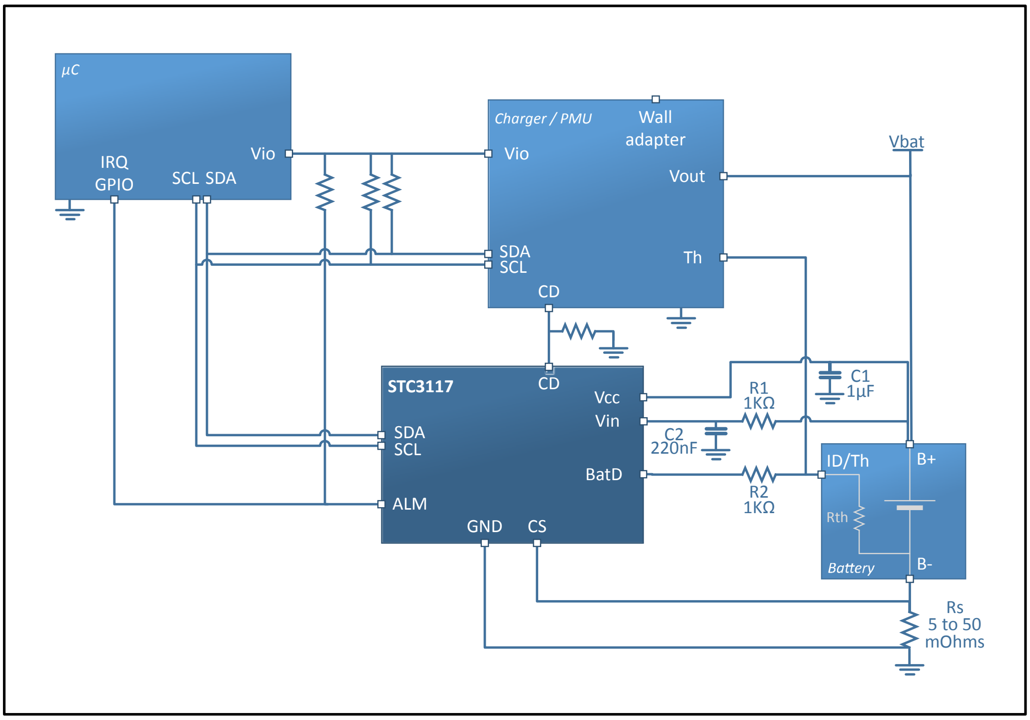
Configuration 2: The CD pin is used and is connected to the charger.

Figure 8: STC3117 application integration, CD pin connected to charger



4 Full application schematic including BatD and CD pin connections

Figure 9: Full application schematic

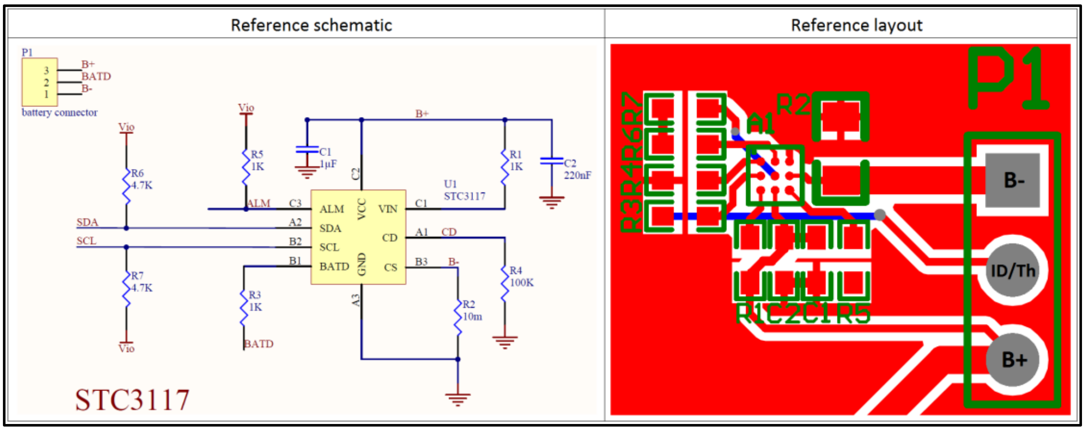


5 Place and route guidelines

The four layout "rules" listed below should be followed to achieve the best STC3117 accuracy in an application.

1. Connect the Vin signal to the battery connector with a specific track and not directly through a Vbat+ plan.
2. Connect an external sense resistor if the STC3117 is used in mixed mode.
3. Consider the temperature with regard to the placement of the battery, the STC3117, and the components of the application.
4. Use the STMicroelectronics PCB design recommendations.

Figure 10: Reference design figures: schematic and layout



5.1 Independent Vin connection

The Vin pin should be connected to the battery connector with a specific track and not directly through a Vbat+ plan or power track. This is to improve the voltage measurement accuracy during the first open circuit voltage (OCV) and during the life of the application.

The voltage drop coming from track impedance and the amount of application current is not negligible compared to the targeted accuracy of the voltage measurement. This rule aims to minimize the current impact on the voltage measurement by minimizing the voltage drop due to track impedance combined with current level.

The STC3117 voltage algorithm is used in low power mode as well as in mixed mode. In both configurations and in every application state (sleep, normal, fast) the voltage measurement accuracy is critical. The lower the current flowing in the Vin track, the more accurate the measurement.

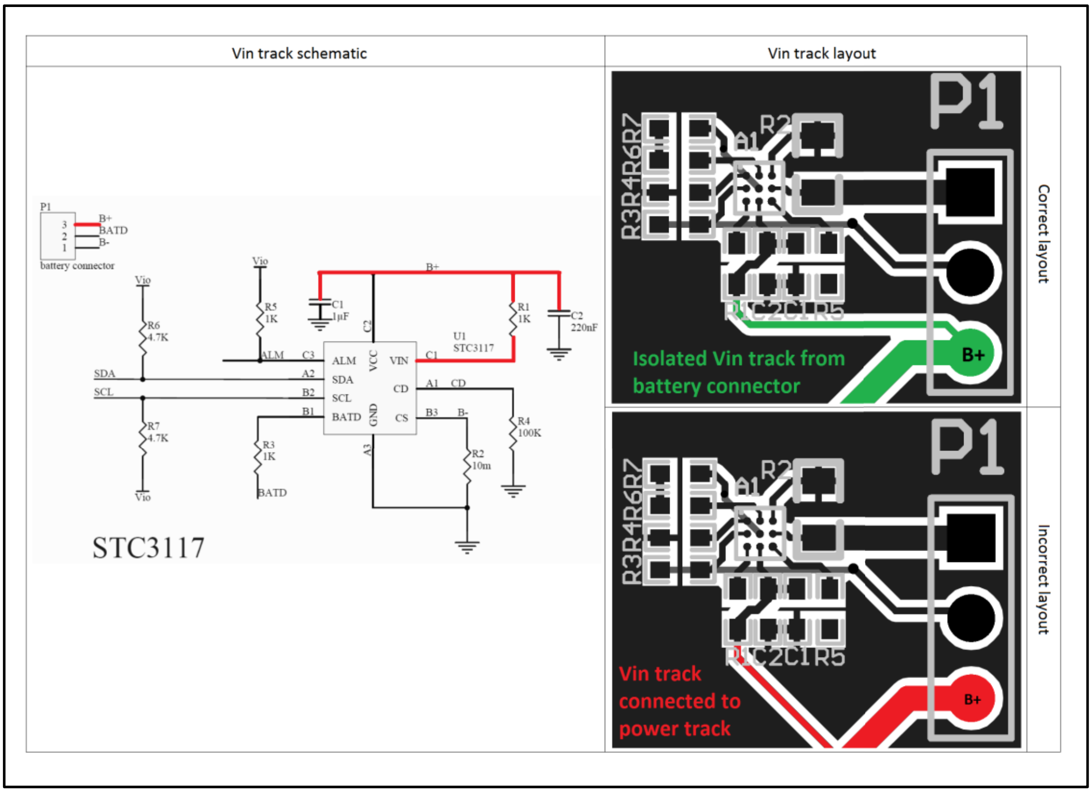
Rule description:

1. Minimize the track and connector impedances
2. Minimize the current to take into account the voltage drop calculation

This is not a mandatory rule in mixed mode. If "voltage mode only" is used to track the battery state, this rule becomes mandatory.

In the diagram below, the image on the right is the correct implementation since the single Vin track involves a very low current level in the voltage measurement track. The maximum value of this current is equal to the STC3117 consumption when the V_{cc} is also powered by this track. This involves a quasi-null voltage drop and guarantees the best voltage measurement accuracy in every application consumption condition.

Figure 11: Vin separated track: schematic and layout



5.2 External sense resistor

If the STC3117 is used in mixed mode, an external sense resistor has to be connected between the CS and the GND pin. The aim of this resistor is to sense the battery flowing current properly by connecting the STC3117 dedicated pins to the system through the sense resistor. The protocol for doing this is as follows (priority 1 indicates a mandatory rule):

1. Place the sense resistor close to the battery connector with the minimum track length to minimize the equivalent impedance. In the same logic, ensure the track wideness is in line with the application power consumption (priority 2).
2. Connect the STC3117 CS pin directly to the sense resistor without any direct connection with the battery minus pin (VBat-). The CS pin connection with the VBat-pin is performed through the sense resistor pad (see *Figure 12*) (priority 1).
3. Connect the STC3117 GND pin directly to the sense resistor with a track. The sense resistor has to be independently connected to the ground to avoid any voltage reference issue. The GND pin connection with the application ground plane is performed through the sense resistor pad (see *Figure 13*) (priority 1).
4. Connect the sense resistor to the ground plane with the minimum equivalent impedance to minimize the sense resistor effect on the application behavior (priority 1).

Figure 12: CS track schematic and layout

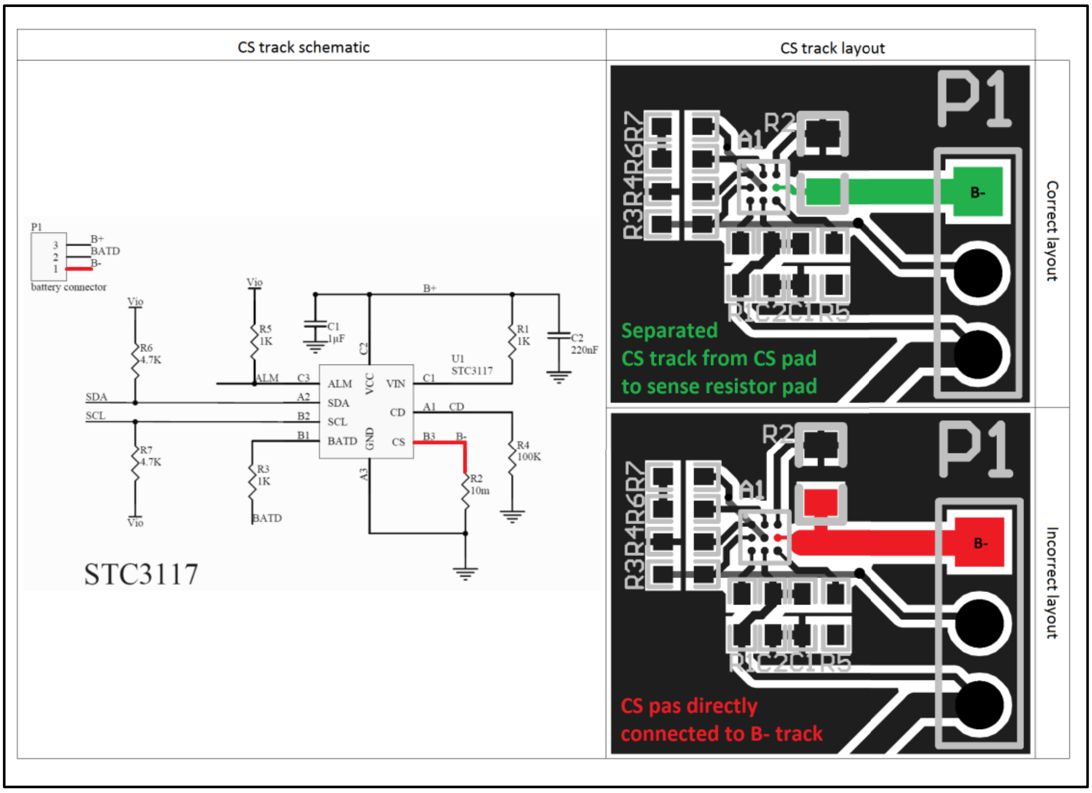
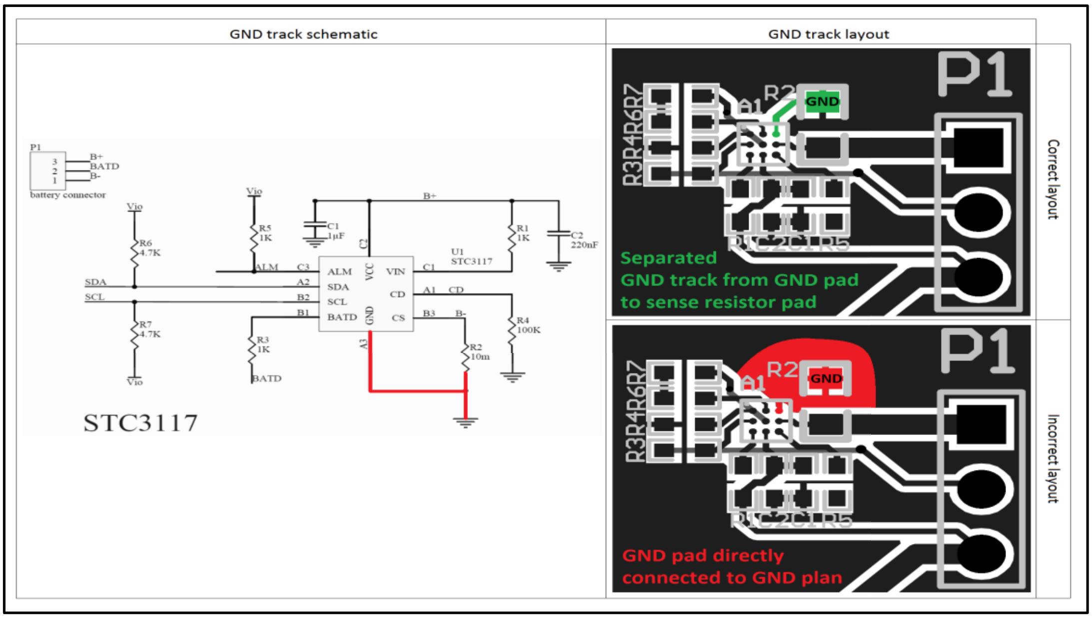


Figure 13: GND track schematic and layout



5.3 Temperature considerations

The STC3117 provides an accurate temperature sensor feature. If this internal temperature sensor is used for battery temperature compensation, the position of the STC3117 with respect to the battery position impacts slightly the accuracy of the temperature measurement. The closer the STC3117 is to the battery, the more accurate the temperature measurement and compensation is.

The other heating components of the application should be placed as far from the STC3117 as possible to avoid big temperature measurement errors. For example, during application charging mode, the IC charger temperature can be increased by several degrees.

Note: if there is a temperature measurement error, the state of charge (SOC) temperature compensation is erroneous.

5.4 STMicroelectronics PCB design recommendations

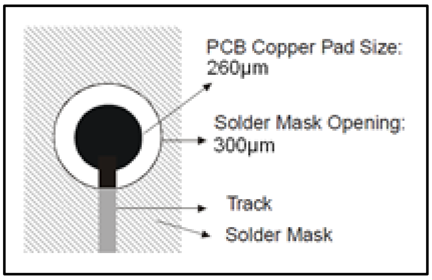
For optimum electrical performance and highly reliable solder joints, STMicroelectronics recommends the PCB design guidelines listed below.

Table 3: PCB design recommendations with "solder mask defined"

PCB pad design	Non solder mask defined, micro via under bump allowed
PCB pad size	Ø = 260 µm max (circular) – 220 µm recommended
Solder mask opening	Ø = 300 µm min (for 260 µm diameter pad)
PCB pad finishing	Cu – Ni (2-6 µm) – Au (0.2 µm max)

To optimize the natural self-centering effect of Flip-Chips on the PCB, the PCB pad positioning and size have to be properly designed (see *Figure 14*).

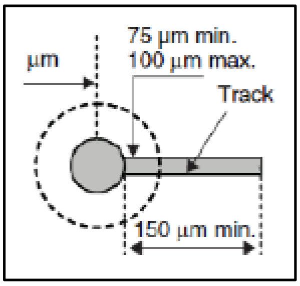
Figure 14: PCB design recommendations



A thick gold layer finishing on the PCB pad is not recommended (low joint reliability).

The PCB tracks close to the CSP footprint have to be designed to ensure a correct mechanical assembly (see *Figure 15*).

Figure 15: PCB track recommendations



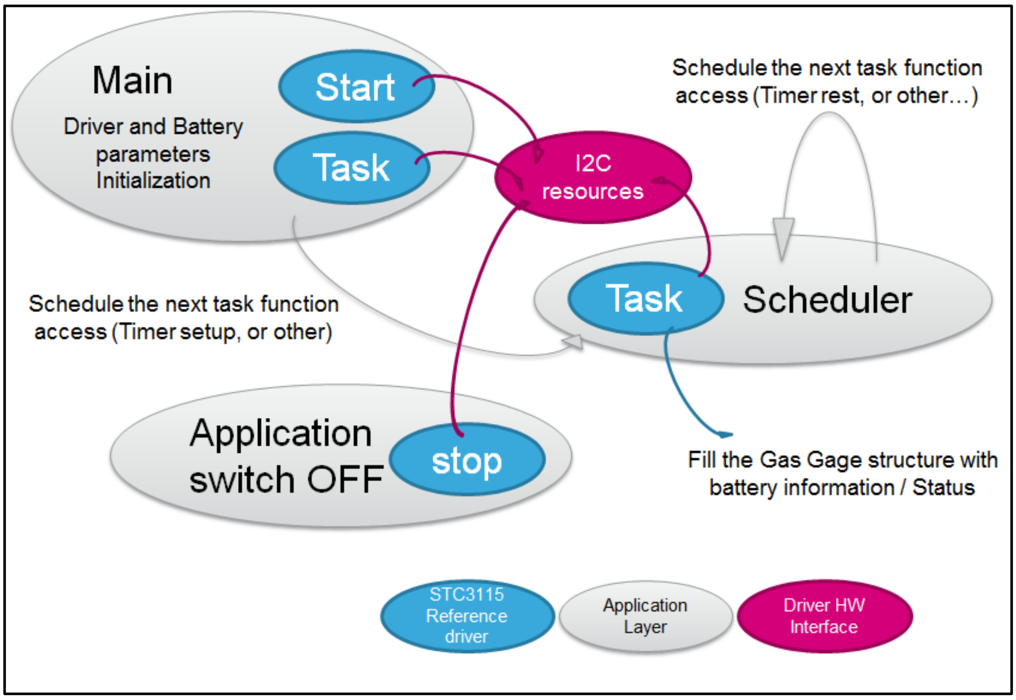
6 Software

The STC3117 hardware architecture can autonomously track the battery status. From battery plug-in to battery plug out, a simple initialization is enough to provide an accurate battery state of charge (SOC). From this initial point, the system can access when the SOC has to be refreshed and displayed to the user without any real time constraints.

An additional software layer can also be used to compensate the temperature and aging effects. This layer uses the STC3117 hardware features for accuracy and to be independent from the host system.

The software architecture can be separated into several blocks (e.g. start, task, and API) which are described below.

Figure 16: Software architecture overview



The STC3117 start block initializes and configures the STC3117. Effectively, this block updates the register reset values so that the values are in line with the battery that has to be tracked.

The STC3117 periodic task is the software block which periodically reads the STC3117 registers to update the system with the new battery status. This block can be enriched to compensate temperature and aging effects.

The STC3117 software API is the software block between the main application software and the driver itself. This block allows communication between each layer.

6.1 Chip initialization

Chip initialization is very sensitive and is the most important task of the software. This software option allows the first battery state of charge (SOC) to be reported accurately. For the best accuracy, the STC3117 registers have to be initialized from specific values based on the battery characteristics. The initialization process is different depending on the history of the application before startup.

1. The battery has not been removed. When the battery has not been removed since the last application switch off, the last recorded data can be recovered and stored in the STC3117 using the data saved in the RAM memory.
2. New battery plug-in. Full battery initialization has to be performed starting from the default software point and using the initial battery model. The initial open circuit voltage (OCV) measurement and SOC have to be used as the initial battery state. This information can be retrieved from the STC3117 registers.

6.1.1 Battery has not been removed

When the battery has not been removed, the STC3117 can be initialized with the memorized data. This is called the STC3117 restoration process. It depends on the RAM memory which is used to retrieve recorded data from the last application switch off. In addition, the battery model tuning, performed by the software during previous operations, can be saved and restored.

Before checking if the STC3117 can be restored or not, the test and CRC words to be located in the RAM memory are used to secure and check the validity of the RAM content i.e. to check if the RAM data can be used to initialize the STC3117 register or not.

The RAM test word is a fixed word but, the CRC word has to be calculated after each RAM write operation. The CRC calculation can be made using any standard algorithm.

Table 4 reflects a RAM organized proposal which is in line with the STC3117 reference driver.

Table 4: STC3117 RAM stack, 16 bytes @0x20

RAM register	Content proposal
REG_RAM0	RAM test word
REG_RAM1	Battery SOC_LOW
REG_RAM2	Battery SOC_HIGH
REG_RAM3	Battery CC_CNF_LOW
REG_RAM4	Battery CC_CNF_HIGH
REG_RAM5	Battery VM_CNF_LOW
REG_RAM6	Battery VM_CNF_HIGH
…	…
REG_RAM15	CRC

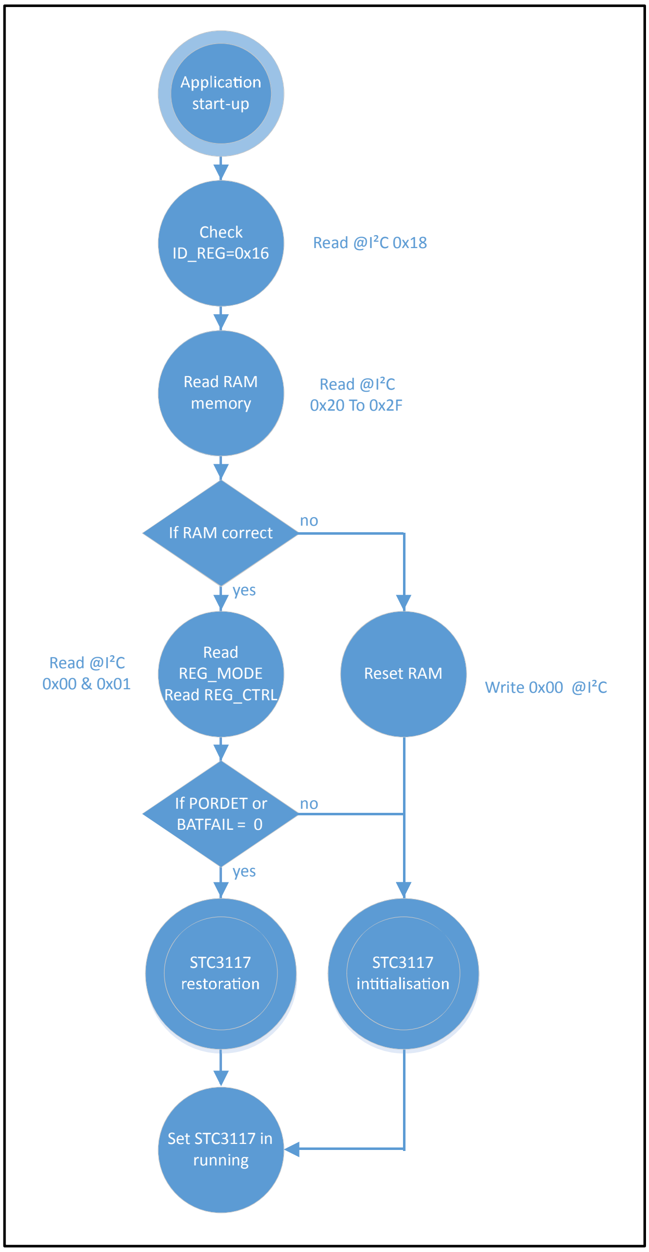
Note that 16 RAM bytes can be used to save some other information/events.

Once the validity of the RAM content has been checked, the STC3117 can then be checked to see if it can be restored or not. This is done as follows:

1. Read the chip ID (REG_ID @ 0x18 = 0x16)
2. Check the RAM memory status
3. If the RAM content is incorrect, the RAM and the chip have to be initialized with default parameters
4. If the RAM content is correct, check the PORDET and BATFAIL bits
5. If the BATFAIL and PORDET = 0, the battery has not been removed and chipset configuration can be recovered from the RAM data information (e.g. from the RAM memory content description). STC3117 restoration is possible; the saved RAM data can be used to restart the STC3117 from a previous application status.

6. If BATFAIL = 1, the battery voltage was previously decreased below 2.6 V (UVLO threshold), the BatD pin detected a battery removal/switch event, and the chip has to be fully initialized. If PORDET = 1, the battery voltage was previously decreased below 2 V (POR threshold), the battery was removed/switched, and the chip has to be fully initialized. The STC3117 cannot be restored and full STC3117 initialization is required.

Figure 17: Software initialization diagram: application startup



6.1.2 New battery plug-in

The software descriptions are accessed when no data can be retrieved from the STC3117 RAM memory. These descriptions (see below) describe the first two I²C communication commands to be executed.

1. The first block initializes the STC3117. All registers have to be initialized using default values and the first open circuit voltage (OCV) measurement result.
2. The second block restores the STC3117. Some registers have to be initialized with the recorded RAM data using the improved battery model from the last working period. Other registers have to be initialized using default values.

STC3117 initialization steps:

1. Read the OCV register. The first OCV measurement reflects the initial battery state of charge. It has to be read and saved into a variable.
2. Read the current register. The first current measurement reflects the current flowing in and out of the battery during the first OCV measurement. It has to be read and saved into a variable.
3. Set the STC3117 parameters (ensuring first that the GG_RUN bit is set to 0). The REG_OCV_TABx registers have to be filled with their values as well as the values of the REG_CC_CNF and REG_VM_CNF registers (e.g. chipset configuration parameters). Next, the application limits have to be configured by setting the REG_ALARM_SOC and REG_ALARM_VOLTAGE registers as required by the application to provide a HW interruption signal on the ALM pin when one of the measurements is detected below the defined threshold. The IODATA bit has to be written to 1 to enable the ALM pin report. The REG_CMONIT_TRES and REG_CMONT_MAX registers are used to configure the internal behavior of the STC3117. Default values are enough to provide good accuracy.
4. Compensate the OCV value with the current value:

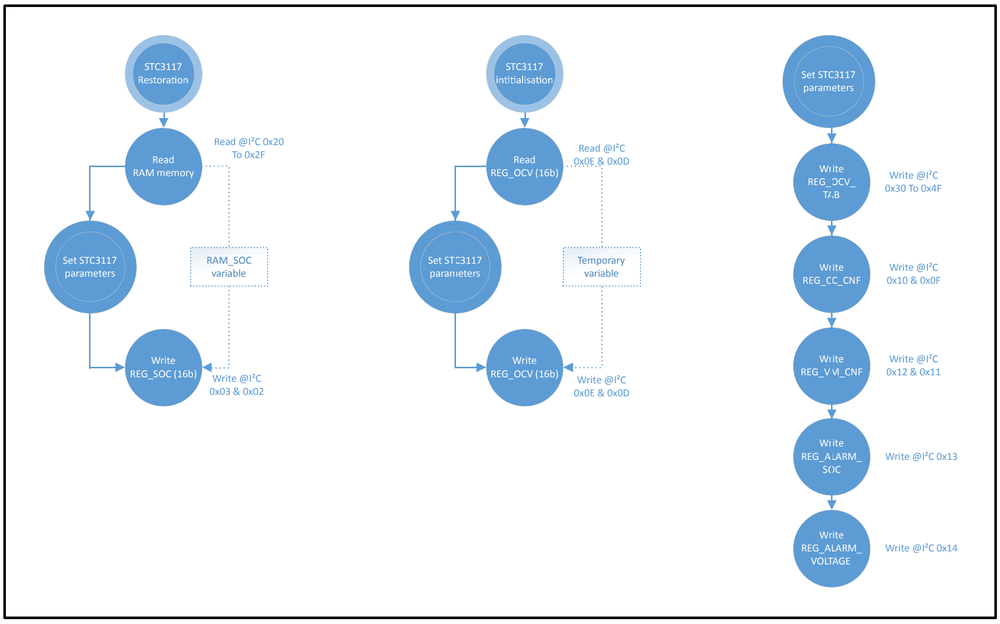
OCV (mV) = OCV (mV) – current (mA) * battery internal impedance (Ohms) **(1)**

From this point, the battery model characteristics and application threshold are initialized and battery tracking can start. To define the battery-starting point, write-back the OCV value (calculated using equation 1) into the REG_OCV register (16 bits) using the variable content. From this point, the battery model characteristics and application threshold are initialized and battery tracking can start. To define the battery starting point, write back the OCV value into the REG_OCV register (16 bits). From this operation, the first SOC of the battery is available in the REG_SOC register 100 ms later. At this time, the provided state of charge (SOC) is based on the fine-tuned OCV curve defined by the REG_OCVTABx table and the REG_SOCTABx.

STC3117 restoration steps:

1. Read the RAM memory. In this case, the last recorded SOC reflects the initial battery state of charge. It has to be read and saved as a temporary variable.
2. Set the STC3117 parameters (ensuring first that the GG_RUN bit is set to 0). The REG_OCV_TABx registers have to be filled with their values. In addition, the REG_CC_CNF and REG_VM_CNF registers have to be initialized using RAM information that can be different and more accurate regarding battery ageing. Next, the application limits have to be configured by setting the REG_ALARM_SOC and REG_ALARM_VOLTAGE registers as required by the application to provide a HW interruption signal on the ALM pin when one of the measurements is detected below the defined threshold. The IODATA bit has to be written to 1 to enable the ALM pin report. The REG_CMONIT_TRES and REG_CMONIT_MAX registers are used to configure the internal behavior of the STC3117. Default values are enough to provide good accuracy.
3. Write the REG_SOC register. From this point, the battery model characteristics and application threshold are initialized and battery tracking can start. To start battery monitoring, the GG_RUN bit has to be set, as well as the VMODE bit if power saving mode is selected. To define the battery starting point, write-back the SOC value (read previously in the RAM) into the REG_SOC register (16 bits). From this operation, the first SOC of the battery is available in the REG_SOC register without delay.

Figure 18: Software initialization diagram: restoration initialization (set parameters)



6.2 STC3117 running configuration

During initialization, the STC3117 is in standby mode. The initial state of charge (SOC) information has been evaluated but, the SOC has to be updated by the gas gauge during normal application operation. The STC3117 has to be configured in run mode to track the battery SOC. The GG_RUN bit in the REG_MODE register must be set to 1. At same time, STC3117 run mode has to be selected, by setting the VMODE bit in the REG_MODE register.

- If VMODE is set to 0, the STC3117 is in power saving mode and it uses only the voltage measurement to track the battery state. This mode is recommended for low power applications or during a low power application state.
- If VMODE is set to 1, the STC3117 is set to mixed mode using the internal OptimGage algorithm to provide the battery SOC status using voltage and current measurements simultaneously. This mode is recommended to provide the best accuracy whatever the application consumption.

Note that the IODATA bit has to be written to 1 to enable the ALM pin report using previously initialized parameters.

6.3 Periodic driver access

To refresh the battery state that is reported to the application, the STC3117 needs to be accessed. The period duration and frequency accuracy do not impact the accuracy of the report.

6.3.1 First STC3117 access after initialization

Depending on whether the STC3117 was fully initialized or restored, the state of charge (SOC) availability can be delayed:

1. After full initialization, the REG_SOC register is updated 100 ms after the end of the REG_OCV write operation.
2. After restoration, the REG_SOC register is directly available, and even the temporary variable content can be reported to the user.

Note that a simple way to manage the SOC delay is to use the STC3117 conversion counter information available in the REG_COUNTER register. This conversion counter is increased by 1 every 500 ms (after setting the GG_RUN bit to 1). To ensure availability of the first SOC, the software can wait to read a counter higher than 2 (REG_VOLTAGE, REG_CURRENT, REG_OCV registers filled).

6.4 STC3117 task procedure

The task software block reports the battery state. From a simple system status check to a full battery report, the STC3115 task software block is called periodically at any frequency to be in line with the system power consumption.

6.4.1 How often the STC3117 is called in run mode

A delay between two STC3117 accesses does not change the report accuracy. For example, when the state of charge (SOC) is moving faster, an access delay can be scaled by the application consumption to update more frequently the SOC reported to the user. This can be achieved by using the REG_CURENT value. The STC3117 access delay can also be decreased accordingly to the SOC level. The lower the SOC is, the lower the access delay is.

Alternatively, the STC3117 can be accessed with a fixed period. A fixed period has to be in line with application consumption and battery capacity. For example, the period could be equal to a minimum delay of 1 % SOC discharge: (battery capacity (mAh) / application maximum current (mA)) x 0.01 = STC3117 access period (h). By default, a delay of 30 seconds to 1 minute between two STC3117 accesses is considered adequate for most standard applications.

6.4.2 STC3117 periodical tasks

During the periodic access to the STC3117, several simple checks have to be performed to verify system stability. Consequently, the task function needs to read the STC3117 to report the information to the system.

The application checks verify if any battery swaps or removal events have occurred since checking the BATFAIL and PORDET bits. The RAM memory integrity can also be checked to ensure chipset stability and reliability between two driver accesses. From these bits and the RAM status, the driver can either report the updated battery status or it can restart and reset the data in the exchange structure.

At the end of the task procedure, the exchange data structure has to be updated to report the last battery status to the system. At this time, the RAM memory registers have also to be updated to save their status for the next driver access.

An additional feature of this procedure is to track the application state. Based on current and voltage measured values, charging, end of charging, idle discharging, and low battery status can also be reported to the system.

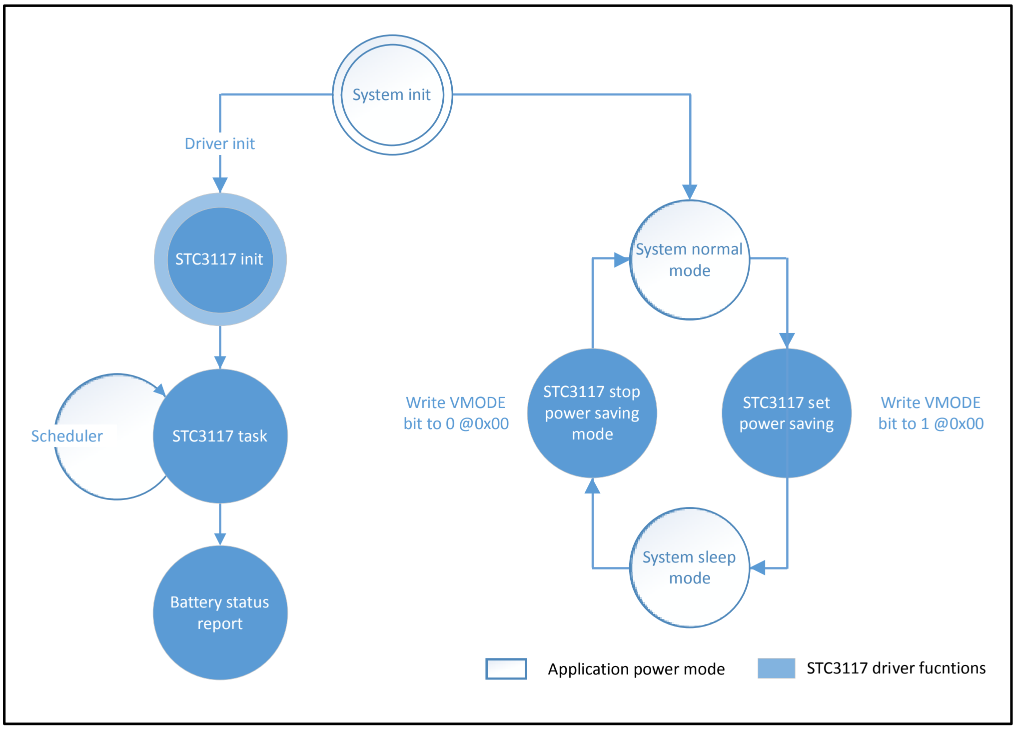
6.5 STOP procedure

To stop the device correctly if the application switches off, the GG_Run bit has to be cleared. The RAM can also be updated with the last updated battery state. This information can then be used during the START procedure if the battery has not been removed.

6.6 Power saving mode software management

The power saving mode of the STC3117 can be set dynamically during normal application running. To maintain the best accuracy from the device, the driver should be called to use power saving integrated functions (see *Figure 19*). Power saving management can be done in parallel with the normal application workflow.

Figure 19: Power-saving software diagram



The global behavior of the driver remains the same irrespective of the STC3117 power mode: it reports the battery state information to the system after each scheduled task.

From the application software point of view, the STC3117 driver power saving functions have to be called to enter power saving mode and to go back to normal mode. When the application goes to sleep mode, the "STC3117 set power saving" and the "STC3117 stop power saving" can be called according to *Figure 19*.

Even if the STC3117 power saving mode is ON by default and used as the main mode for reporting the battery state over the application life, the STC3117 mixed mode remains more accurate. Mixed mode allows the user to get the best performance from the STC3117.

To obtain the best accuracy in every application state, power saving mode should be setup during application sleep mode.

7 Revision history

Table 5: Document revision history

Date	Revision	Changes
09-May-2016	1	Initial release