User Manual

DA1468x State of Charge Functionality

UM-B-075

Abstract

The user manual describes the State of Charge concept and how its functionality is realized in the DA1468x system. It describes the list of features provided by the related driver code and how it is incorporated into the target. It also introduces a tool which is used to characterize a battery and check the performance of the State of Charge driver.

Contents

Figures

Tables

1 Terms and Definitions

SOC	State of Charge
SOCF	State of Charge Function
VBAT	Voltage of Battery
IBAT	Current of Battery
FG	Fuel Gauge
GPADC	General Purpose Analog to Digital Converter
SDK	Software Development Kit
BPT	Battery Profiler Tool
BLE	Bluetooth® Low Energy

2 References

[1] DA14680, Datasheet, Dialog Semiconductor

3 Introduction

The State of Charge (SOC) is an important parameter in modern day applications that will provide the energy level of a rechargeable battery. It is an abstract energy concept which is typically the ratio of the available capacity to the attainable capacity of a battery pack. This document first introduces the SOC concept followed by the algorithm implemented in our system. Then, it describes an overview of the State of Charge Function (SOCF) driver code included in the DA1468x SDK and how to integrate SOC in an user application. In addition, this document explains how to use the Battery Profiler Tool (BFT) which is used to characterize the battery and also check the performance metric of the SOCF driver. It is expected that anyone reading this document should have a good understanding of DA1468x system [1].

4 State of Charge Concept and Algorithm

As earlier defined, the SOC is the ratio of the available capacity to the total capacity of a battery pack. However, modelling a battery is a complex task. If you look into the spec sheet of a battery, it gives you its capacity only for a given discharge rate. If one draws a different current from a battery which in turn means a different discharge rate, then the capacity of the battery changes. In Figure 1, the battery voltage versus the capacity of the battery is plotted for different discharge rate.

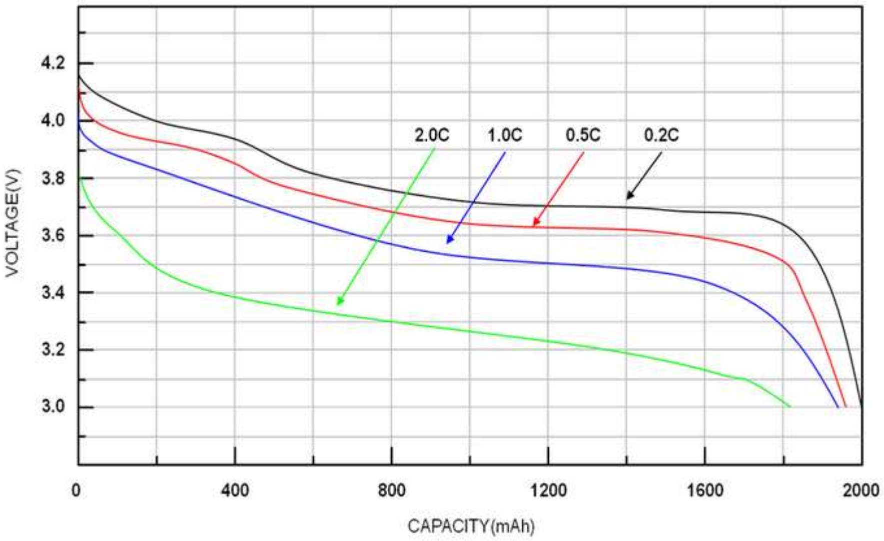


Figure 1: Battery Voltage versus Battery Capacity for Different Discharge Rates

A C-rate is a measure of the rate at which a battery is discharged relative to its maximum capacity. A 0.5C rate means that the discharge current will discharge the entire battery in 2 hour. In the graph, given that the cut off voltage of the battery is 3V, it is evident that the battery has a lower capacity for a higher discharge rate. Hence, in order to estimate the SOC, one has to estimate the current drawn from the battery.

This is realized by the built in hardware fuel gauge (FG) circuit in the DA1468x. The block diagram of the circuit is shown in Figure 2. The main function of this block is to continuously integrate a voltage which is the result of a current flowing through a sensing resistor. The sensing resistor is recommended to have an accurate value of 0.1 Ω and is placed between the SOCp and SOCn pins. For more details of the circuit, please look in the DA1468x datasheet given in [1].

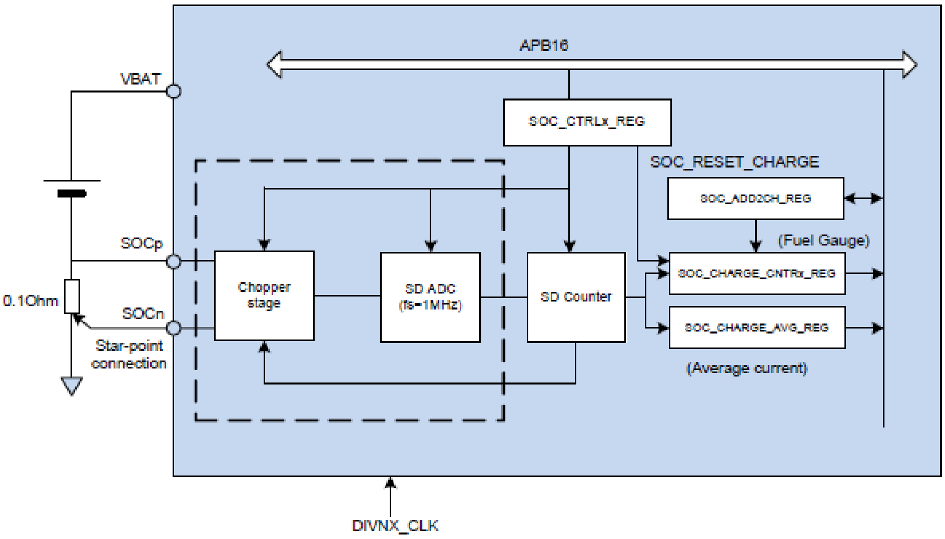


Figure 2: Fuel Gauge Circuit Block Diagram [1]

This FG circuit maintains the coulomb counting continually and the software works out the SOC from the coulomb counting result directly. Since, this circuit is off during sleep mode, it is only used to estimate the SOC during the charging phase i.e. when the VBUS is connected. The flow diagram outlining the SOC calculation procedure is shown in Figure 3.

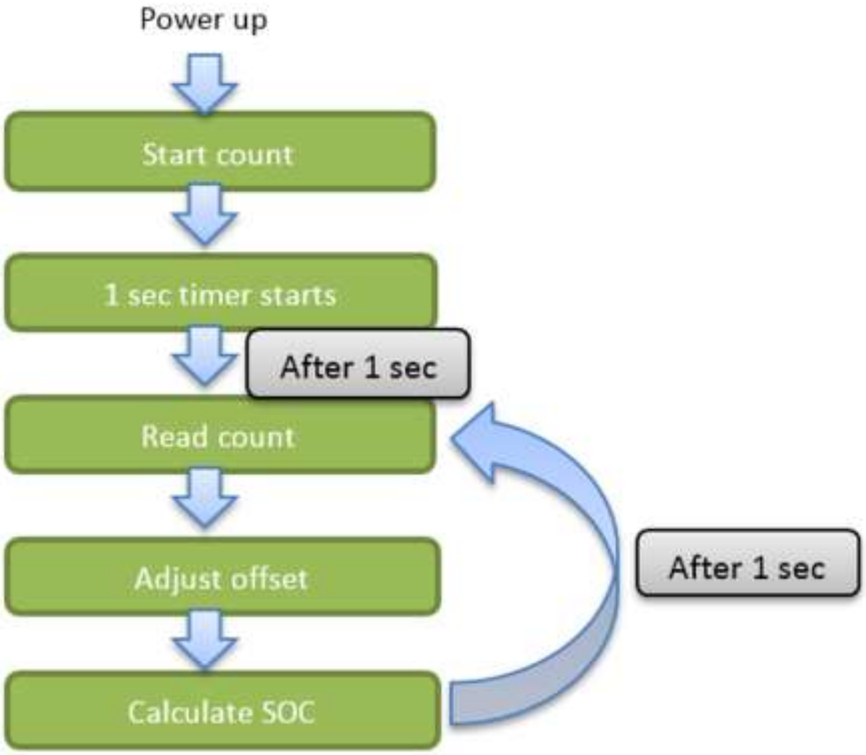


Figure 3: State Diagram of SOC Calculation Using the Fuel Gauge Circuit

To cover requests from all applications, apart from the charging state, a SOCF using VBat+IBat has been implemented. Before describing the algorithm, the relationship between the battery voltage and SOC is analyzed. This is depicted in Figure 4.

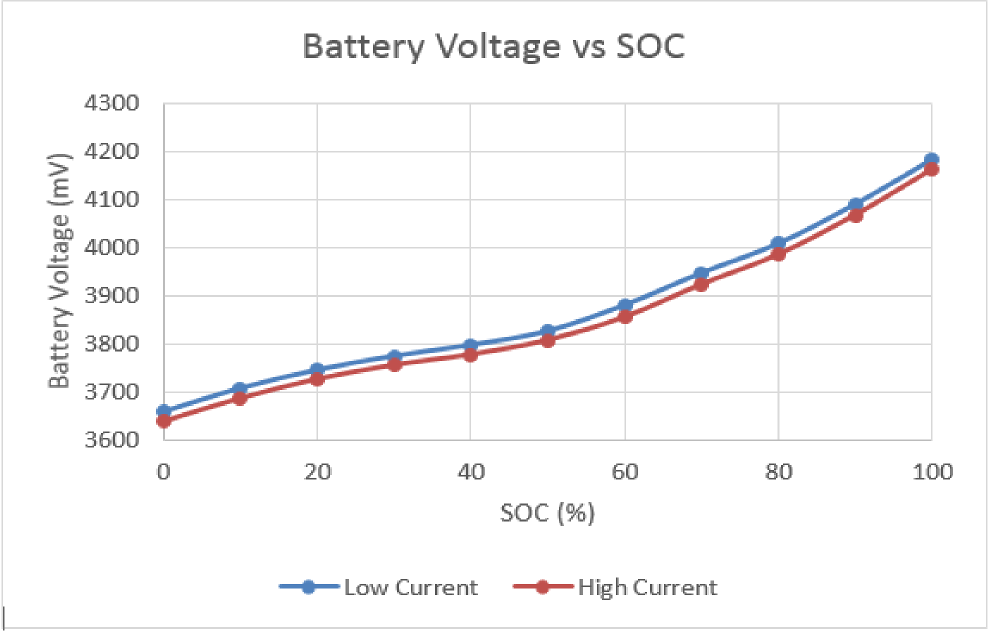


Figure 4: Battery Voltage versus SOC for a 230 mAh Sample Battery

If one measures the battery voltage, then from a look up table created while characterizing the battery which maps the defined graph, the SOC can be estimated. However, the characteristics changes depending on the current drawn from the battery. In Figure 4, the curve is different for a low current and a high current. Hence, two look up tables are created, one for a lower current and one for a higher current. The higher limit corresponds to the maximum average cuurent in the application during the active state and the lower limit is set by default to 1 mA.

Similar to the charging state, the FG circuit is used to calculate the average current every 1 second which is working only during the active state. For the battery voltage measurement, a timer is initialized. Just before preparing for sleep, it checks if over 10 seconds has elapsed after the timer was initialized. If yes, then it measures the battery voltage with the help of the on-chip General Purpose Analog to Digital Convertor (GPADC). Using the battery voltage and the latest averaged current measurements during the active period, one compares this to the look up table to get the SOC value. If the current measured is between the lower and higher limit, then the SOC is estimated using a linear interpolation. The state diagram of this method is shown in Figure 5.

This method is very low power as the fuel gauge circuit is not active all the time. However, to enable this concept, the battery has to be characterized in order to create the look up tables. This is done with a BFT which is described in detail in 6. The characterization process takes the battery parameters and the range of current limits into account to generate a look up table which can be incorporated in the application code. The BPT also has an option to do a performance test. It basically compares the SOC value from the SOCF function implemented in the application to the one accurately read through precise laboratory instruments to check the accuracy of our solution.

4.1 Known Limitations

There are a few known limitations of the above described solution. They are as follows -

- There is presently no provision explicitly for temperature and aging compensation.
- If the average current calculated in the active period is beyond the higher or lower limit, then the SOC is calculated by clipping the current value to their respective higher or lower limit.
- A short call to get the SOC value from the application periodically is not supported. It is recommended that periodicity of the call by the application should be over 30 seconds at least.

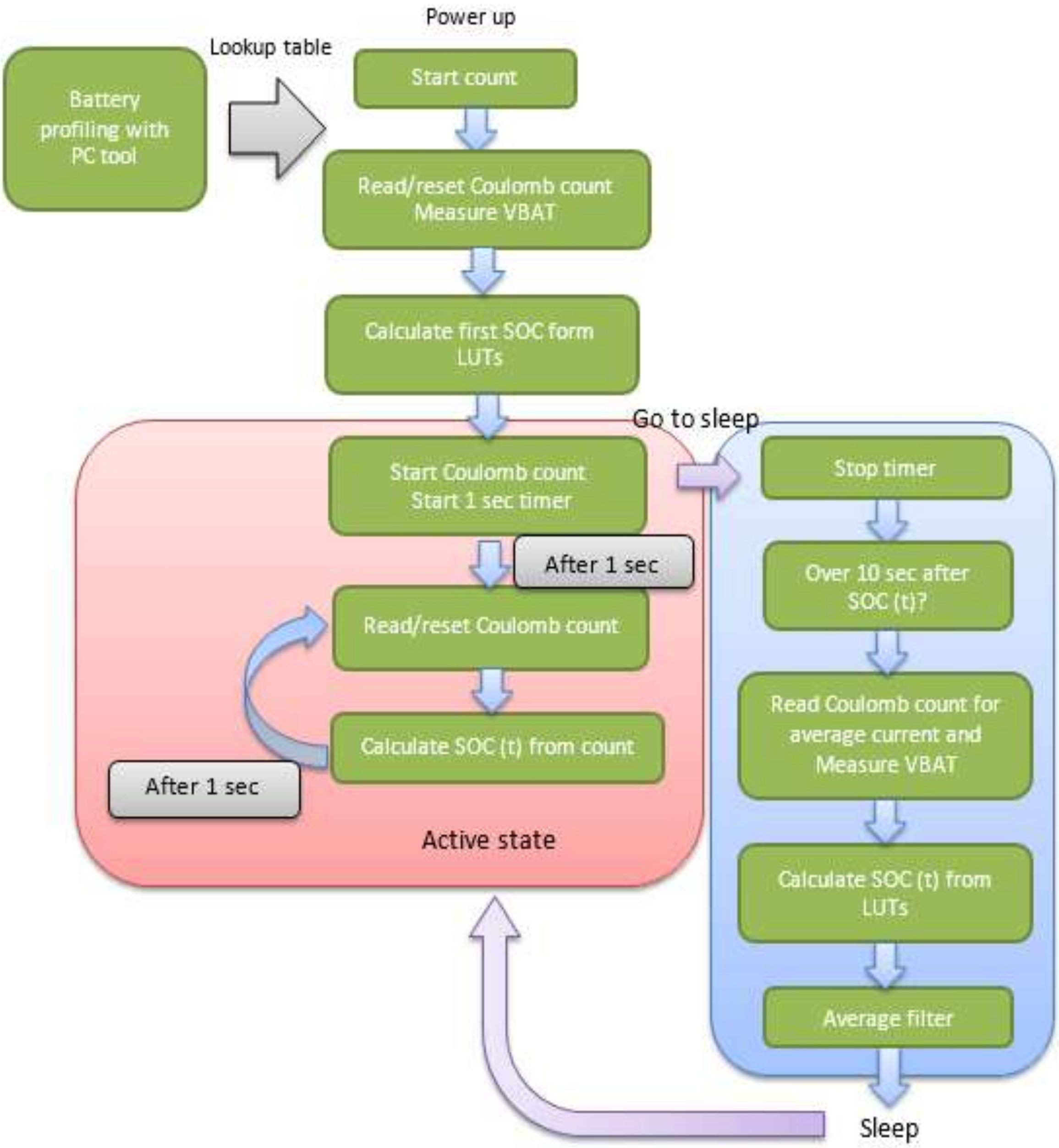


Figure 5: State Diagram of Calculating SOC using Vbat+Ibat Method

5 SOCF Driver

SOC driver is located in sdk\bsp\system\sys_man\. It consists of one source (sys_socf.c) and one header file (sys_socf.h). There is also a config file namely custom_socf_battery_profile.h in the config directory of each projects. All driver functions are in sys_socf.c. All battery dependent data including the look up tables are in custom_socf_battery_profile.h. All external function for application is defined in sys_socf.h.

5.1 List of Driver Functions

Table 1: List of Driver Functions

Name	Function
soc_chk_lut	Interpolation function
soc_measure_vbat	Measure ADC for VBAT
soc_start_count	Reset and start the built-in coulomb count
soc_compensation_count	Adjust offset of count from the built-in coulomb count
soc_get_avg_current	Get average current from count
soc_get_soc_vbat_ibat	Get SOC in VBAT+IBAT method
soc_get_count	Get count from the built-in coulomb count
soc_full_charged_notification	Capacitance compensation
soc_get_soc	External function for application
ad_socf_wake_up_ind	Operations after sleep
ad_socf_prepare_for_sleep	Operations before sleep. SOC is calculated in case of VBAT+IBAT.
soc_init_soc	Initial SOC is calculated.
soc_timer_cb	Timer callback for periodic calculation of the coulomb count.
soc_init	The built-in coulomb count starts after boot
soc_set_capacity	Set capacitance intentionally
soc_get_capacity	Get capacitance
soc_report_cb	Report timer callback for reporting SOC during performance test.
soc_task	Task for reporting SOC during performance test.
soc_report_init	Define task for reporting SOC during performance test.

5.2 Required Information in custom_socf_battery_profile.h

Table 2: Required Information in sys_socf_batt.h

Name	Description	Value
SOC_BATT_LOW_CURRENT	Low IBAT during battery profile.	Example Value 1000 (1mA)
SOC_BATT_HIGH_CURRENT	High IBAT during battery profile.	Example Value 20000 (20mA)
SOCF_BATTERY_CAPACITANCE	Battery capacitance	Example Value 120 (120mAh)
SOCF_CHARGING_CURRENT	Max charging current	Example Value 60 (60mA)

Name	Description	Value
vol_dis_low_0	LUTs for low IBAT at discharging state.	Copy from battery profile
vol_dis_high_0	LUTs for high IBAT at discharging state.	Copy from battery profile
vol_chg_0_0	voltage at 0% at charging state	Copy from battery profile
SOCF_BATT_CAPACITANCE_ADJ	Enable a compensation of the capacitance whenever charging operation is done.	0 or 1
SOCF_SECONDARY_BATT	If two type of battery is used, ID pin is existed for differentiation and battery profile is different between two batteries, the definition is needed. If defined, the following configuration for ID pin must be set. SOC_BATT_TYPE_TH : voltage threshold for differentiation SOC_BATT_ID_PORT : GPIO port for ID pin SOC_BATT_ID_PIN: GPIO number SOC_BATT_ID_INPUT : ADC pin	0 or 1

5.3 Additional modification in SDK

To enable SOCF, dg_configUSE_SOC must be defined as “1” in custom_config_qspi.h.
Already modified sources for SOCF are listed in.

Table 3: Already Modified Sources

File	Location	
sys_power_mgr.c	sdk\bsp\system\sys_man	soc_init for the start of coulomb count. soc_start_timer is added for always active mode.
Sys_charger.c	sdk\bsp\system\sys_man	soc_init_soc for initial SOC soc_full_charged_notification for a notification of full charged.

5.4 How to Enable SOCF Driver

Follow below instruction to enable SOCF driver.

1. Execute battery profile with a battery. (See 6.7)
2. Enable SOCF driver.

#dg_configUSE_SOC 1

3. To apply ADC compensation value, below feature must be enabled.

dg_configUSE_ADC_GAIN_ERROR_CORRECTION 1

4. For UART output logging the following must be defined.

#define DEBUG_SOC

#define CONFIG_RETARGET

(Not recommend during release builds, the device is trigger to wake up every sec for serial output)

5. Define battery capacitance.

SOCF_BATTERY_CAPACITANCE

6. Define low and high IBAT and the chagrin current.

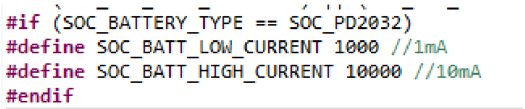
SOCF_CHARGING CURRENT

SOCF_BATT_LOW_CURRENT

SOCF_BATT_HIGH_CURRENT

This value must be matched to the input IBAT in battery profiling (See 6.7).

Example below:



```
#if (SOC_BATTERY_TYPE == SOC_PD2032)
#define SOC_BATT_LOW_CURRENT 1000 //1mA
#define SOC_BATT_HIGH_CURRENT 10000 //10mA
#endif
```

7. Request the battery profiling.

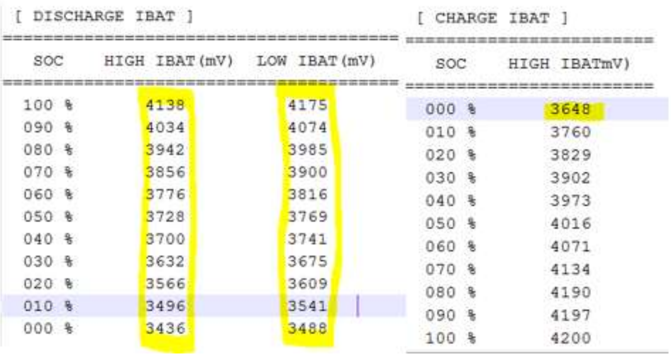
Copy LUTs to vol_dis_low_0, vol_dis_high_0 and vol_chg_0_0

After battery profiling, result.txt file would be generated (See 6.7).

Example below:

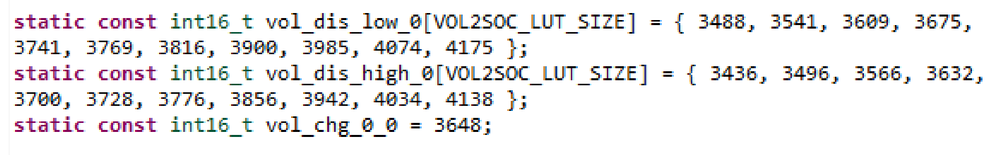
The value of vol_dis_low_0. vol_dis_high_0 and vol_chg_0_0 in result.txt must be copied to the array in custom_socf_battery_profile.h.

In result.txt



```
[ DISCHARGE IBAT ]                            [ CHARGE IBAT ]
========================================      =========================
  SOC      HIGH IBAT(mV)   LOW IBAT(mV)         SOC       HIGH IBATmV)
========================================      =========================
  100 %        4138            4175             000 %        3648
  090 %        4034            4074             010 %        3760
  080 %        3942            3985             020 %        3829
  070 %        3856            3900             030 %        3902
  060 %        3776            3816             040 %        3973
  050 %        3728            3769             050 %        4016
  040 %        3700            3741             060 %        4071
  030 %        3632            3675             070 %        4134
  020 %        3566            3609             080 %        4190
  010 %        3496            3541             090 %        4197
  000 %        3436            3488             100 %        4200
```

In custom_socf_battery_profile.h



```
static const int16_t vol_dis_low_0[VOL2SOC_LUT_SIZE] = { 3488, 3541, 3609, 3675,
3741, 3769, 3816, 3900, 3985, 4074, 4175 };
static const int16_t vol_dis_high_0[VOL2SOC_LUT_SIZE] = { 3436, 3496, 3566, 3632,
3700, 3728, 3776, 3856, 3942, 4034, 4138 };
static const int16_t vol_chg_0_0 = 3648;
```

5.5 Get SOC

The application can get SOC value through the function soc_get_soc. The range of the return value is 0 (0%) ~1000 (100%).

To report SOC from DA1468x to host, BAS (Battery Service) profile should be enabled.

Please refer to proximity sample application in SDK to know how to register BAS in application.

There is only an interface function to get SOC from SOCF driver. Function soc_get_soc will return soc. The available range is from 0(0%) to 1000(100%). So this function can be called from BAS function like below.

```
Ex) bas_set_level(svc, (soc_get_soc()+5)/10, true);
```

Please refer to the projects\dk_apps\demos\pxp_reporter\readme_soc.md for further details.

6 Battery Profiler Tool

This section describes the Battery Profiler Tool which is designed to generate a profile for a battery, incorporate information about your target platform, and hence enable this platform to run the Dialog DA1468x SOC functions with optimal accuracy for the reported state of charge of the battery used. This tool is installed together with your Smart Snippets Studio installation.

It includes descriptions about how to:

- Configure the Battery Profiler tool through the various stages of the process.
- Connect the various equipment required for each stage.
- Adjust the battery to a particular state of charge when required.

6.1 Environments

The Battery Profiler application has been verified to work in the following environment.

- Operating system: Windows 7
- .NET Framework 4.5 (Included in the installer)
- NI-488.2 with .NET framework 4.5 language support (Installed by user, See 6.3).

6.2 Required Instruments

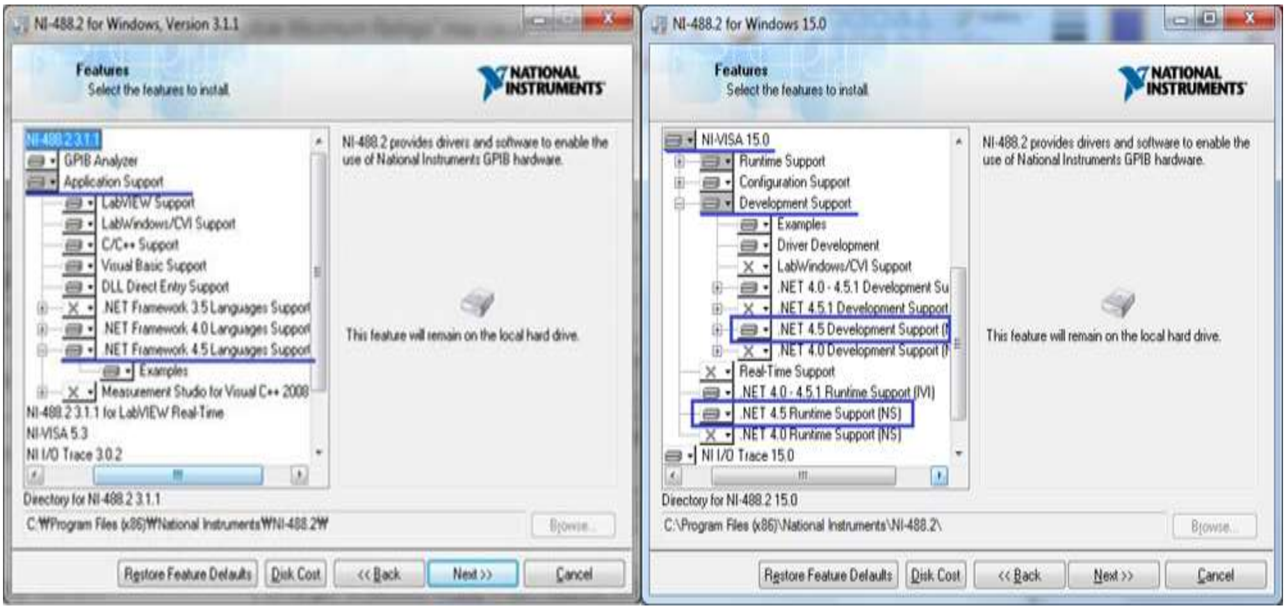
Table 4 lists the type of equipment's required, the specific items that have been tested with the Battery Profiler application, and the various interface options available on the devices. Note that only 9600 baud for RS232 is supported.

Table 4: Instruments Description for the Battery Profiler Tool

Equipment Type	Device / Specification	Interface options	
		NI GPIB-USB-HS	**Agilent 82357B GPIB/USB**
Source meter	Keithley 2400	Y	Y
Voltmeter	Agilent 34401A	Y	Y
	Tektronix DMM4040	Y	Y
Shunt Resistor	0.1 Ohm, 25mW, 01% error	-	-

6.3 Required Software

NI-488.2 software installation is required to run the Battery Profiler application on your Windows machine. Especially “.NET Framework 4.5 Language Support” should be installed. Figure 6 shows reference snapshots for version 3.1.1. and 15.0 respectively. Please be aware that the option structure is different for each version.


NI-488.2 for Windows, Version 3.1.1
Features
Select the features to install.
NATIONAL INSTRUMENTS
NI-488.2 3.1.1
GPIB Analyzer
Application Support
LabVIEW Support
LabWindows/CVI Support
C/C++ Support
Visual Basic Support
DLL Direct Entry Support
.NET Framework 3.5 Languages Support
.NET Framework 4.0 Languages Support
.NET Framework 4.5 Languages Support
Examples
Measurement Studio for Visual C++ 2008
NI-488.2 3.1.1 for LabVIEW Real-Time
NI-VISA 5.3
NI I/O Trace 3.0.2
Browse...
Restore Feature Defaults
Disk Cost
<< Back
Next >>
Cancel
NI-488.2 for Windows 15.0
Features
Select the features to install.
NATIONAL INSTRUMENTS
NI-VISA 15.0
Runtime Support
Configuration Support
Development Support
Examples
Driver Development
LabWindows/CVI Support
.NET 4.0 - 4.5.1 Development Su
.NET 4.5.1 Development Support
.NET 4.5 Development Support (I
.NET 4.0 Development Support (I
Real-Time Support
.NET 4.0 - 4.5.1 Runtime Support (IVI)
.NET 4.5 Runtime Support (NS)
.NET 4.0 Runtime Support (NS)
NI I/O Trace 15.0
Browse...
Restore Feature Defaults
Disk Cost
<< Back
Next >>
Cancel

NI-488.2 provides drivers and software to enable the use of National Instruments GPIB hardware.

This feature will remain on the local hard drive.

Directory for NI-488.2 3.1.1

C:₩Program Files (x86)₩National Instruments₩NI-488.2₩

NI-488.2 provides drivers and software to enable the use of National Instruments GPIB hardware.

This feature will remain on the local hard drive.

Directory for NI-488.2 15.0

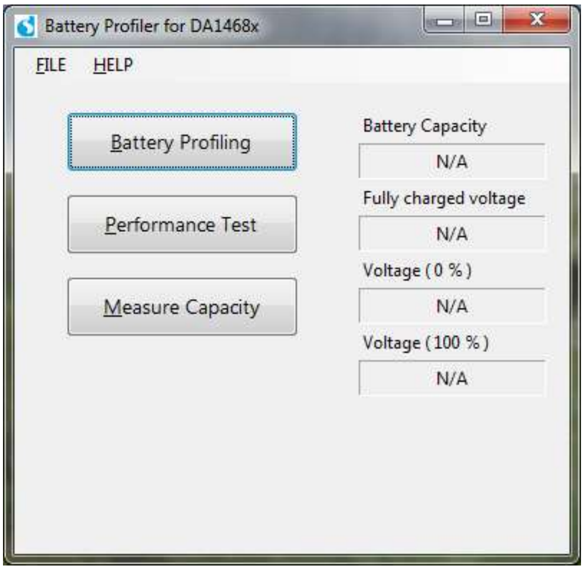
C:\Program Files (x86)\National Instruments\NI-488.2\

Figure 6: Reference Snapshot for NI-488.2 Version 3.1.1 (Left) and 15.0 (Right)

6.4 Main Screen

Figure 7 depicts how the main screen of the tool looks like. It has three options, which are as follows:

- Battery Profiling - This generates the look up table which characterizes the battery under consideration. It may take about a few hours or days to generate a battery profile. It depends on battery capacity, charging/discharging current and relaxation time.
- Performance Test - This measures the performance of SOCF driver at target system.
- Measure Capacity - This measures a battery capacity during specified voltage.



Battery Capacity

N/A

Fully charged voltage

N/A

Voltage (0 %)

N/A

Voltage (100 %)

N/A

Figure 7: Main Screen Window of the Battery Profiler Tool

6.4.1 File → Set working Directory

A work directory can be set through selecting FILE → set working directory. While the Battery Profiler is running, all files will be stored in the chosen directory. If you select the Battery Profiling or Performance test button without the setting working directory, then “C:\Dialog Semiconductor\BatteryProfile_YYYYMMDD” directory will be created as the working directory.

6.5 Test Sequence Diagram

Figure 8 shows the process of data generation and data flow in case of the target system. The measure capacity is an independent function as shown in the figure.

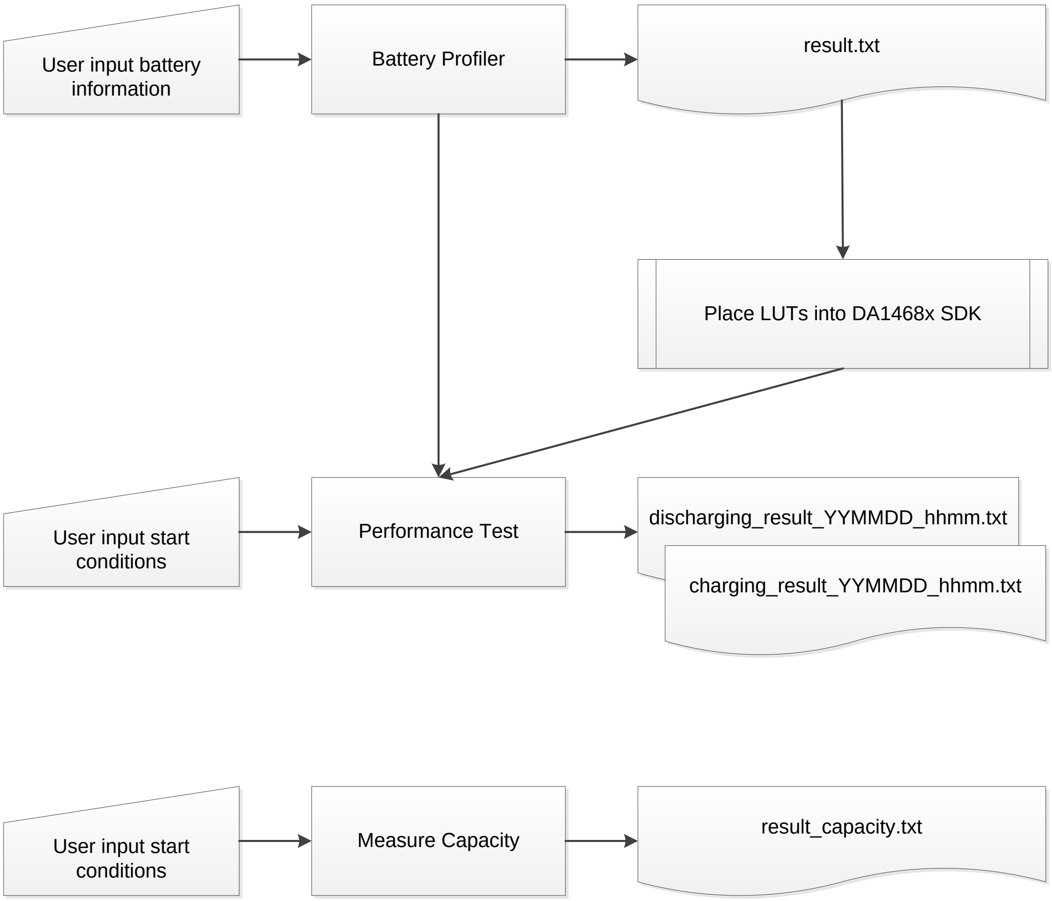

User input battery information
Battery Profiler
result.txt
Place LUTs into DA1468x SDK
User input start conditions
Performance Test
discharging_result_YYMMDD_hhmm.txt
charging_result_YYMMDD_hhmm.txt
User input start conditions
Measure Capacity
result_capacity.txt

Figure 8: Test Sequence for Different Possibilities with Battery Profiler Tool

6.5.1 Sequence for Profiling the Battery and Testing the Target

6.5.1.1 Battery profiling

This stage analyses the characteristics of the battery and generates a parameter that involve the target device.

First, the voltage of the battery is measured through both discharge and charge cycles under different conditions which reflect the expected range of currents that the target device will be able to apply.

Therefore, it is necessary to have data regarding:

- The battery voltage and usable capacity (subject to target start-up requirements).
- The target system peak and minimum demands.
- The target charger supply and charging configuration.

This stage takes the most time of all in the process - in the order of days - but is entirely automated. The length of this stage will depend on both the capacity of the battery (higher is slower), and the expected load of the target (higher is quicker).

6.5.1.2 Performance tests

To check accuracy of SOCF driver on the target system, a performance test will be useful.

Whenever a performance test is done, logs will be generated with names such as:

- charging_result_YYYYMMDD_HHMM.txt
- discharging_result_YYYYMMDD_HHMM.txt

6.6 Caution for HW Setup

There are a number of areas that can introduce resistance and hence reduce the performance of the DA1468x fuel gauge. Please regard the following points:

- The line between the battery and the target, or shut resistor, must be shorter than 10cm.
- Banana connectors must be used. Crocodile connectors must be avoided.
- The contact points at the target system and battery must be soldered solidly.

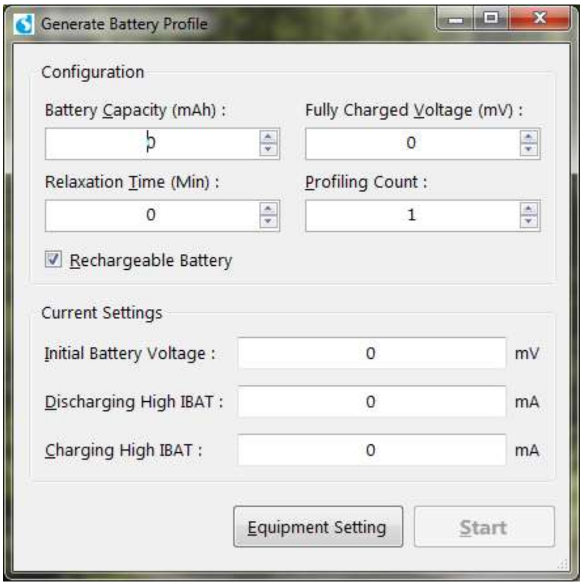
6.7 Battery Profiling

This section describes in detail how we create a profile for the battery using standard test equipment. The shunt resistor which you will use for measurement of current should be as low as possible because it could impact the accuracy while battery profiling. Accuracy of the resistor must be high, at least 0.1%.

First some information about the battery and the performance of the target platform is required, and then the identity of the equipment to be used. Once the stage is started, it works automatically.

6.7.1 Configuration

Figure 9 depicts the option that pops up once you have clicked the battery profiling option. The meaning of each of those options are described below.



Generate Battery Profile

Configuration

Battery Capacity (mAh) : 0

Fully Charged Voltage (mV) : 0

Relaxation Time (Min) : 0

Profiling Count : 1

☑ Rechargeable Battery

Current Settings

Initial Battery Voltage : 0 mV

Discharging High IBAT : 0 mA

Charging High IBAT : 0 mA

Figure 9: Configuration setup for the Battery Profiling option

6.7.1.1 Battery Capacity (mAh)

This is the capacity of the battery which will be used by the target system. Normally the specification of the battery provides a total capacitance figure, but it has a tolerance (normally +/- 5 to10 %) and the usable capacity is related to the maximum discharging current of the target system due to the voltage drop. If battery capacitance is unknown, the capacitance can be measured using the Measuring Capacity option in the tool which is describe in 6.9.

The minimum voltage the target requires to operate must be sustained at the maximum current draw. So, it is necessary to select the capacity of the fully charged battery which can be used without a risk of reaching the minimum voltage requirement, even under full load. If customer has not certain information for the capacitance, minimal capacitance in the specification is preferred.

- Unit : mAh
- Input Range : 0 ~ 10000 mAh
- Increment : 100

6.7.1.2 Fully Charged Voltage (mV)

This is the voltage level of the battery which is specified in the battery specification and when the battery is charged fully. This value would be used to avoid over charging during test.

- Unit : mV
- Input Range : 0 mV ~ 4500 mV
- Increment : 50 mV

6.7.1.3 Rechargeable Battery

This checkbox is for rechargeable battery. When profiling has started, battery is run under discharging at first and then again charging. But, in case of non-rechargeable battery, it isn’t able to charge. So, you must uncheck the checkbox when you do not use rechargeable battery.

6.7.1.4 Relaxation Time (Min)

Select a relaxation time to apply, after adjustment of the battery SOC, before measuring the battery voltage during profiling.

- Unit : Minutes
- Range: 0 ~ 30 Minutes
- Increment : 1 Minute
- Recommended : 20 Minutes

6.7.1.5 Profiling Count

Input counts of how many times you want to generate battery profile. This item will be available when the rechargeable battery item is checked. During profiling test, the result could be slightly different due to temperature, an error of the equipment and battery state. So an averaged profile from a few generations would be more reliable. It is highly recommended to do it at least 2 times in order to check that the two results are consistent with no major differences.

- Input Range: 1 ~ 5

6.7.1.6 Initial Battery Voltage

Input a voltage corresponding to 100% SOC from the target system’s point of view. It should be a value within the usable range of battery. The usable range can be found in the specification of a battery. This checkbox is only available for rechargeable battery. Recommended value is (Full Charged Voltage - 15mV).

- Unit : mV

6.7.1.7 Discharging High Ibat

The high value should be the maximum average discharging current during the active period that the target system can apply to discharge the battery.

- Unit : mA
- Range : 0 mA ~ 500 mA

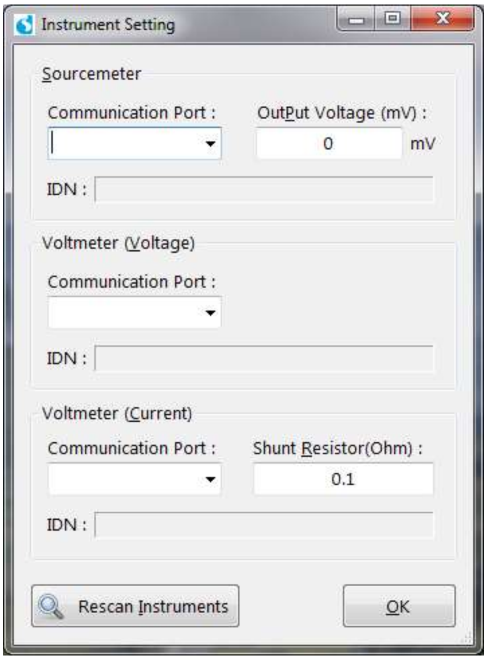
6.7.1.8 Charging High Ibat

Normally this value depends on the ability of the charger IC and the battery. The value should be defined as the charging current of the charger IC in the target.

- Unit : mA
- Range : 0 mA ~ 500 mA

6.7.2 Instrument Setting

Figure 10 shows the instrument setting possibilities while using the Battery Profiling option. Each setting possibility is described in detail below.



Instrument Setting

Sourcemeter

Communication Port :

OutPut Voltage (mV) : 0 mV

IDN :

Voltmeter (Voltage)

Communication Port :

IDN :

Voltmeter (Current)

Communication Port :

Shunt Resistor(Ohm) : 0.1

IDN :

Figure 10: Instrument Setting for the Battery Profiling Option

6.7.2.1 Interface Type and Communication Port

Each device can be communicated with using different methods, so the “Interface Type” option can be used to select the method chosen.

For each device using an RS232 serial interface, the appropriate ‘COM’ port will need to be chosen.

For each device using the GPIB interface, the “Communication Port” list box will contain a list of device addresses. The correct address is usually shown by the device when it is powered on.

When the port is selected, the IDN information for the device will be shown below. This can be used to verify the selection

6.7.2.2 Output Voltage (mV)

A charging voltage for power supply should be input. This should reflect the voltage charging IC delivers in the target device.

6.7.2.3 Shunt Resistor (Ω)

A shunt resistor is used to measure the current. If an ammeter instead of the shunt resistor is used, input value zero would make it possible. But the internal resistance of the ammeter must be around 100mΩ.

6.7.2.4 Scan Instrument

This is used to scan all instruments again. If the port for an instrument is still not shown, the power, connection and status of the instrument should be checked.

6.7.3 Hardware Connection

Figure 11 shows how the hardware should be setup while using the Battery Profiling option. The Voltmeters are placed across the Battery and the sense resistor to measure the battery voltage and the current drawn from the battery respectively. The source meter typically is used to dictate the current in the setup during the charging and discharging phase.

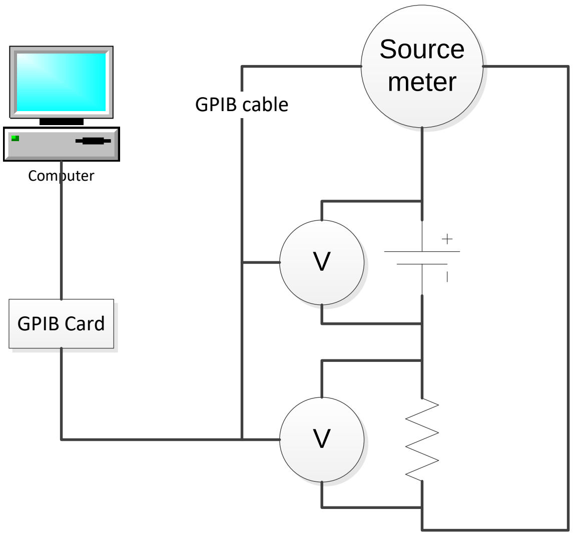
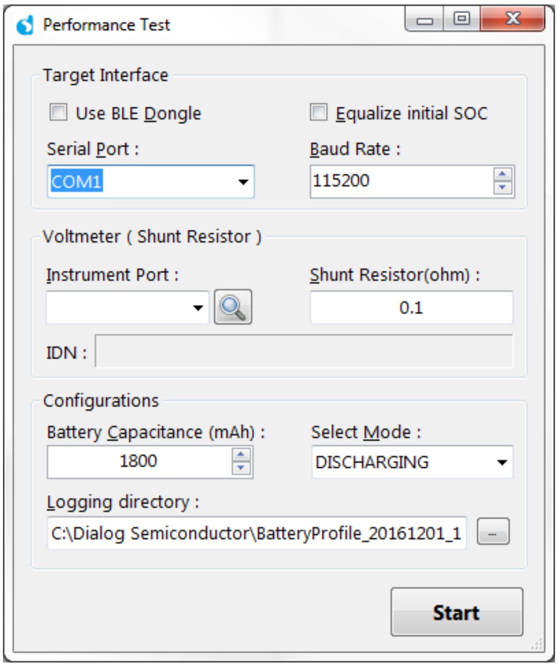


Figure 11: Hardware setup for Battery Profiling Option

6.8 Performance Test

This stage provides the opportunity to describe in detail the Performance Test and establish the performance of the software on the target system. The actual state of charge of the battery is calculated with measurements of the current flow, and drawn into a chart along with the state of charge as it is reported by the software running on the target. Once the whole process is completed, this stage can be repeated to show the final performance of SOC functions; the error rate against the true condition of the battery. Figure shows the test set-up screen of the Performance Test Option. These option are described in detail in the subsequent sections.



Performance Test

Target Interface

□ Use BLE Dongle □ Equalize initial SOC

Serial Port : COM1

Baud Rate : 115200

Voltmeter (Shunt Resistor)

Instrument Port :

Shunt Resistor(ohm) : 0.1

IDN :

Configurations

Battery Capacitance (mAh) : 1800

Select Mode : DISCHARGING

Logging directory : C:\Dialog Semiconductor\BatteryProfile_20161201_1

Figure 12: Test Setup for Performance Test Option

6.8.1 Selection Items

6.8.1.1 Target Interface

Select the port and baud rate for communication between PC and target device. “Use BLE Dongle” option enable to get SOC value from BLE Dongle. Please refer Annex A section for a complete description. "Equalize initial SOC" option takes the SOC value from the target and tool calibrates itself to start the performance test from that point. This offers flexibility as the battery does not have to be completely charged or discharged for the test. However the use of this feature is not recommended if one would want to test the complete charge or discharge cycle.

6.8.1.2 Voltmeter (Shunt Resistor)

- Port for communication between instrument and personal computer.
- A shunt resistor value for current measurement.

6.8.1.3 Configurations

- Battery Capacitance (mAh) : Usable battery capacity for target system (see 6.9)
- Select Mode: charging or discharging
- Logging directory: A directory for a log file to be stored during performance test. It is filled in with a default.

6.8.2 Experiment Sequence

1. Adjust voltage of battery for initial SOC (see 6.10 for further details) or use equalize SOC option.
2. Connect all equipment and target system, which stays at power down.
3. Select all items in the set-up menu.
4. Confirm all connections are correct.
5. Insert battery to battery holder.
6. Switch on target system.
7. Press ‘Start’ button.
8. Confirm the target SOC and gauge SOC is shown.

9. Completed' pop-up will be shown if the test is done.
10. After test is finished, a log and a chart will be generated automatically.

6.8.3 Hardware Connection

Figure 13 shows the how the required hardware needs to be connected for using the Performance Test option.

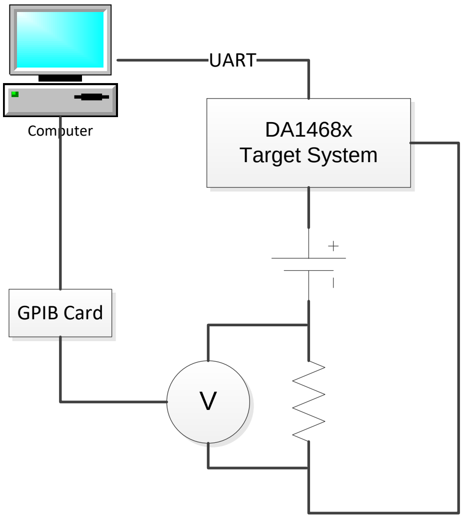


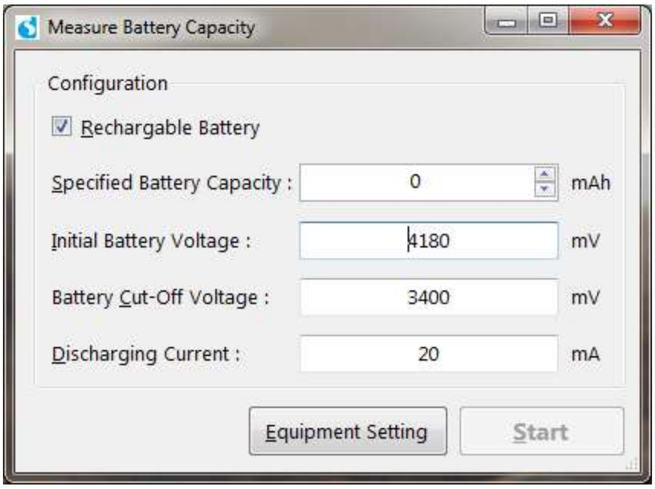
Figure 13: Hardware Setup for Performance Test Option

6.9 Measure Capacity

In this stage, you can measure capacity of a battery on specified voltage range. Capacity is calculated with measurements of current flows while discharging battery from the initial battery voltage to the battery cut-off voltage. Adjustment of battery voltage will be required if the initial battery voltage which you input is different with measured voltage. It will be run automatically. If you adjust battery voltage manually, you may have advantage that measurement time is reduced.

6.9.1 Configuration

Figure 14 shows the Configuration setup for the Measure Capacity option. The various possibilities are explained in the subsequent sections.



☑ Rechargable Battery

Specified Battery Capacity : 0 mAh

Initial Battery Voltage : 4180 mV

Battery Cut-Off Voltage : 3400 mV

Discharging Current : 20 mA

Figure 14: Configuration for the Measure Capacity Option

6.9.1.1 Rechargeable Battery

This checkbox is for rechargeable battery. When measurement has started, battery is run under discharging. In case of non-rechargeable battery, it isn’t able to charge. So, you must check the checkbox when you use rechargeable battery. If you clear the checkbox, then the initial battery voltage item will be greyed out and will not be available.

6.9.1.2 Specified Battery Capacity

This is the capacity of the battery which is described on battery specification. There is no issue even though the capacity is not accurate. It is just for reference.

- Unit : mAh
- Input Range : 0 ~ 10000 mAh
- Increment : 100

6.9.1.3 Initial Battery Voltage

Input a voltage corresponding to 100% SOC from the target system’s point of view. It should be a value within the usable range of battery. The usable range can be found in a specification of battery. Recommended value is (Full Charged Voltage - 15mV).

- Unit: mV

6.9.1.4 Battery Cut-Off Voltage

Input a voltage to stop capacity measurement. It should be a value within the usable range of battery. The usable range can be found in a specification of battery.

- Unit: mV

6.9.1.5 Discharge Current

The discharging current should be the maximum average load during the active state. The available values are described in a specification of battery and the value would be the same to the value in section 6.7.1.7.

- Unit : mA
- Input Range : 0 mA ~ 500 mA

6.9.1.6 Equipment Setting

It is the same as what is described in Section 6.7.2.

6.9.2 Hardware Connection

The H/W connection is the same as described in Section 6.7.3.

6.10 Adjust Voltage of Battery

For performance test, the voltage of battery must be adjusted properly. For example, if discharging mode is to be tested, the battery voltage must be adjusted to Open Circuit Voltage of SOC 100%. The Open Circuit Voltage for SOC of 100% is that same as the "Initial Battery Voltage". This can be achieved by the methods described below for the Discharging and Charging tests.

6.10.1 For Discharging Test

The battery can be connected to a DC power supply directly, with the settings of the supply like this.

- Voltage: 4186mV (4185 (Open Circuit Voltage of SOC 100%) + 1)
- Current limit: 50mA (typical charging current of battery)

Normally the current should be under 1mA when battery reaches the voltage. Before performance test, the voltage of the battery must be measured with voltage meter and checked if the value is proper for test because an error of power supply could make the adjustment incorrect. If the voltage is higher or lower than 4185mV, the voltage setting in power supply has to be adjusted according to the difference and retry should be done.'

6.10.2 For Charging Test

In the battery profile data, the Open Circuit Voltage of 0% SOC is not available. But the voltage with low IBAT should be used. The settings of the DC power supply are

- Voltage: 3649 mV (3650(the voltage with low IBAT at SOC 0%) - 1)
- Current limit: 50mA (typical charging current of battery)

The current should be under 1mA when battery reaches the voltage. Again before performance test, the voltage of the battery must be measured with voltage meter and checked if the value is proper for test. If the voltage is higher or lower than 3649mV, the voltage setting in power supply has to be adjusted according to the difference and retry should be done.

Appendix A "Use USB Dongle" Option

A.1 Introduction

This option enable to get SOC value from BLE dongle (DA1458x evaluation kit) via air. USB Dongle connected to PC communicates with target device using BSS (Battery Service Specification) profile of BLE.

A.2 Experiment Sequence

1. Adjust voltage of battery for initial SOC. (see 6.10 Adjust voltage of battery)
2. Connect all equipment and target system, which stays at power down.
3. Select all items in the set-up menu.
4. Confirm all connections are correct.
5. Insert battery to battery holder.
6. Switch on target system.
7. Press ‘Start’ button.
8. Select com port number connected to USB Dongle.
9. Press ‘SCAN’ button and wait 10~15 seconds when it shows device list.
10. Select target device name to be connected.
11. Confirm the target SOC and gauge SOC is shown.
12. For discharge test, it is necessary to start a tool to stop target from going to sleep, which would cease communications to test application, and to represent user activity. See next section.
13. ‘Completed’ pop-up will be shown if the test is done.
14. After test is finished, a log and a chart will be generated automatically.

A.3 Hardware Connection

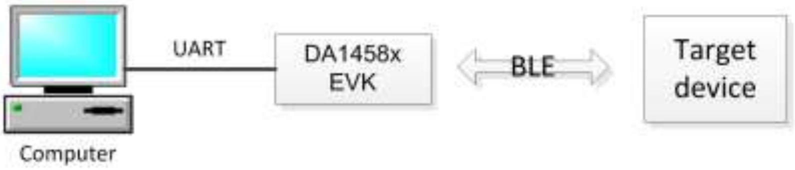


Figure 15: Hardware Setup

Revision History

Revision	Date	Description
1.0	18-Oct-2016	Initial version.
1.2	02-Dec-2016	Updated with Battery Profiler Tool
1.3	19-Jan-2022	Updated logo, disclaimer, copyright.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.