BLE Battery Level

1.0

Features

- Custom profile in Server role operation
- Battery Service usage
- Measures battery level voltage and converts it to percent
- Simulates battery charging
- Low Power mode
- LED status indication

General Description

This project demonstrates measurements of the battery voltage using PSoC 4 BLE/PRoC BLE's internal ADC and notifies the BLE central device of any change in the battery voltage.

Development Kit Configuration

1. Connect J2 pin P3[0] to J3 pin VREF.
2. Build the project and program the hex file into the CY8CKIT-042 BLE Pioneer Kit.

Project Configuration

The example project consists of the following components: an ADC_SAR_Seq, BLE, digital output pin, digital input pin, analog input pin. The ADC_SAR_Seq is used to measure the battery voltage. The output pins are used to reflect the line signal output on the LED. The input pin is configured to the resistive pull up mode and used to wake device from low power hibernate mode. The top design schematic is shown in **Figure 1**.

BLE Battery Level Example project

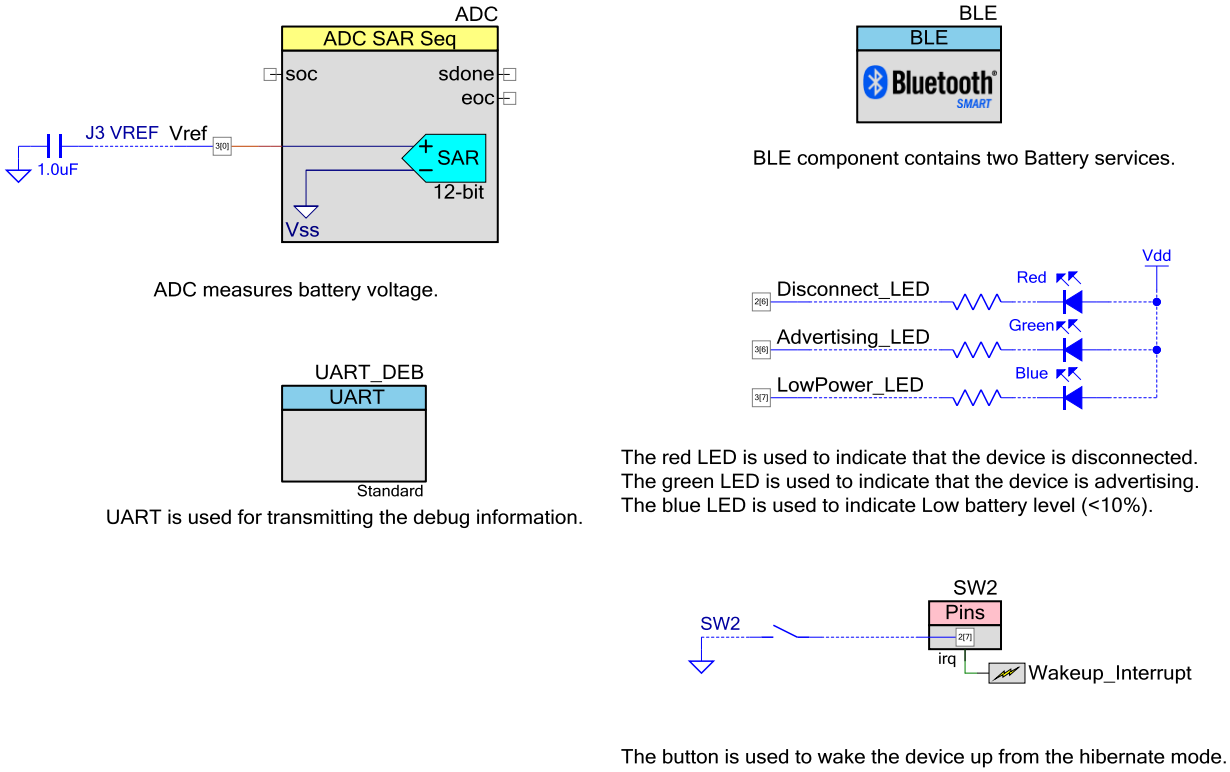

Figure 1. Top design schematic

The Vref analog input pin is locked to P3[0]. The ADC is configured for a single ended measurement with a sample rate of 1.60 kbps.

The BLE component is configured as Custom Profile in Peripheral Role. Device Information Service added with the Manufacture Name String characteristic. There are two Battery Services. The first one (Battery Service Measure) sends a measured battery level, the second is for software simulation of the battery level (Battery Service Simulate). Simulated battery level value is changed from 2 to 20 percent.

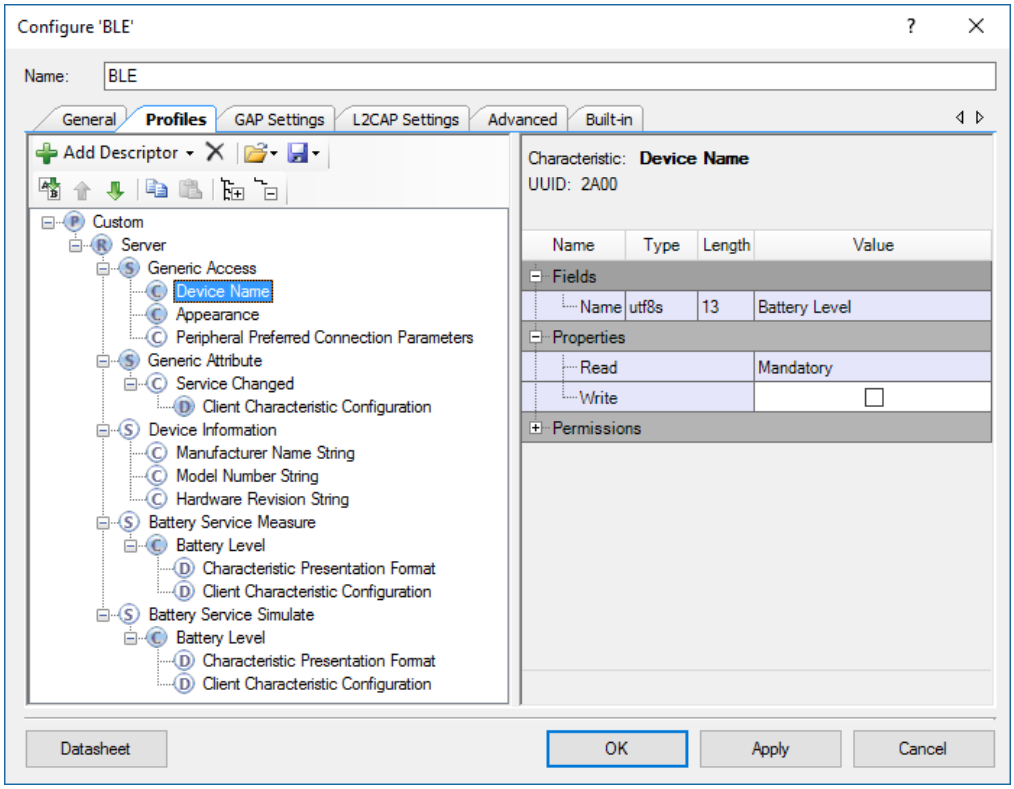

Figure 2. GATT settings

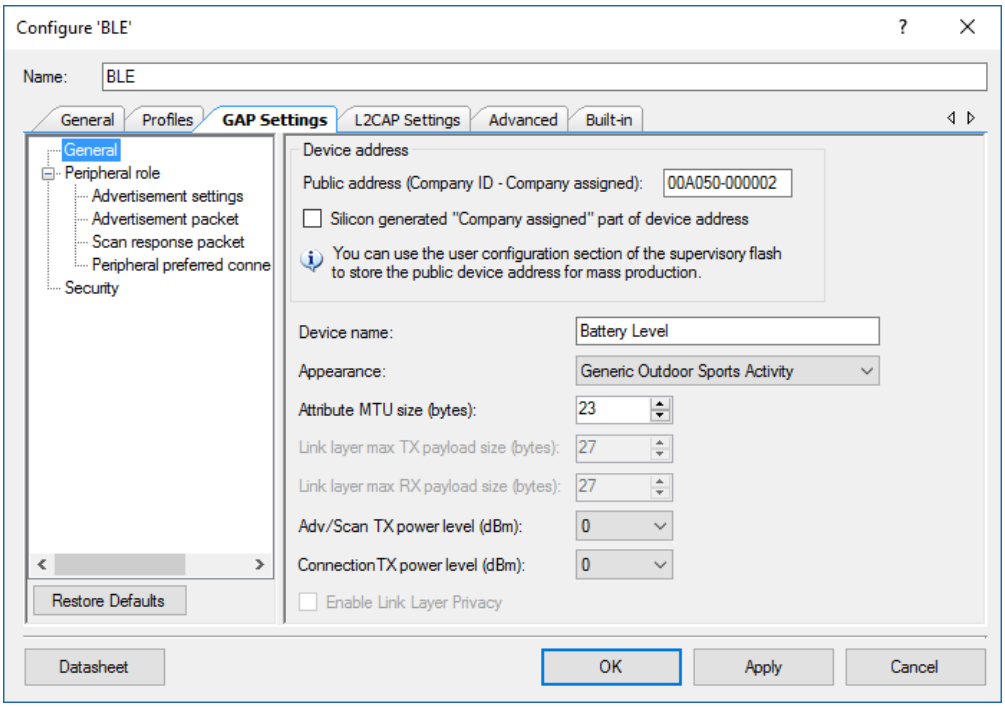

Figure 3. GAP settings

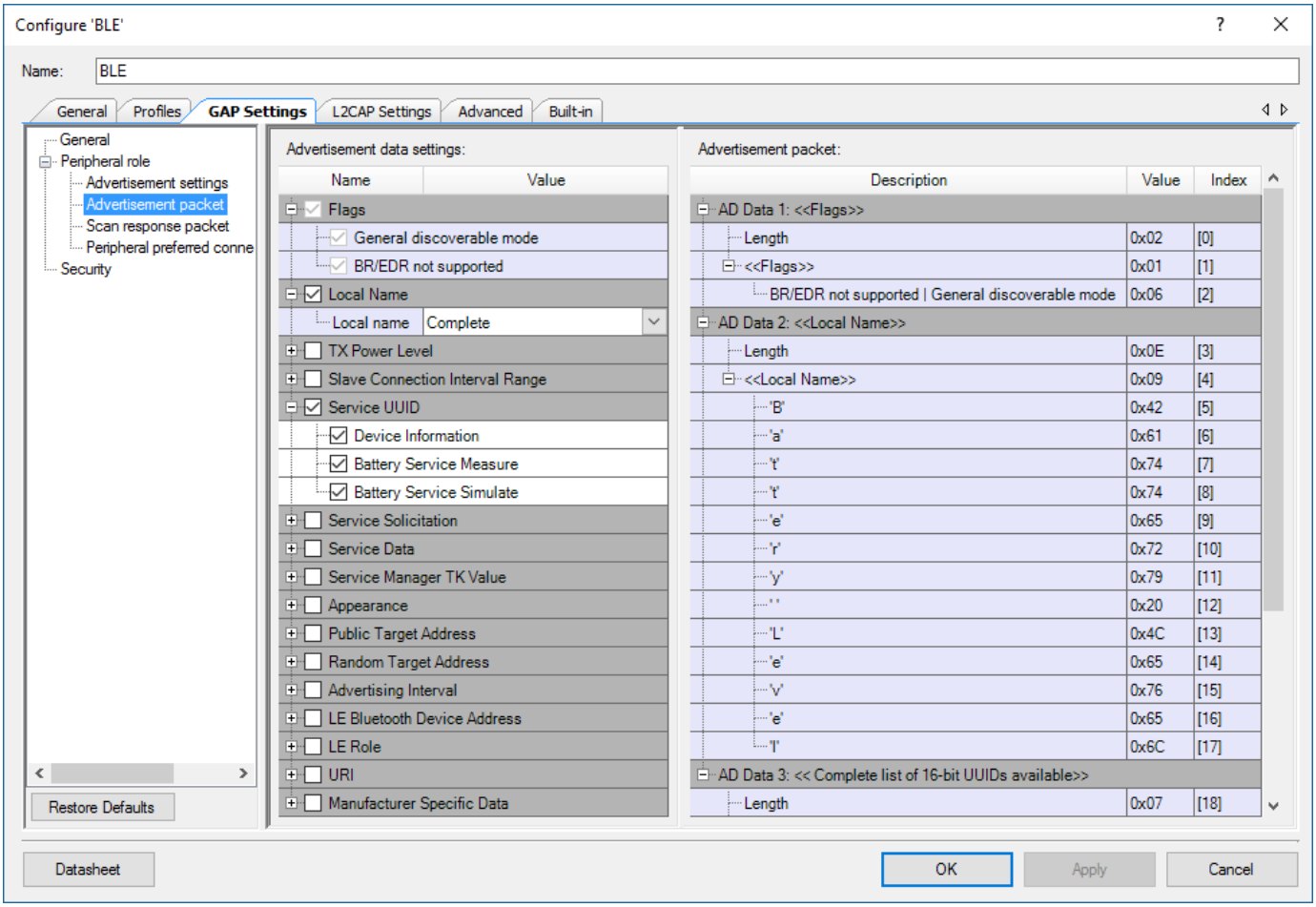

Figure 4. GAP settings -> Advertisement packet

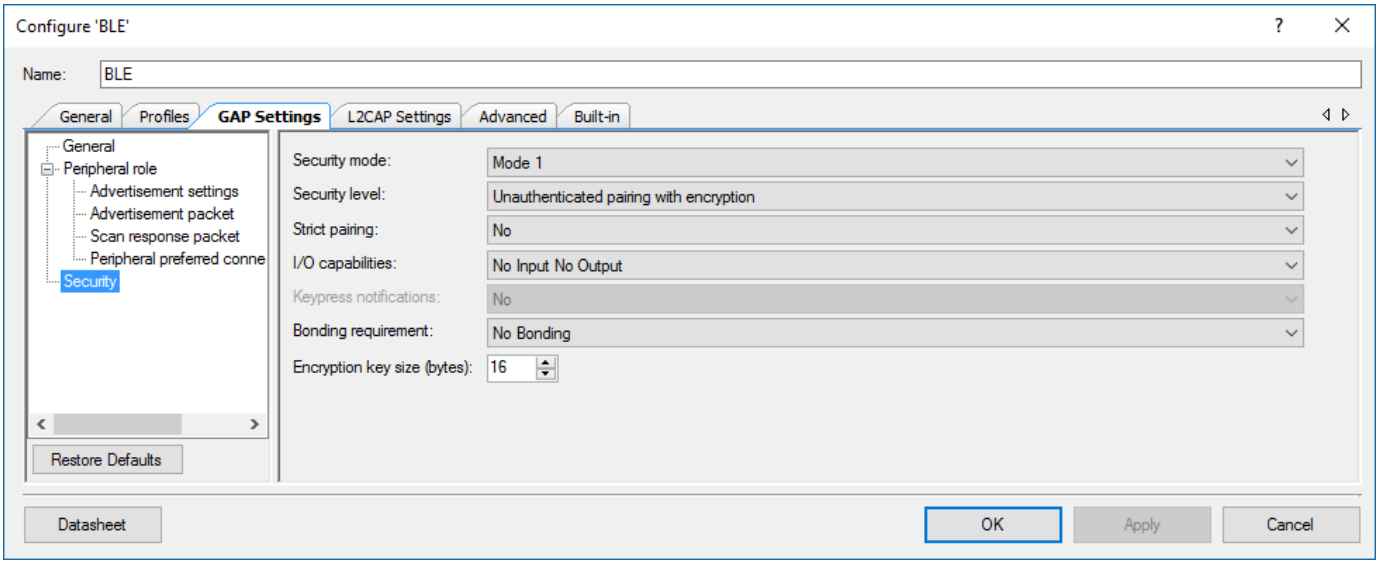

Figure 5. Security settings

Project Description

This project demonstrates functionality of the Battery Service of the BLE component.

The reference to the ADC in PSoC 4 can be either internal 1.024V or VDD. A simple method to measure VDD in PSoC4 would be to use a resistive divider and scale the VDD to 1.024V and set the ADC reference to internal 1.024V. This project measures the battery voltage which is equal to the VDD voltage without an external resistive divider.

When VREF is selected as reference to the ADC, and if the reference bypass is enabled, this will bring out VREF (through an internal series resistance).

Following is the procedure to measure VDD that exploits this feature.

- Make sure that J2 pin P3[0] is connected to J3 pin VREF.
- Set the reference of ADC to VREF and enable a bypass. Keep the bypass enabled for a short time (25 ms in this test) and the external capacitor is charged to VREF.
- Change the reference of the ADC to VDD and measure the voltage on P3[0].
- Calculate the VDD voltage using the formula:

  VDD = (1.024 * Full Scale Counts) / ADC Counts P3[0]

  where Full Scale Counts = 2047. With the ADC configured for a single ended measurement, the effective resolution is 11 bits and the full scale count is 2047.

Right after the device is started it performs the BLE component initialization. Two callback functions are required in this project for BLE operation. One callback function (AppCallBack()) is required to receive generic events from BLE Stack. The other callback function (BasCallBack()) is required for receiving events from the BAS services. CyBle_GappStartAdvertisement() API is called after CYBLE_EVT_STACK_ON event to start advertising with the packet shown in **Figure 4**. To indicate that the device is advertising, the green LED is blinking. The red LED will be lighted on after disconnection to indicate that no Client is connected to the device. When a Client is connected successfully, both red and green LEDs will turn off. When the measured battery voltage drops below 10% limit, the blue LED will be on.

On an advertisement event timeout, if the device is not connected to any Client, the device goes to the Low Power mode (Hibernate mode) and waits for a SW2 button press to wake up the device again and start advertising. While the device will be in Hibernate mode the red LED will be turned on.

Expected Results

You can use CySmart app on a Windows PC, Android or iOS BLE-compatible device as client for connection to Battery Service.

To use CySmart Windows application as Battery Service client:

- Connect the CySmart BLE dongle to a USB port on the PC.

- Launch CySmart app and select connected dongle in dialog window.
- Reset development kit to start advertising by pressing SW1 button.
- Click **Start Scan** button to discover available devices.
- Select **Battery Level** in the list of available devices and connect to it.
- Click **Pair**, then **Discover All Attributes**, then **Read All Characteristics**, and finally **Enable All Notifications** in CySmart app.
- Notice values of Battery Level descriptors:

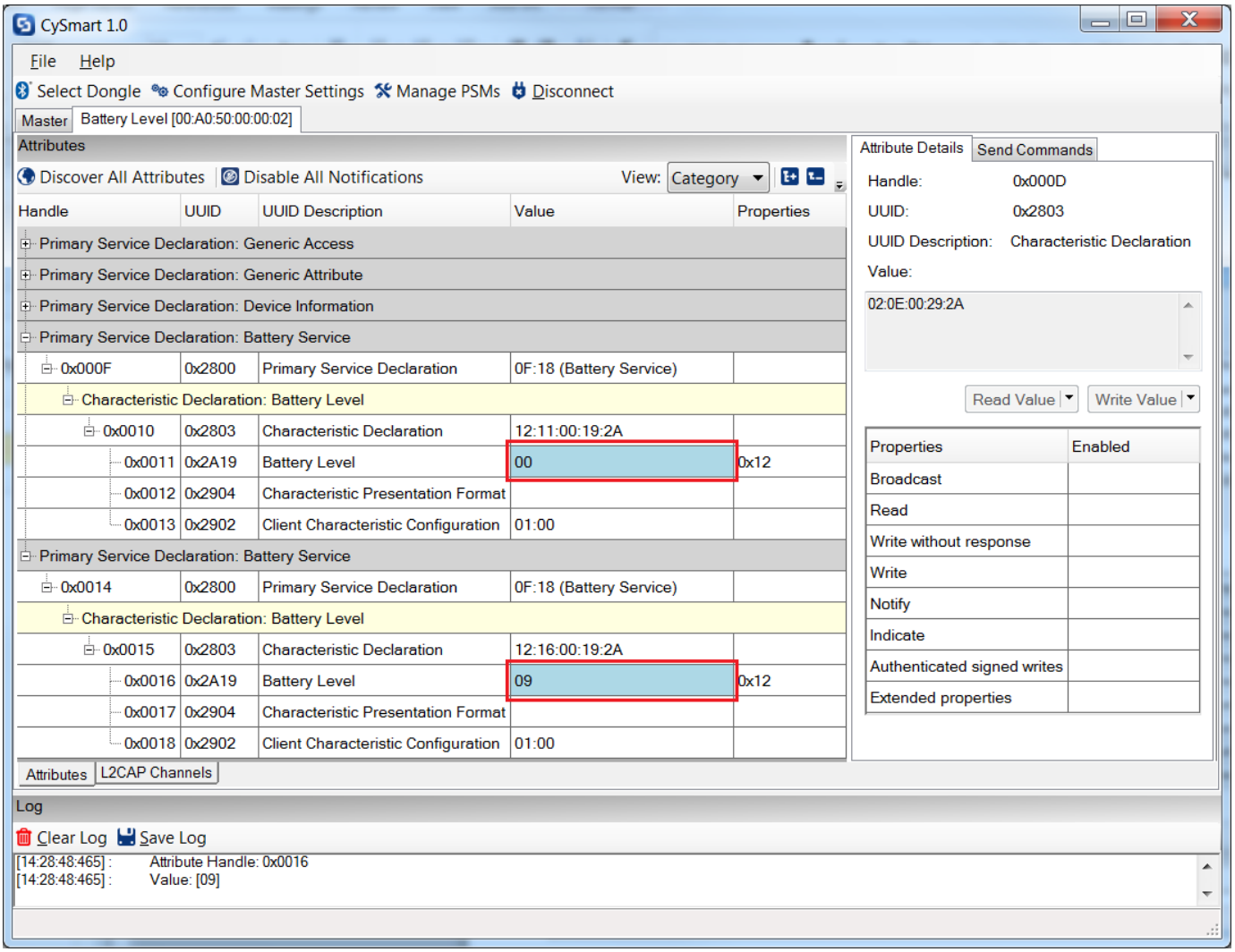

Figure 6. CySmart Windows app

If you have some problems with usage of CySmart app, please refer to CySmart User Guide.

To use CySmart mobile app as Battery Service client:

- Launch CySmart mobile app on the Android or iOS BLE-compatible device.
- Swipe down to refresh the list of found BLE devices.
- Connect to “Battery Level” device and select “Battery Service” in service list.
- Notice the value of Battery Level.

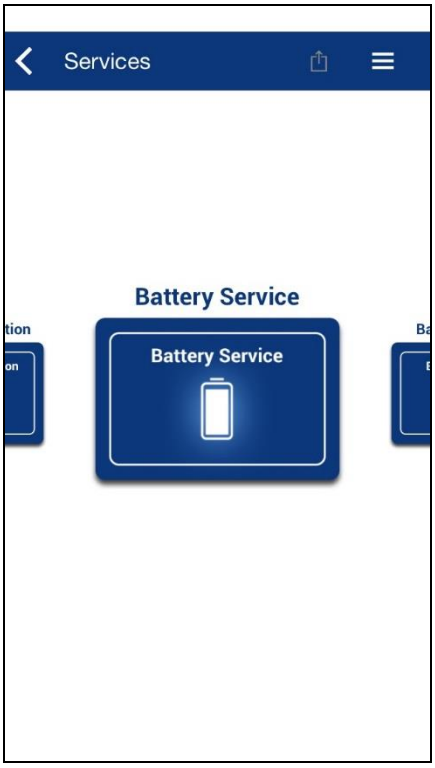

Figure 7. CySmart iOS app recognized Battery Service

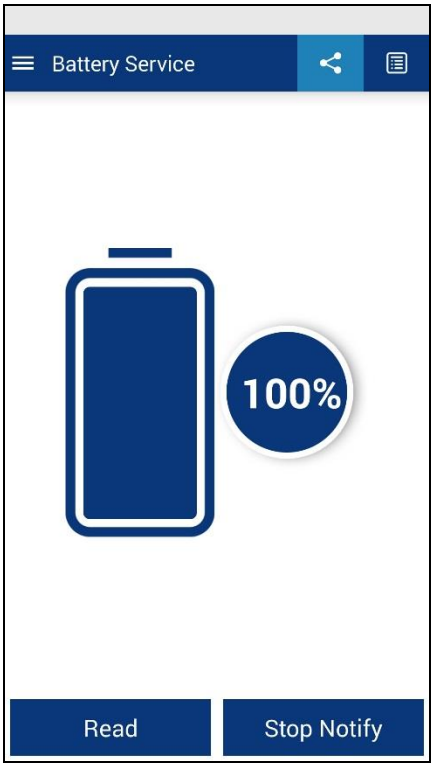

Figure 8. CySmart Android app shows battery level

Also, you can notice that the blue color LED will be on when a measured battery level is less than 10%.