BLE Running Speed and Cadence Profile

1.0

Features

- BLE Running Speed and Cadence Service GATT Server (Sensor) role operation in GAP Peripheral role
- Support “Running” and “Walking” profiles
- DeepSleep mode support
- Reporting the workflow status through UART
- LED status indication

General Description

This example project demonstrates the Running Speed and Cadence Sensor operation of the BLE PSoC Creator Component. The device simulates running/walking data measurements and sends it over the BLE Running Speed and Cadence Service.

Development Kit Configuration

Configure your device as follows:

- The UART RX pin is connected to port 1 pin 4.
- The UART TX pin is connected to port 1 pin 5.
- A mechanical button (port 2 pin 7) is used to wake up the device and start re-advertising.
- The red LED (port 2 pin 6) is used to indicate the BLE disconnection state.
- The green LED (port 3 pin 6) is used to indicate the advertising state.
- The blue LED (port 3 pin 7) is used to indicate “Running” or “Walking” profile.

Project Configuration

The top design schematic is shown in **Figure 1**.

BLE Running Speed and Cadence Profile Example Project

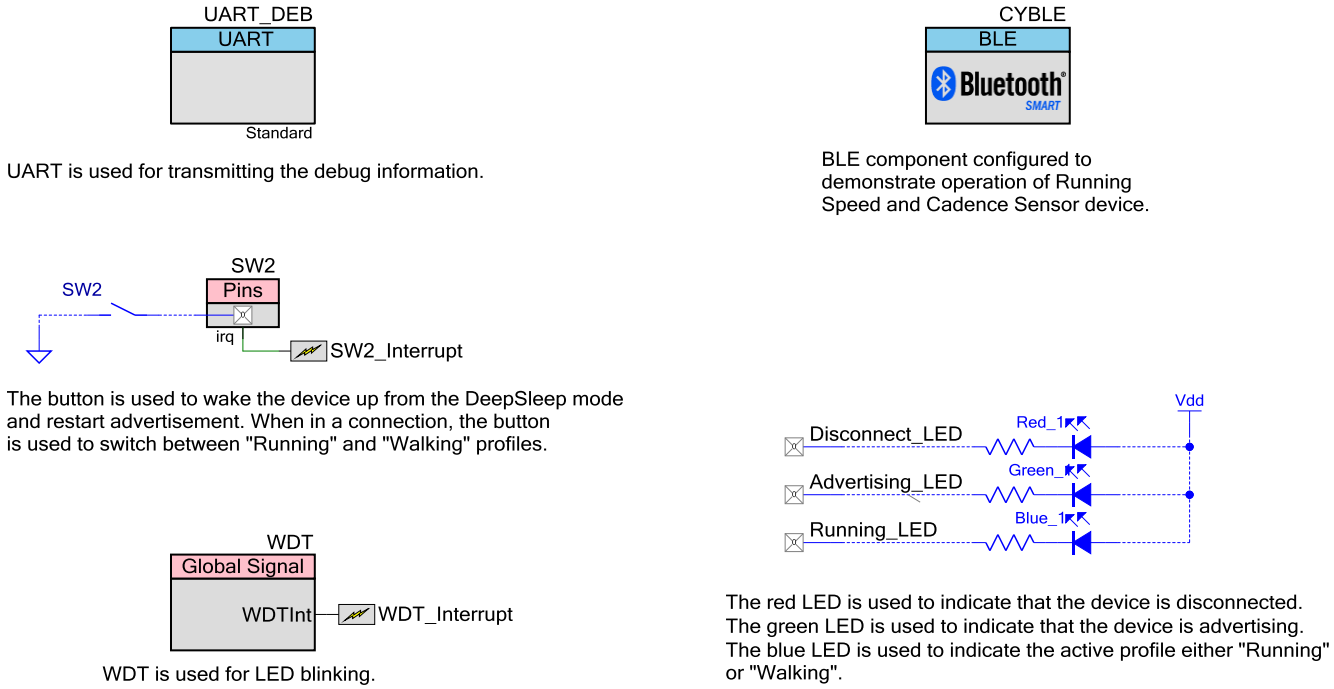

Figure 1. Top design schematic

The BLE (CYBLE) component is configured as Running Speed and Cadence Sensor in the GAP Peripheral role.

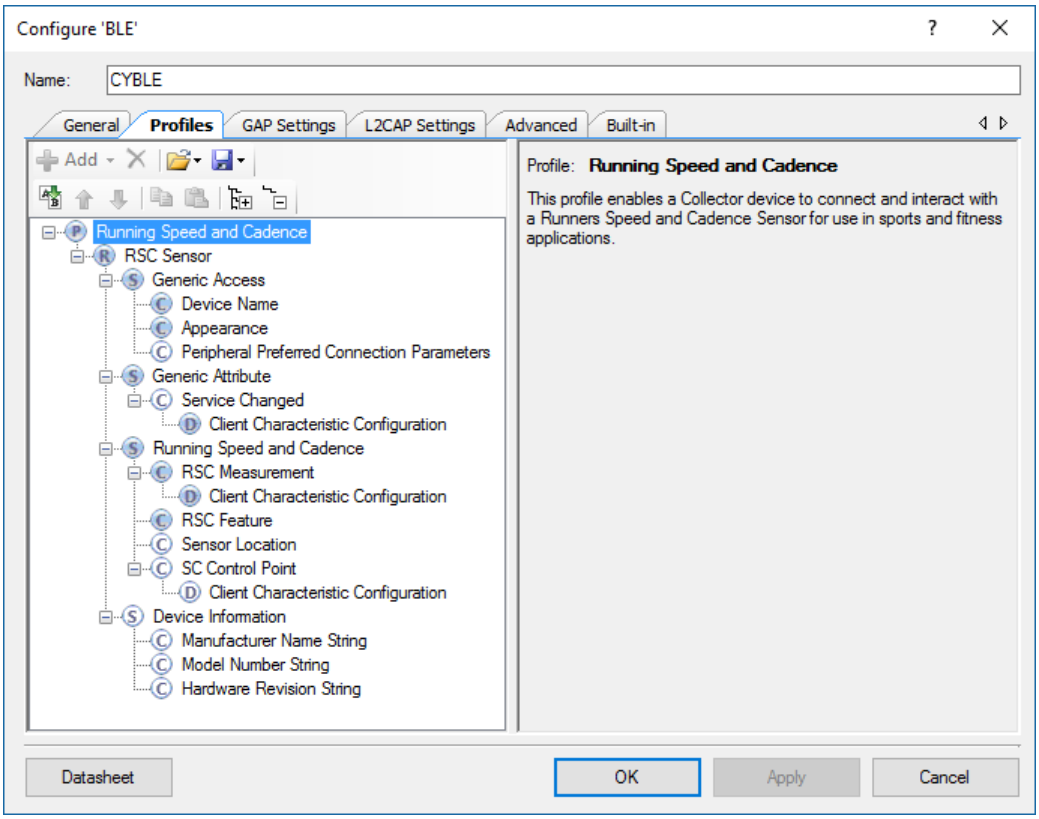

Figure 2. GATT settings

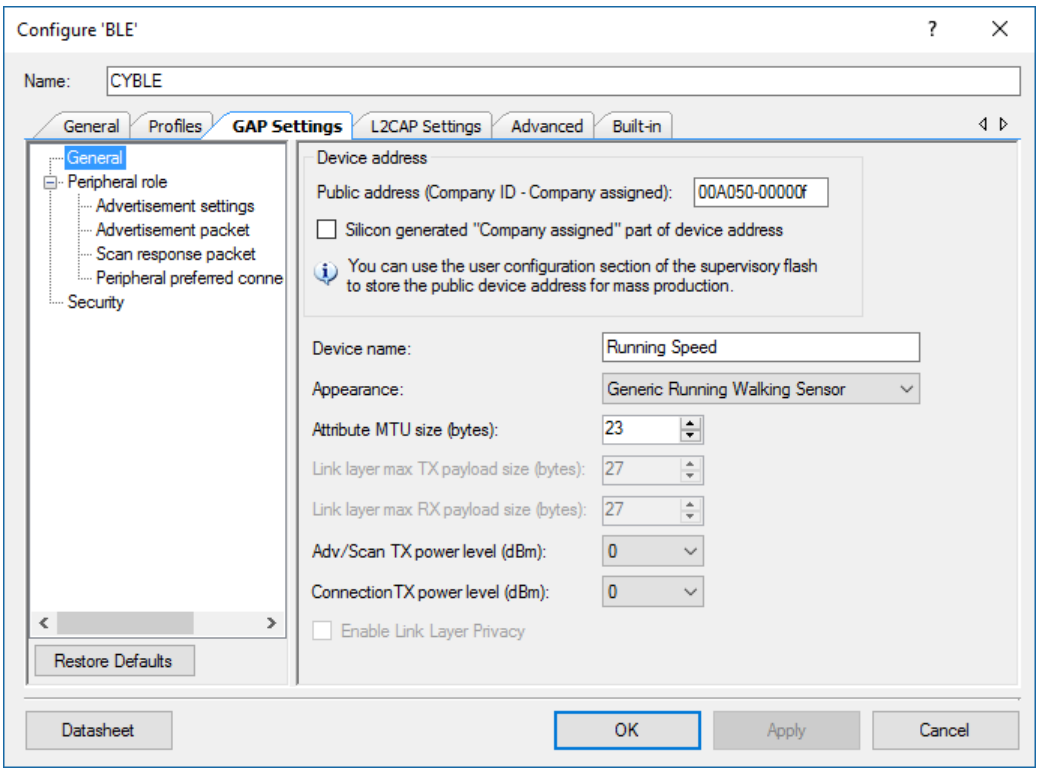

Figure 3. GAP settings

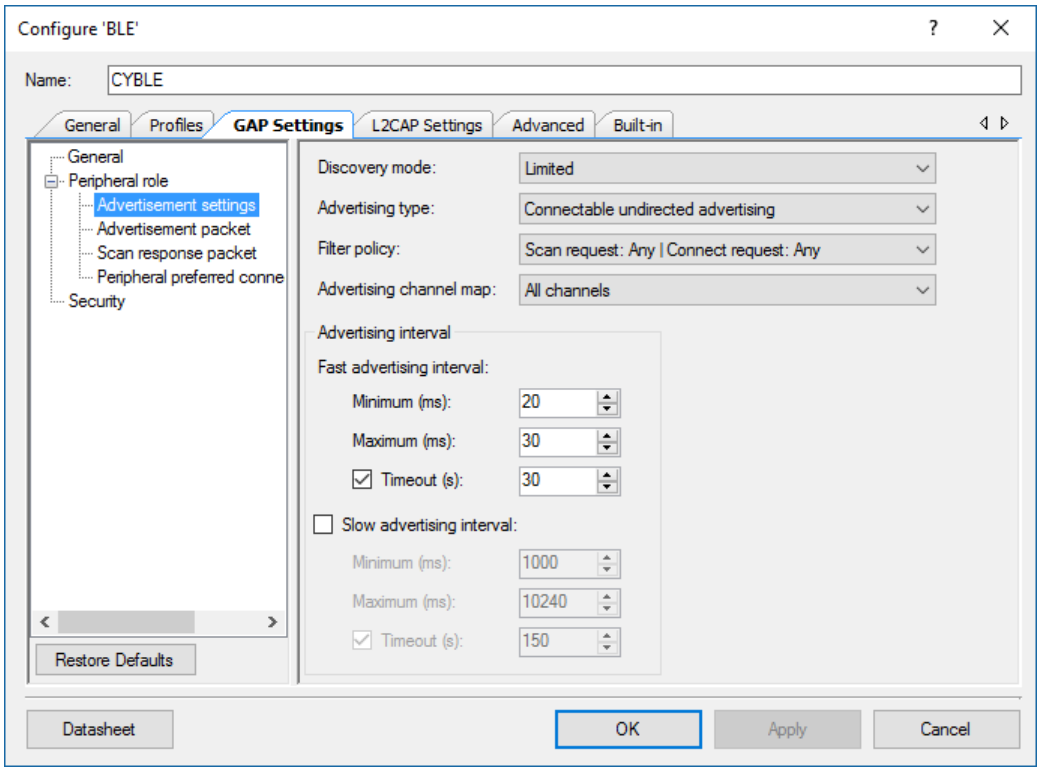

Figure 4. GAP settings -> Advertisement settings

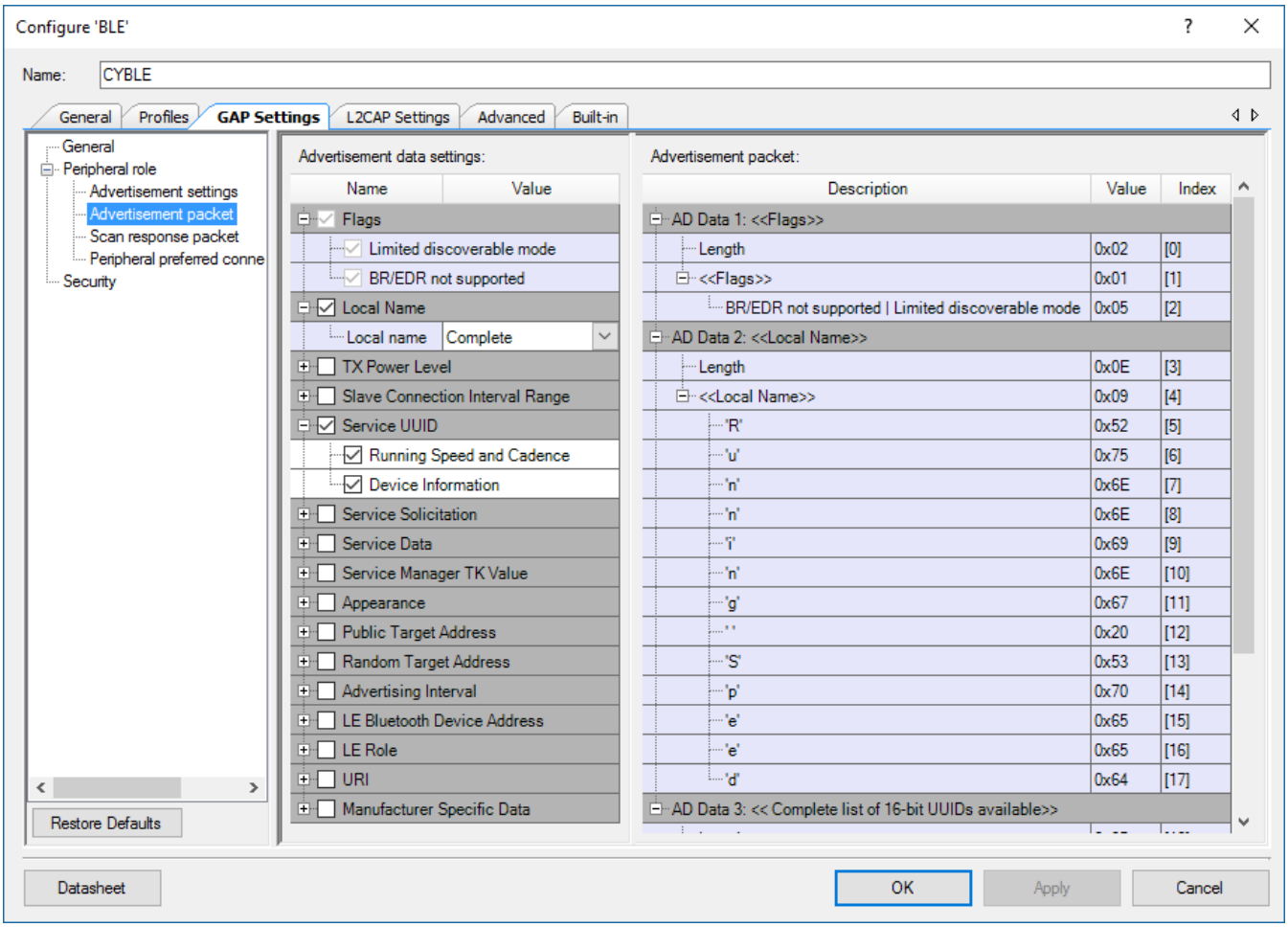

Figure 5. GAP settings->Advertisement packet

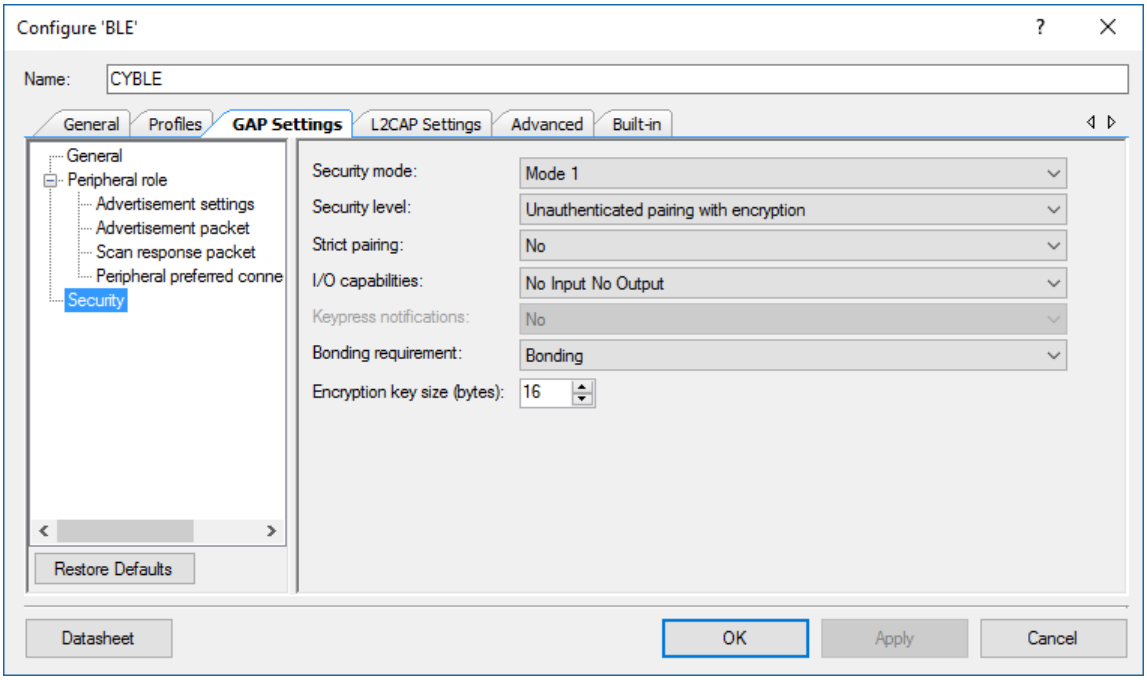

Figure 6. GAP Settings -> Security

Project Description

The project demonstrates the core functionality of the BLE component configured as a Running Speed and Cadence Sensor. The example project requires CY8CKIT-042-BLE Pioneer Kit with PSoC 4100-BL, PSoC 4200-BL or PRoC BLE devices.

For operation the example project uses two callback functions: AppCallBack() and RscServiceAppEventHandler(). One callback function (AppCallBack()) is required for receiving the generic events from the BLE Stack, and the second (RscServiceAppEventHandler()) is required for receiving the events from the Running Speed and Cadence Service.

The example project uses UART component for displaying debug information.

To start the example project operation, build it and program onto CY8CKIT-042-BLE Pioneer Kit with PSoC 4100-BL, PSoC 4200-BL or PRoC BLE device. Right after the startup the BLE, UART and ISR components will be initialized. After the initialization the BLE component begins its operation that can be seen on the RGB LED which starts blinking with a green color. This indicates that the device has started advertising and it is available for the connection with a central device. For advertisement the example project uses the packet structure as shown in **Figure 5**. This example requires an external GAP Central device configured in GATT Client role for operation. To connect to the Running Speed and Cadence Sensor device, send a connection request when the device is advertising. After 30 seconds, if no central device has connected to the Running Speed and Cadence Sensor, the Sensor device will stop advertisement and red LED will be turned on indicating the disconnection state.

While connected to a Client and between connection intervals, the device is put into the DeepSleep mode.
After a Client device is connected to the Running Speed and Cadence Sensor, the Client device should enable notifications for the RSC Measurement Characteristic. When notifications are enabled, the Sensor will be sending notifications to the peer Client periodically. The period of the notifications is dependent on the connection interval used by the Client. In case of 30 ms connection interval, the period of the notifications is approximately 3 seconds. The notifications sent from the Sensor contain “Instantaneous Speed”, “Instantaneous Cadence”, “Instantaneous Stride Length”, “Total Distance”, and “Walking or Running Status bit” fields. The example project also periodically increases the value of “Instantaneous Cadence”, “Instantaneous Stride Length”, and, therefore, “Instantaneous Speed”. The period of values increasing is approximately 10 seconds for 30 ms connection interval. After the “Instantaneous Cadence” and “Instantaneous Stride Length” reached their maximum allowed values, they will be reset back to minimum values. By default, the project simulates the “Walking” profile, but it can simulate the “Running” profile as well. To switch to the “Running” profile, press and hold SW2 button on CY8CKIT-042-BLE Pioneer Kit. When “Running” profile becomes active, the blue LED is turned on indicating this.

Expected Results

The project is intended to work with any BLE-compatible device (e.g. phone, tablet). Appropriate software with Running Speed and Cadence Profile support should be installed on client OS. CySmart mobile app (available for Android and iOS) has support for this profile and can be used as a client for Running Speed and Cadence.

To use CySmart mobile app as a client for Running Speed and Cadence Service, launch it and swipe down the screen to refresh the list of BLE devices available nearby. Make sure that development kit is advertising (green LED is blinking): you may need to press SW1 button in order to wake up device from hibernate mode. Once “Running Speed” device appears in BLE devices list, you can connect to it and choose “Running Speed & Cadence Service” in service selector.

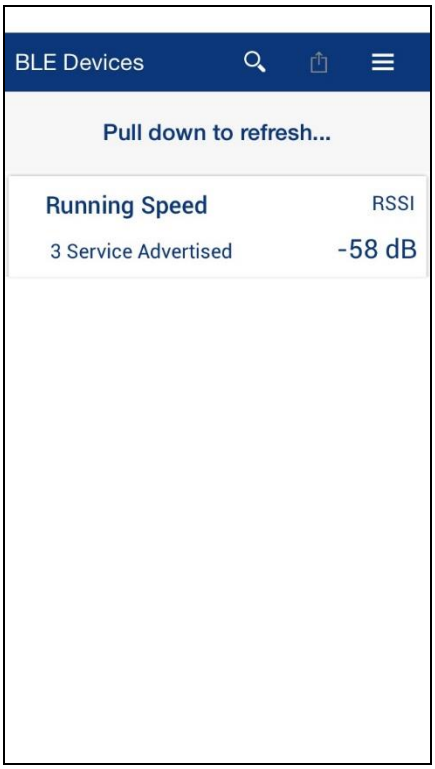

Figure 7. CySmart iOS app recognized BLE kit as Running Speed Reporter

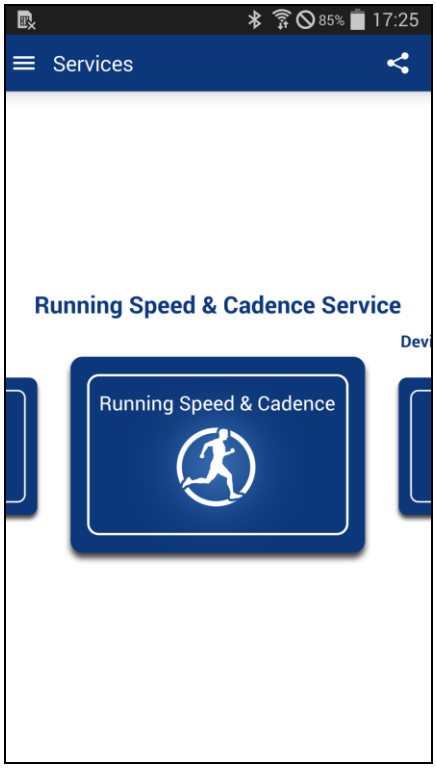

Figure 8. CySmart Android app shows "Running Speed & Cadence Service" in service selector

Once connected, CySmart mobile application provides interface for measurement of distance being run, calories burnt and average speed based on running speed and cadence values obtained from development kit. To attain this functionality, please enter runner's weight into appropriate textbox and press "Start" button:

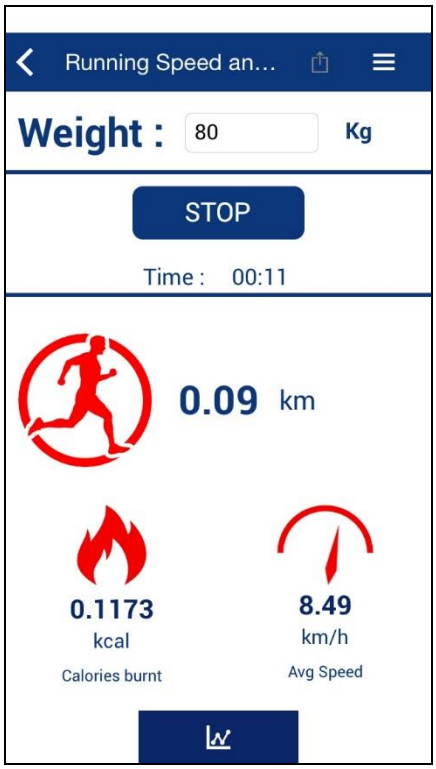

Figure 9. Running Speed and Cadence Service Interface on CySmart app provides possibility to enter runner's weight

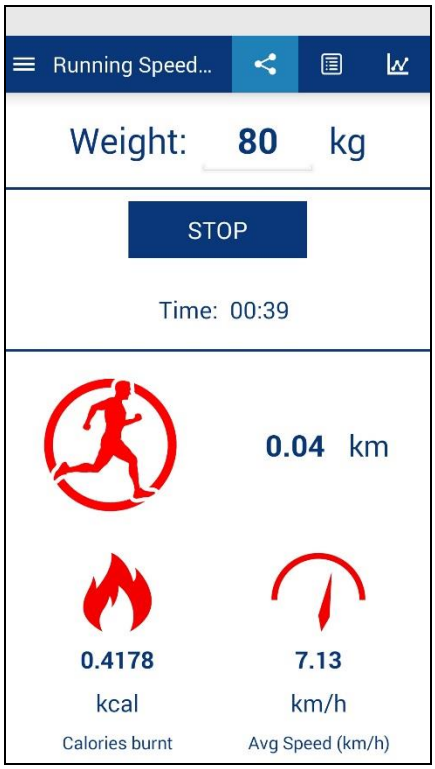

Figure 10. CySmart Android app simulates measurement of total distance being run, amount of calories burnt and average speed

Also, the project can be used together with CySmart app for Windows. To connect to Running Speed and Cadence Service, this application requires USB Bluetooth Low Energy dongle installed (included with CY8CKIT-042-BLE Pioneer Kit). For further instructions on how to use CySmart application, see CySmart User Guide.

To use the CySmart Windows application as a Running Speed and Cadence Client:

- Connect the CySmart BLE dongle to a USB port on the PC.
- Launch the CySmart app and select the connected dongle in the dialog window.
- Reset the development kit to start advertising by pressing the SW1 button.
- Click the **Start Scan** button to discover available devices.
- Select **Running Speed** 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 the CySmart app. As a result you should see that the **Running Speed Reporter** will start notification of RSC Measurement Characteristic:

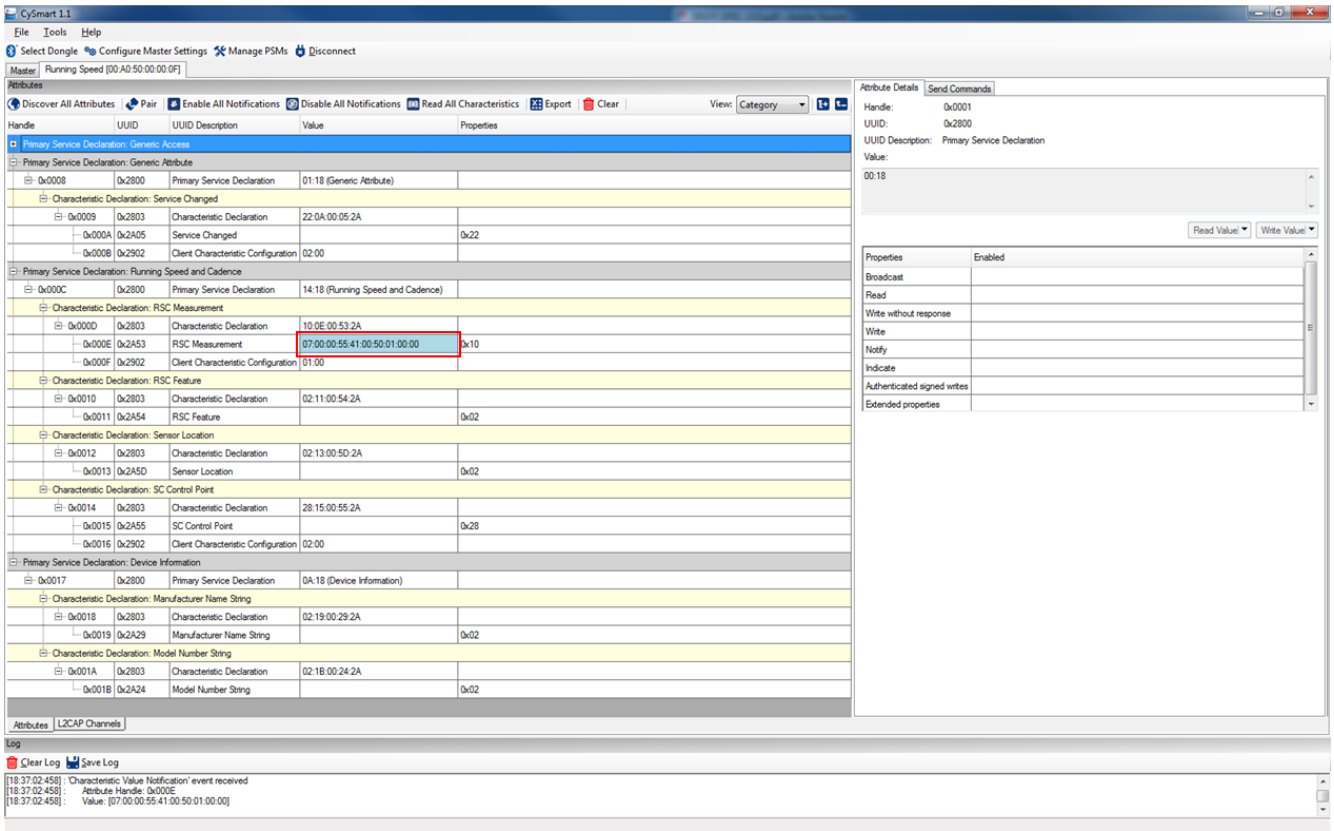

Figure 11. RSC Measurement Characteristic notification

- Send the **Set Cumulative Value** Op Code to the **SC Control Point**. To do that select the SC Control Point and write the **01:XX:XX:XX:XX**, where 01 is a **Set Cumulative Procedure** Op Code XX:XX:XX:XX is a four byte cumulative value, then press **Write Value.**

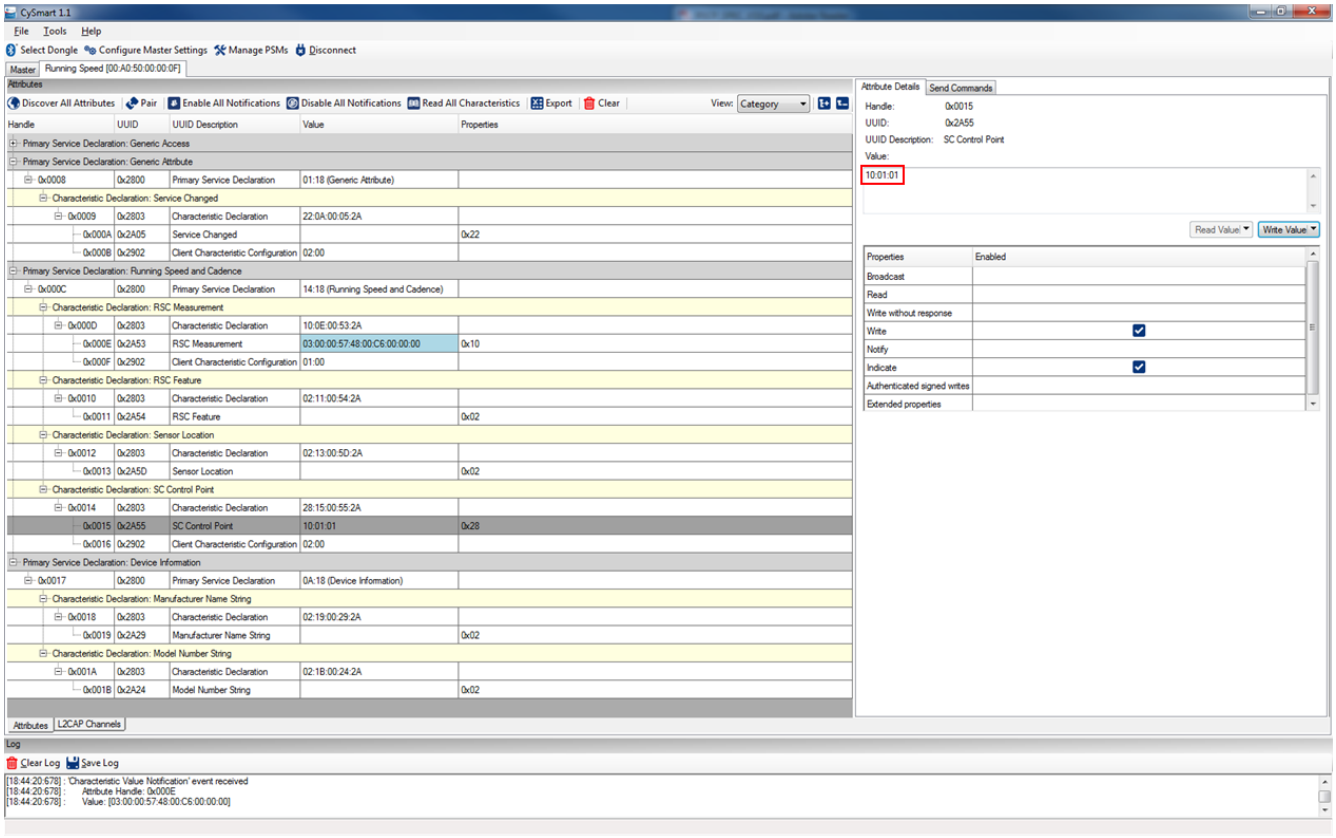

Figure 12. Set Cumulative Value Procedure response

- Observe a response from the **SC Control Point**. The general format of a response is following – XX:YY:ZZ, where XX - response Op Code, YY- requested Op Code, ZZ – status byte. Status byte can be set to one of the following values: 0x01 – Success, 0x02 – Op Code is not supported, 0x03 – Invalid parameter, 04 – Operation failed.
- Select the **Sensor Location Characteristic** and press **Read Value**. Observe the sensor location value:

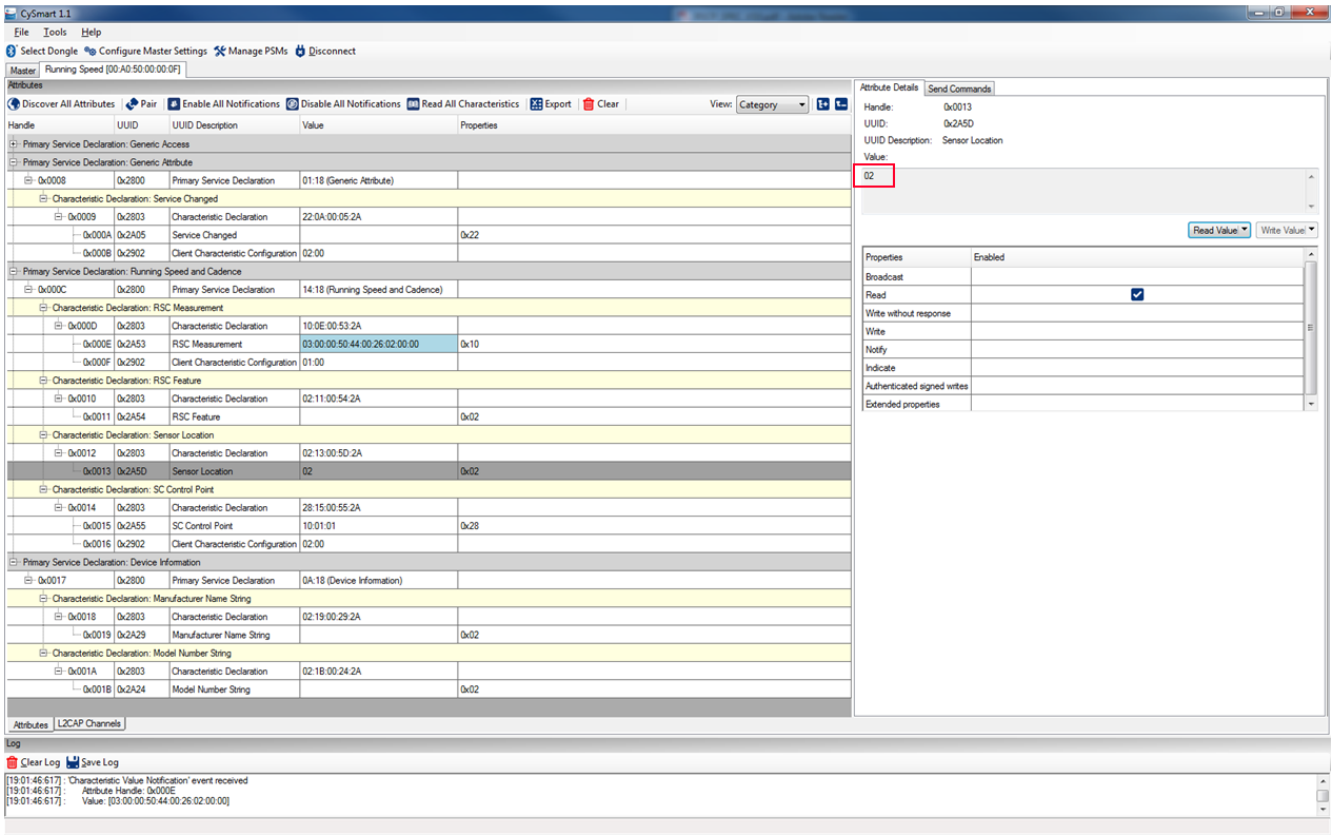

Figure 13. Read Sensor Location Characteristic

- Send the **Request Sensor Locations** Op Code to the SC Control Point. Write 04 in the **Value** field then press **Write Value.**

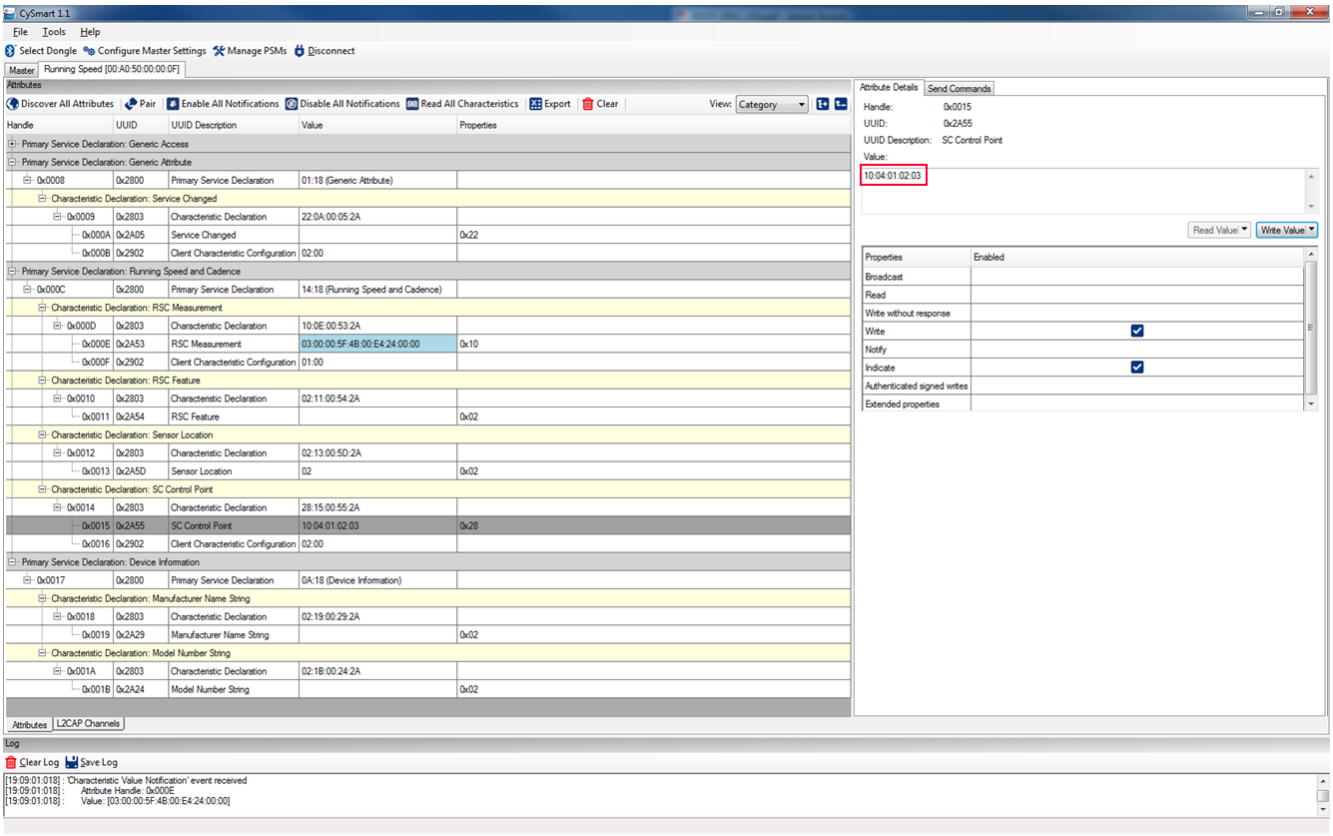

Figure 14. Request Sensor Locations response

- Observe a response from the **SC Control Point**. In case of successful execution of the **Request Sensor Locations** procedure the first three bytes of the response have the same meaning as it is shown for **Set Cumulative Value** procedure and all the remaining bytes are identifying the supported sensor locations. In case of a failure there will be only three bytes in a response.
- Send the **Update Sensor Location** Op Code to the SC Control Point. Write 03:03 in the **Value** field then press **Write Value** and observe a response. Read the **Sensor Location Characteristic** value again and observe that the new sensor location was set.