

# BLE Blood Pressure Sensor Example Project

1.0

## Features

- BLE BLS Service GATT Server role operation
- DeepSleep mode support
- Reporting the workflow status through UART
- LED status indication

## General Description

This example project demonstrates the BLE Blood Pressure Sensor application workflow. The Blood Pressure Sensor application utilizes the BLE Blood Pressure profile to report blood pressure measurement records to the Client. Also the Blood Pressure Sensor application utilizes the Battery Service to notify the battery level and the Device Information Service to assert the Device Name, etc.

## 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.
- The VREF pin (J3) is connected to port 3 pin 0 to battery voltage measurement.
- 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 the battery discharge (low power).

## Project Configuration

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

## BLE Blood Pressure Sensor Example project

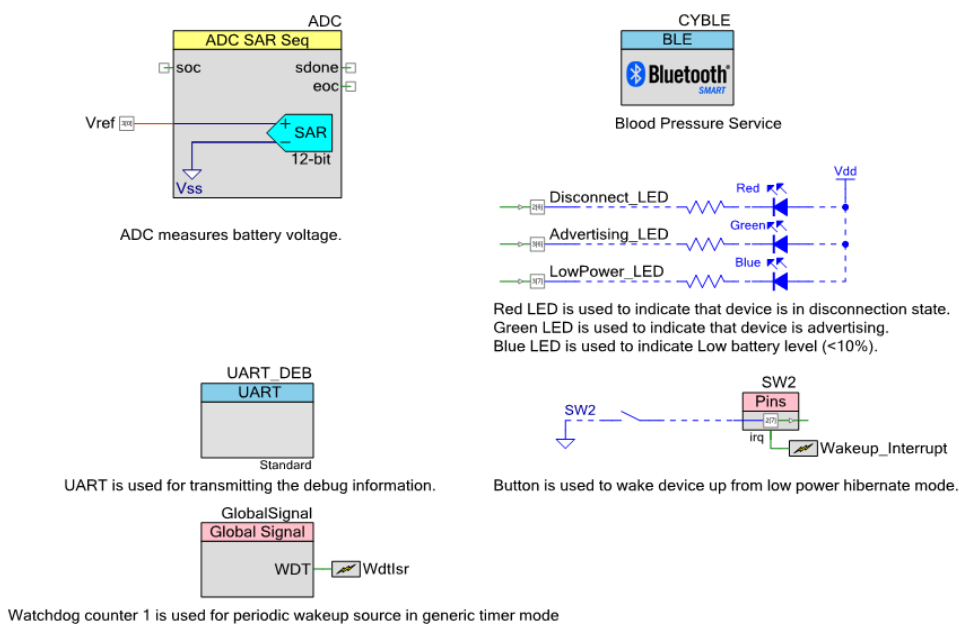


Figure 1. Top design schematic

The BLE component is configured as Blood Pressure Server in the GAP Peripheral role. Also BAS and DIS services are included.

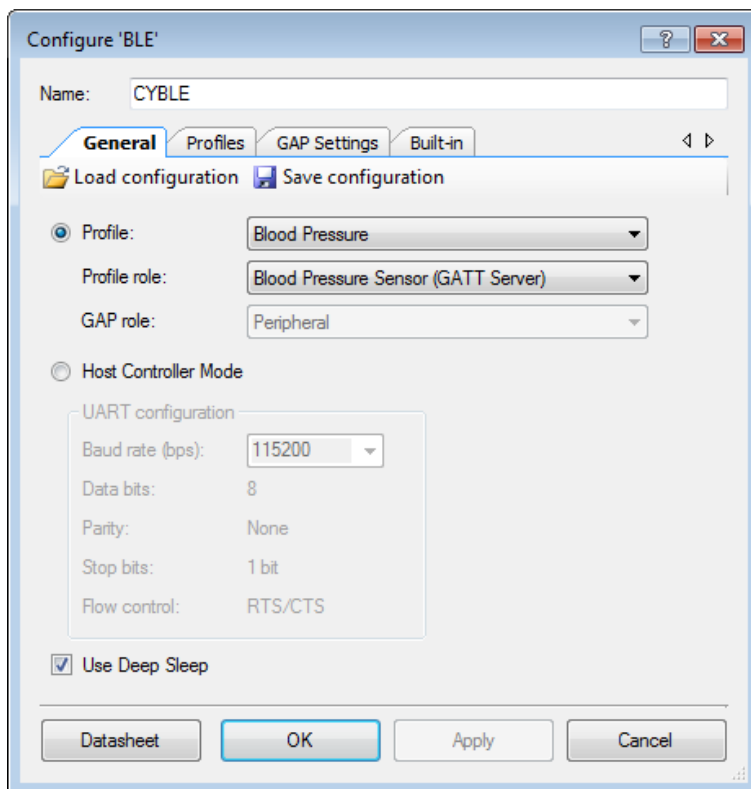


Figure 2. General settings

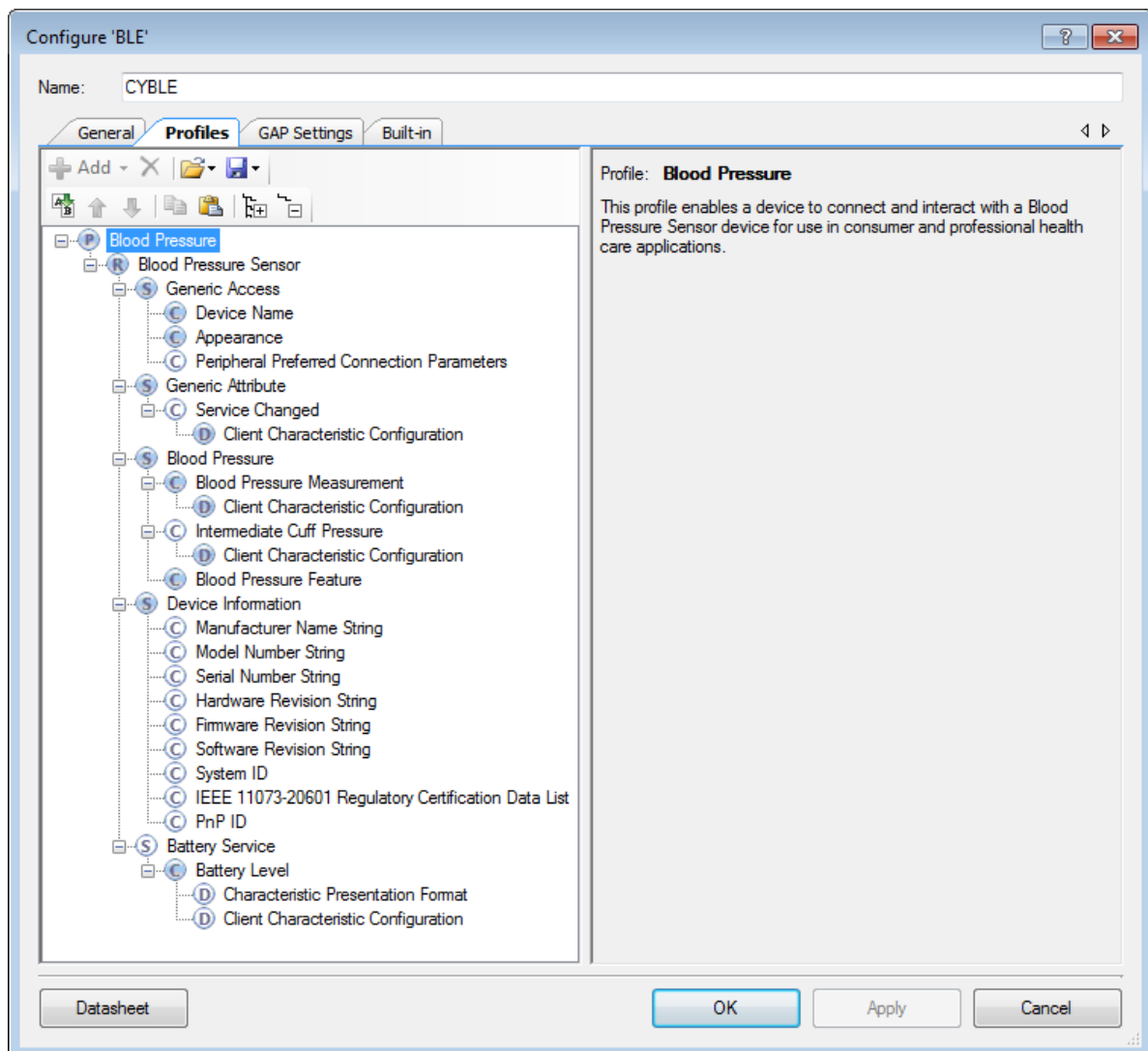


Figure 3. GATT settings

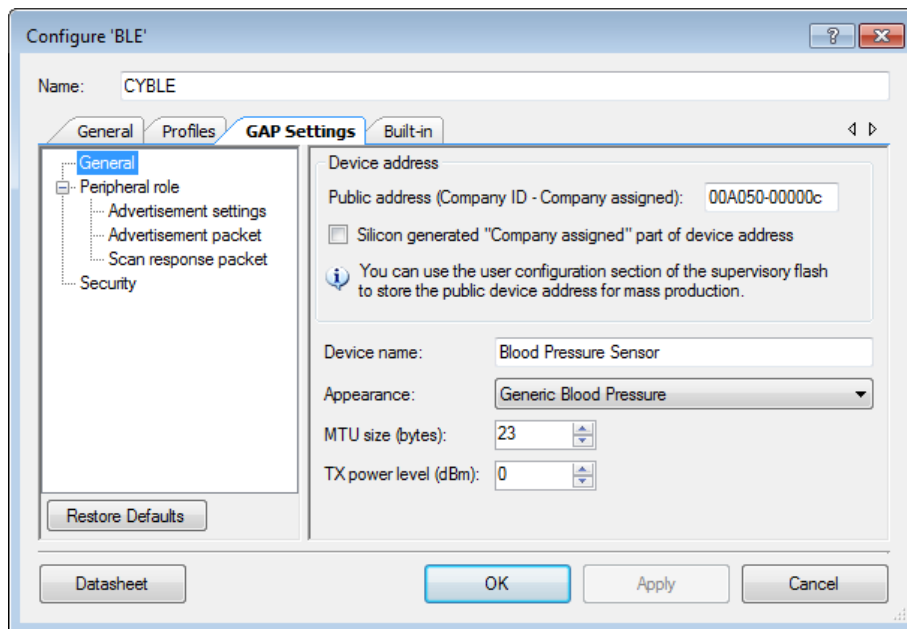


Figure 4. GAP settings

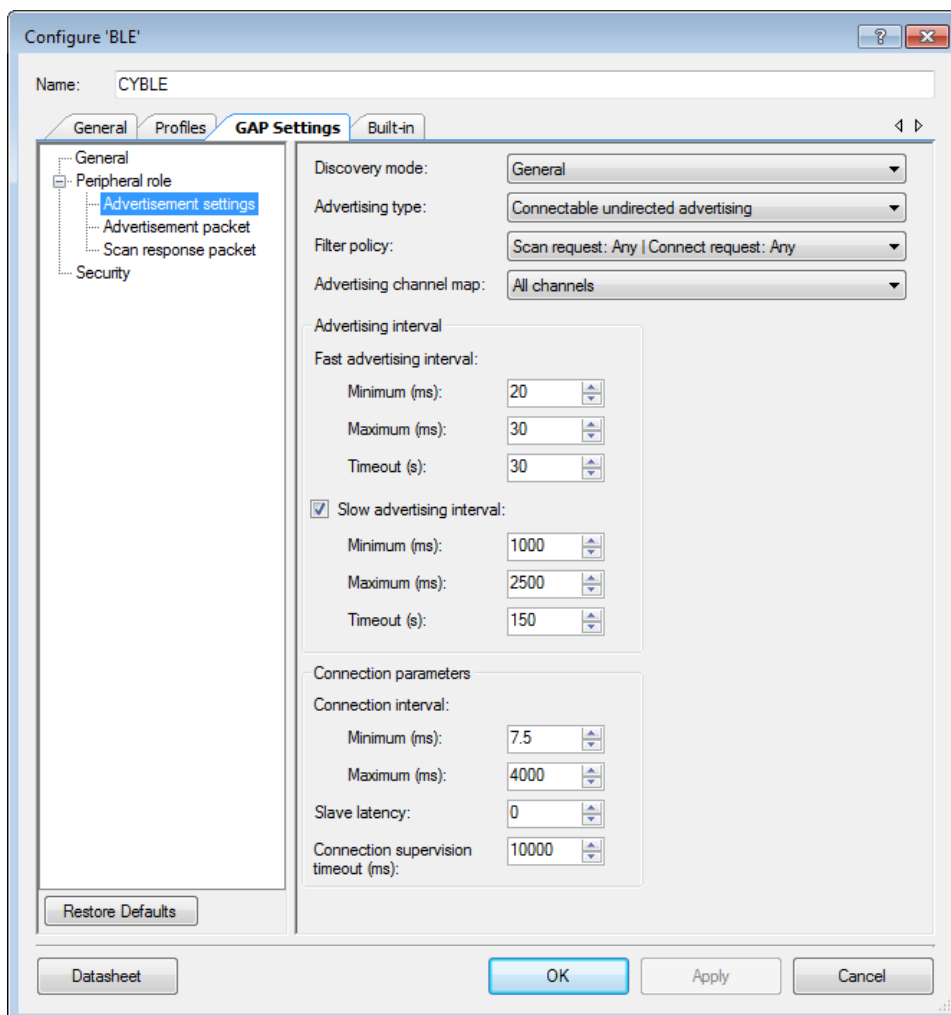


Figure 5. GAP settings -&gt; Advertisement settings

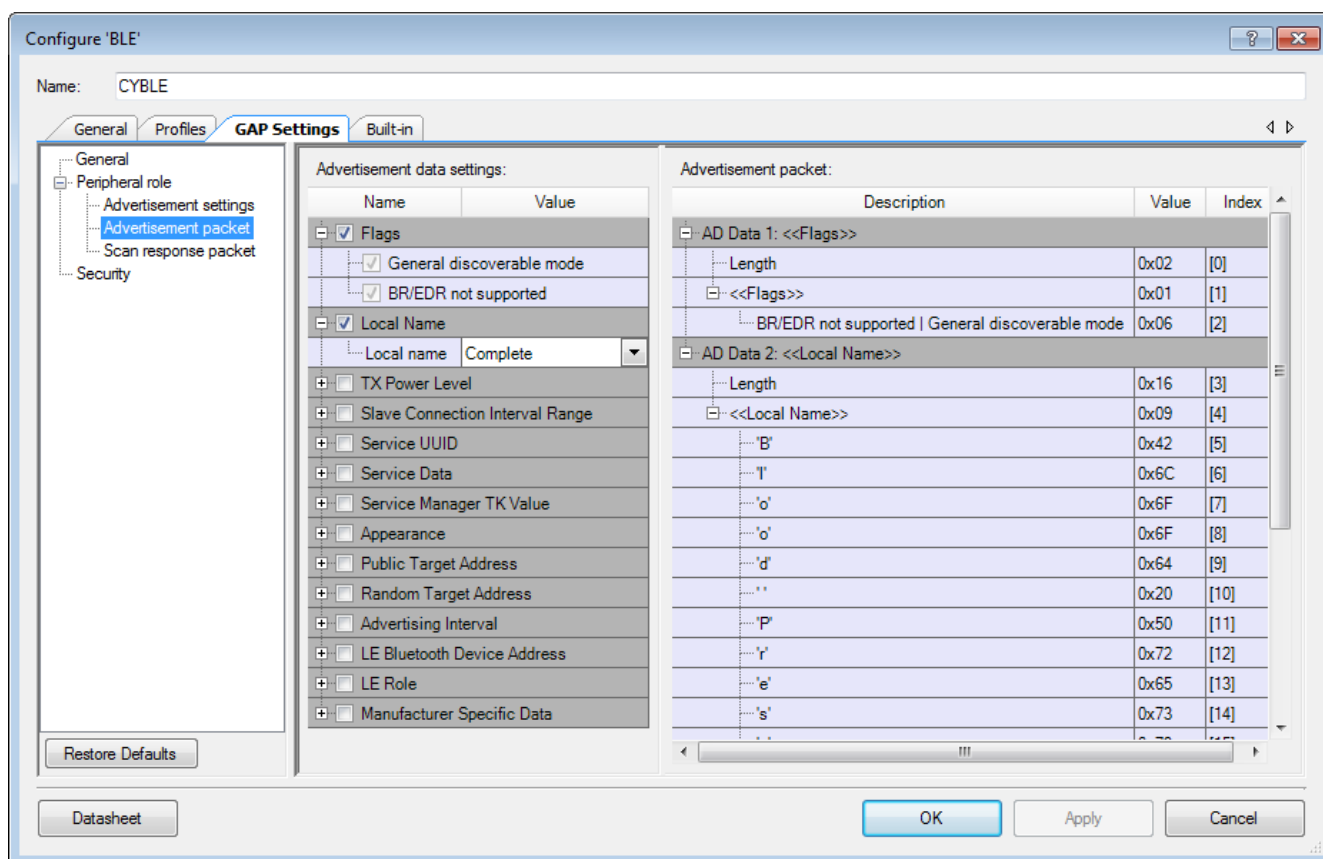


Figure 6. GAP Settings -&gt; Advertisement packet

The Scan response packet settings are also configured to include the Local Name and all the service UUIDs into the Scan response packet.

## Project Description

The project demonstrates the core functionality of the BLE component configured as a Blood Pressure Server.

Right after startup the device performs BLE component initialization. In this project three callback functions are required for the BLE operation. One callback function (AppCallBack()) is required to receive generic events from BLE Stack and the service specific callbacks BasCallBack() and BlsCallBack() for Battery and Blood Pressure service-specific events accordingly. The CYBLE\_EVT\_STACK\_ON event indicates a successful initialization of BLE Stack. After this event is received the component starts advertising with the packet structure as configured in BLE component customizer (see **Figure 6**). The BLE component stops advertising once 180 seconds advertising period expires.

The Blood Pressure Sensor device can be connected with any BLE (4.0 or later) compatible device configured as GAP Central role and GATT Client which supports Blood Pressure Profile. Also the Battery and Device Information Services may be optionally used. To connect to the Blood Pressure Sensor device, just send a connection request to the device while the device is advertising. The green LED is blinking while the device is advertising. The red LED is turned on after disconnection to indicate that no Client is connected to the device. When

the client connects successfully, both red and green LEDs are turned off. If the Client is connected to the Blood Pressure Sensor and the Blood Pressure Measurement (BPM) characteristic indication and/or the Intermediate Cuff pressure (ICF, if it is supported by client) characteristic notifications is/are enabled then the device is simulating blood pressure measurement process continuously and periodically (once a second) sends the ICF notification and/or BPM indication. The WDT is used to timing the blood pressure measurement simulations, battery level measurement and LED blinking. While connected to a Client and between connection intervals, the device is put into Sleep Mode.

## Expected Results

The project sends the Blood Pressure Service characteristic's notifications/indications and Battery Level notifications to the Central Client device which can show them for user. LEDs are blinking as described in Project Description section.

The project is intended to work in pair with any BLE-compatible device (e.g. phone, tablet). Appropriate software with Blood Pressure Profile support should be installed on client OS. CySmart mobile app (available for [Android](#) and [iOS](#)) can act as a client for Blood Pressure Service.

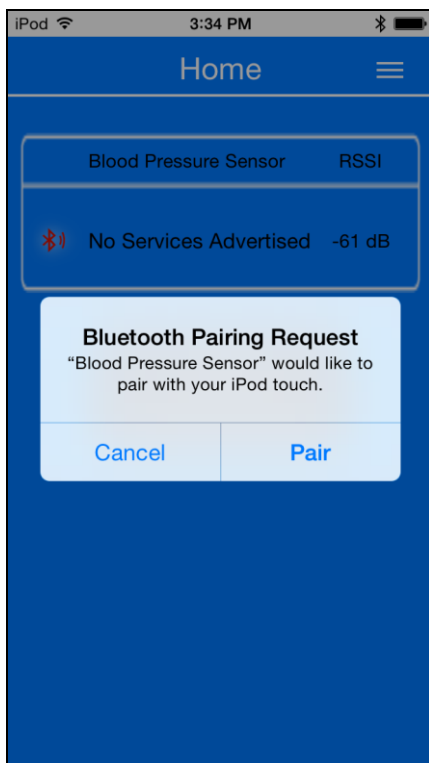


Figure 7. CySmart iOS app pairs with Blood Pressure Sensor Service

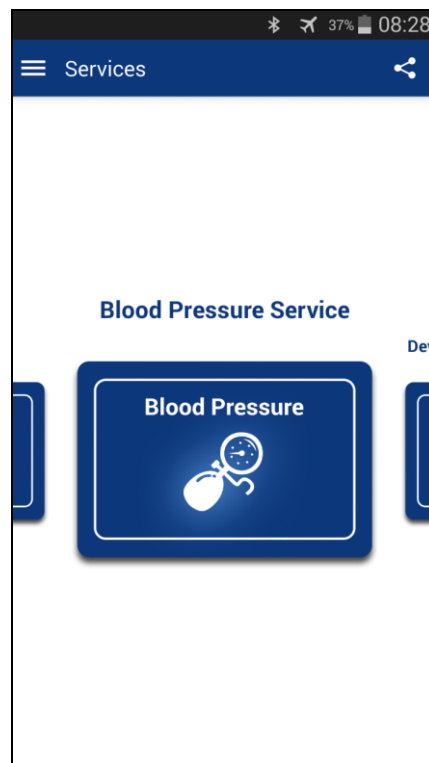


Figure 8. CySmart Android app recognized Blood Pressure Sensor Profile

You can observe simulated values of Blood Pressure with CySmart mobile app:

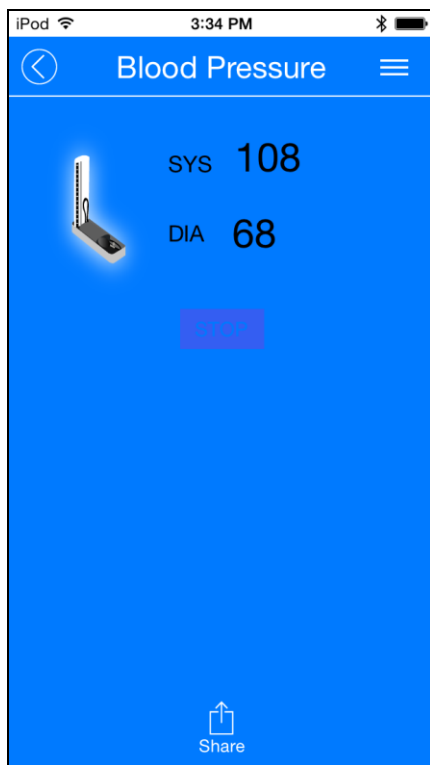


Figure 9. CySmart app on iOS displays simulated Blood Pressure values

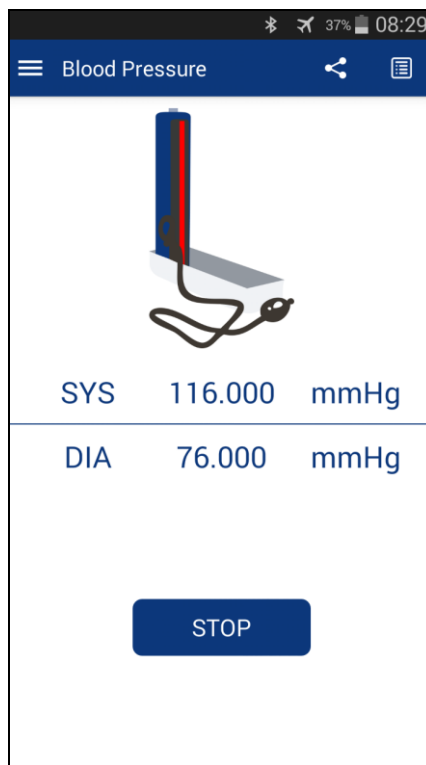


Figure 10. CySmart app on Android displays Blood Pressure values

Also, the Blood Pressure Sensor can be used together with [CySmart app for Windows](#). It is required to match the security settings between Blood Pressure Sensor and CySmart Client and perform pairing (bonding) before any writing (enabling notifications etc.) into Server's GATT database. For further instructions on how to use CySmart application, see [CySmart User Guide](#).

© Cypress Semiconductor Corporation, 2009-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

