PSoC BLE Project Report

University of Western Ontario | ECE 9047

Proximity Sensing Device

Jiayuan Hong (250868589) | Tanushree Das (250877760)

July 12th 2016

PSoC BLE Project Report

University of Western Ontario | ECE 9047

Objectives:

1. Sensing proximity data by a single coil (proximity sensor).

2. Transfer proximity data between BLE Pioneer Kit and Cysmart mobile app.

3. Advertisement with timeout.

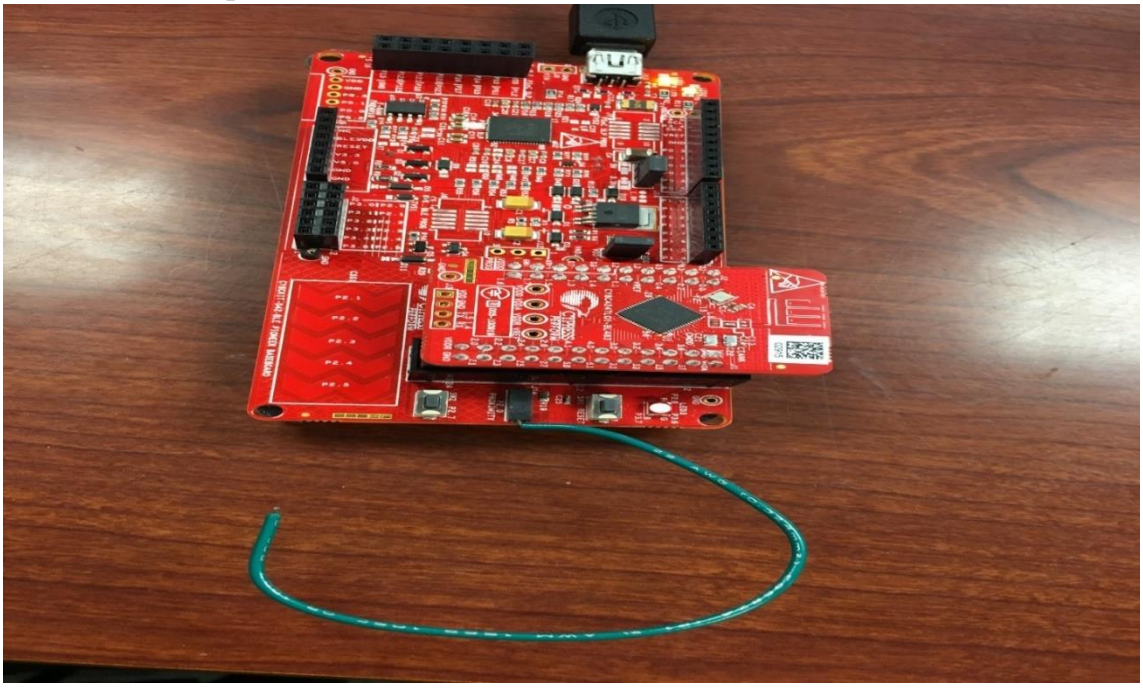
4. Hibernate Mode: For low power implementation, once the device gets disconnected, it enters the hibernate mode; in that case we have to press Switch 2 to exit from Hibernate mode and start working again.

Abstract:

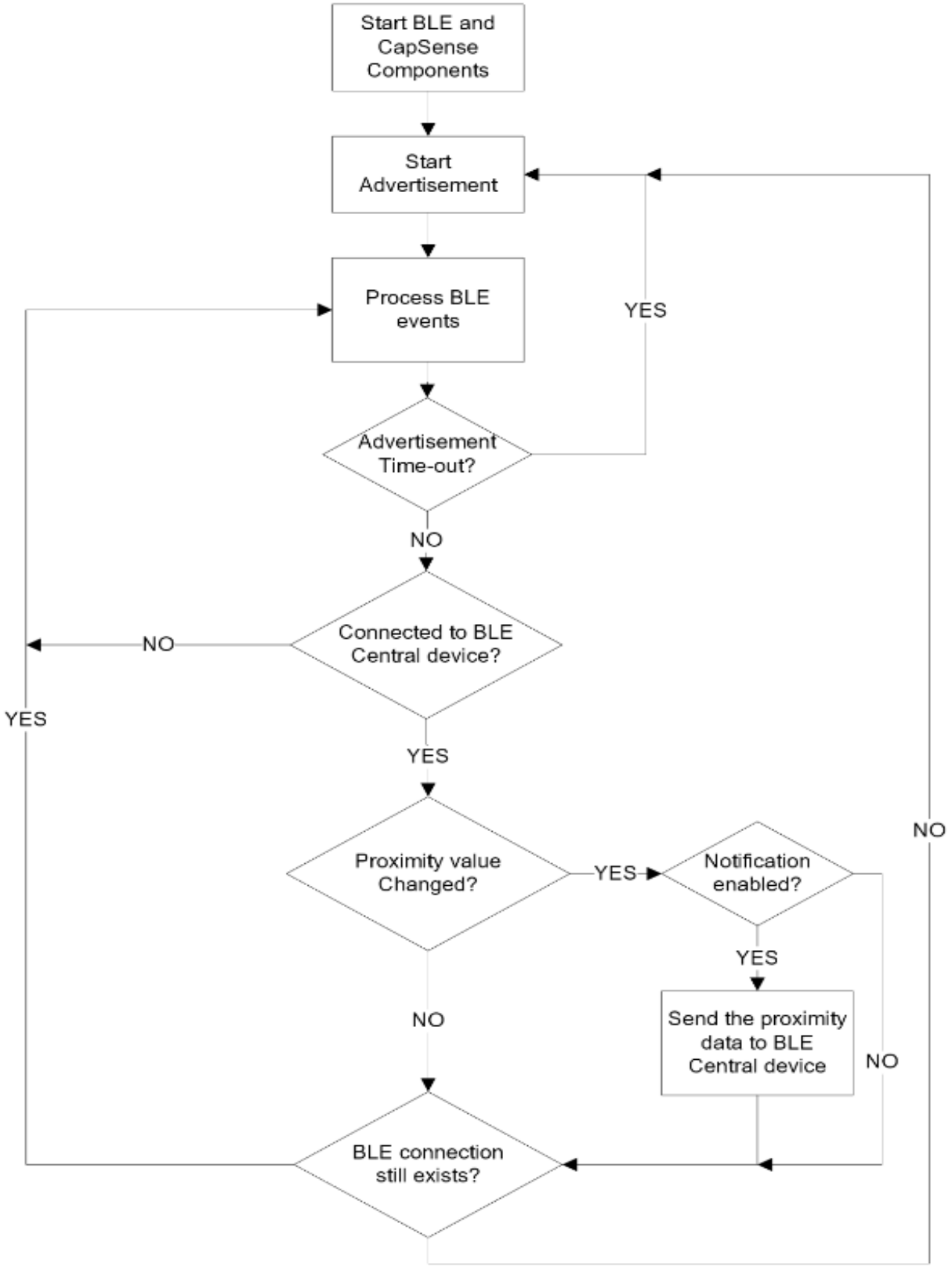
Custom BLE profile implement Capsence service to send one byte's proximity data (range from 0 to 255) to GATT Client device (Cysmart mobile app). The proximity sensor is connected to P2.0 on the kit.

Method:

Hardware Setup:

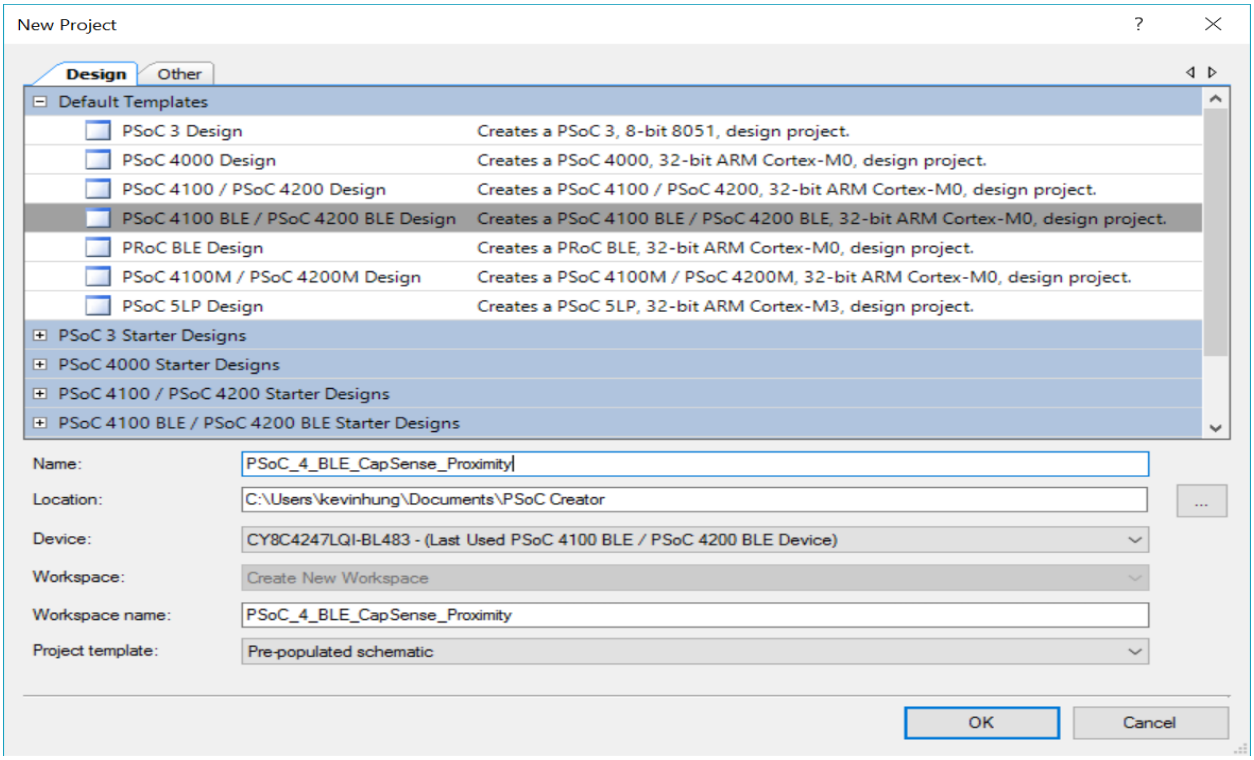


BLE Project Flow Diagram:

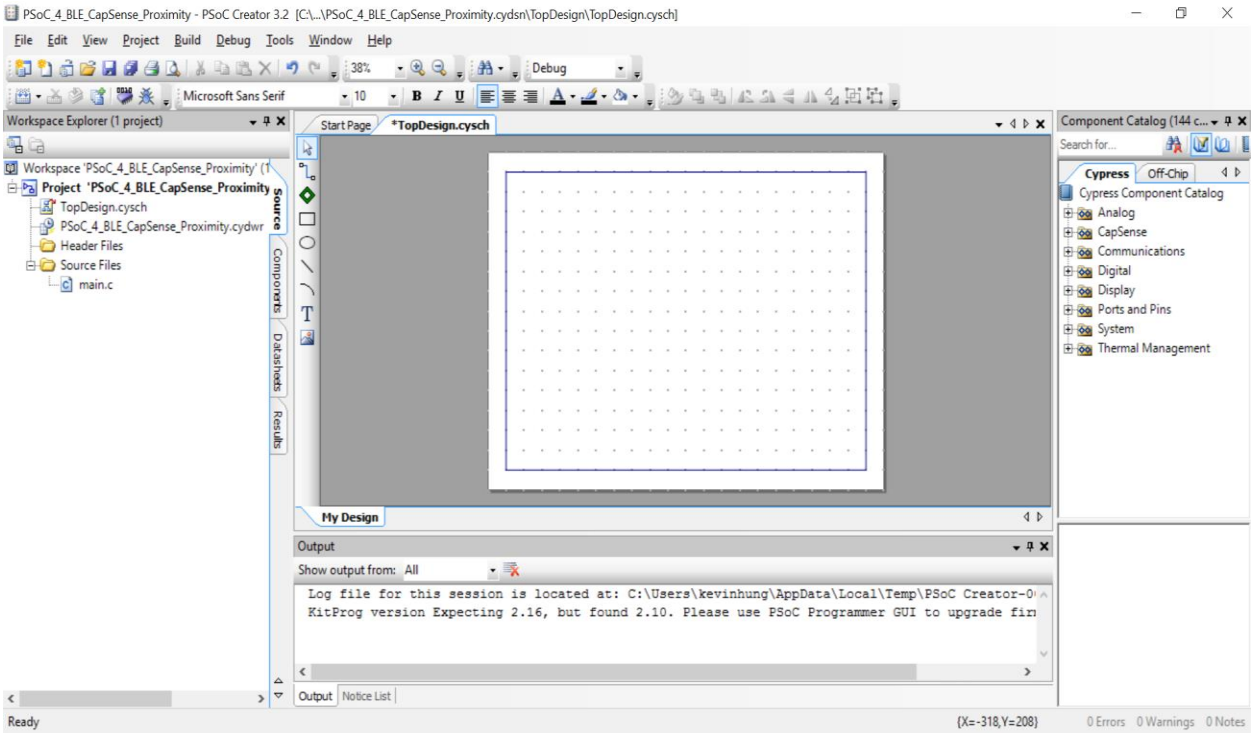


Build Schematic:

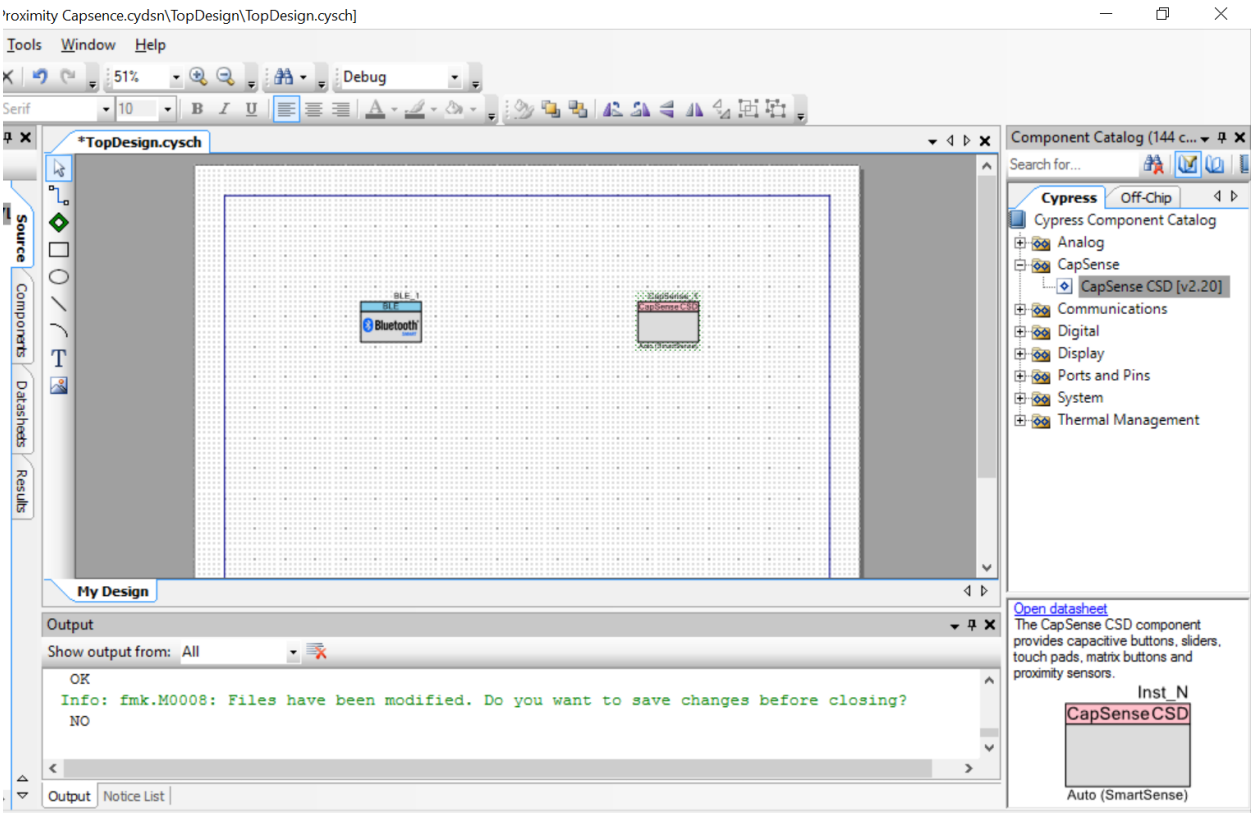
1. New a project name: PSoC_4_BLE_CapSense_Proximity



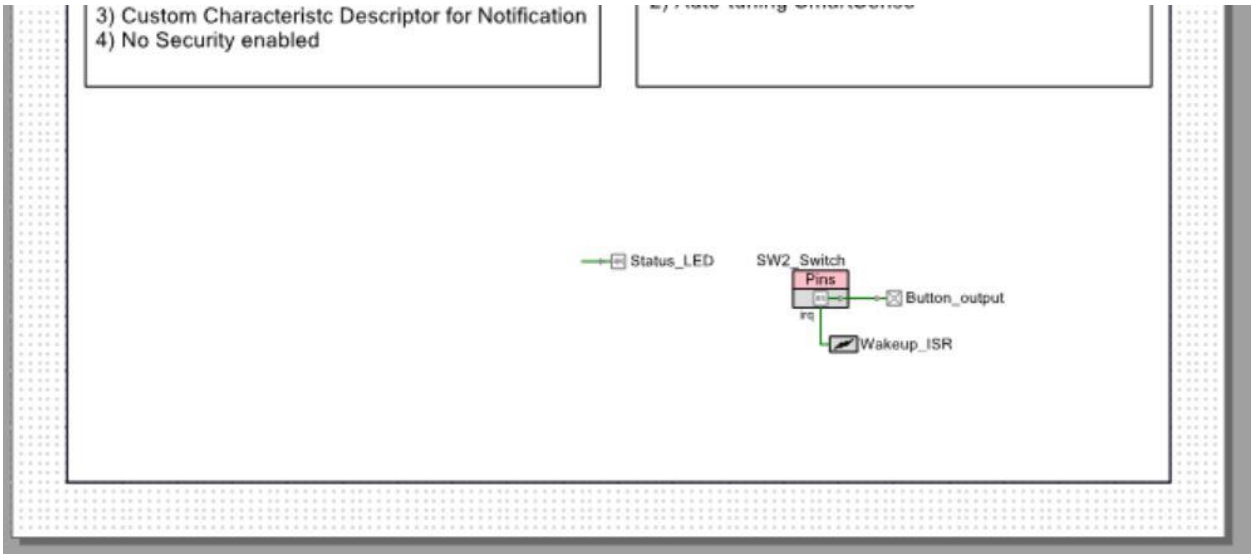
2. Clear the other pages, left Mydesign page.



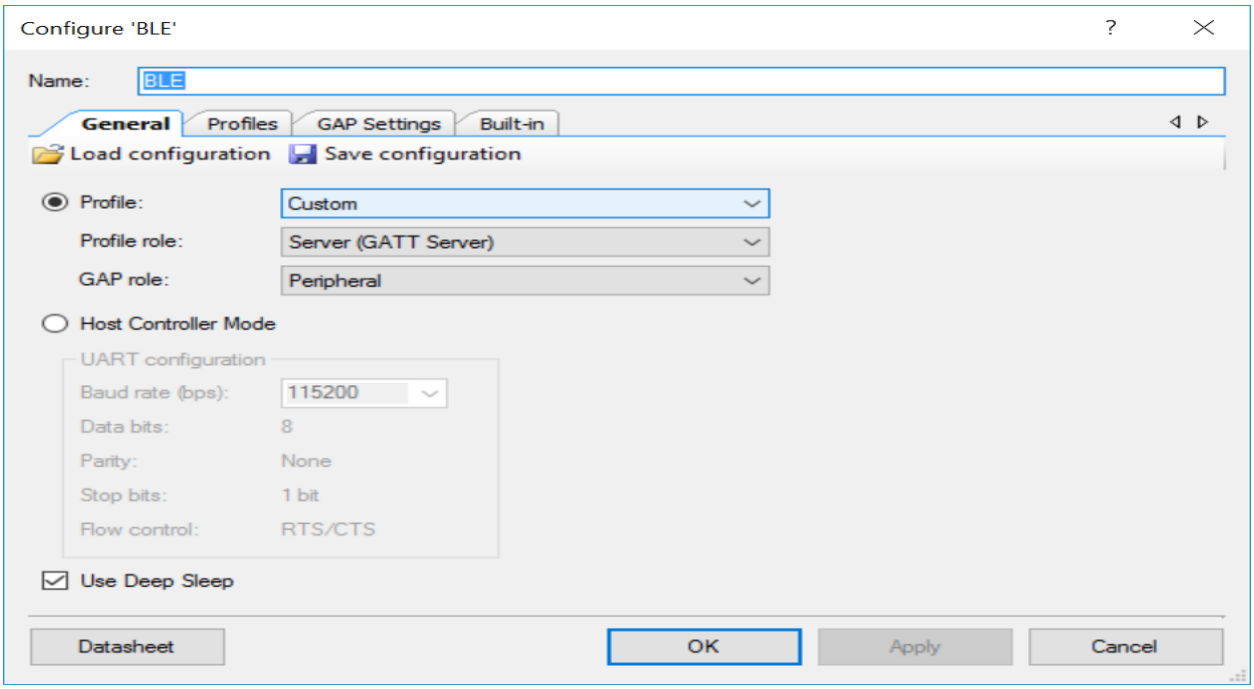
3. Drag BLE and CapSence components to the page:



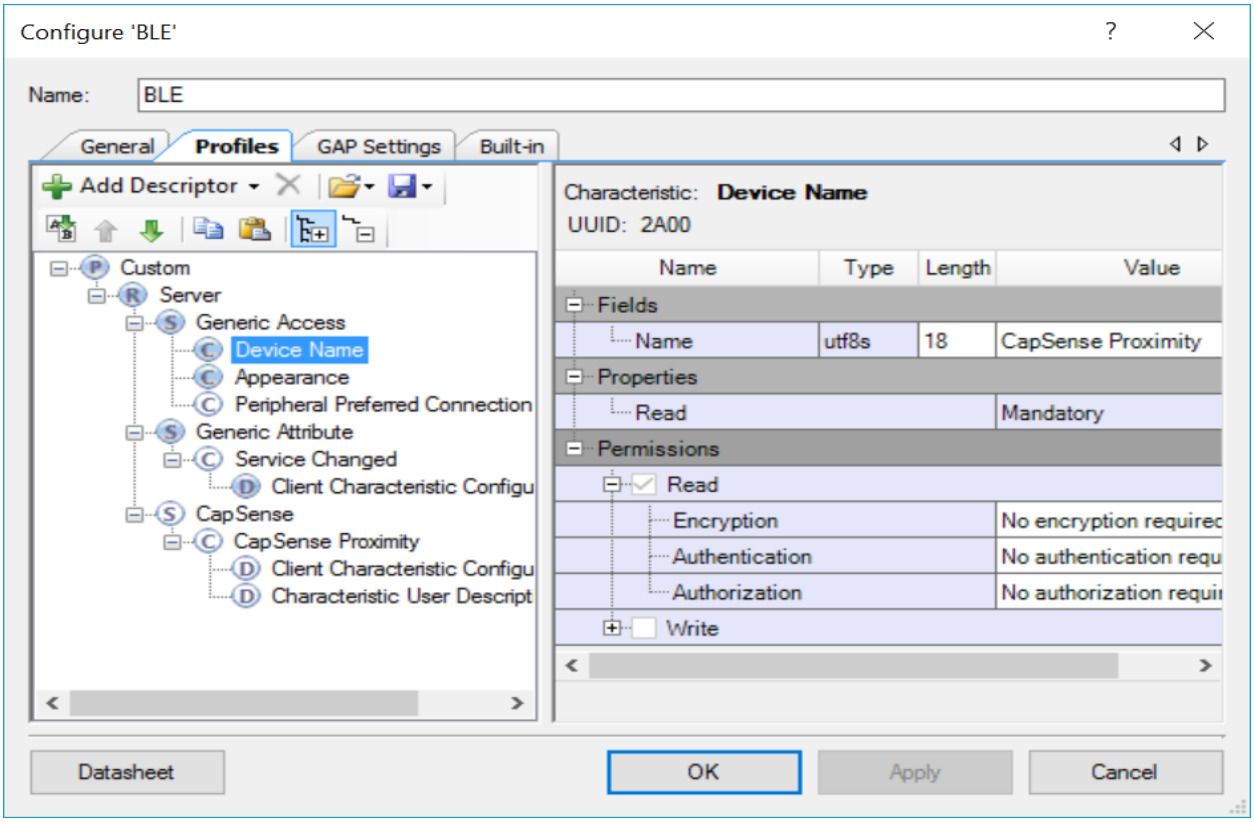
4. Add output pin to become a statue LED for the kit and interrupt switch:



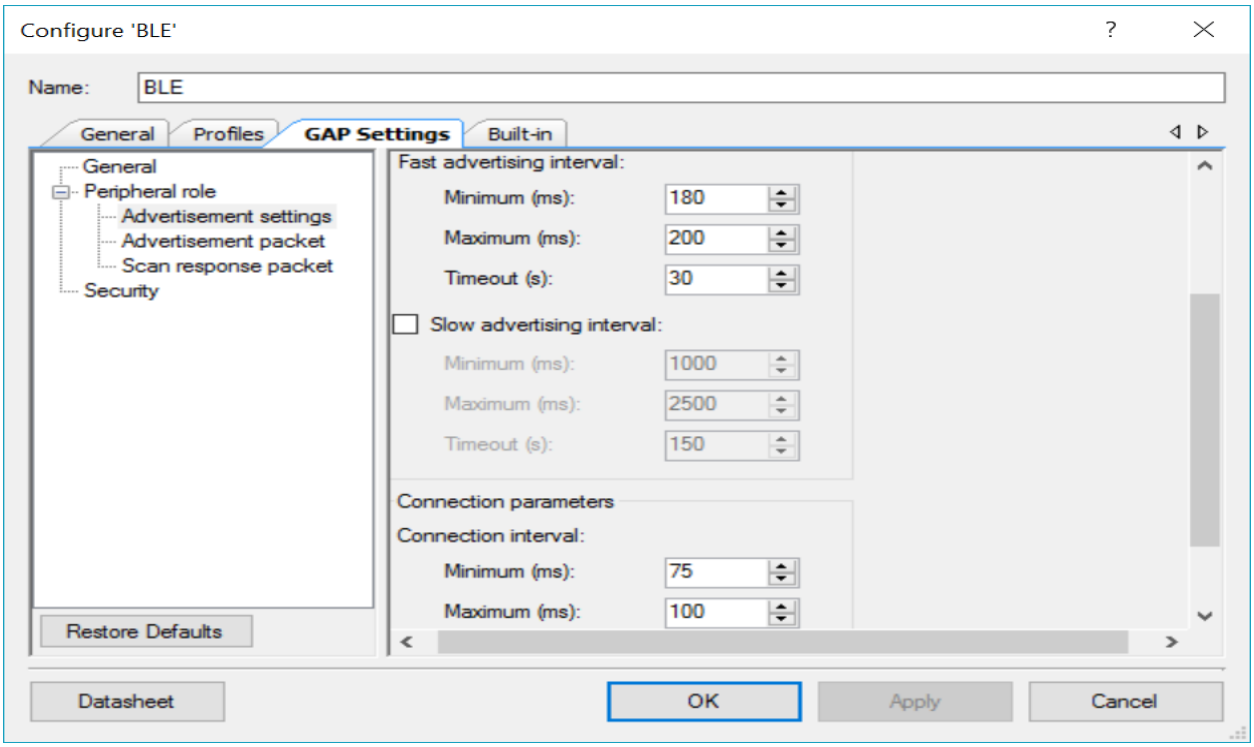
5. Configure BLE component, select custom Profile and profile role act as GATT server:



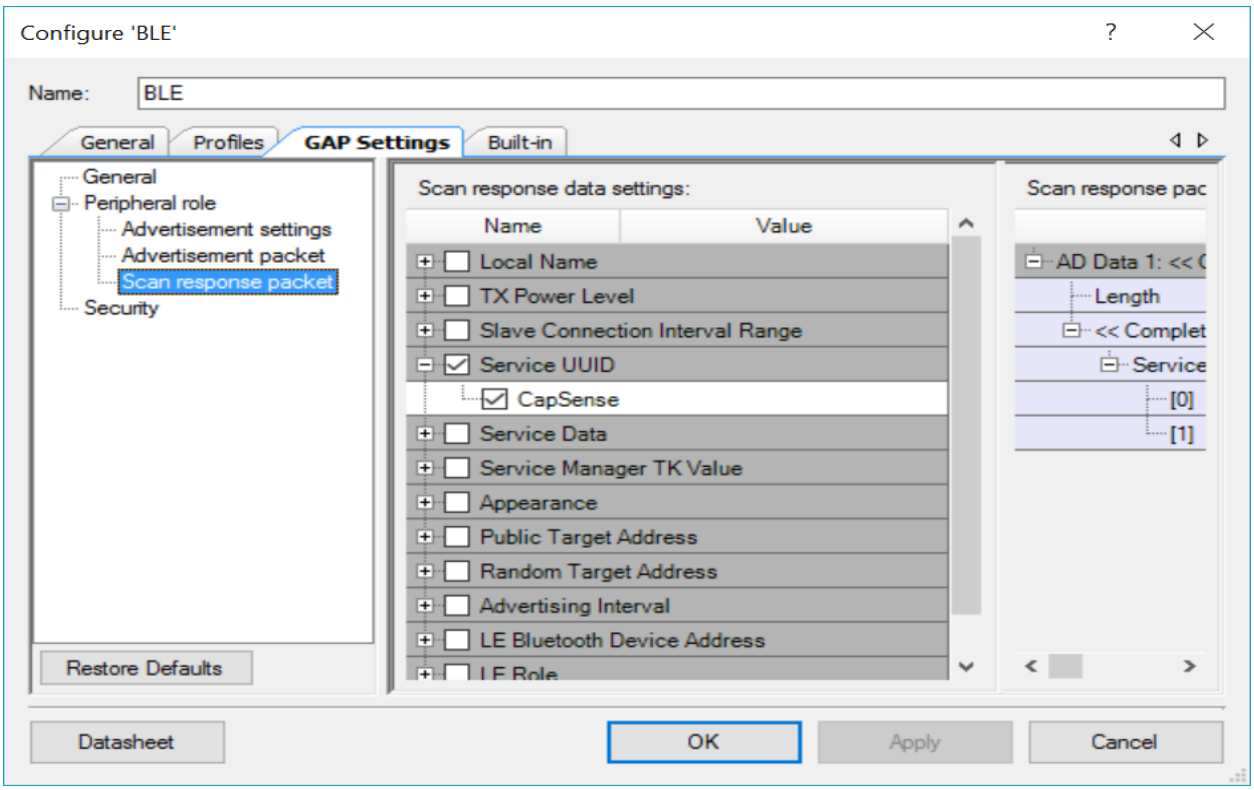
6. Set BLE device' name as CapSence Proximity:



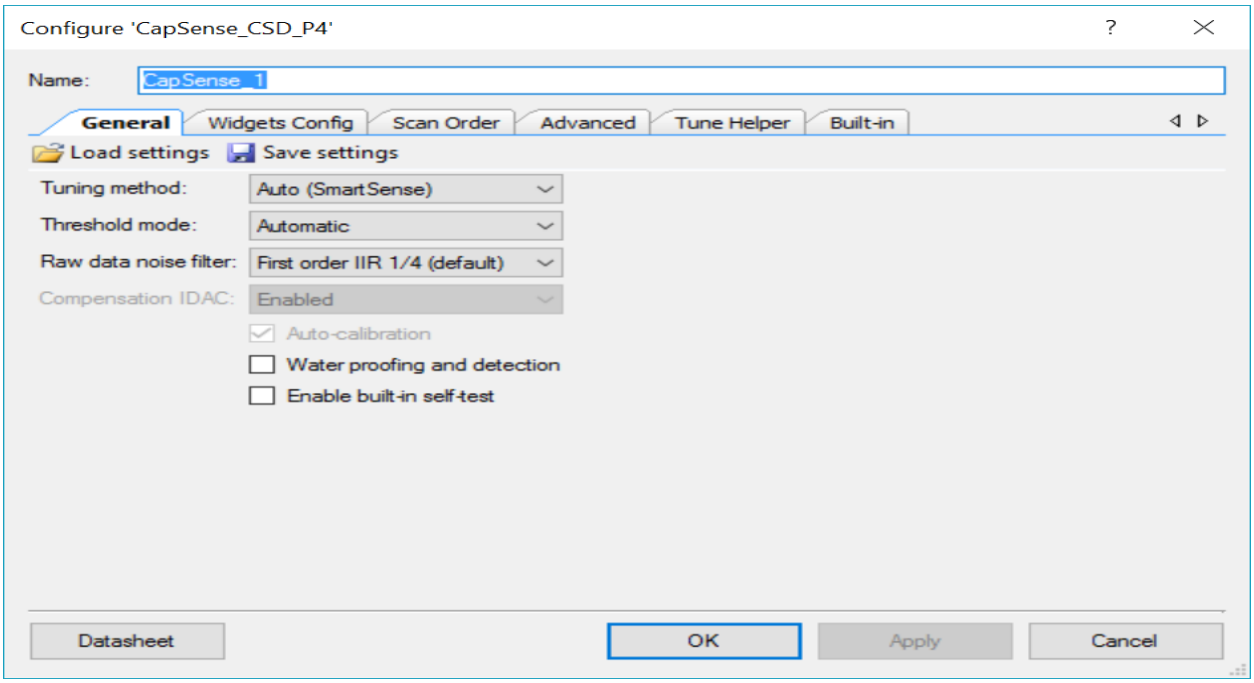
7. Configure GAP Setting in advertisement Setting, set up interval time and timeout:



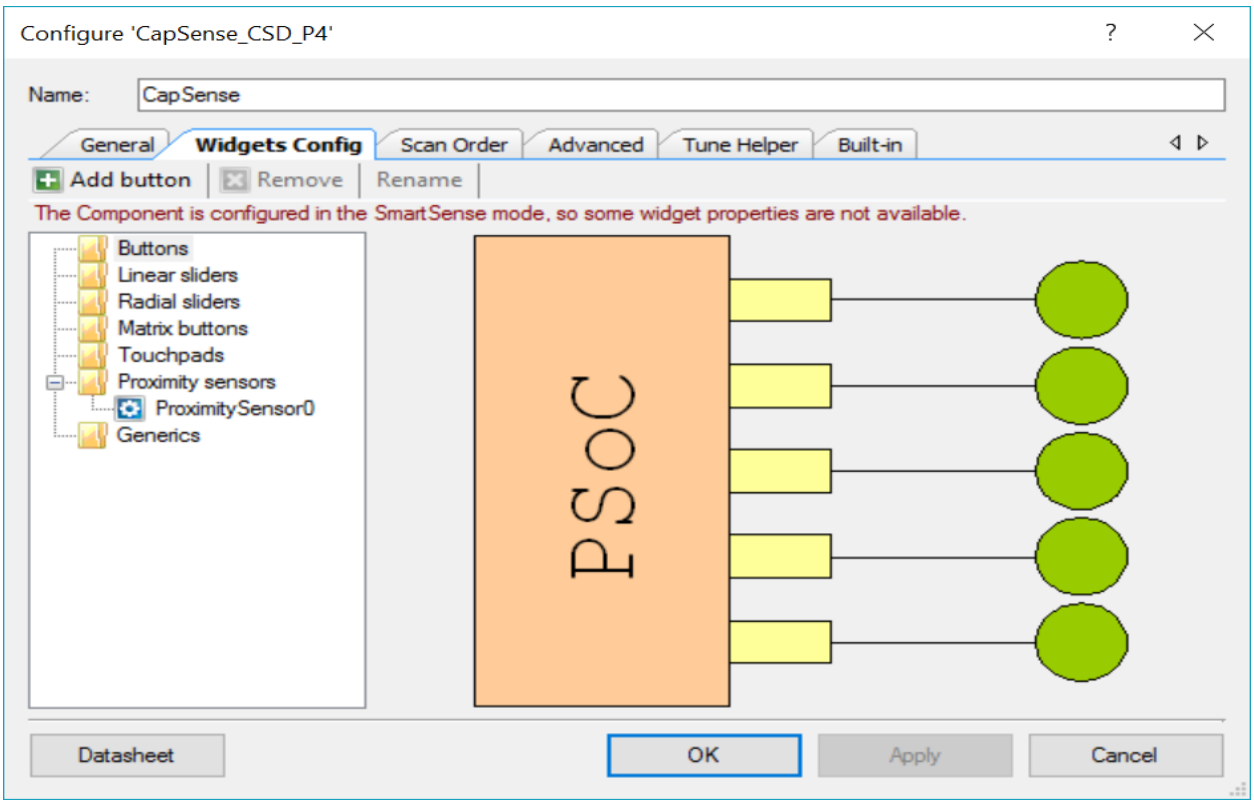
8. Select Scan response packet to CapSence:



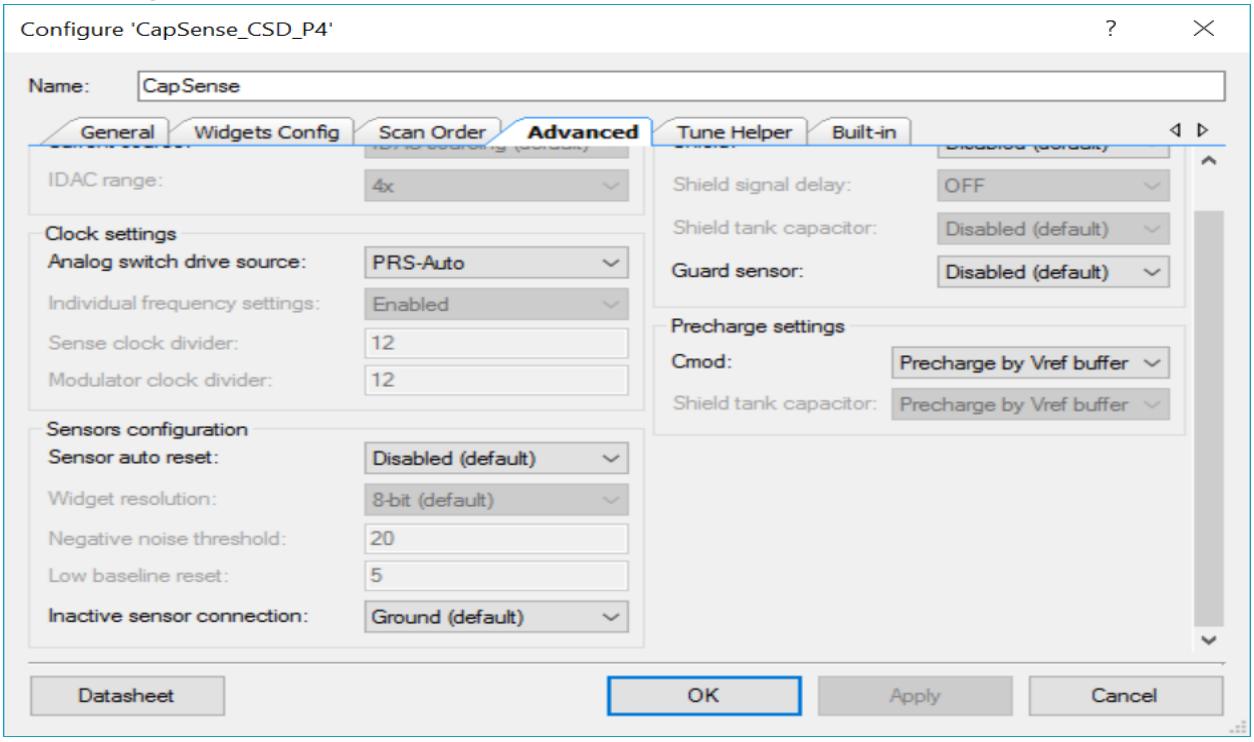
9. Configure CapSence component, rename it to CapSence and select auto turning method:



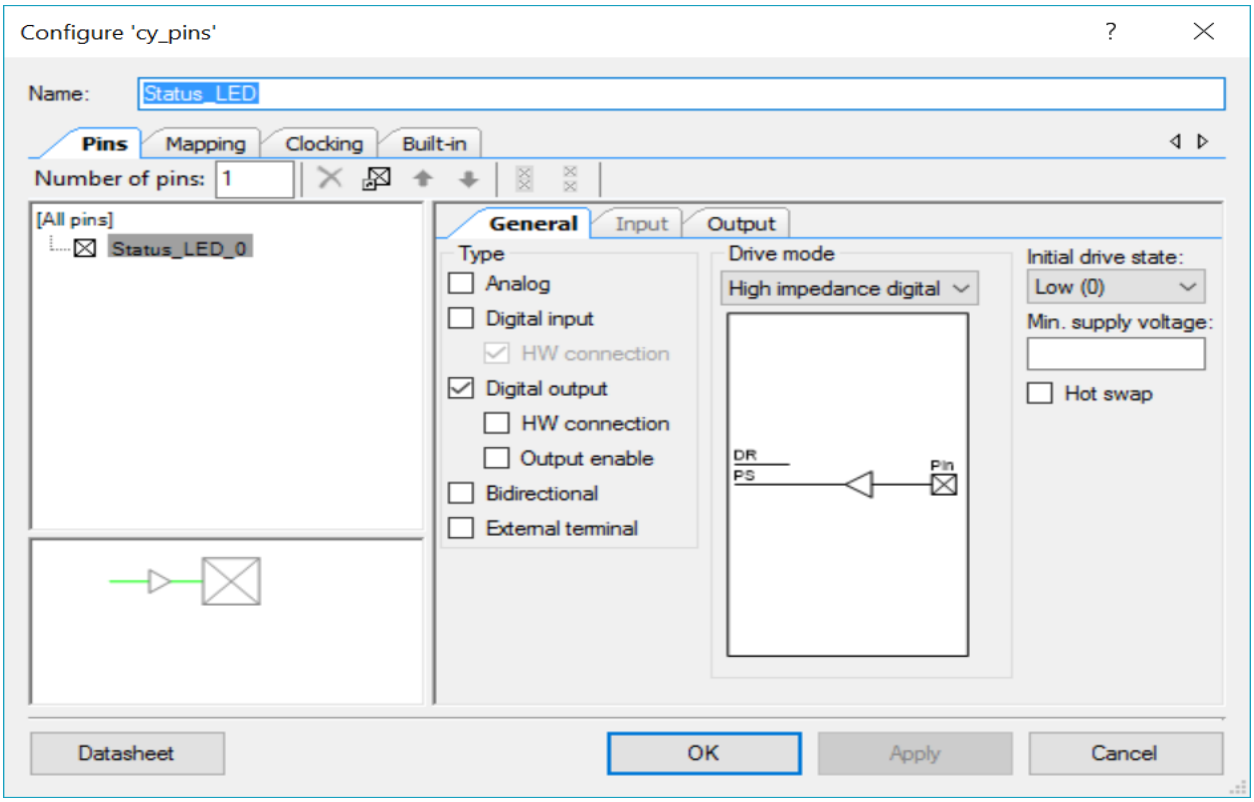
10. Configure widget config in CapSence, add Proximity Sensor() to proximity sensors.



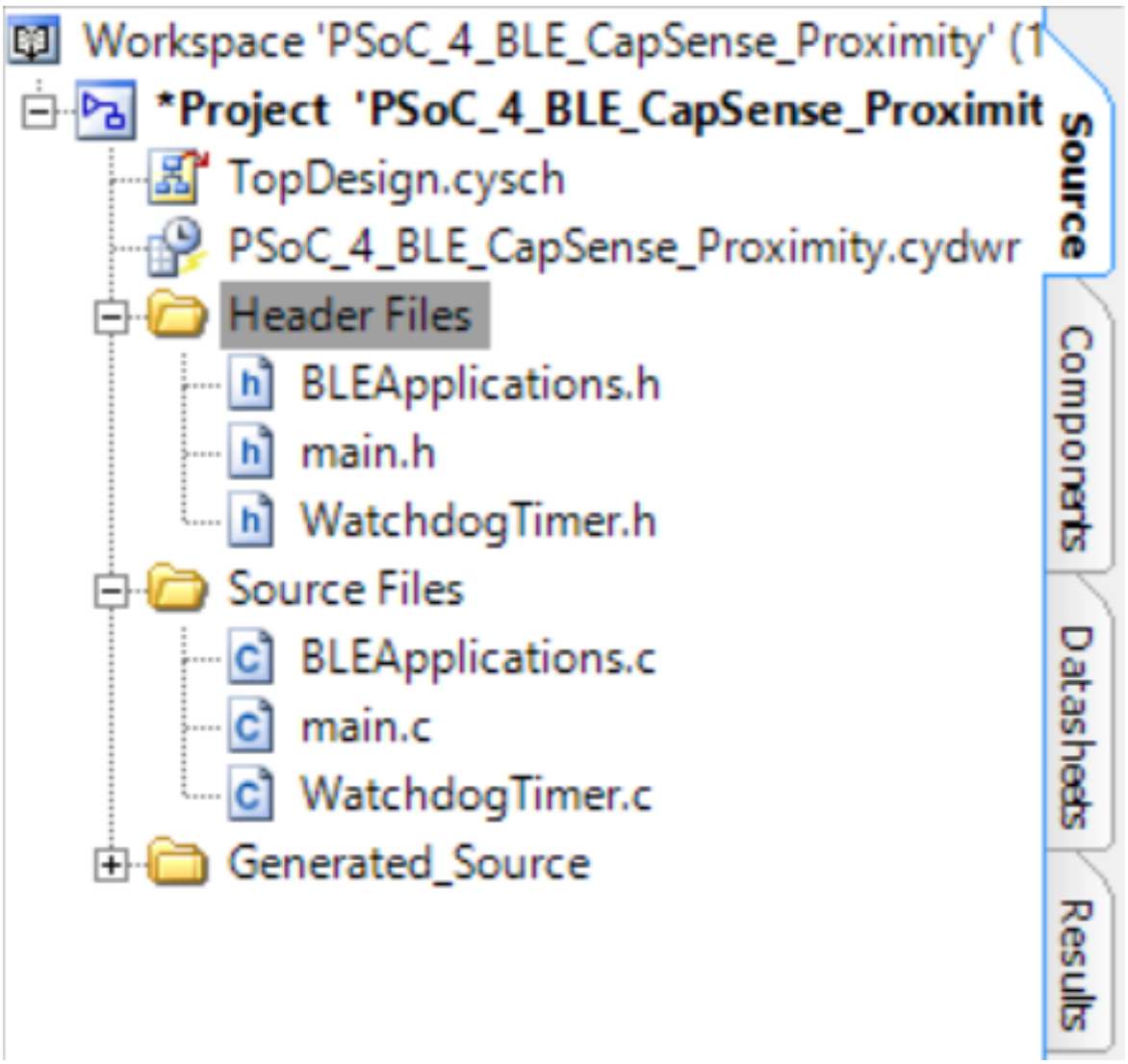
11. Configure advanced tab, select as below:



12. Configure output pin, change the name and select Drive mode to ‘High impedance digital’:



13. Add BLEApplication code and WatchdogTimer code to Source file and header file:



In BLEApplication, the code defines the LED statue.

In main.c, the code initializes the system and implement the LED statue and proximity data transfer.

In WatchdogTimer, the file handles the watchdog timer functionality and keeps track of the system time. It has these functions:

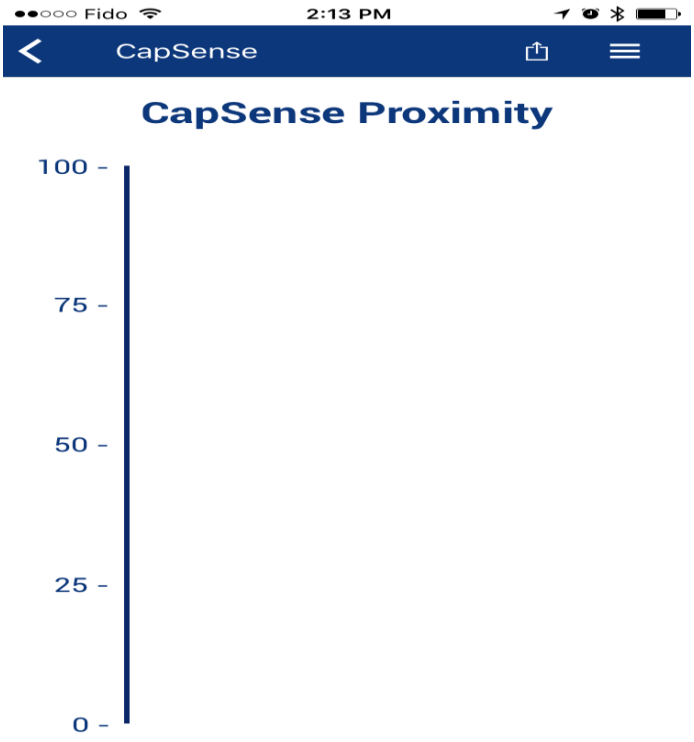
WatchdogTimer_Start() – Starts the watchdog timer (WDT0) with a 10 ms period and interrupt on match.

WatchdogTimer_Isr() – The ISR for the WDT; it increments the system timestamp variable by 10 ms. This function is a Callback from the watchdog timer.

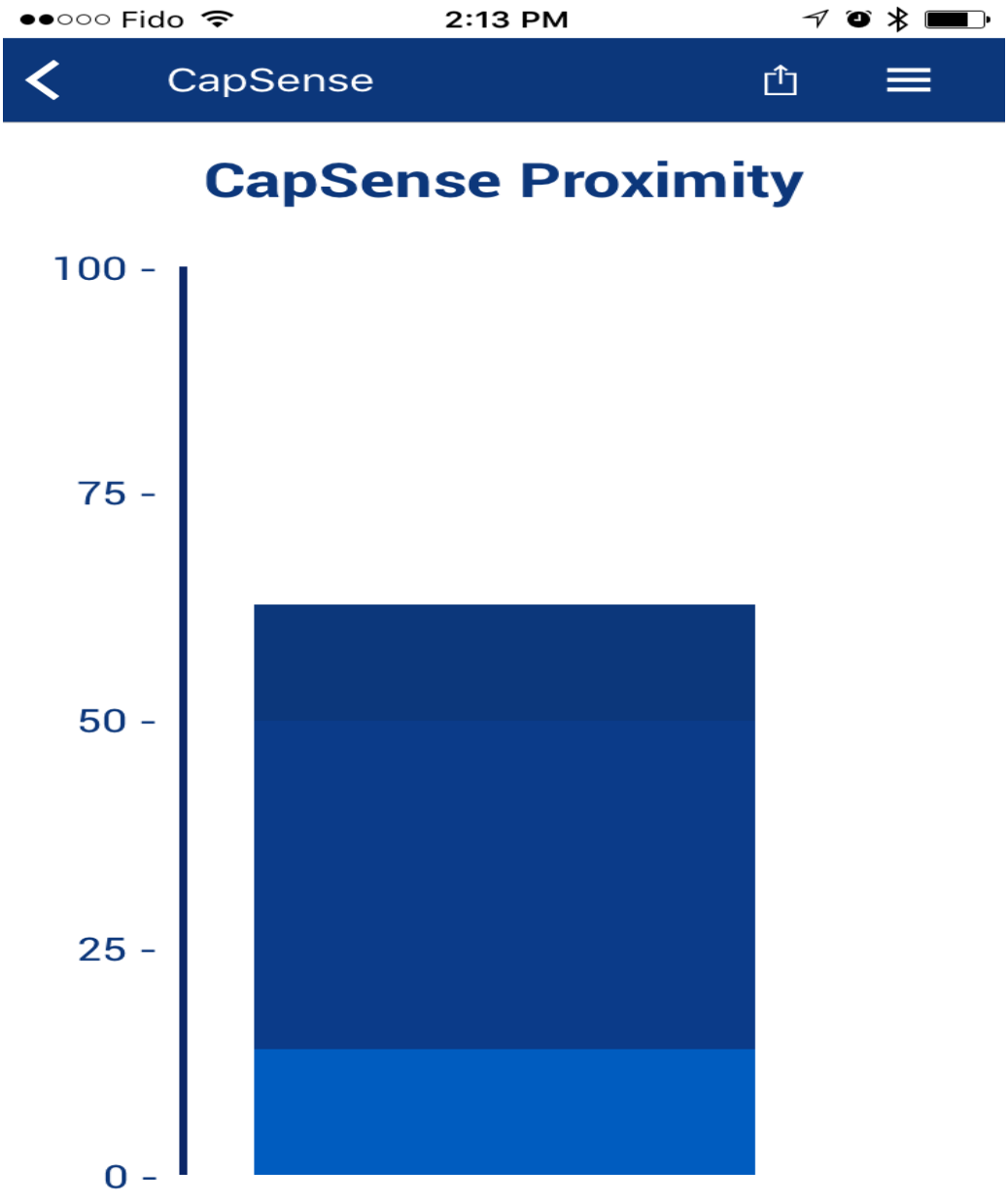
WatchdogTimer_GetTimestamp() – Returns the current timestamp to the caller for any time-keeping purposes of the application.

Test:

1. Open Cysmart app on mobile device, connect ‘CapSence Proximity’ and open the Proximity service:



2. Put the hand on the coil to make the proximity data change, the result is shown on the app:


●●○○○ Fido
2:13 PM
CapSense
CapSense Proximity
100 -
75 -
50 -
25 -
0 -

Conclusion:

In BLE Proximity project, the BLE Pioneer Kit act as both GAP Peripheral and GATT Service device. The kit communicates with mobile app by using Custom BLE Profile which uses CapSence to implement the service.

In every BLE lab, BLE component and ARM Cortex-M0 are the core. The processor can receive and process different types of data to implement different services and communicate with BLE terminal devices by using BLE component.