PROJECT REPORT
CLIENT AND SERVER

Director: Dr. Ahmed Rafaey Hussein

Group Members: Mohammadamin Saburruhmonfared & Chaoao Shi

July 15, 2016

OBJECTIVES

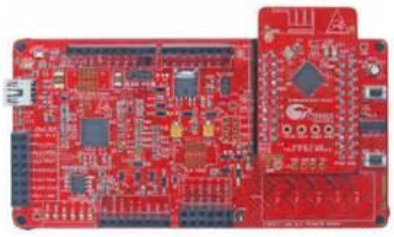
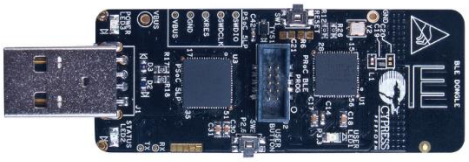
1. Configure Cypress pioneer kit and BLE dongle as both GATT Client and Server;
2. Implement profiles of Device Information Service (DIS) and Battery Level Service;
3. Use UART commands to realize the functions like enabling notifications, disabling notifications, read characteristics etc.;
4. Use the LED to present different communication statuses.

DESCRIPTIONS

In this project, the Cypress Pioneer kit initiates connection to the BLE dongle. After the connection is built up, there are two optional states, which can be switched by SW2 on the kit. When the function of Battery Level Service is activated, the dongle can request the information about the counter from the pioneer kit. When the Device Information Service is activated, the kit has the right to get access to the device information of the dongle.

In this project we want to implement a counter on the kit, which counts from 40 to 1. Since there is no counter service is in a standard, we use the battery level service in order to communicate data between two modules. The LED embedded on the board shows their connection status as well as the status of counter by changing the color of LED and making the LED constant or blinking. The color of the LED is the other parameter which can be changed by sending commands to the kit via UART. In sum, we need to build two projects. One is for the Cypress Pioneer kit and the other is for the BLE dongle.

Table 1. Profiles of Pioneer Kit and BLE Dongle

<table>
<tr><th></th><th colspan="2">Pioneer Kit</th><th colspan="2">BLE Dongle</th></tr>
<tr><td>GAP role</td><td colspan="2">Peripheral</td><td colspan="2">Central</td></tr>
<tr><td>GATT role</td><td colspan="2">Client & Server
Client for Device Information
Server for Battery Level</td><td colspan="2">Client & Server
Server for Device Information
Client for Battery Level</td></tr>
<tr><td rowspan="13">Command</td><td>manufacturer name</td><td>1</td><td rowspan="4">read the battery level</td><td rowspan="4">r</td></tr>
<tr><td>model number</td><td>2</td></tr>
<tr><td>serial number</td><td>3</td></tr>
<tr><td>hardware Rev</td><td>4</td></tr>
<tr><td>firmware Rev</td><td>5</td><td rowspan="4">enable the notification on the GATT server</td><td rowspan="4">e</td></tr>
<tr><td>software Rev</td><td>6</td></tr>
<tr><td>system ID</td><td>7</td></tr>
<tr><td>IEEE 11073-20601</td><td>8</td></tr>
<tr><td>PNP-ID characteristics</td><td>9</td><td rowspan="4">disable the notification on the GATT server</td><td rowspan="4">d</td></tr>
<tr><td>Charge Battery</td><td>q</td></tr>
<tr><td>Battery Level Notification</td><td>n</td></tr>
<tr><td>Change Color</td><td>+</td></tr>
<tr><td>Objective</td><td colspan="2">  </td><td colspan="2">  </td></tr>
</table>

CENTRAL PART

The BLE dongle plays the role of GAP central part.

1. Profiles

- Server for Device Information

The server profile is included in a standard service called device information service which exposes manufacturer and/or vendor information about a device. In this function, when the client requests the device information of the server, the server would provide 9 characteristics: Manufacturer Name, model number, serial number, hardware Rev, firmware Rev, software Rev, system ID, IEEE 11073-20601, and PNP-ID characteristics.

- Client for Battery Level

The client profile is included in a standard service named battery service. As a client, the dongle could request the Battery Level information from the server, which is the Cypress Pioneer kit in this case. The functions appointed to the client for Battery Level Service are listed below.

Table 2. Functions for Battery Level Client

Functions	Descriptions
CyBle_BascSetCharacteristicValue	This function is used to read the Characteristic value from a server which is identified by charIndex.
CyBle_BascSetCharacteristicDescriptor	Sends a request to set Characteristic Descriptor of specified Battery Service Characteristic on the server device.
CyBle_BascGetCharacteristicDescriptor	Sends a request to get Characteristic Descriptor of specified Battery Service Characteristic from the server device.

As we mentioned, we replace the counter information with power level information.

2. Hardware

- Schematics

Fig. 1 shows the PSoC creator schematics of "Client and Server-Central". It includes a BLE and a UART. BLE is used to exchange data between two devices, and UART is used to send and operate commands for the Device Information Service and Battery Level Service.

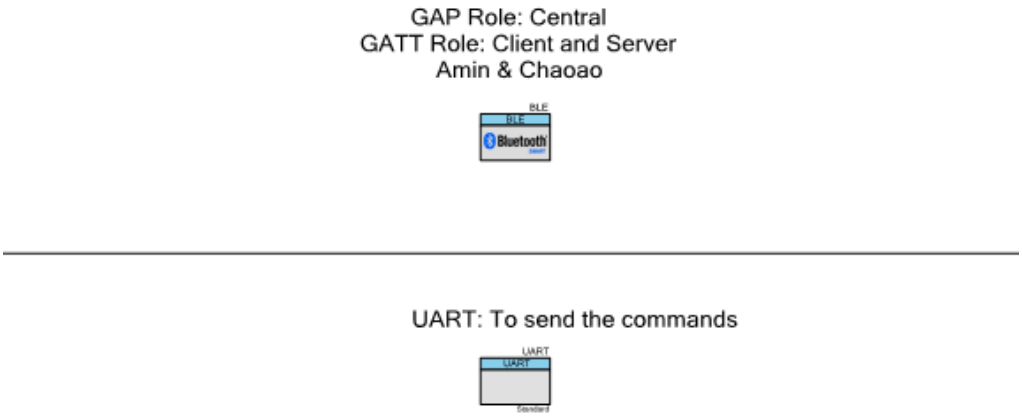


Figure 1. Schematic of the central device

- BLE Setting-General tag

As the central device, the dongle is set as the GAP Central and GATT Client and Server. The GAP Central means it accepts connection. The GATT client configuration for battery level service allows it to request the battery status from the Cypress Pioneer kit. Moreover, the GATT server configuration for DIS requires it to execute the request from the Cypress Pioneer kit about its device information.

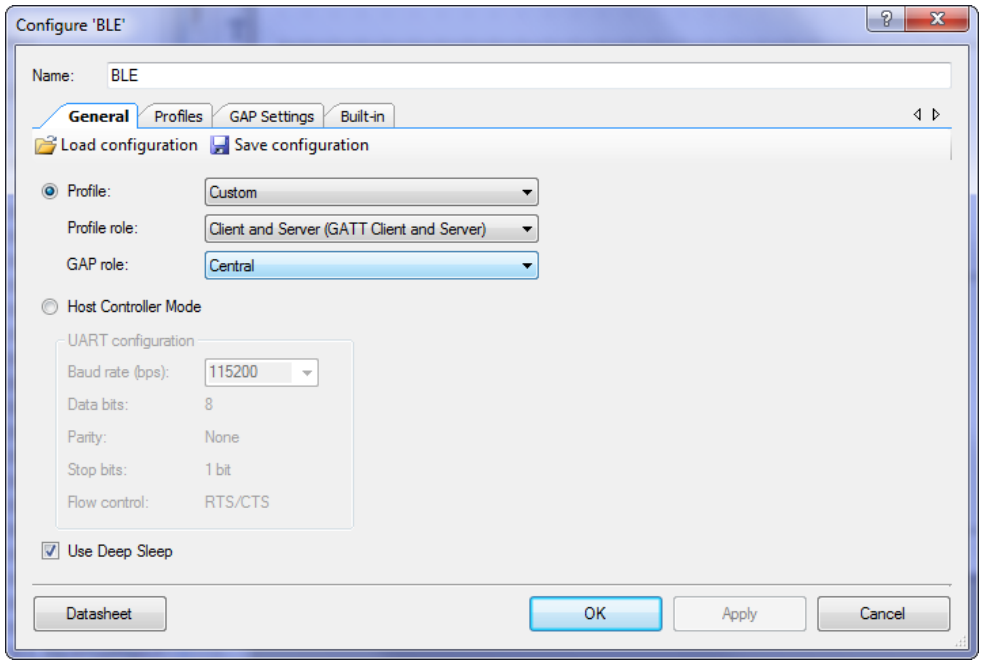


Figure 2. BLE Setting-General Tag

- BLE Setting-Profiles tag

The dongle plays the role of GATT server for DIS. Therefore, we need to add a function named Device Information. Besides, some characteristics can be modified in the **Profiles Tag** of the BLE component.

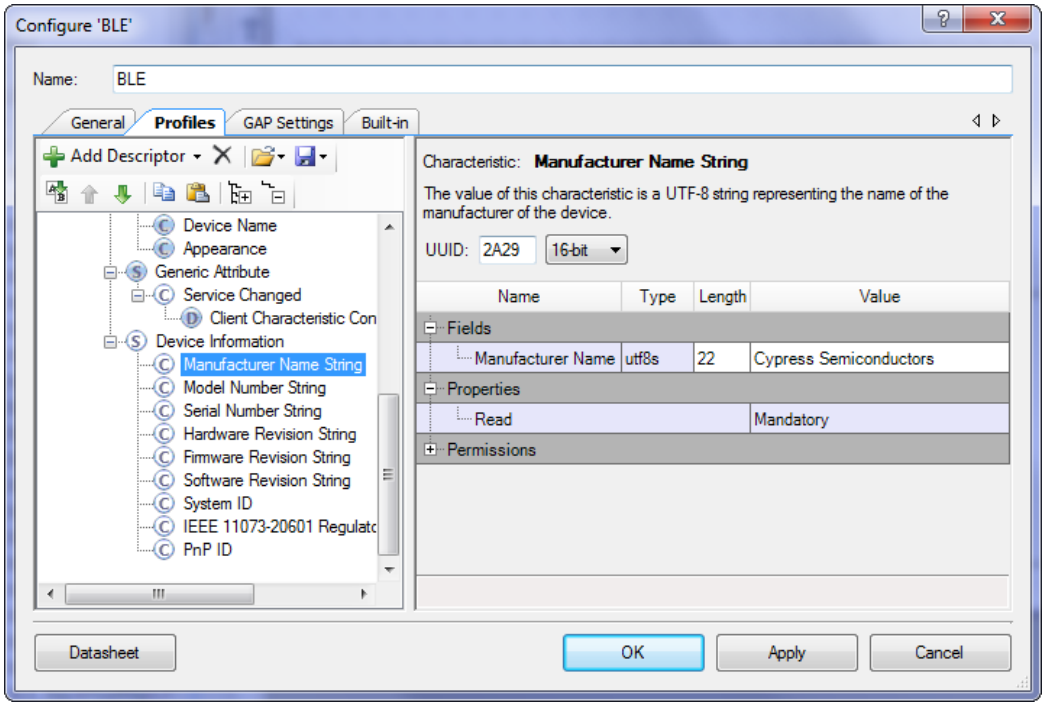


Figure 3. BLE Setting-Profiles Tag

- UART

The UART component is configured to send commands by the serial port of the PC and receive the information from the Cypress pioneer kit. Fig. 4 shows the configuration of UART. The baud rate is selected to be 115200 and the data bits should be 8 bits. The oversampling rate should be set as 12.

In addition, a Hyper Terminal is used in this project to write the command or show the information.

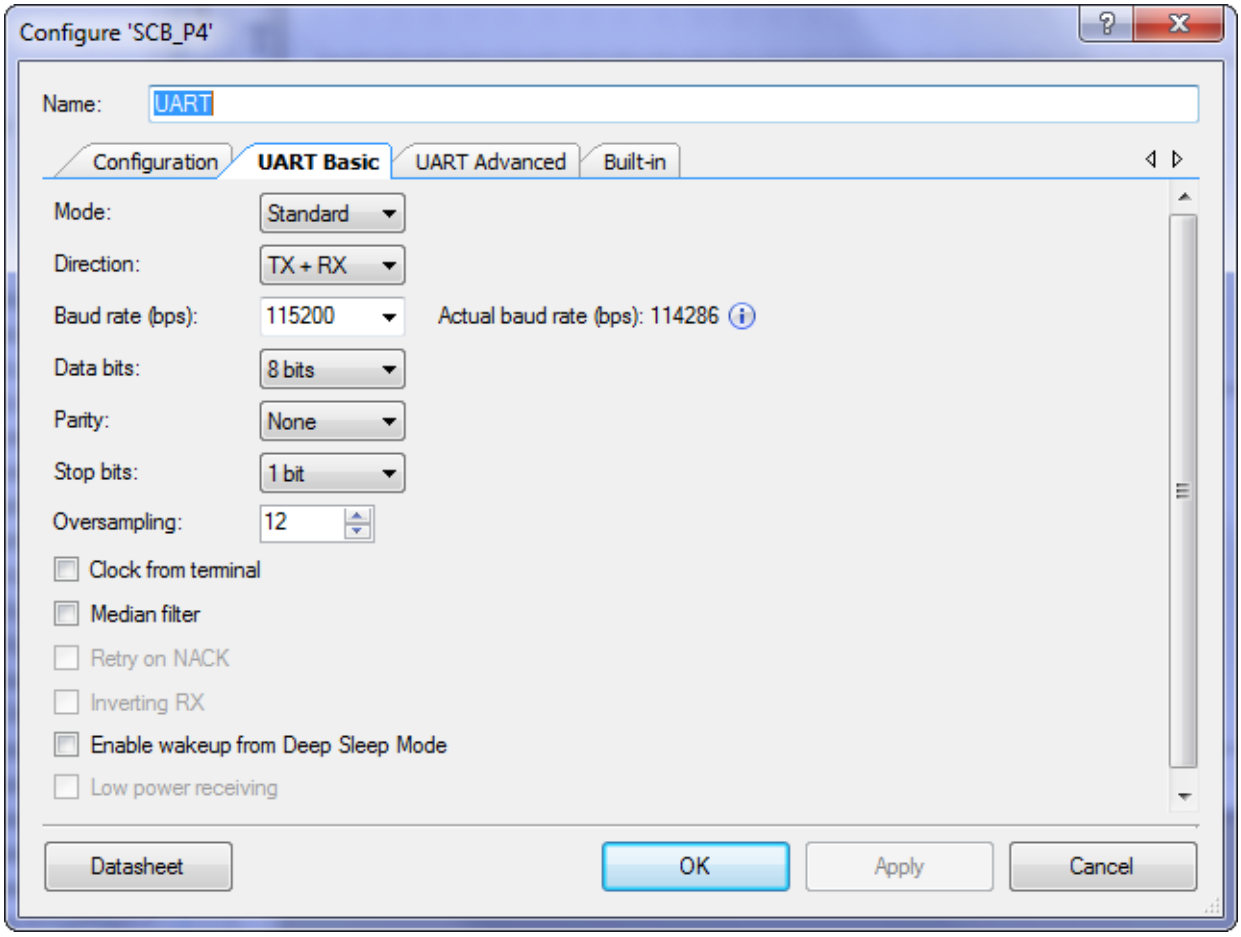


Figure 4. UART Setting

3. Firmware

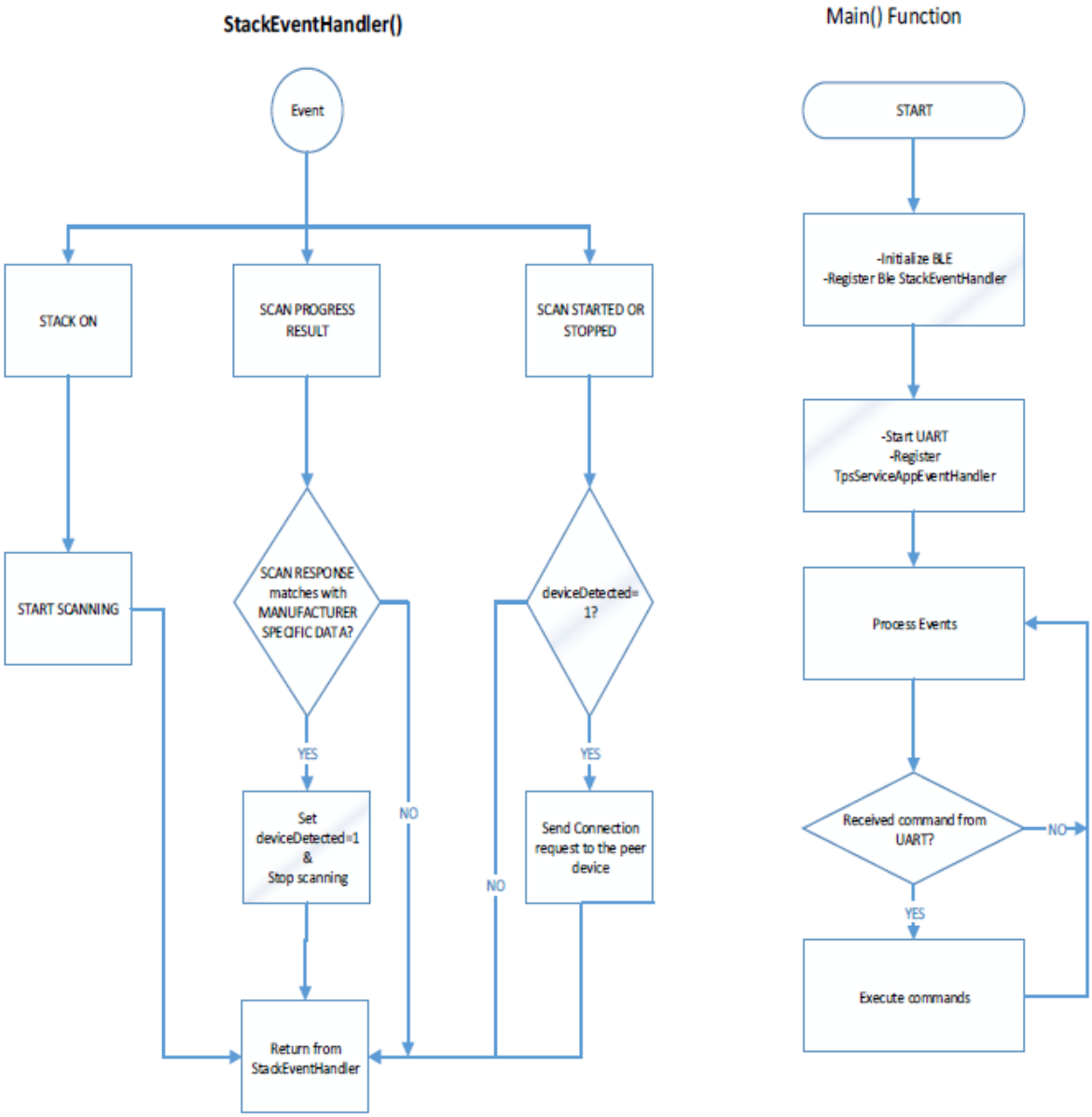


Figure 5. Flowchart of the firmware for Central

The firmware includes three major files: bas.c, common,c and main.c. They complete the functions of UART communication, Device Information Service, and Battery Level Service. It is further discussed as follows.

- Main Function

The main function contains the initialization of the components, registering the event handler for BAS service, and check the commands from UART and execute them. Commands from UART terminal for the central part can be found in Table 1.

- StackEventHandler () function

When a BLE component is started, this function is used to handle the common events generated for the BLE Stack.

Table 3. Functions in StackEventHandler for the Central Part

Functions	Descriptions
CYBLE_EVT_STACK_ON	This event happens when BLE is initialized and turned on.
CYBLE_EVT_GAP_DEVICE_CONNECTED	This event happens when connection of a central device to a peer device is completed.
CYBLE_EVT_GAP_DEVICE_DISCONNECTED	This event happens when central device is disconnected from peer device and starts to scan again.
CYBLE_EVT_GATTC_DISCOVERY_COMPLETE	This event happens when discovery of the peer service is completed. Here the discovered service is BAS.

PERIPHERAL PART

The Cypress Pioneer kit plays the role of GAP peripheral.

1. Profiles

- Server for Battery Level

The server profile is included in a standard service called Battery Level service. The Battery Service exposes the battery level of a single battery or set of batteries in a device. Depending on the chosen GATT role in the GUI, we can use a subset of the supported APIs. The BAS API names begin with CyBle_Bas. The functions appointed to the server for Battery Level Service are listed below.

Table 4. Functions for Battery Level Server

Functions	Descriptions
CyBle_BassSetCharacteristicValue	Sets a Characteristic value of the service in the local database.
CyBle_BassGetCharacteristicValue	Gets a Characteristic value of the Battery service, which is identified by charIndex.
CyBle_BassGetCharacteristicDescriptor	Gets a Characteristic Descriptor of a specified Characteristic of the Battery service from the local GATT database.

CyBle_BassSendNotification	This function updates the value of the Battery Level Characteristic in the GATT database. If the client has configured a notification on the Battery.

- Client for Device Information

The Device Information Service exposes manufacturer and/or vendor information about a device. The client can use appointed functions to read the characteristic values from a server which is identified by char index. The response returns the characteristic value in the Attribute Value parameter.

2. Hardware

- Schematics

Fig. 6 shows the PSoC creator schematics of "Client and Server-Peripheral. It includes a BLE, a UART, a counter, a switcher and some output pins. BLE is used to exchange data between two devices, and UART is used to send and operate commands for the Device Information Service and Battery Level Service. The switch component is used to connect an interrupt to the switch. Thus, when the switch is pressed an interrupt routine is executed and the device start to search for the services of the server. It helps to switch the GATT profile between Device Information Service and Battery Level Service.

In addition, we use a timer to simulate the battery level. When the timer counts one period, an interrupt event happens and the counter decreases in the interrupt routine. Those output pins are used to control the RGB LED embedded on the board. It helps to demonstrate different communication status. The pinout of the device is shown in Fig. 7. In continue the configuration of each component is explained.

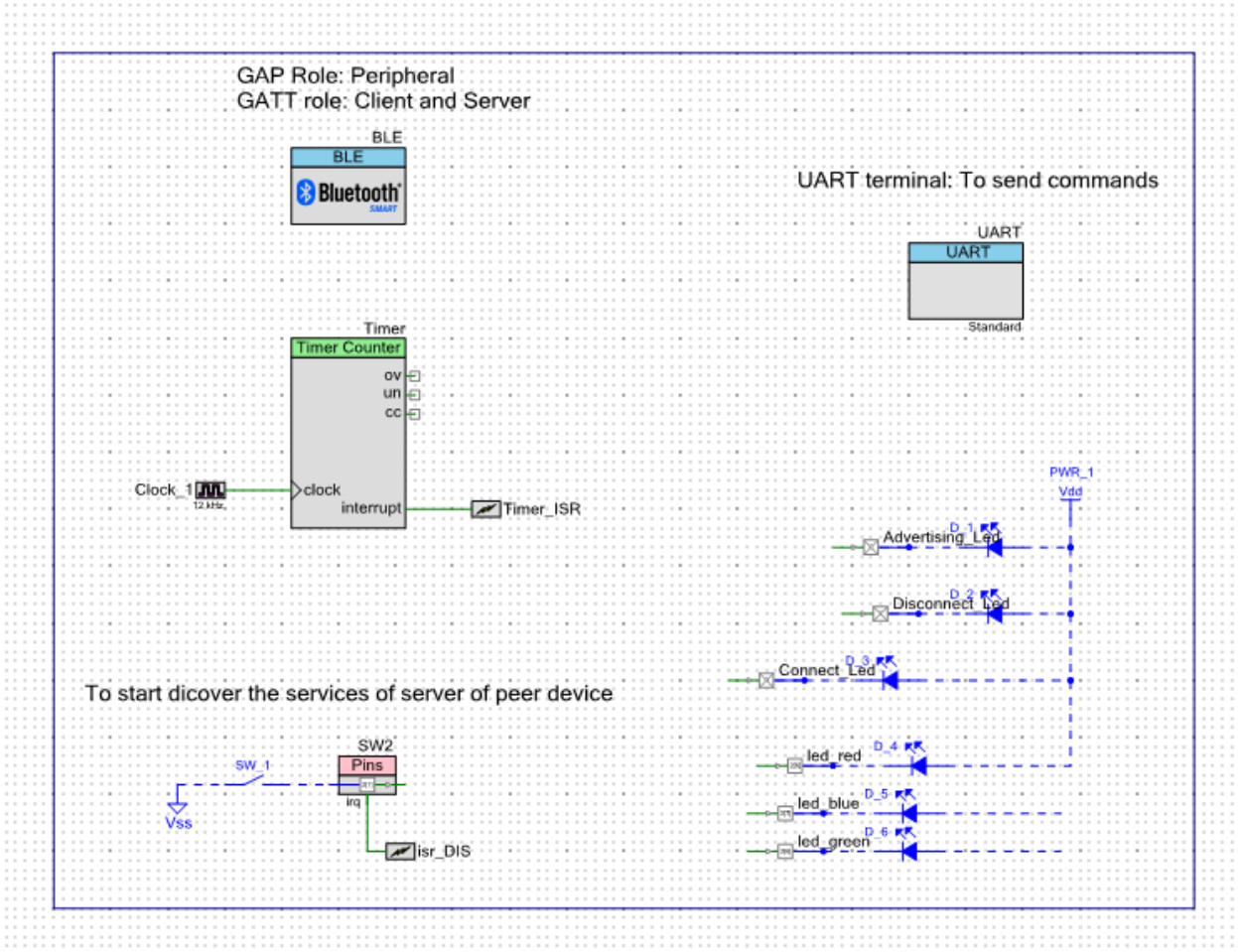


Figure 6. Schematic of the peripheral device

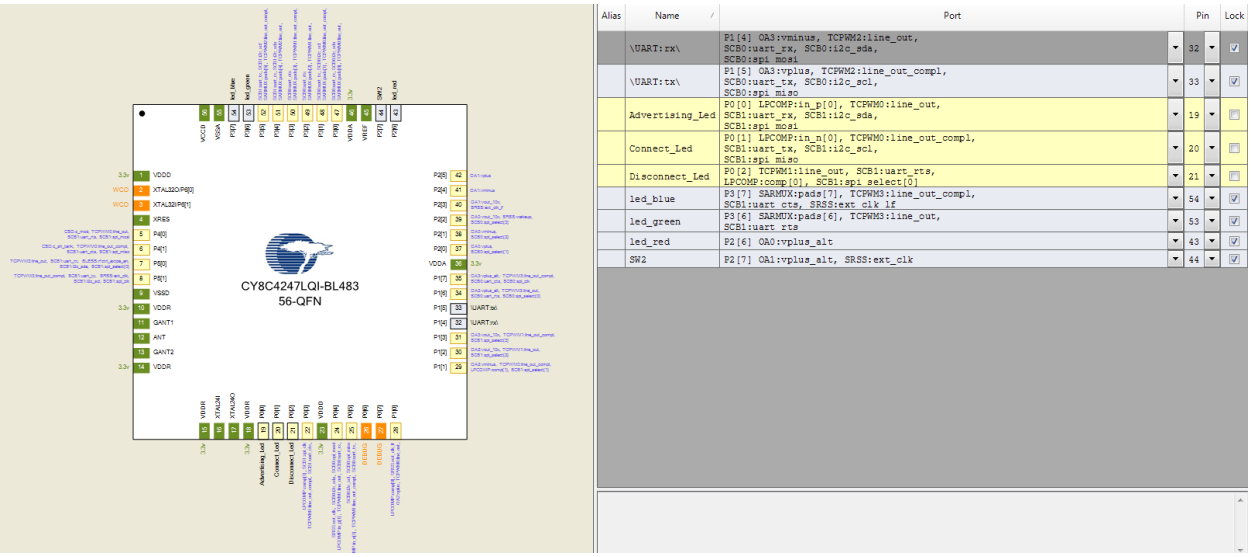


Figure 7. Pinout assignment for the peripheral device

- BLE Setting-General tag

As the Peripheral device, the kit is set as the GAP Peripheral and GATT Client and Server. The GAP Central means it initiates connection. On one hand, the GATT server configuration for battery level service requires it to report the status of counter to BLE dongle. On the other hand, the GATT client configuration for DIS allows it to request from the dongle about its device information.

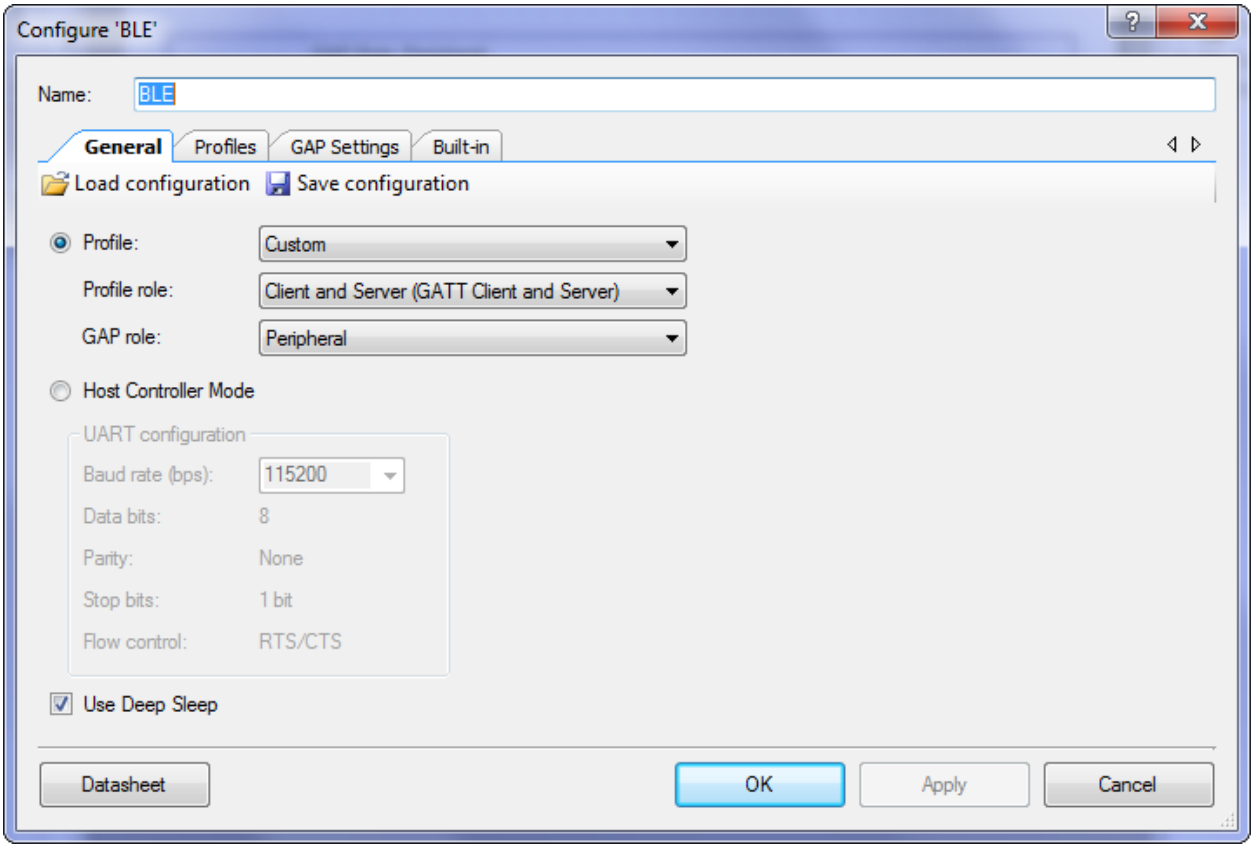


Figure 8. BLE Setting-General Tag

- BLE Setting-Profiles tag

The dongle plays the role in DIS and BAS. Therefore, we need to add functions named Device Information and Battery Level. The Characteristic Presentation Format should be deleted for this project. Besides, some characteristics can be modified in the **Profiles Tag** of the BLE component.

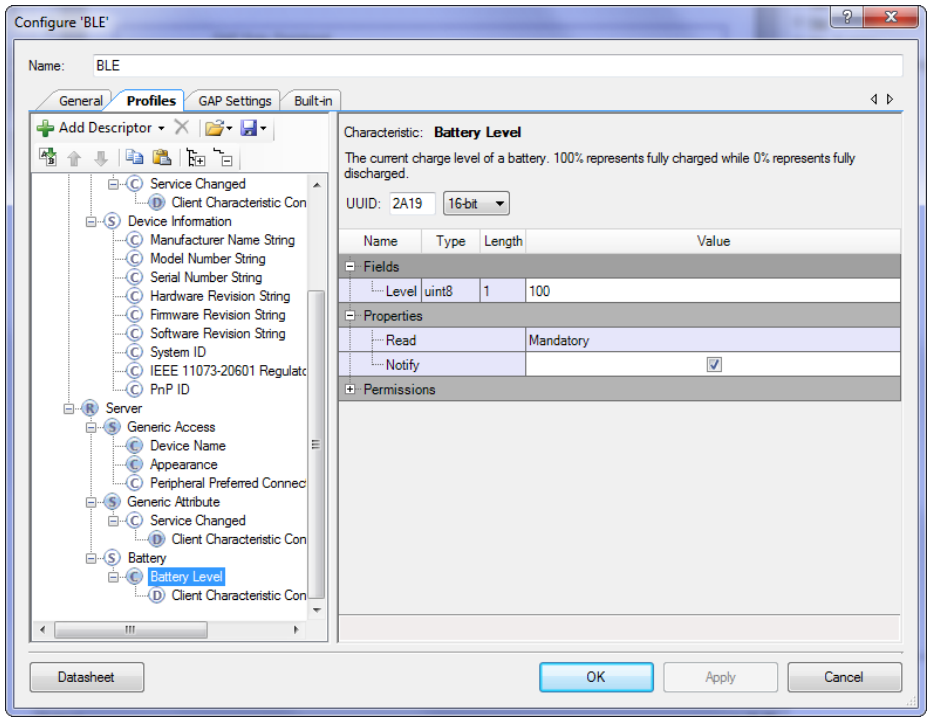


Figure 9. BLE Setting-Profiles Tag

- UART

The UART component should configure as the both transmitter and the receiver. Please see Fig. 10

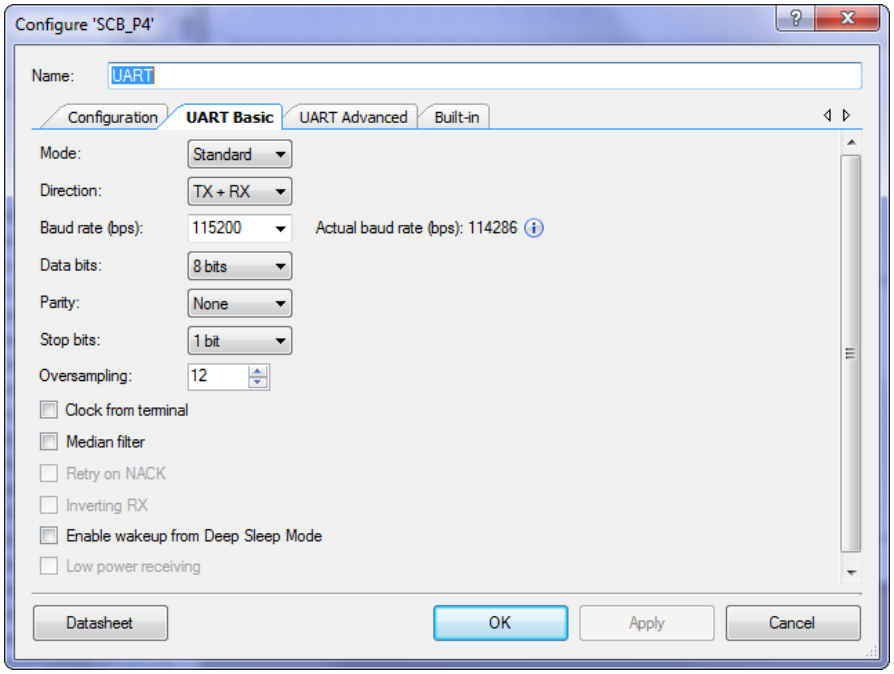


Figure 10. UART Setting

- Timer configuration

Fig. 11 shows the configuration for the timer. The counter should be set as an up counter. The interrupt should be enabled at on the terminal count of the counter. The period of 12000 is set here to generate an interrupt event every 1 second, since the it works with clock frequency of 12 KHz in this project.

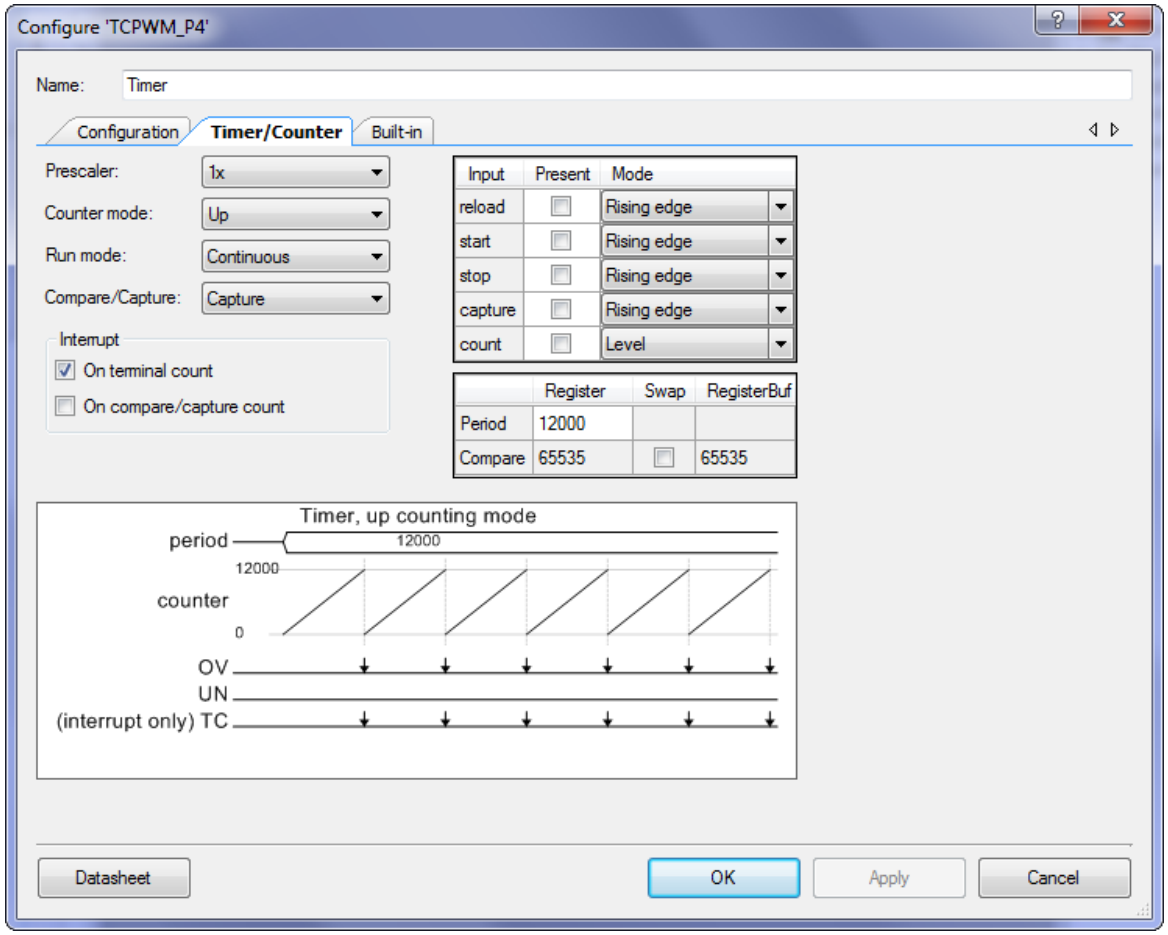


Figure 11. Timer Configuration

- Output Pin

The digital output pin which is connected to the onboard switch SW2 is configured in Fig. 12

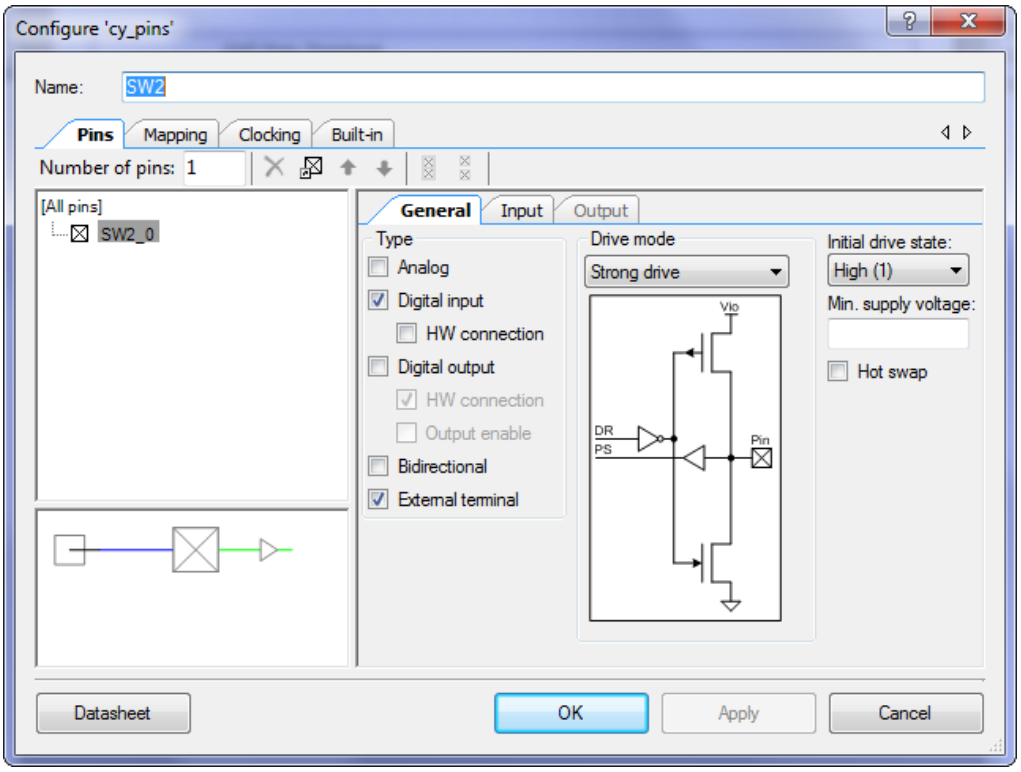


Figure 12. Output Pin Configuration

3. Firmware

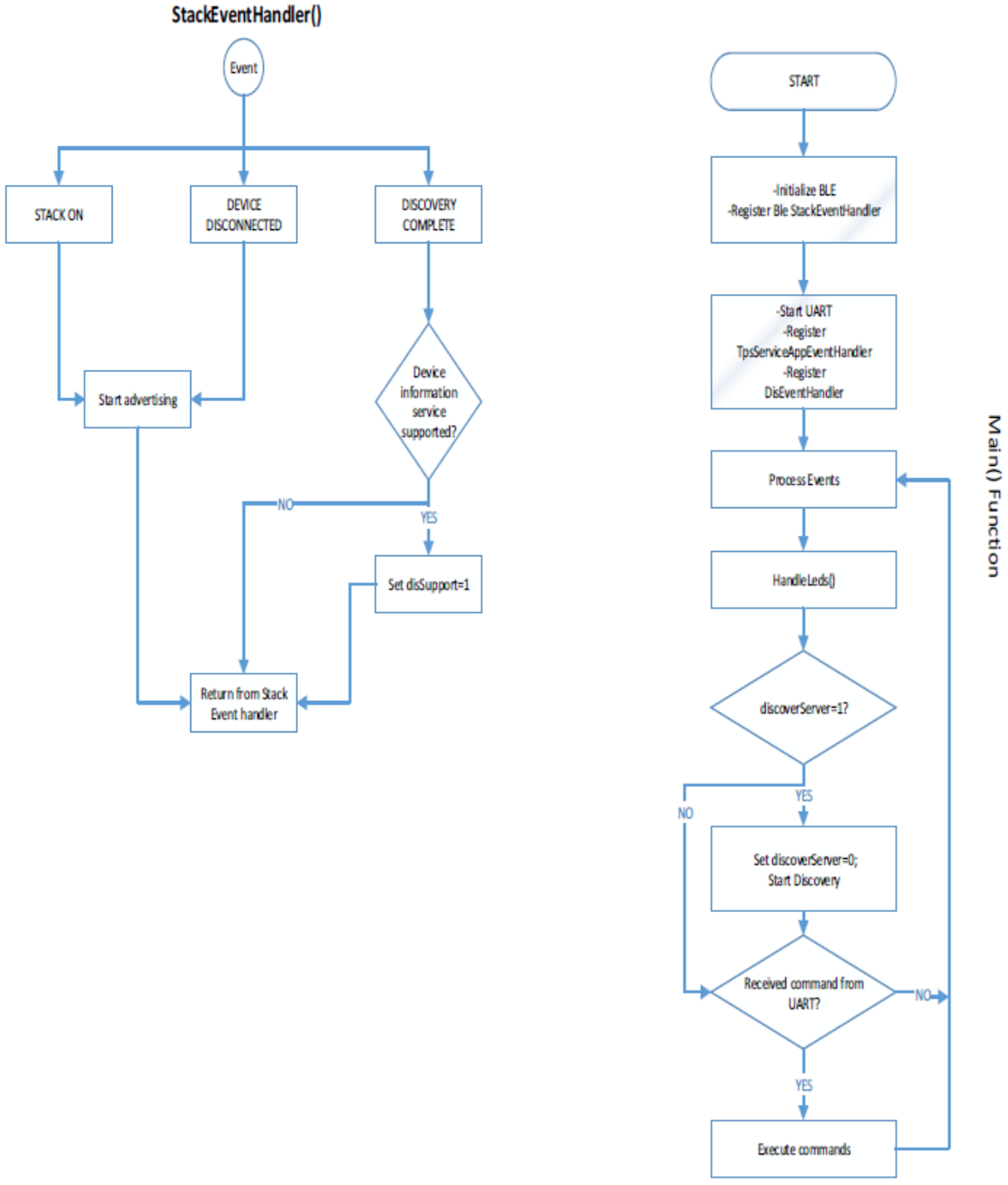


Figure 5. Flowchart of the firmware for Central

The firmware includes UART communication, DIS, and Battery Level. The main.c file can be divided into four main functions.

- Main Function

The main () function performs the initialization of the BLE stack, executes the necessary routines to process the BLE events and maintain the connection. When the system is initialized, main() function continuously operates in a while (1) loop which executes UART commands and send notifications, read characteristic etc. At the same time, it will process the events received by the BLE stack and enables application layer to use them and take the appropriate action. Commands from UART terminal can be found in Table 1.

- StackEventHandler () function

When a BLE component is started, this function is used to handle the common events generated for the BLE Stack.

Table 5. Functions in StackEventHandler for the Peripheral Part

Functions	Descriptions
CYBLE_EVT_STACK_ON	This event happens when BLE is initialized and turned on.
CYBLE_EVT_GAP_DEVICE_CONNECTED	This event happens when connection of a central device to a peer device is completed.
CYBLE_EVT_GAP_DEVICE_DISCONNECTED	This event happens when central device is disconnected from peer device and starts to scan again.
CYBLE_EVT_GATTC_DISCOVERY_COMPLETE	This event happens when discovery of the peer service is completed. Here the discovered service is BAS.

- CY_ISR(DIS_Interrupt) function

It is used to handle the interrupt routine when SW2 is pressed as well as the interrupt routine caused by the timer. An interrupt event is generated every one second when the timer resets, which leads decrease counter by 1.

- LED Handle Function:

This function is used to turn on or turn off the LEDs based on the BLE state. The detailed mechanism is described in the following LED PART.

LED PART

In order to demonstrate different communication status, the LED is fully developed as shown below. When the counter ends at 1, the LED starts to blinking in order to alert the user.

Table 6. Different Statuses indicated by the LED

Status	Color	
Advertising	White	
Connected	If the counter is working	If the counter is finished
	Constant Green	Blinking Green
	Constant Blue	Blinking Blue
	Constant Red	Blinking Red
Disconnected	LED OFF	

RESULTS

For testing our project, both kid and dongle should be programed. We also need the HyperTerminal software in order to connect to the devices by UART terminal.

- Connection

At the starting time, both devices announce their connection on UART Terminal as shown in Fig. 14 and Fig. 15

```
BLE Stack ON:
00a050021c32
Started to Advertise
Device Connected
)
Discovery complete.
Discovered services:
Peer device supports Device Information Service
```

Fig. 14 Principal device starting message

```
Started to Scan
peer address:
00a050021c32
Stop scanning:
success

Discovery complete.
Peer device supports the Service
```

Fig.15 Principal device starting message

- Reading the counter

There two ways for reading the counter.

1) By typing ‘r’ in the UART terminal of dongle.
2) By typing ‘n’ on the kit UART terminal to send notification by the peripheral device. But first notification from peripheral device should be enabled by typing ‘e’ on the dongle UART terminal. (To disable notification from peripheral device, type ‘d’ on the dongle UART terminal)

Fig. 16 shows the received notification by the dongle.

```
Write Descr Response
Write Descr Response
Write Descr Response
Write Descr Response
Notification received. Battery Level:21
Notification received. Battery Level:20
Notification received. Battery Level:19
Notification received. Battery Level:18
Notification received. Battery Level:18
Read Char Response. Battery Level: 16
Read Char Response. Battery Level: 14
Read Char Response. Battery Level: 13
Read Char Response. Battery Level: 12
Read Char Response. Battery Level: 2
Notification received. Battery Level:1
_
```

Fig. 16 received notification by the dongle

- Reset counter & Change Color

The counter can be reset by typing ‘q’ and also the color of LED can be change by typing ‘+’ on the kit UART terminal. Fig. 17 shows the received notification by the kit.

```
 BAS notification disabled
BAS notification enabled
The colour of LED has been changed
The colour of LED has been changed
The Counter has been Reset
 The colour of LED has been changed
```

Fig. 17 received notification by the kid

- Switch Server

Peripheral device will start to discover peer service which is Dis service when switch 1 is pressed on the pioneer kit. Fig. 18 shows what will be displayed on the pioneer kit URT terminal when the switch is pressed.

```
Discovery complete.
Discovered services:
Peer device supports Device Information Service
```

Fig. 18 peripheral start to discover peer service

- Read Device Information

For reading the characteristics related to the DIS, ‘1’ to ‘9’ should be typed on the pioneer kit UART terminal. The characteristics of the DIS will be displayed on the pioneer kit UART terminal as shown in the Fig. 19.

```
Manufacturer Name:Cypress Semiconductors
Model Number:DIS v1.0
Serial Number:
Hardware Rev:CY8CKIT-042 BLE
Firmware Rev:
Software Rev:CySmart tool
System ID:
IEEE 11073-20601:
PNP ID:0
```

Fig. 19 the characteristics of DIS read by the server

CONCLUSION

In this study, BLE is used to build up the communication channel between the Cypress Pioneer Kit and the dongle. UART is implemented to send commands. In the DIS feature, the kit can read the device information from the dongle. In the Battery Level Service, the dongle can get access to the battery level of the kit. In addition, the dongle can enable and disable the notification function of the kit. There are various connection status and status of counter. The LED embedded on the

board shows their connection status as well as the status of counter by changing the color of LED and making the LED constant or blinking.