ELEC5514 Lab 4: BLE Real-time Communications

From the last three weeks, we have learnt the UART Tx/Rx and how to add a BLE service/characteristic at BLE server side. Besides, we also learned how to use the code to complete a certain BLE application. Besides the BLE server that we have learnt, from now, we will start to learn how to construct a BLE network through programming both the BLE server and BLE client.

Before the connection established, the role of a server is to broadcast its local information periodically, such as local name and MAC address, which makes itself can be seen by another client. The role of client is to find the interested server and initialize the connection. Once the server is connected by one client, it will stop broadcasting local information.

At the last lab, we have learnt how to add a simple application to the server and achieve the data communication from the mobile phone (BLE client) to a BLE board (BLE server). In this lab, we will learn how to achieve the two-way communication between server and client (BLE server → BLE client and BLE client → BLE server).

This week's experiment will be divided into two parts:

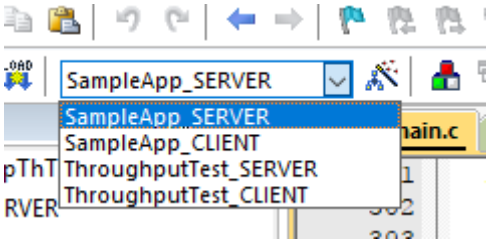
1. Build the connection between two BLE devices using the allocated MAC address, local name and device name.
2. Achieve the LED control and real time chatting between two BLE devices.

Section 1

To establishment the connection, please use the code in '**Lab4_BLEChat**':

SourceCode\Lab4_BLEChat\Projects\Multi\Applications\SampleAppThT\MDK-ARM\STM32L053R8-Nucleo\SampleAppThT.uvprojx

Programming the device as the BLE client. You can choose the role of BLE device as the figure showing below. In this lab, you can set the device as a BLE server or BLE client by selecting "SampleApp_SERVER" or "SampleApp_CLIENT" respectively.



Once you finish programming both the BLE client and the BLE server, your BLE client may connect to the BLE server of another group. To avoid this issue, please:

a) Identify your BLE server using the allocated local name, device name and MAC address (as we did in previous lab).

b) Set the dedicated connection address for the BLE client.

Group No.	Local name	Device name	MAC address
1	BLE_LOC_G1	Dev1	0xF1, 0x33, 0x01, 0xE1, 0x80, 0x03
2	BLE_LOC_G2	Dev2	0xF2, 0x33, 0x01, 0xE1, 0x80, 0x03
3	BLE_LOC_G3	Dev3	0xF3, 0x33, 0x01, 0xE1, 0x80, 0x03
4	BLE_LOC_G4	Dev4	0xF4, 0x33, 0x01, 0xE1, 0x80, 0x03
5	BLE_LOC_G5	Dev5	0xF5, 0x33, 0x01, 0xE1, 0x80, 0x03
6	BLE_LOC_G6	Dev6	0xF6, 0x33, 0x01, 0xE1, 0x80, 0x03
7	BLE_LOC_G7	Dev7	0xF7, 0x33, 0x01, 0xE1, 0x80, 0x03
8	BLE_LOC_G8	Dev8	0xF8, 0x33, 0x01, 0xE1, 0x80, 0x03
9	BLE_LOC_G9	Dev9	0xF9, 0x33, 0x01, 0xE1, 0x80, 0x03
10	BLE_LOC_G10	Dev10	0xF0, 0x33, 0x01, 0xE1, 0x80, 0x03
11	BLE_LOC_G11	Dev11	0x11, 0x33, 0x01, 0xE1, 0x80, 0x03
12	BLE_LOC_G12	Dev12	0x12, 0x33, 0x01, 0xE1, 0x80, 0x03
13	BLE_LOC_G13	Dev13	0x13, 0x33, 0x01, 0xE1, 0x80, 0x03

To observe how the connection is built through the programming language, please follow the "**User_Process()**" function within the while loop of main function in "**main.c**" file. Within the function, you can follow the "**Make_Connection()**" to see how the connection is built in detail. After you finished programming and opening the serial monitor, you should see the graphs below:

For the client:

```
HWver 49, FWver 1811
BLE Stack Initialized.
CLIENT: BLE Stack Initialized
Client Create Connection
Connected to device:02-80-E1-00-00-AA
Start reading TX Char Handle
Disconnected
Client Create Connection
Connected to device:02-80-E1-00-00-AA
Start reading TX Char Handle
Disconnected
Client Create Connection
Connected to device:02-80-E1-00-00-AA
Start reading TX Char Handle
EVT_BLUE_GATT_DISC_READ_CHAR_BY_UUID_RESP
TX Char Handle 000D
Start reading RX Char Handle
EVT_BLUE_GATT_DISC_READ_CHAR_BY_UUID_RESP
RX Char Handle 0010
Disconnected
Client Create Connection
```

For the server:

```
BLE Stack Initialized.
SERVER: BLE Stack Initialized
Sample Service added.
TX Char Handle 000D, RX Char Handle 0010
Service added successfully.
General Discoverable Mode 0
Connected to device:02-80-E1-00-00-BB
Disconnected
General Discoverable Mode 0
Connected to device:02-80-E1-00-00-BB
```

Please remind that in your screenshot in this section, I hope that the client connects to the device with the **allocated MAC address**.

Please note to monitor the UART port of two devices at the same time, you may need to open **two** Arduino serial monitors.

Once you finished, please save the screenshot for both the server and client.

Section 2

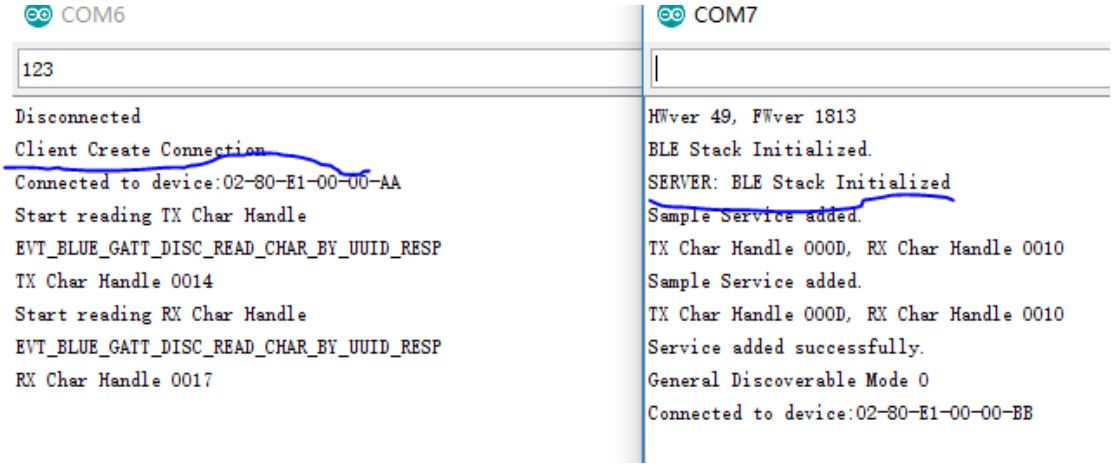
Instead of using the Mobile phone and one BLE device we did last week, this time we will use two BLE boards to communicate with each other:

a) Add a BLE service

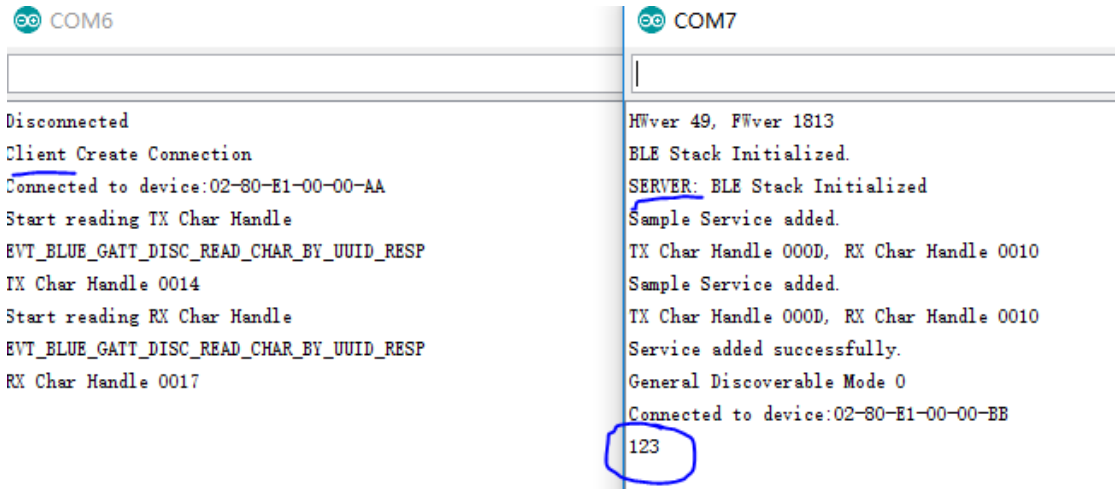
b) This service should be able to achieve the **LED control application** through typed in some specific characters, and **real-time chatting** between two BLE devices.

For example, the left interface is from the client, and the right one is from the server.

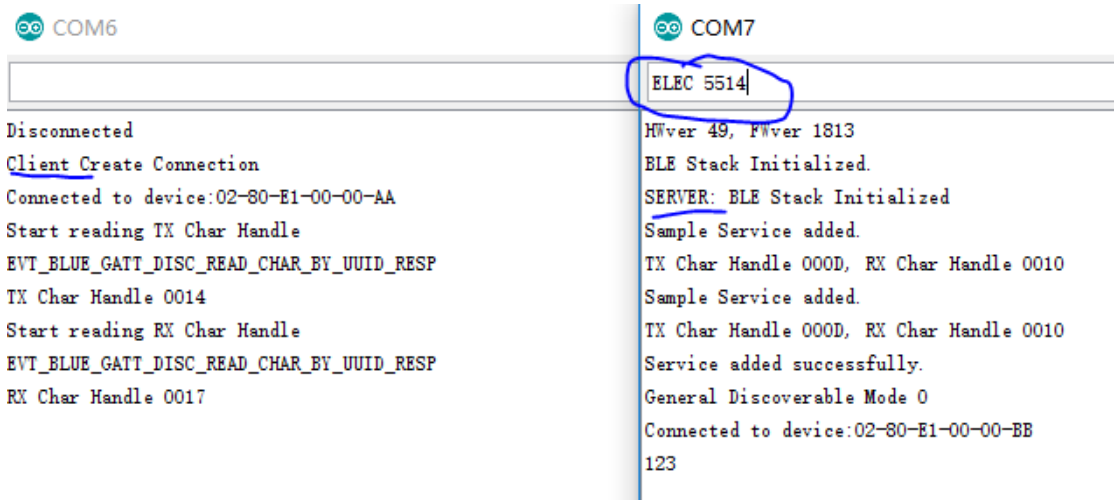
When type '123' to the client:



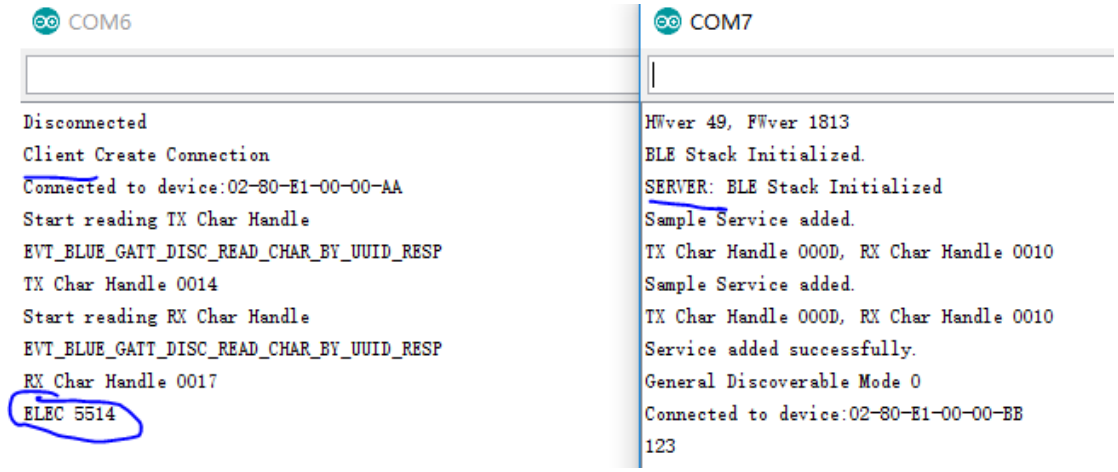
The server should receive this message and display it:



This time type ‘ELEC 5514’ to the server:



The client can display the message sent from the server:



Hints: Please note that the controlling commends can be sent from either client or server (**Two-way LED control**). For the real time chatting, we hope the students to get the result as the above graphs shown.

In this lab, since the programming target is not the smart device, it cannot provide the options that which device you want to connect to, which service/characteristic you

want. Thus, you need to pre-define your target device and target service/characteristic. In “main.c”, the function “**User_Process()**” tells how the connection is established, how the client device connects to the interested server, and how the client select the interested service/characteristic. After the connection is established, the sample service provides a simple example: once the developer finished download the code and identified the role of the devices, press the blue button of one device, it will send a fixed string to the other device and at the same time, the state of LED will be changed.

Please remind that **the data transmission of client --> server will be different from that of server-->client**. Recall the content of lab 3, we have learnt how the **BLE server** executes the action that the attribute is modified through the function “**Attribute_Modified_CB()**”.

Please read the example firstly and try to understand how the communication is achieved in both directions.

Once you finished, please raise your hand and call your tutor to check the result. Remember to save the screenshot and the photo of your result then you can use them in your report.