

## I. Content

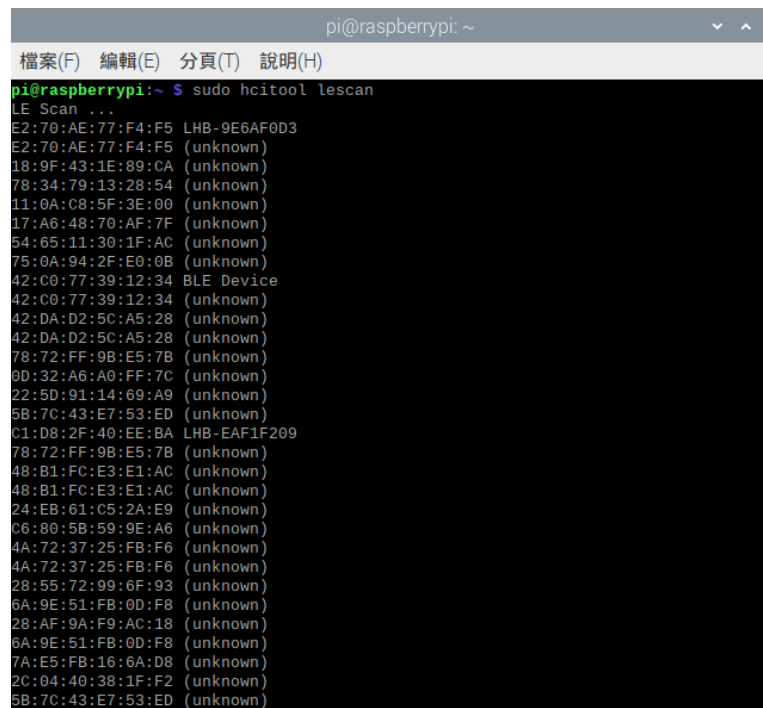
Use Python BLE program on Rpi to communicate with BLE Scanner APP on the mobile phone, including write / read operation.

## II. Discussion

0. We download the image and set up Rpi with external monitor and mouse.

Client Characteristic Configuration Descriptor is a kind of descriptor that GATT client can use it to control what kind of packets the GATT server can send to it. So, we use the BLE Scanner APP on the mobile phone to be the GATT server and set the property to be read and writable. Rpi would be the GATT client.

1. After set up Rpi blue tooth, we can use `$sudo hcitool lescan` to scan the BLE devices in the nearby. Note that we have to use `$sudo hciconfig hci0 leadv 0` to set Rpi to be discoverable.



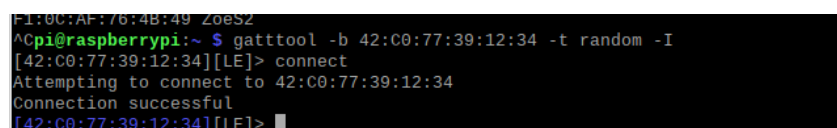
```

pi@raspberrypi: ~
檔案(F) 編輯(E) 分頁(T) 說明(H)
pi@raspberrypi:~ $ sudo hcitool lescan
LE Scan ...
E2:70:AE:77:F4:F5 LHB-9E6AF0D3
E2:70:AE:77:F4:F5 (unknown)
18:9F:43:1E:89:CA (unknown)
78:34:79:13:28:54 (unknown)
11:0A:C8:5F:3E:00 (unknown)
17:A6:48:70:AF:7F (unknown)
54:65:11:30:1F:AC (unknown)
75:0A:94:2F:E0:0B (unknown)
42:C0:77:39:12:34 BLE Device
42:C0:77:39:12:34 (unknown)
42:DA:D2:5C:A5:28 (unknown)
42:DA:D2:5C:A5:28 (unknown)
78:72:FF:9B:E5:7B (unknown)
0D:32:A6:A0:FF:7C (unknown)
22:5D:91:14:69:A9 (unknown)
5B:7C:43:E7:53:ED (unknown)
C1:D8:2F:40:EE:BA LHB-EAF1F209
78:72:FF:9B:E5:7B (unknown)
48:B1:FC:E3:E1:AC (unknown)
48:B1:FC:E3:E1:AC (unknown)
24:EB:61:C5:2A:E9 (unknown)
C6:80:5B:59:9E:A6 (unknown)
4A:72:37:25:FB:F6 (unknown)
4A:72:37:25:FB:F6 (unknown)
28:55:72:99:6F:93 (unknown)
6A:9E:51:FB:0D:F8 (unknown)
28:AF:9A:F9:AC:18 (unknown)
6A:9E:51:FB:0D:F8 (unknown)
7A:E5:FB:16:6A:D8 (unknown)
2C:04:40:38:1F:F2 (unknown)
5B:7C:43:E7:53:ED (unknown)

```

From the above figure, we can find our BLE Device address.

2. Use `gatttool` to connect a peripheral. Note that if the peripheral use random BLE address, we have to specify it in the command, or the connection might fail. Furthermore, the connection is easy to disconnect, so we have to reconnect the device frequently.



```

F1:00:AF:76:4B:49 ZoesZ
^Cpi@raspberrypi:~ $ gatttool -b 42:C0:77:39:12:34 -t random -I
[42:C0:77:39:12:34][LE]> connect
Attempting to connect to 42:C0:77:39:12:34
Connection successful
[42:C0:77:39:12:34][LE]>

```

3. After connection, use *\$char-desc* to find currently available handles, which are the connection points where we can read / write data.

```

pi@raspberrypi: ~
檔案(F) 編輯(E) 分頁(T) 說明(H)
[42:C0:77:39:12:34][LE]> char-desc
handle: 0x0001, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0002, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0003, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0004, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0005, uuid: 00002a01-0000-1000-8000-00005f9b34fb
handle: 0x0006, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0007, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0008, uuid: 00002a05-0000-1000-8000-00005f9b34fb
handle: 0x0009, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x000a, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x000b, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x000c, uuid: 8667586c-9a37-4c91-84ed-54ee27d90049
handle: 0x000d, uuid: 00002900-0000-1000-8000-00005f9b34fb
handle: 0x000e, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x000f, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0010, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0011, uuid: af6badb1-5b99-43cd-917a-a77bc549e3cc
handle: 0x0012, uuid: 00002900-0000-1000-8000-00005f9b34fb
handle: 0x0013, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x0018, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0019, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x001a, uuid: 00002819-0000-1000-8000-00005f9b34fb
handle: 0x001b, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x001c, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x001d, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x001e, uuid: 00002a2b-0000-1000-8000-00005f9b34fb
handle: 0x001f, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x0020, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0021, uuid: 00002a0f-0000-1000-8000-00005f9b34fb
handle: 0x0022, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0023, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0021, uuid: 00002a0f-0000-1000-8000-00005f9b34fb
handle: 0x0022, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0023, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0024, uuid: 00002a29-0000-1000-8000-00005f9b34fb
handle: 0x0025, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0026, uuid: 00002a24-0000-1000-8000-00005f9b34fb
handle: 0x0027, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0028, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0029, uuid: 69d1d8f3-45e1-49a8-9821-9bbdfdaad9d9
handle: 0x002a, uuid: 00002900-0000-1000-8000-00005f9b34fb
handle: 0x002b, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x002c, uuid: 9fbf120d-6301-42d9-8c58-25e699a21dbd
handle: 0x002d, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x002e, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x002f, uuid: 22eac6a9-24d6-4bb5-be44-b38ac67c7bfb
handle: 0x0030, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x0031, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x0032, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0033, uuid: 9b3c81d8-57b1-4a8a-b8df-0e56f7ca51c2
handle: 0x0034, uuid: 00002900-0000-1000-8000-00005f9b34fb
handle: 0x0035, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x0036, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x0037, uuid: 2f7cabce-888d-411f-9a0c-bb92ba96c102
handle: 0x0038, uuid: 00002900-0000-1000-8000-00005f9b34fb
handle: 0x0039, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x003a, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x003b, uuid: c6b2f38c-23ab-4ed8-a6ab-a3a879bbd5d7
handle: 0x003c, uuid: 00002900-0000-1000-8000-00005f9b34fb
handle: 0x003d, uuid: 00002800-0000-1000-8000-00005f9b34fb
handle: 0x003e, uuid: 00002803-0000-1000-8000-00005f9b34fb
handle: 0x003f, uuid: 61b2220b-f162-4e0b-afc5-137498281443
handle: 0x0040, uuid: 00002900-0000-1000-8000-00005f9b34fb
[42:C0:77:39:12:34][LE]>
(gatttool:4388): GLib-WARNING **: 10:40:29.700: Invalid file descriptor.
  
```

We can find the correlated handles *0x003f* to the custom setting *61B220B-F162....* on the mobile phone.



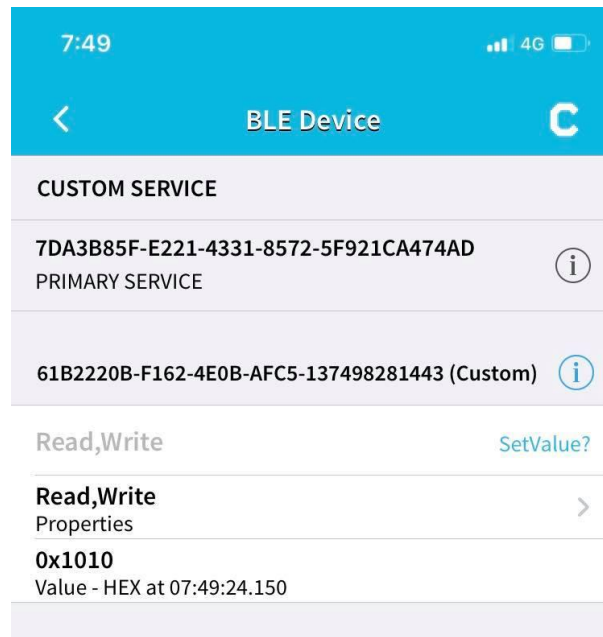
4. We first try to manually communicate with the mobile phone.

write: write data *1010* to handle *0x003f*.

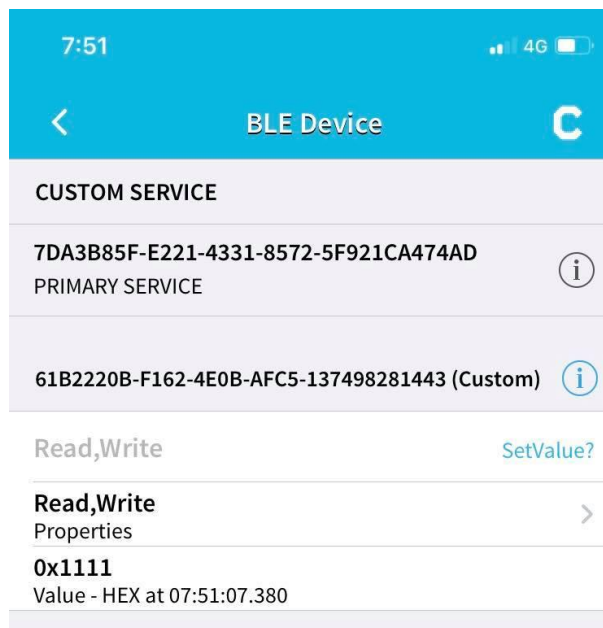
*\$char-write-req 0x003f 1010*

```

[42:C0:77:39:12:34][LE]> connect
Attempting to connect to 42:C0:77:39:12:34
Connection successful
[42:C0:77:39:12:34][LE]> char-write-req 0x003f 1010
Characteristic value was written successfully
[42:C0:77:39:12:34][LE]>
  
```



read: Send data *1111* from the mobile phone, and read data on Rpi from handle *0x003f*  
*\$ char-read-hnd*



```
[42:C0:77:39:12:34][LE]> char-read-hnd 0x003f
Characteristic value/descriptor: 11 11
[42:C0:77:39:12:34][LE]>
```

5. Then, we use python to wrap the write and read command, using *bluepy* module.

```

Enter your device number: 0
('Device', 0)
7a:c1:54:ee:df:1f
Connecting...
Services...
Service <uuid=Generic Attribute handleStart=1 handleEnd=3>
Service <uuid=Generic Access handleStart=20 handleEnd=26>
Service <uuid=fff0 handleStart=40 handleEnd=65535>
Characteristic <fff1>
Characteristic <fff2>
Characteristic <fff3>
Characteristic <fff4>
efg
1: Device 69:c6:9b:fb:f3:3d (random), RSSI=-69 dB
Flags = 1a
Enter your device number: 0
('Device', 0)
7a:c1:54:ee:df:1f
Connecting...
Services...
Service <uuid=Generic Attribute handleStart=1 handleEnd=3>
Service <uuid=Generic Access handleStart=20 handleEnd=26>
Service <uuid=fff0 handleStart=40 handleEnd=65535>
Characteristic <fff1>
Characteristic <fff2>
Characteristic <fff3>
Characteristic <fff4>
hip
Tx Power = 07
Enter your device number:

```

From the above figure, we can see the received message *efg* and *hip* have been successfully decoded using ASCII code.

6. Then we try to refine the program for better user interaction. It can print out the message received. The user can also choose whether or not they want to write and the content of the message. Note that the handle number should be correlated to the custom characteristic shown on the app.

```

pi@raspberrypi: ~
檔案(F) 編輯(E) 分頁(T) 說明(H)
#39: c5:ab:e4:c7:bd:31 (random), RSSI=-74 dB
#40: 18:b8:64:33:46:55 (random), RSSI=-85 dB

Enter your device number: 32
Connecting to #32: 7c:97:08:cc:b0:73

Available Services:
Service Service <uuid=Generic Attribute handleStart=1 handleEnd=3>
Service Service <uuid=Generic Access handleStart=20 handleEnd=26>
Service Service <uuid=fff0 handleStart=40 handleEnd=65535>
Characteristic <fff1>
Characteristic <fff2>
Characteristic <fff3>
Characteristic <fff4>

Data in channel: b'Hello~'
Write?y
To Write:Hi!
pi@raspberrypi:~ $

```

Custom Characteristic	
(0000fff4-0000-1000-8000-00805f9b34fb)	
Property : 0 0 0 0 1 0 1 0 b	
<b>R</b>	Hi! 48 69 21
<b>W</b>	Hello~ 48 65 6C 6C 6F 7E
descriptor :	
(00002904-0000-1000-8000-00805f9b34fb)	
: 19 00 00 00 00 00 00	
(00002901-0000-1000-8000-00805f9b34fb)	
: 52 65 61 64 28 4E 6F 74 69 66 79 29 20 2F 20	
57 72 69 74 65	
: Read(Notify) / Write	

### III. GitHub

[https://github.com/paying45292/Embedded\\_HW3.git](https://github.com/paying45292/Embedded_HW3.git)