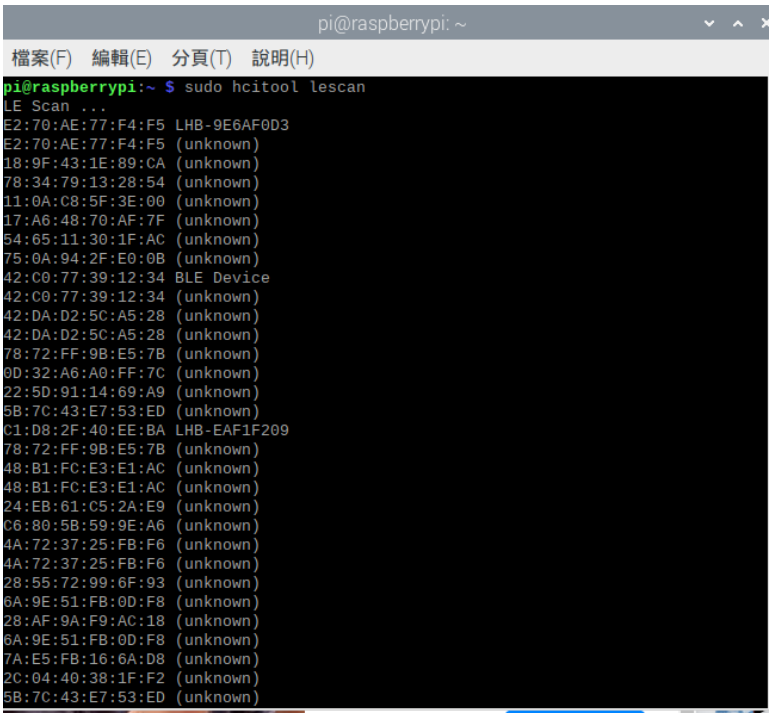
I. Content

Use Python BLE program on Rpi to communicate with BLE Scanner APP on the mobile phone.

II. Discussion

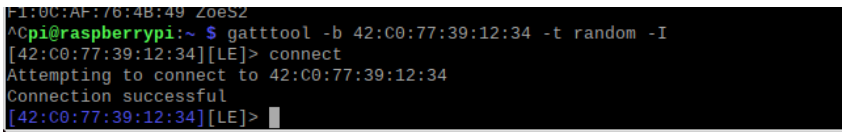
0. We download the image and set up Rpi with external monitor and mouse.

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.

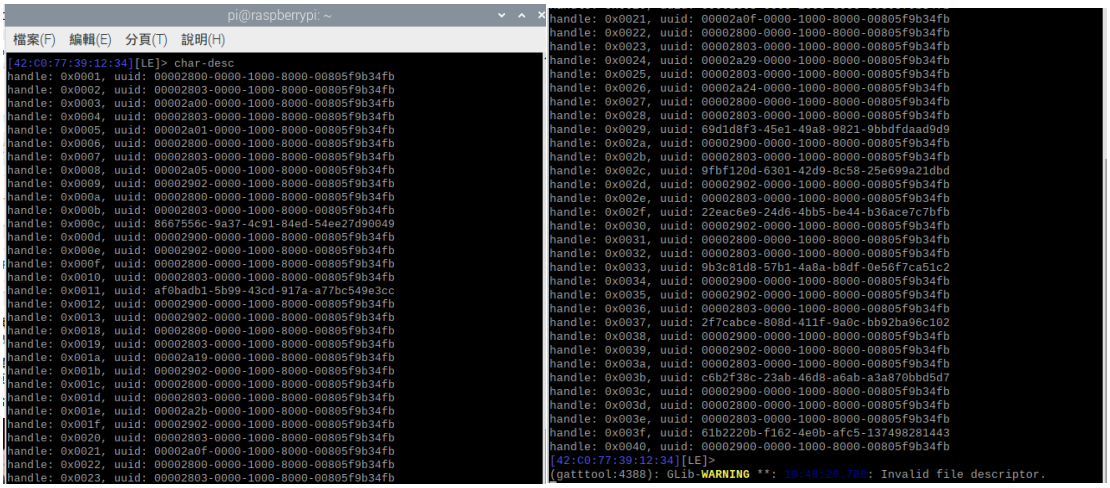


From the above figure, we can find our BLE Device address.

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.



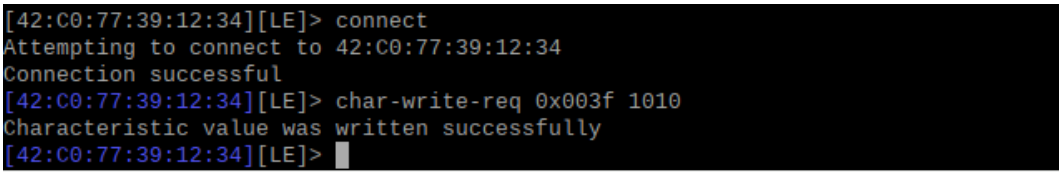
3. After connection, use *$char-desc* to find currently available handles, which are the connection points where we can read / write data.

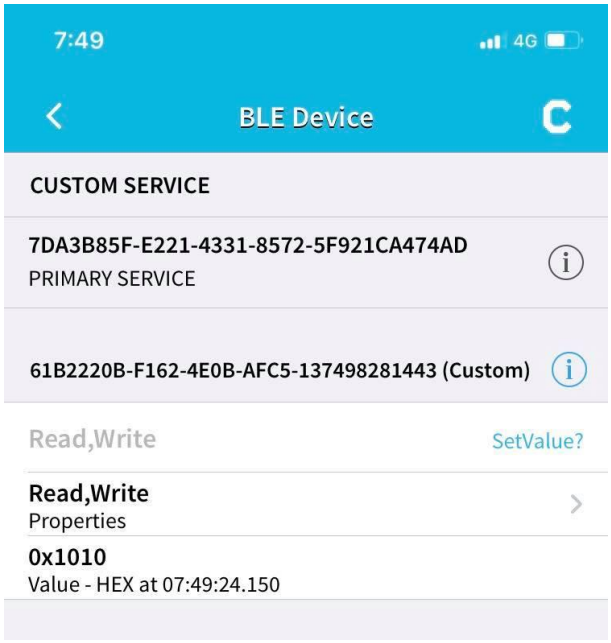


We can find the correlated handles *0x003f* to the custom setting *61B220B-F162….* on the mobile phone.

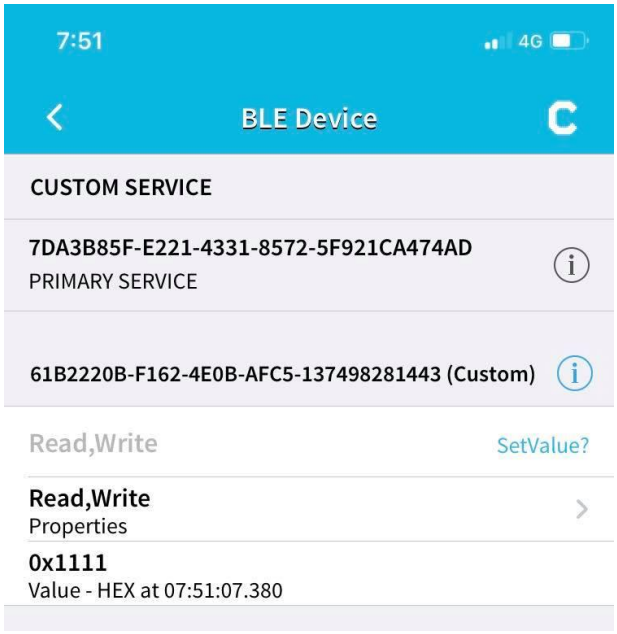
4. We first try to manually communicate with the mobile phone.

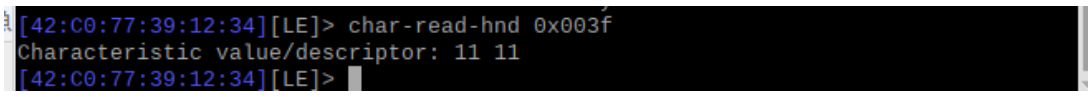
write: write data *1010* to handle *0x003f.*
$char-write-req 0x003f 1010



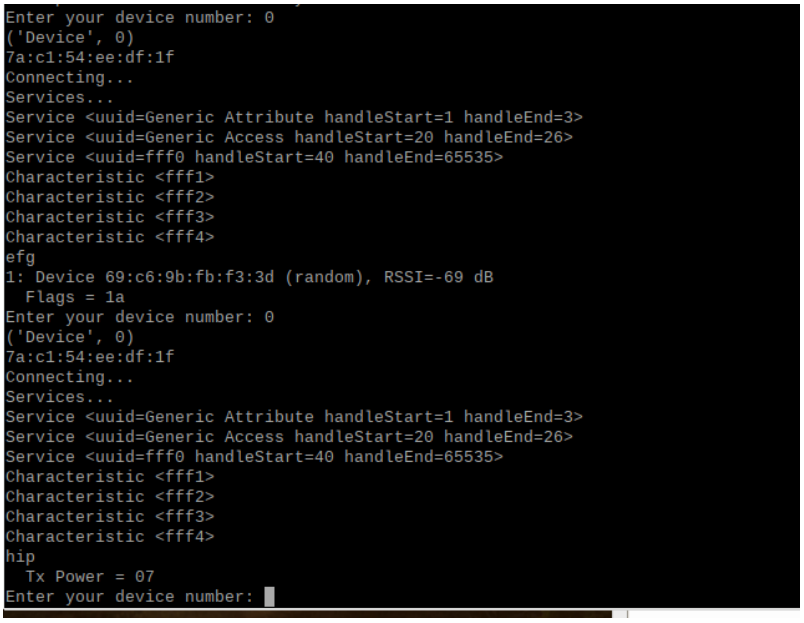


read: Send data *1111* from the mobile phone, and read data on Rpi from handle *0x003f*
$ char-read-hnd



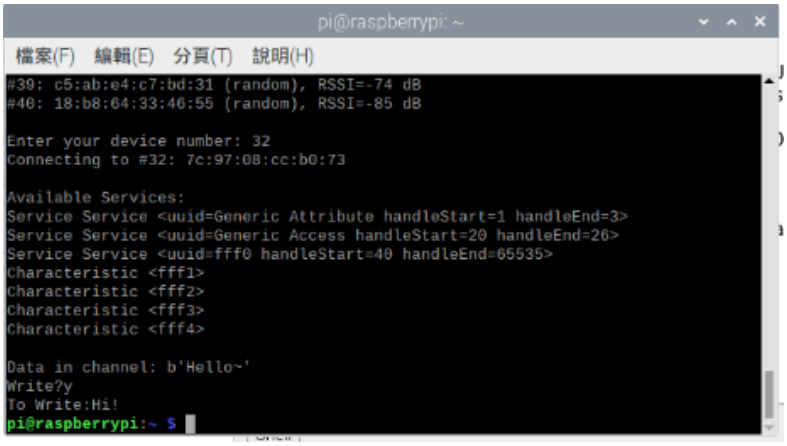


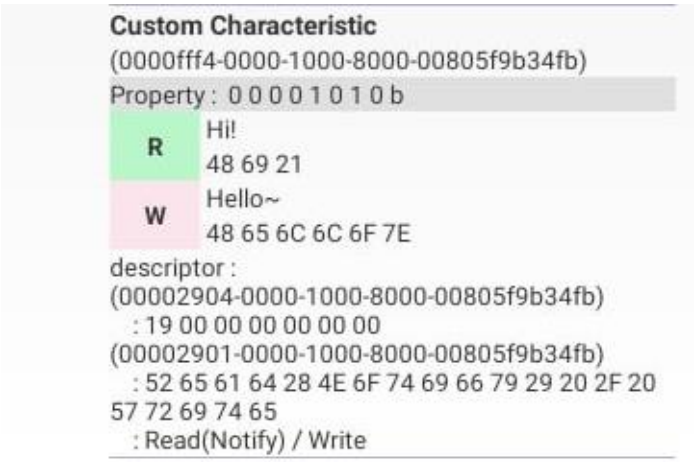
5. Then, we use python to wrap the write and read command, using *bluepy* module.



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.





V. GitHub

https://github.com/paying45292/Embedded_HW3.git