

Documentation For ESP_BLE Client

Custom PC Application for BLE Client using python

Bleak: <https://bleak.readthedocs.io/en/latest/index.html>

Its GitHub codes: <https://github.com/hbldh/bleak>

Requirement:

- VS Code with Python extension.
Guide: https://www.youtube.com/watch?v=9o4gDQvVklU&ab_channel=KennyYipCoding
- Pip library installation for Bleak.
Guide: https://www.youtube.com/watch?v=ENHnfQ3cBQM&ab_channel=TheCodeCity
- Remember to **Turn ON the Bluetooth** of computer.

Note: The examples like the `service_explorer.py` or `enable_notifications.py` will not directly execute in the terminal, it need some inputs based commands in the terminal as followings.

Example: **service_explorer.py**(An example showing how to access and print out the services, characteristics and descriptors of a connected GATT server)

Execution: Run the command "python service_explorer.py --address 30:C6:F7:20:D2:66" in the terminal. (if not in examples directory then "cd .\examples\) This command will only execute in ..\bleak-master\examples> directory.

Checking the MAC address of BLE Server, one can run the example of **discover.py** for showing how to scan for BLE devices.

Example: **enable_notifications.py**(Example showing how to add notifications to a characteristic and handle the responses.)

Execution: Run the command: python service_explorer.py --address 30:C6:F7:20:D2:66 "beb5483e-36e1-4688-b7f5-ea07361b26a8" in the terminal. The argument <notify uuid> refers to the *UUID of a characteristic* that supports notifications. (if not in examples directory then "cd .\examples\) This command will only execute in ..\bleak-master\examples> directory.

Our Use Example: **async_callback_with_queue.py**(An example showing how async notification callbacks can be used to send data received through notifications to some external consumer of that data.)

Execution Process: ...\\bleak-master\\examples> python async_callback_with_queue.py --address 34:85:18:98:1C:95 "beb5483e-36e1-4688-b7f5-ea07361b26a8"

One thing more, first the ble server device must be disconnected, then connected back for getting the data, otherwise the client will not retrieve the data by showing "ERROR: could not find device with address '34:85:18:98:1C:95'".

Issue: Using the `ble_server.ino` file, is that once the `BLEServerCallbacks()` executes it set the `deviceConnected` to true, and the server starts sending data, now on the other hand the ble Client is showing connecting to ble server(which take some time), so due to this instant transfer of data from the Sever side, we lost the data capturing on the client side.

Temporary Solution: Once the ble Communication starts establishing with client, we introducing some delay of 5 sec before sending the data over ble, so that the client proper connects to its server.

Efficient Solution: In such scenario, before send data from Server to Client, once the client should first send some data to Sever(Which will mean that the connection is build successfully), when the Sever received the data, it will check it and then start sending back the original data to client.

Ble_Server:

- Creating a new Characteristic(for writing data) with a Callback function, which will get the value from using onWrite() function, and will check a condition if the first element is '1', if it is then will it will call the sendDataToClient() function.
- The function must be defined above its calling area, otherwise it'll give error of not declare scope, which will just check the ble connectivity and send the data.
- One more thing to note, using this read and write communication the notification property is ignored, so it can be commented from READ characteristic.
- Now in loop function, we'll getting data and storing it to array, and the READ & WRITE is just depends upon the callback functions.