TQ5 CO2 Logger

# Introduction

The following documentation will serve as the steps taken by the group members to complete the following project. The environment used to program the ESP32 is ESP-IDF on Visual Studio. The reason for us using ESP-IDF rather than Arduino IDE was because we ran into some problems when connecting to the Nordic Thingy:52. Not only did we consider the problems, but also found more documentation on the handling BLE GATT client / servers. It is recommended for the ESP to hold the BLE connections to 3, even though it can reach 9.

# Steps

## GATT Client

As we are not so familiar with the ESP-IDF framework, this project had to be divided into certain steps to complete it. The steps go as follows:

1. Connect to Nordic Thingy:52 with the ESP
   1. Find Service of GATT Server
   2. Find Characteristic of GATT Server
   3. Subscribe to a notify characteristic (air quality)

This was a little challenging because it required a 128-bit char UUID, whilst the example codes provided only an 8-bit UUID. Note: Read through walkthrough on GitHub of espressif examples. Solution is there

1. Connect to multiple Nordic Thingy:52’s
   1. Get average of all air quality characteristics

Having read the walkthrough of a single connection, setting up multiple connections was easier. First the git hub example was used as a boiler plate, rather than creating a project from scratch. Like the first step, the second step consisted of creating a profile for each Server and changing their UUID for both service and characteristics. The walkthrough provides a flow chart of how the 2nd step works:

## Multi-Connection GATT Client FlowchartGATT Server / Client

Share the value found with another device (Raspberry Pi) – Multiple ways to target this approach

* 1. ESP32 can coexist as Client / Server
  2. Raspberry Pi can act as another Server with a Write type Characteristic

1. Client / Server advertising data collected from one thingy

References

* Single Server - <https://github.com/espressif/esp-idf/blob/master/examples/bluetooth/bluedroid/ble/gatt_client/tutorial/Gatt_Client_Example_Walkthrough.md>
* Multiple servers - <https://github.com/espressif/esp-idf/tree/master/examples/bluetooth/bluedroid/ble/gattc_multi_connect>
* Client / Server - <https://github.com/espressif/esp-idf/tree/master/examples/bluetooth/bluedroid/coex/gattc_gatts_coex>
* 34. ESP BLE connection limit - <https://docs.espressif.com/projects/espressif-esp-faq/en/latest/software-framework/ble-bt.html>