

Our IoT Car!

2022/05/13 鄭至盛

OUR IOT CAR

報告流程：

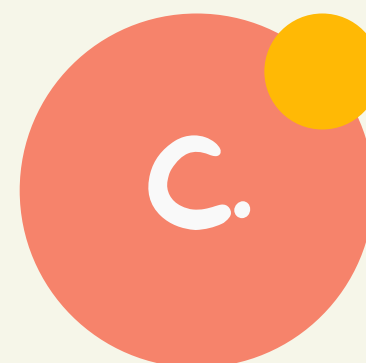
接下來的簡報會依序介紹我們的進度與遇到的問題等等的，如果大家有任何建議或回饋也歡迎在Q&A Time與我們分享。



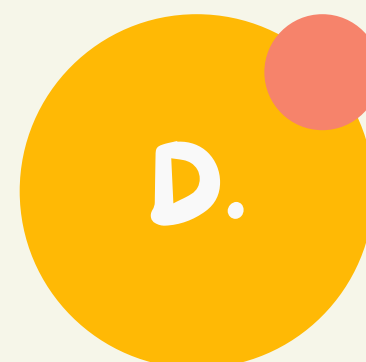
目前進度：



實作方法：



研究中的內容：



問題整理：



QA TIME

專案結構：

Canva

RASPBERRY端

使用PiCamera、OpenCV與Mediapipe
作為手勢辨識的主要Module。

01

SERVER端

串連兩端的Client，接收、發送資訊
並進行對應處理，與資料庫溝通。

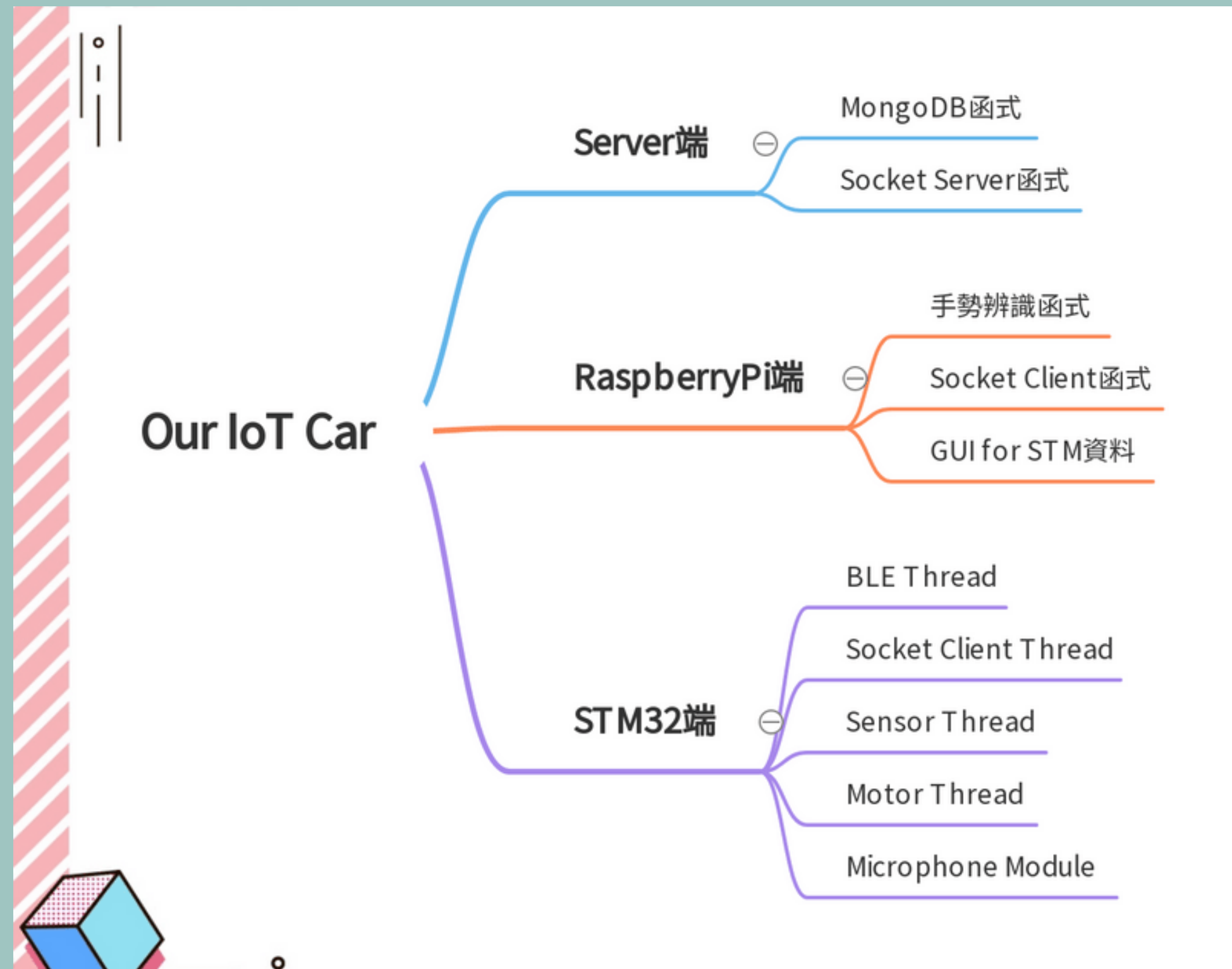
02

STM32端

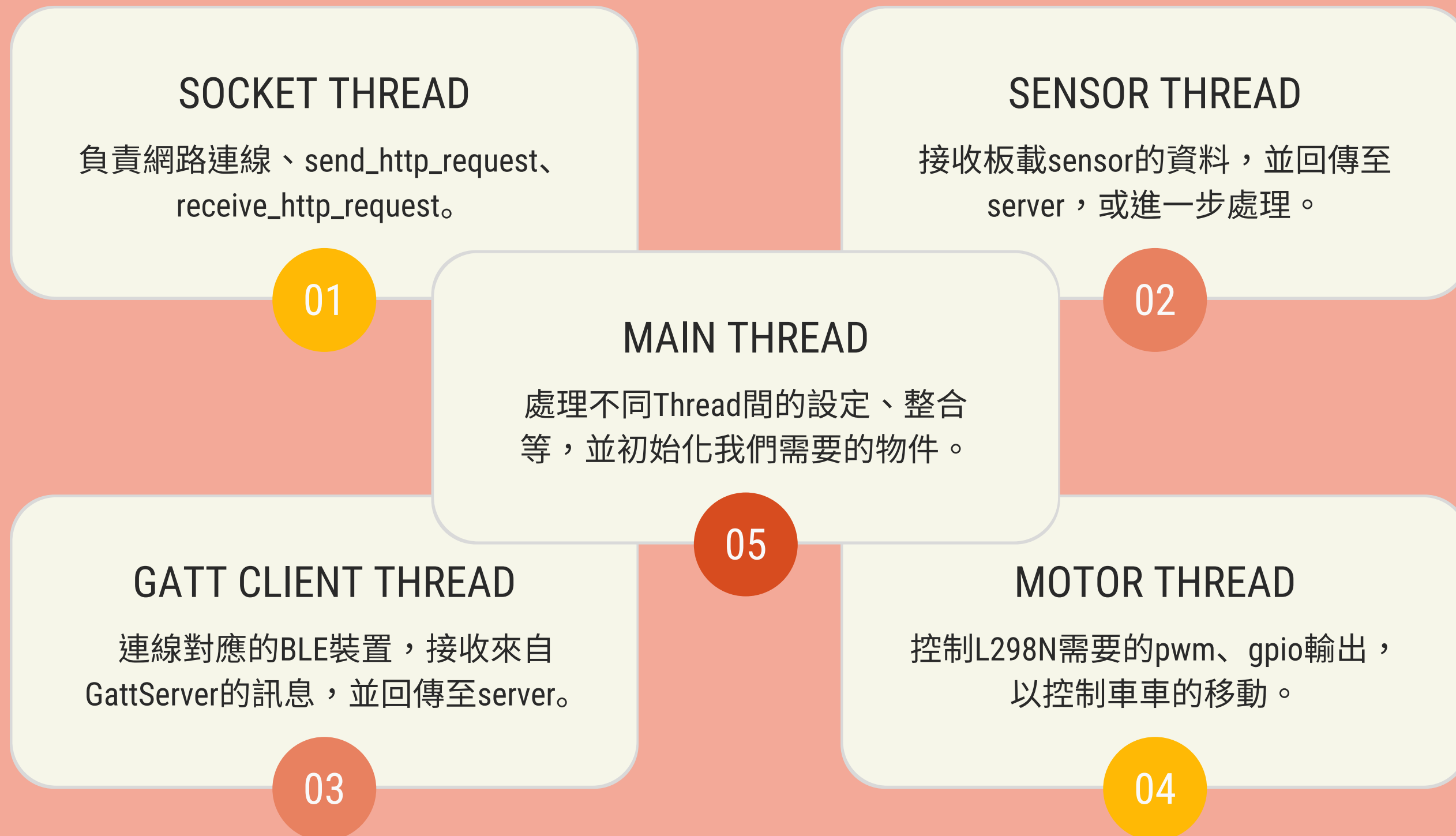
發覺板子上不同功能，透過mbed os
提供的工具進行對應功能設計。

03

專案結構：



Mbed端的結構：



```
mbded-os-final-project
/Users/zhengzhisheng/Mbed Programs/mbded-os-final-project/BSP_B-L475E-IOT01
> BUILD
  include
    GattClientModule.h
    MyGattClient.h
    MyMotorControl.h
    MySensor.h
    MySocket.h
    root_ca_cert.h
    SocketModule.h
    wifi_helper.h
  mbed-os
  mbed-os-ble-utils
  source
    GattClientModule.cpp
    MyGattClient.cpp
    MyMotorControl.cpp
    MySensor.cpp
    MySocket.cpp
    SocketModule.cpp
  wifi-ism43362
  .gitignore
  .mbed
  main.cpp
  mbed_app.json
```

目前進度：



Mbed功能實作：

- BLE的Gatt Client
- Socket發送與接收訊息
- Motor的控制 (PWM控速、GPIO控制馬達轉彎)
- 不同Thread間的整合

RaspberryPi功能實作：

- 整合手勢辨識與攝影機輸入
- 在OS安裝對應模組

實作方法(整合功能):

1

建立不同的執行序，並封裝成獨立的物件。

2

透過main.cpp傳入對應的EventQueue、ConditionVariable、Mutex、Semaphore

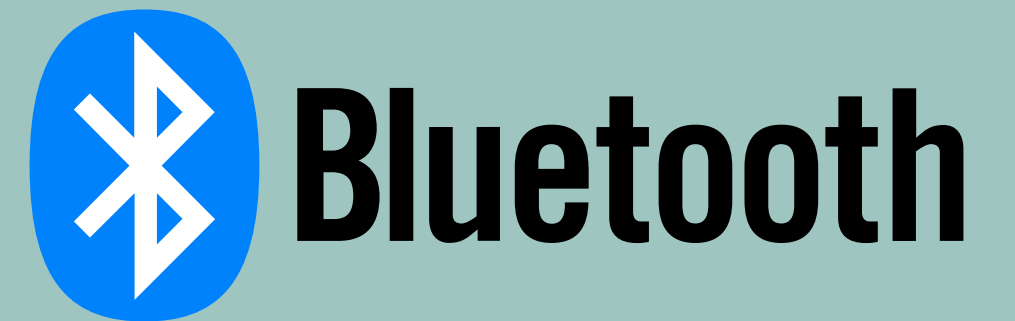
3

針對不同功能的同步、非同步的整合設計採取對應的OS功能。



實作方法(GattClient):

- 1 初始化BLE物件，並設定對應Callback函式。
- 2 接收對應訊息，並透過Condition Variable實現與Socket Thread的同步。
- 3 根據需求連結不同BLE裝置。

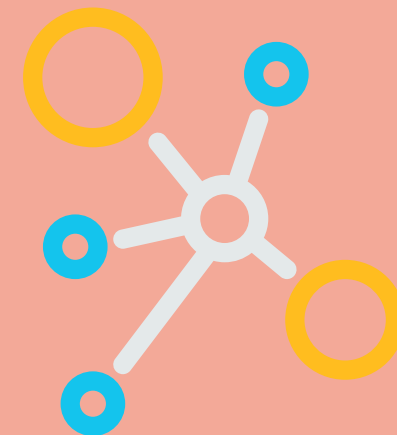


實作方法(GattClient):

```
void MyGattClient::modify_BLE_device(char* device_name){  
    _ble->gap().disconnect(ble_module.get_connection_handle_t(),  
        ble::local_disconnection_reason_t::USER_TERMINATION);  
  
    ble_module.stop();  
  
    ble_process.modify_peer_device_name(device_name);  
}
```

實作方法(Socket):

```
void GattClientModule::when_characteristic_changed(const GattHVXCallbackParams* event)
{
    _mutex->lock();
    int len = sprintf(_buffer, "Change on attribute %u: new value = ", event->handle);
    printf("Change on attribute %u: new value = ", event->handle);
    for (size_t i = 0; i < event->len; ++i) {
        if(len<200){
            len += sprintf(_buffer+len, "0x%02X ", event->data[i]);
        }
        printf("0x%02X ", event->data[i]);
    }
    printf(".\r\n");
    _cond->notify_all();
    _mutex->unlock();
    ThisThread::wait_us(1000);
}
```



研究中的內容：

麥克風模組：

- 透過stm3214提供的HAL庫控制板子上的板子上的MP34DT01麥克風模組
- 針對麥克風模組的訊號進行對應的DSP處理
- 應用場域：
 - 接收環境音訊，處理雜訊，並回傳至Server
 - 訓練模型，進行簡單的關鍵詞辨識



Canva

參考資料：

1

STM32效果器：<https://youtu.be/VDhmVrbSpqA>

2

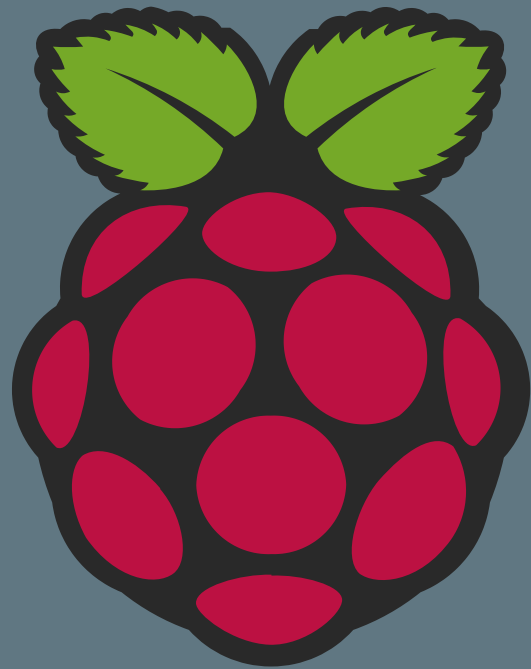
Mbed CLI的範例：<https://github.com/janjongboom/b-1475e-iotOla-audio-mbed.git>

3

關於BSP庫怎麼應用、結合至mbed studio已經實測成功，有機會再跟大家分享。



問題整理：



RPI 3B+的算力不足：

- 實際執行手勢辨識時Latency很大，效率大概只有 1 fps。

解決方法：

- 更換為RPI 4。
- 研究程式碼可以優化之處。

Q&A
Time 