

# Mbed HW 3 Report

## eRPC and MQTT

109033130 唐振家

## 一、Lab Description

### 1、eRPC setting

說明：

我將此次Homework切成幾個部分：

- 1、eRPC
- 2、MQTT
- 3、accelerometer
- 4、matplotlib

首先先利用Lab9創建一個新的Program，並完成eRPC的功能，我直接用當時給的blink\_led.h、blink\_led\_server.cpp、blink\_led\_server.h檔案去做修改，以下是我的start、stop function，當client call on/off時，type變數就會因為輸入的值而做出改變。

```
client.led_on(1)
print("ON")
sleep(2.0)
client.led_off(1)
print("OFF")
```

```
/****** erpc declarations *****/
void led_on(uint8_t a) {
    if(a){
        type = 1;
    }
}

void led_off(uint8_t a) {
    if(a) type = 0;
}
```

### 2、MQTT

說明：

會這樣做的原因是因為，eRPC的部分，為了能夠持續接收訊息而會進入一個無窮迴圈的部分，保持能夠一直傳輸的功能，所以我只能將eRPC以及MQTT用不同的thread做區隔，使兩者都能夠穩定並持續執行，並且這意味著我只能使用全域變數去做到兩者結合的動作。

```
rpc_server.run();
```

## 一、Lab Description

### 3、accelerometer

說明：

接下來我加入accelerometer的部分，與Lab10很像，我將其加入在publish\_message中，並且透過改變全域變數type去控制是否pub出去，我讓程式在一開始的時候就會一直不斷call publish\_message(每0.1秒一次)，差別只在於當call on時，開始pub，當call off時，則甚麼都不做。

```
mqtt_thread.start(callback(&mqtt_queue, &EventQueue::dispatch_forever));  
mqtt_queue.call_every(100ms, &publish_message, &client);
```

```
void publish_message(MQTT::Client<MQTTNetwork, Countdown>* client) {  
    if (type){  
        MQTT::Message message;  
        BSP_ACCELERO_AccGetXYZ(pDataXYZ);  
        char buff[100];  
        sprintf(buff, "%d %d %d", pDataXYZ[0], pDataXYZ[1], pDataXYZ[2]);  
        message.qos = MQTT::QOS0;  
        message.retained = false;  
        message.dup = false;  
        message.payload = (void*) buff;  
        message.payloadlen = strlen(buff) + 1;  
        int rc = client->publish(topic, message);  
        // printf("rc: %d\r\n", rc);  
        printf("%s\n", buff);  
    }else{}  
}
```

### 4、matplotlib

說明：

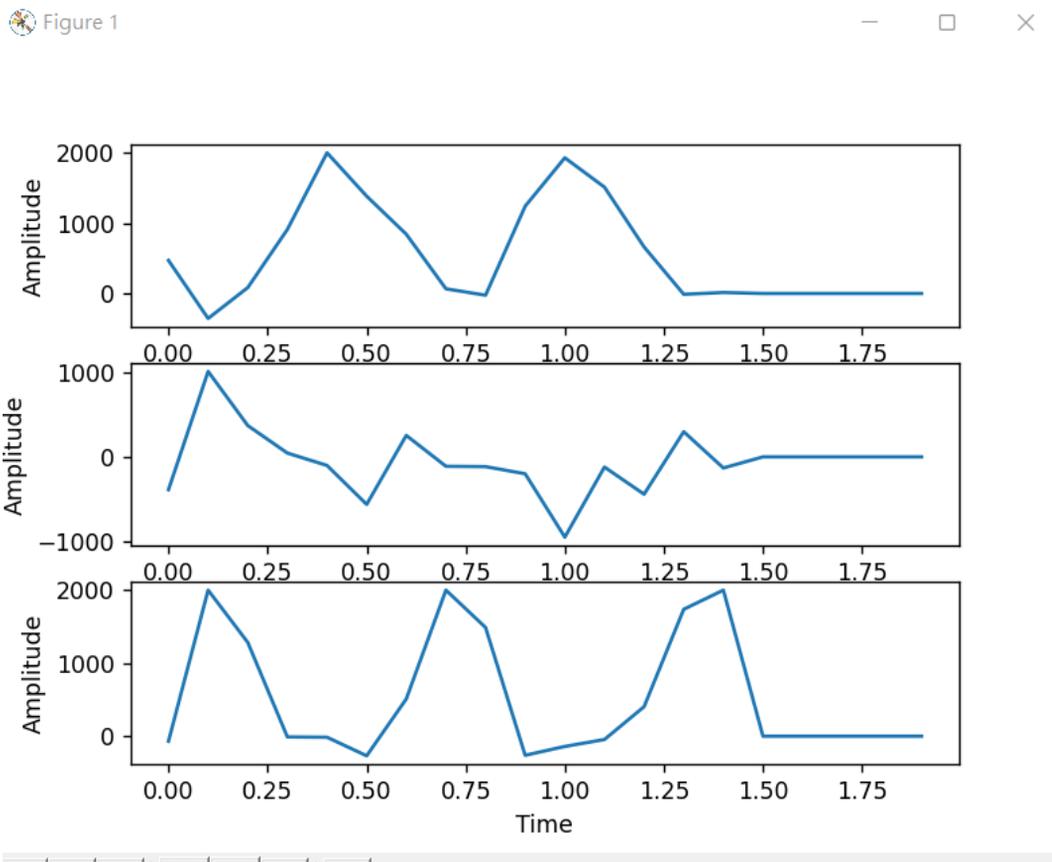
將accelerometer數字data取下來之後需要做圖，但是取下來的資料型態為string，而且包含了很多沒辦法在server裡就去除掉的符號，所以必須在python中處理掉，將message取出來之後，先去掉尾端'\x00'的特殊字符，之後再分別取出xyz的data，轉成int之後做圖。利用loop函數，在for迴圈中每一次都執行subscribe，並且轉換、存入陣列之中，最後存完資料後將圖印出來。

```
line = str(msg.payload)  
line = line.rstrip("")  
line = line.rstrip("0")  
line = line.rstrip("x")  
line = line.rstrip("\*")
```

```
for i in range(0, 20):  
    mqttc.loop()  
    x[i] = x[num]  
    y[i] = y[num]  
    z[i] = z[num]
```

## 二、Result

### 最後呈現結果



## 三、遇到的問題

### Python整數、字串轉換問題

最一開始使用以下方法做數字data的存取，但是此方式會轉換變成整數的list，目前正在查如何將list轉換為array，如果成功，說不定能夠用多維的矩陣，簡化程式並做圖。

```
line = str(msg.payload)
numbers = str([int(s) for s in re.findall(r'-?\d+\.?\d*', line)])
# numbers = numbers.strip(', 0']')
# numbers = numbers.strip('[')
# numbers = numbers.replace(', ' ' ')
array = np.asarray(line)
print("Message: " + numbers);
```

### 三、遇到的問題

#### mbed出現ERROR

猜測可能是因為在傳輸過程中，可能接收到眾多函式中或是thread的return，導致出現異常error。

```
>_ DISCO-L4S5I (B-L4S5I-IOT01),Baud rate 9600
++ MbedOS Fault Handler ++
FaultType: HardFault
Context:
R0 : 00000000
R1 : 200A0000
R2 : 00000000
R3 : 0800E9A5
R4 : 20003E00
R5 : 20002D74
R6 : 20002D78
R7 : 00000010
R8 : 20002D73
R9 : 00000000
R10 : 00000000
R11 : 00000000
```

#### 圖形無法生成

如果range沒有設定好，那麼會出現傳輸筆數作完前就出現圖形，或是傳輸完成後並沒有圖形產生，需要重複執行直到for迴圈跑滿。

```
for i in range(0, 20):
    matplotlib.pyplot.plot()
```

### 四、討論

#### 結合過去的lab

本次 HW 我覺得難度非常大，因為必須結合之前學的eRPC、MQTT、accelerometer、matplotlib的概念，如果沒有融會貫通，或是紮實將Lab搞懂，那麼這次就會花很大把時間完成，並且也會變的非常熟悉這些應用。