

# Lab4

105072123 黃海茵

## ● 程式流程

```
#define SSID      "yin"
#define PASSWORD  "0988140518"
#define HOST_NAME "192.168.137.1"
#define HOST_PORT (16619)
```

首先 define 好自己的 SSID、PASSWORD 等等資料

```
void setup() {
    Serial.begin(9600);
    pinMode(LED, OUTPUT);

    /* Now set up two tasks to run independently */
    /*xTaskCreate( target, name, stack size, parameter_pointer, priority, task_handle);*/
    xTaskCreate(TaskWifi, "Wifi", 256, NULL, 1, NULL);
    xTaskCreate(TaskBlink, "Blink", 64, NULL, 1, &xHandle);
}
```

把 LED 設為 OUTPUT 後，create 兩個 task

```
void TaskWifi(void* pvParameters){

    Serial.print("FW Version:");
    Serial.println(wifi.getVersion().c_str());

    if (wifi.setOprToStationSoftAP()) {
        Serial.print("to station + softap ok\r\n");
    } else {
        Serial.print("to station + softap err\r\n");
    }

    if (wifi.joinAP(SSID, PASSWORD)) {
        Serial.print("Join AP success\r\n");
        Serial.print("IP:");
        Serial.println(wifi.getLocalIP().c_str());
    } else {
        Serial.print("Join AP failure\r\n");
    }

    if (wifi.disableMUX()) {
        Serial.print("single ok\r\n");
    } else {
        Serial.print("single err\r\n");
    }

    Serial.print("setup end\r\n");
}
```

TaskWifi 負責送收，使用的是 Arduino 內部的 TCPClientSingle 的範例，先與筆電 WiFi 建立連線

```

uint8_t buffer[8] = {0};

if (wifi.createTCP(HOST_NAME, HOST_PORT)) {
    Serial.print("create tcp ok\r\n");
} else {
    Serial.print("create tcp err\r\n");
}

vTaskResume(xHandle);

for(;;){
    String prVal = String(int(analogRead(A2)));

    char *pr = prVal.c_str();
    wifi.send((const uint8_t*)pr, strlen(pr));

    uint32_t len = wifi.recv(buffer, sizeof(buffer), 10000);
    if (len > 0) {
        Serial.print("Received:");
        for(uint32_t i = 0; i < len; i++) {
            Serial.print((char)buffer[i]);
            led_blink = buffer[i];
        }
        Serial.print("]\r\n");
    }
}

```

這部分也在 TaskWifi 中，因為要收的資料只有 LED 閃或不閃，所以把 buffer 改小一點，並且在 TCP 連線建立完成後，開始送收 data 時再把 handle LED 的 task 喚醒。然後我把送收 data 的部分寫在 for loop 中，讓 Arduino 和筆電間可以一直溝通，不會傳一次就停了，並且每次都把 led\_blink 更新成接收到的回傳值

```

def on_new_client(clientsocket,addr):
    global currentRing
    while True:
        # [TODO] decode message from Arduino and send to AWS
        recv = clientsocket.recv(3).decode("utf-8")
        print("recv: " + recv)
        if int(recv) > 150:
            blink = 0
        else:
            blink = 1
        conn.send(str(blink).encode("utf-8"))
    clientsocket.close()

```

由筆電的 python code 來接收 arduino 傳過來的資料，基本上跟上次 Lab 一樣，只是多了判斷傳過來的光敏電阻值，大於 150 表示光線充足不用閃爍，回傳 0，反之回傳 1

```

char led_blink = '0';

void TaskBlink(void *pvParameters){
    vTaskSuspend(xHandle);

    for(;;){
        if(led_blink == '1'){
            digitalWrite(LED, HIGH);
            vTaskDelay(500/portTICK_PERIOD_MS);
        }
        digitalWrite(LED, LOW);
        vTaskDelay(500/portTICK_PERIOD_MS);
    }
}

```

TaskBlink 用來控制 LED 閃爍，在一開始我就先讓它進入休眠階段，等待另一個 task 把它喚醒，之後只要 led\_blink 為 '1' 就開始閃爍

### ● 實作過程遇到的困難及解決方式


一開始我把 suspend 放在 setup 中，但跑程式就會卡住，移到 TaskBlink 中最上面就可以正常跑了

還有 WiFi 一下正常一下 error，有時候要重跑很多次才會成功，有時候又很順利 (◡ ◡ ◡)

### ● 如何讓 task 休眠及喚醒

使用 vTaskSuspend 休眠 vTaskResume 喚醒

### ● 運行結果



```

FW Version:
to station + softap ok
Join AP success
IP:
single ok
setup end
create tcp ok
Received:[0]
Received:[0]
Received:[0]
Received:[0]
Received:[0]
Received:[0]
Received:[0]
Received:[0]
Received:[0]
Received:[0]
Received:[0]

```

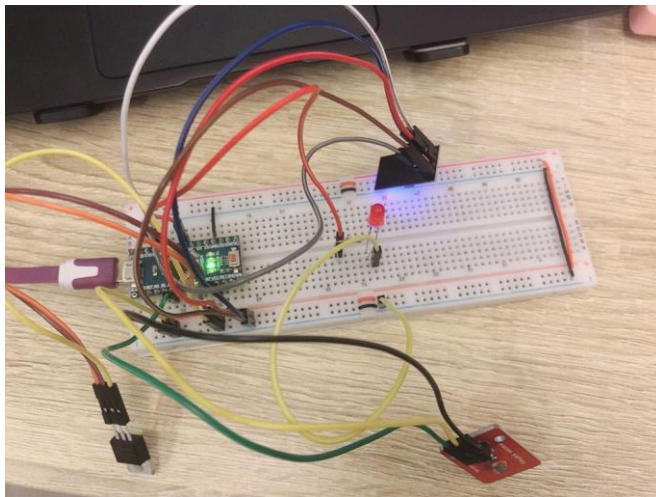
```

C:\Users\User\Desktop>python Lab4.py
server start at: 0.0.0.0:16619
wait for connection...
connected by ('192.168.137.110', 38761)
recv: 464
recv: 465
recv: 464
recv: 469
recv: 470
recv: 470
recv: 470
recv: 469
recv: 469
recv: 469

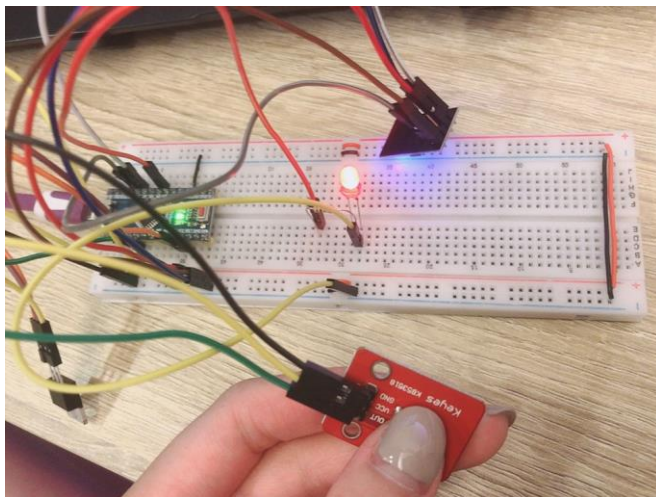
```

我的 IP (圖左) 一直顯示不出東西，但在討論區發問後，助教說沒關係，可以正常送收即可

- 設備照片



光線充足時，LED 熄滅



光線不足時，LED 閃爍