

## ปฏิบัติการที่ 3

### Python MQTT and pyQT HW control/monitoring

#### วัตถุประสงค์

1. สามารถใช้งาน JSON บนภาษา python ได้
2. สามารถใช้งาน MQTT ด้วยภาษา python ได้
3. สามารถออกแบบ GUI เพื่อควบคุมอุปกรณ์ผ่านทาง MQTT ได้
4. สามารถออกแบบ GUI เพื่อรับข้อมูลจาก MQTT และแสดงผล หรือบันทึกข้อมูลได้

#### อุปกรณ์การทดลอง

- คอมพิวเตอร์/Qt designer
- MQTT broker
- ESP32 microcontroller
- IO board

#### ปฏิบัติการที่ 3.1 การใช้งาน JSON บนภาษา python

1. จากข้อมูลคะแนนของนักศึกษาต่อไปนี้จึงแปลงให้อยู่ในรูป python-JSON

John	50.3
Marry	75.8
Boy	65.2
Kethy	77.4
Brian	83.4

#### Code:

```
import json
data={}
data["John"] = 50.3
data["Marry"] = 75.8
data["Boy"] = 65.2
data["Kethy"] = 77.4
data["Brian"] = 83.4
datajson=json.dumps(data)
print(datajson)
```

## 2. จากข้อมูลคะแนนของนักศึกษาต่อไปนี้จึงแปลงให้อยู่ในรูป python-JSON

name	john
age	35
phone	0992111111
car	ford,toyota,nissan
pet	cat,dog
nationality	USA

### Code:

```
import json
data={}
data["name"] = "john"
data["age"] = 35
data["phone"] = "0992111111"
data["car"] = ["ford","toyota","nissan"]
data["pet"] = ["cat","dog"]
data["nationality"] = "USA"
datajson = json.dumps(data)
print(datajson)
```

### คำถาม

ก่อนที่จะทำการแปลงให้อยู่ในรูป Json ต้องเก็บข้อมูลด้วยชนิดข้อมูลใด และเมื่อแปลงให้อยู่ในรูป json แล้ว บนภาษา python มองว่าเป็นข้อมูลชนิดใด

Json จะเก็บข้อมูลในรูปแบบ Dictionary เมื่อแปลงให้อยู่ในรูป json แล้ว บน python จะมองว่า เป็นข้อมูลชนิด String

### ปฏิบัติการที่ 3.2 การใช้งาน MQTT-paho บนภาษา python

1. ให้ทำการติดตั้ง paho-mqtt โดยติดตั้งผ่าน pip ตามที่ได้เรียนมาแล้ว
2. ให้เขียนโปรแกรมเพื่อเชื่อมต่อกับ MQTT brokerดังตัวอย่างต่อไปนี้ โดยตั้งชื่อ client เป็นรหัสนักศึกษาของตนเอง

```
import paho.mqtt.client as mqtt
broker_address="electsut.trueddns.com"
broker_port=27860
client = mqtt.Client("Bxxxx")
client.connect(broker_address,broker_port)
```

3. ให้ทำการ publish ข้อมูลผ่านทาง topic inno/bxxxxxx/323 (ใช้รหัสนักศึกษาของตนเอง) โดยให้ส่งข้อมูล โดยใช้ข้อมูลจากข้อที่ 3.1.1 จากนั้นให้ทำการ subscribe ข้อมูลด้วย MQTT box

**Code: เฉพาะบรรทัดที่ publish**

```
client.publish("inno/b6214197/323",datajson)
```

ภาพ MQTTbox ที่ได้รับข้อมูล

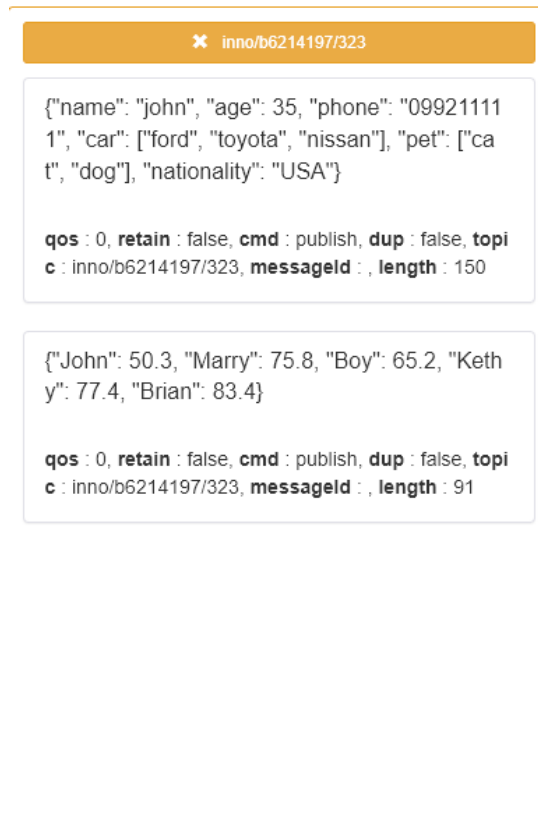


3. ให้ทำการ publish ข้อมูลผ่านทาง topic inno/bxxxxxx/323 (ใช้รหัสนักศึกษาของตนเอง) โดยให้ส่งข้อมูล โดยใช้ข้อมูลจากข้อที่ 3.1.2 จากนั้นให้ทำการ subscribe ข้อมูลด้วย MQTT box

**Code: เฉพาะบรรทัดที่ publish**

```
client.publish("inno/b6214197/323",datajson)
client.publish("inno/b6214197/323",datajson1)
```

ภาพ MQTTbox ที่ได้รับข้อมูล



4. ให้เขียนโปรแกรมเพื่อทำการ subscribe ข้อมูลด้วย topic inno/bxxxxxx/324

- ให้ทำการดีเลย์ 10 วินาที ระหว่าง loop\_start และ loop\_stop
- ให้ทดลองส่งข้อมูลผ่านทาง MQTTbox {"LED":1,"state":"on"}
- ให้ทำการแสดงข้อความที่ได้รับออกมาดังต่อไปนี้
  - Topic
  - Payload
  - Qos

**Code: ทั้งหมด**

```
import paho.mqtt.client as mqtt
import time
import json
```

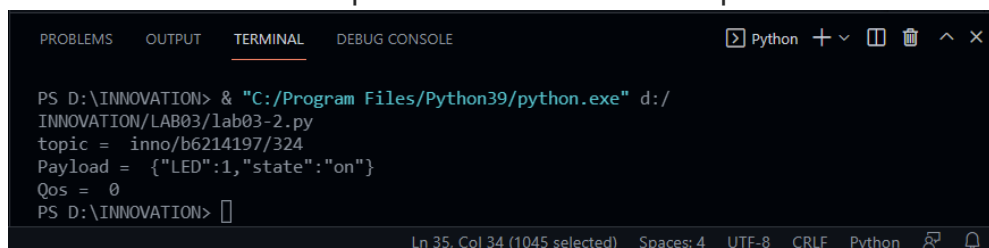
```

broker_address="electsut.trueddns.com"
broker_port=27860
client = mqtt.Client("B6214197")
client.connect(broker_address,broker_port)
def on_message(client, userdata, message):
    print("topic = ",message.topic)
    print("Payload = ",str(message.payload.decode("utf-8")))
    print("Qos = ",message.qos)
    # print("message retain flag=",message.retain)
data={}
data["John"] = 50.3
data["Marry"] = 75.8
data["Boy"] = 65.2
data["Kethy"] = 77.4
data["Brian"] = 83.4
datajson=json.dumps(data)
data1={}
data1["name"] = "john"
data1["age"] = 35
data1["phone"] = "099211111"
data1["car"] = ["ford","toyota","nissan"]
data1["pet"] = ["cat","dog"]
data1["nationality"] = "USA"
datajson1 = json.dumps(data1)
client.publish("inno/b6214197/323",datajson)
client.publish("inno/b6214197/323",datajson1)

client.on_message= on_message
client.subscribe("inno/b6214197/324")
client.loop_start() #start the loop
time.sleep(10) # wait
client.loop_stop() #stop the loop

```

ภาพหน้าจอในส่วนของ output ที่แสดงข้อความจากการ print



```

PS D:\INNOVATION> & "C:/Program Files/Python39/python.exe" d:/
INNOVATION/LAB03/lab03-2.py
topic = inno/b6214197/324
Payload = {"LED":1,"state":"on"}
Qos = 0
PS D:\INNOVATION>

```

5 จากข้อที่ 4 ให้ทำการเตรียมวิธีการรอรับข้อมูลจากวิธีการ delay เป็นการใช้ loop\_forever() โดยให้โปรแกรมหยุดทำงานเมื่อได้รับคำสั่ง system\_stop

**Code: เฉพาะส่วนของ callback**

```
def on_message(client, userdata, message):  
    print("topic = ",message.topic)  
    print("Payload = ",str(message.payload.decode("utf-8")))  
    print("Qos = ",message.qos)  
    myPL = str(message.payload.decode("utf-8"))  
    if myPL == "system_stop" :  
        client.disconnect()
```

**คำถาม**

ต้องใช้คำสั่งใดในการหยุด loop\_forever()

```
client.disconnect()
```

### ปฏิบัติการที่ 3.3 การใช้งาน GUI เพื่อควบคุมและรับค่าจากอุปกรณ์ผ่านทาง MQTT

ให้เขียนโปรแกรมบน ESP32 เพื่อใช้งาน MQTT โดยให้มีการทำงานดังต่อไปนี้

(สามารถประยุกต์ใช้ code จากปฏิบัติการที่ 2 ในสัปดาห์ที่แล้ว)

1. ให้ทำการเขียนโค้ดไมโครคอนโทรลเลอร์เพื่อทำการแปลงข้อมูลที่ได้รับจาก MQTTbox เพื่อสั่งการอุปกรณ์โดยมีรายละเอียดดังต่อไปนี้

**Topic:** "inno/bxxxxxx/lab03/lighting"

หากได้รับข้อความ {"LED":1,"state":"on"} ให้ทำการ on LED ที่ต่ออยู่กับขา 18

หากได้รับข้อความ {"LED":1,"state":"off"} ให้ทำการ off LED ที่ต่ออยู่กับขา 18

หากได้รับข้อความ {"LED":2,"state":"on"} ให้ทำการ on LED ที่ต่ออยู่กับขา 19

หากได้รับข้อความ {"LED":2,"state":"off"} ให้ทำการ off LED ที่ต่ออยู่กับขา 19

2. ให้ทำการเขียนโค้ดไมโครคอนโทรลเลอร์เพื่อทำการแปลงข้อมูลที่ได้รับจาก MQTTbox เพื่อสั่งการอุปกรณ์โดยมีรายละเอียดดังต่อไปนี้ ให้กำหนด PWM ของไมโครคอนโทรลเลอร์ให้มีความละเอียด 8 bit

**Topic:** "inno/bxxxxxx/lab03/pwm"

หากได้รับข้อความ {"PWM":1,"val":255} ให้ทำการปรับความสว่างของหลอด LED ที่ต่ออยู่กับขา 12 ให้เท่ากับค่าที่ส่งไปซึ่งอยู่ในช่วง 0- 255

หากได้รับข้อความ {"PWM":2,"val":255} ให้ทำการปรับความสว่างของหลอด LED ที่ต่ออยู่กับขา 14 ให้เท่ากับค่าที่ส่งไป ซึ่งอยู่ในช่วง 0-255

3. ให้ทำการเขียนโค้ดไมโครคอนโทรลเลอร์เพื่อทำการ publish ข้อมูล โดยให้ publish ข้อมูลและให้ทำการ Subscribe ด้วย MQTTbox โดยมีรายละเอียดดังต่อไปนี้

**Topic:** "inno/bxxxxxx/lab03/sensor"

ให้ทำการจำลองข้อมูลของเซ็นเซอร์จำนวน 2 ตัว โดยให้มีค่าอยู่ในช่วง 0.0-100.0 โดยเป็นเซ็นเซอร์วัดความชื้นและอุณหภูมิตามลำดับ โดยให้ส่งข้อมูลทุก 1 วินาที โดยสลับกันส่งข้อมูล

หากได้รับข้อความ {"sensor":"temp","val":51.5} ให้ทำการพิมพ์ข้อความผ่านทาง Serial Monitor Temp = 51.5

หากได้รับข้อความ {"sensor":"humid","val":32.9} ให้ทำการพิมพ์ข้อความผ่านทาง Serial Monitor Humid= 32.9

ให้ทำการทดสอบการทำงานของโค้ดผ่านทาง MQTT box ให้เรียบร้อย

**Code Arduino: ทั้งหมด**

```
#include <ArduinoJson.h>
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient client;
PubSubClient mqtt(client);
#define WIFI_STA_NAME "worachet"
#define WIFI_STA_PASS "12345678"
#define MQTT_SERVER "electsut.trueddns.com"
#define MQTT_PORT 27860
#define MQTT_USERNAME ""
#define MQTT_PASSWORD ""
#define MQTT_NAME "B6214197"
#define LED_PIN 23
#define sw 5
float temp = 50;
float humid = 50;

DynamicJsonDocument docd(1024); //รับมาจาก mqtt
DynamicJsonDocument docs(1024); //ส่งไปหา mqtt
DynamicJsonDocument docs1(1024); //ส่งไปหา mqtt
DynamicJsonDocument docs2(1024);
char out1[256];
char out2[256];
char out3[256];
char out4[256];
char out5[256];

void callback(char* topic, byte* payload, unsigned int length){
  Serial.print("Topic = ");
  Serial.print(topic);
  deserializeJson(docd,payload,length);
  serializeJson(docd,out3);
  Serial.print(out3);
  Serial.println();
  std::string tp =topic;

  if (tp=="inno/b6214197/lab03/lighting"){

    ///{"LED":1,"state":"on"}
    int LED1 = docd["LED"];
    const char* state = docd["state"];
```



```
std::string s =state;
if (LED1 == 1){
    if (s=="on"){
        digitalWrite(18,1);}
    else if(s=="off"){
        digitalWrite(18,0);}
    }
else if (LED1 == 2){
    if (s=="on"){
        digitalWrite(19,1);}
    else if(s=="off"){
        digitalWrite(19,0);}
    }
}

if (tp=="inno/b6214197/lab03/pwm"){
    ///{"pwm":1,"val":255}
    int pwm = docd["PWM"];
    int val = docd["val"];
    if (pwm == 1){
        ledcWrite(1,val);
        Serial.print("pwm=1");
    }

    else if (pwm == 2){
        Serial.print("pwm=2");
        ledcWrite(2,val);
    }
}
}
```

```
void setup() {

pinMode(18,OUTPUT);
pinMode(19,OUTPUT);
pinMode(12,OUTPUT);
pinMode(14,OUTPUT);
pinMode(4,INPUT_PULLUP);
ledcSetup(1,5000,8);
ledcAttachPin(12,1);
ledcSetup(2,5000,8);
ledcAttachPin(14,2);
```

```
Serial.begin(115200);
pinMode(LED_BUILTIN, OUTPUT);
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(WIFI_STA_NAME);
WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
  digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
}
digitalWrite(LED_BUILTIN, HIGH);
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
mqtt.setServer(MQTT_SERVER, MQTT_PORT);
mqtt.setCallback(callback);
}

void loop() {
  if (mqtt.connected() == false) {
    Serial.print("MQTT connecting... ");
    if (mqtt.connect(MQTT_NAME, MQTT_USERNAME, MQTT_PASSWORD))
    {
      Serial.println("connected");
      mqtt.subscribe("inno/b6214197/lab03/lighting");
      mqtt.subscribe("inno/b6214197/lab03/pwm");

    }
    else {
      Serial.println("failed");
      delay(1000);
    }
  }
  else {
    mqtt.loop();
    float randtemp = random(-100,101)/100.0;
```

```

float randhumid = random(-100,101)/100.0;
temp = randtemp+temp;
if(temp>100){
    temp = 100;
}
else if (temp<0){
    temp = 0;
}
humid = randhumid+humid;
if(humid>100){
    humid = 100;
}
else if (humid<0){
    humid = 0;
}
docs1["sensor"] = "temp";
docs1["val"] = temp;

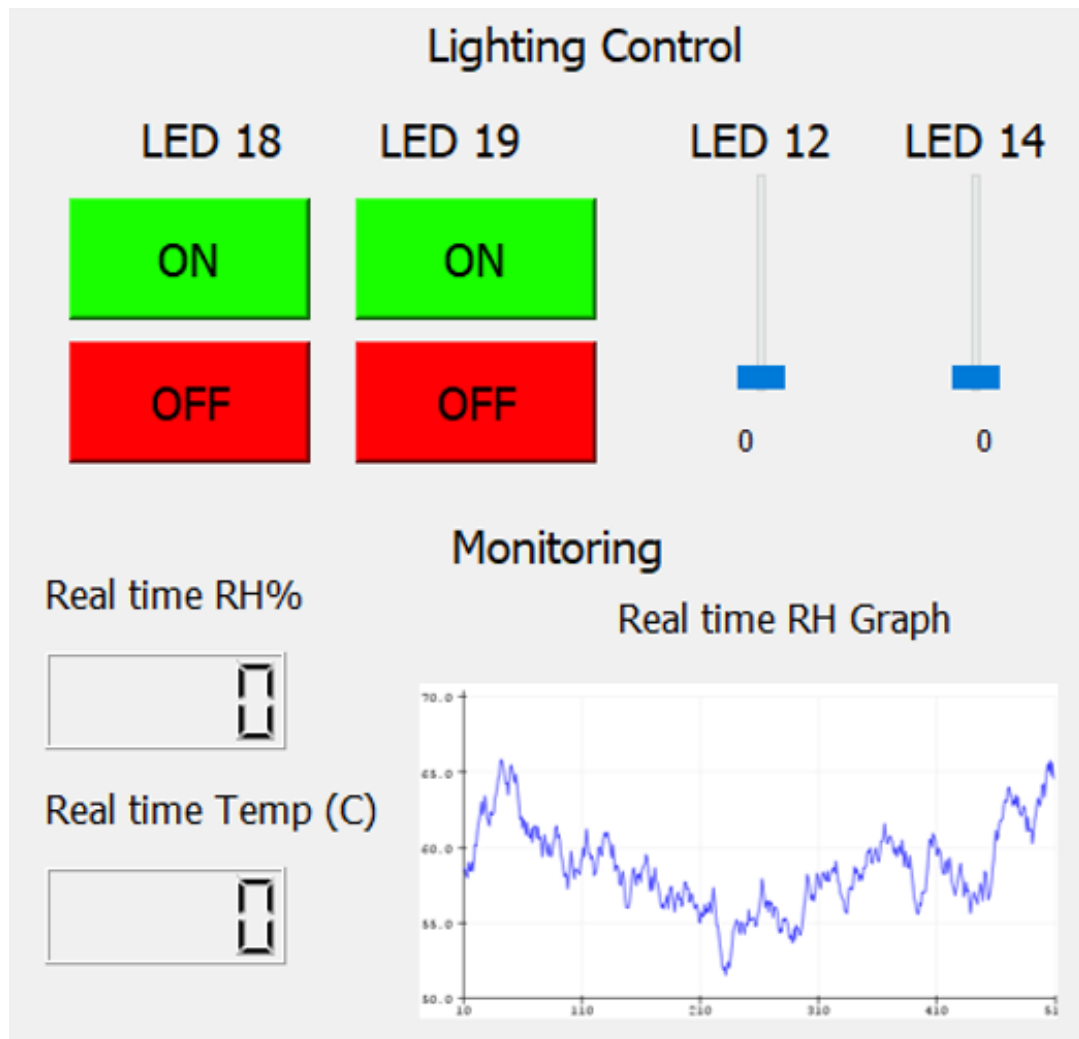
docs2["sensor"] = "humid";
docs2["val"] = humid;

serializeJson(docs1,out4);
serializeJson(docs2,out5);
mqtt.publish("inno/b6214197/lab03/sensor",out4);
delay(1000);
mqtt.publish("inno/b6214197/lab03/sensor",out5);
delay(1000);

}
}

```

4. จงออกแบบ GUI ดังรูป จากนั้นเขียนโปรแกรมเพื่อให้ทำงานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ในข้อที่ผ่านมา โดยมีการทำงานดังต่อไปนี้
- ปุ่มกดชุดที่ 1 จะเป็นการเปิดปิดหลอดแอลอีดีที่ต่ออยู่กับขา 18
  - ปุ่มกดชุดที่ 2 จะเป็นการเปิดปิดหลอดแอลอีดีที่ต่ออยู่กับขา 19
  - สไลเดอร์ชุดที่ 1 ใช้สำหรับปรับความสว่างของหลอด LED ที่ต่ออยู่กับขา 12
  - สไลเดอร์ชุดที่ 2 ใช้สำหรับปรับความสว่างของหลอด LED ที่ต่ออยู่กับขา 14
  - LCD display ใช้สำหรับแสดงค่าความชื้นและอุณหภูมิ
  - Realtime graph ใช้สำหรับแสดงค่าความชื้น



```
from PyQt5 import QtCore, QtGui, QtWidgets
from lab3 import Ui_MainWindow
import sys
import json
import paho.mqtt.client as mqtt
import time
import pyqtgraph as pg
import numpy as np
broker_address="electsut.trueddns.com"
broker_port=27860
client = mqtt.Client("B6222932_1")
client.connect(broker_address,broker_port)
app = QtWidgets.QApplication(sys.argv)
MainWindow = QtWidgets.QMainWindow()
class mylab3(Ui_MainWindow):
```

```

def __init__(self) -> None:
    super().setUpUi(MainWindow)
    self.gcn()
    self.val={}
    self.valsld={}
    self.valsld1={}
    client.on_message = self.mycallback
    client.subscribe("inno/b6214197/lab03/sensor")
    client.loop_start()
    self.valrh=0
    self.time()
    self.mygraph=pg.PlotWidget(self.centralwidget)
    self.mygraph.setGeometry(QtCore.QRect(200,270,261,141))
    self.x1 = np.arange(100)
    self.y1= np.arange(100)
    self.dataline= self.mygraph.plot()
    self.mygraph.setYRange(0,100)

def gcn(self):
    self.on18.clicked.connect(self.onclick18)
    self.off18.clicked.connect(self.offclick18)
    self.on19.clicked.connect(self.onclick19)
    self.off19.clicked.connect(self.offclick19)
    self.slider12.sliderReleased.connect(self.fslider12)
    self.slider14.sliderReleased.connect(self.fslider14)

def autoplot(self):
    self.x1=np.append(self.x1[1:], self.x1[-1]+1)
    self.y1=np.append(self.y1[1:],self.valrh)
    self.dataline.setData(self.x1,self.y1)

def time(self):
    self.mytm1=QtCore.QTimer()
    self.mytm1.timeout.connect(self.autoplot)
    self.mytm1.setInterval(10)
    self.mytm1.start()

def onclick18(self):
    print("onclick18")
    self.val["LED"]=1
    self.val["state"]="on"
    self.valjson = json.dumps(self.val)
    client.publish("inno/b6214197/lab03/lighting",self.valjson)

def offclick18(self):

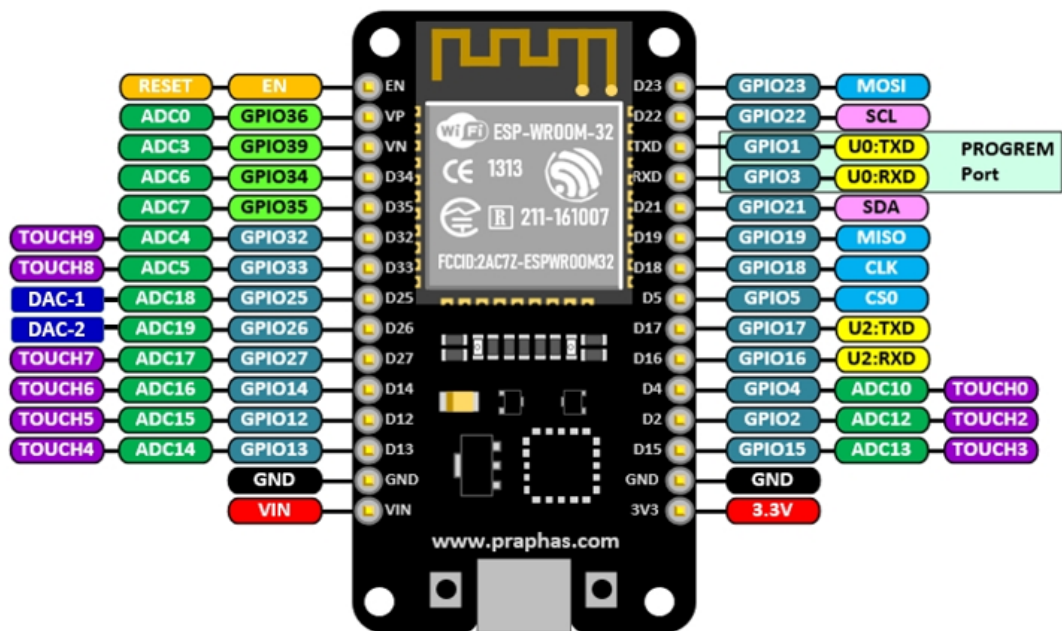
```

```

        print("offclick18")
        self.val["LED"]=1
        self.val["state"]="off"
        self.valjson = json.dumps(self.val)
        client.publish("inno/b6214197/lab03/lighting",self.valjson)
def onclick19(self):
    print("onclick19")
    self.val["LED"]=2
    self.val["state"]="on"
    self.valjson = json.dumps(self.val)
    client.publish("inno/b6214197/lab03/lighting",self.valjson)
def offclick19(self):
    print("offclick19")
    self.val["LED"]=2
    self.val["state"]="off"
    self.valjson = json.dumps(self.val)
    client.publish("inno/b6214197/lab03/lighting",self.valjson)
def fslider12(self):
    self.valsld["pwm"]=1
    self.valsld["val"]= self.slider12.value()
    self.valsldjson=json.dumps(self.valsld)
    print(self.valsldjson)
    client.publish("inno/b6214197/lab03/pwm",self.valsldjson)
def fslider14(self):
    self.valsld["pwm"]=2
    self.valsld["val"]= self.slider14.value()
    self.valsldjson=json.dumps(self.valsld)
    print(self.valsldjson)
    client.publish("inno/b6214197/lab03/pwm",self.valsldjson)
def mycallback(self,client,userdata,message):
    self.valpayload = str(message.payload.decode("utf-8"))
    self.send = json.loads(self.valpayload)
    self.sensorval = self.send["sensor"]
    self.val12 = self.send["val"]
    if self.sensorval == "temp":
        self.realtimetemp.display(self.val12)
    if self.sensorval == "humid":
        self.valrh = self.val12
        self.realtimerh.display(self.val12)
if __name__ == "__main__":

```

```
ui = mylab3()
MainWindow.show()
sys.exit(app.exec_())
```



GPIO	Input	Output	Notes	ADC	Touch	PWM
0	pulled up	OK	outputs PWM signal at boot			OK
1	TX pin	OK	debug output at boot			OK
2	OK	OK	connected to on-board LED	ADC12	2	OK
3	OK	RX pin	HIGH at boot			OK
4	OK	OK		ADC10	0	OK
5	OK	OK	outputs PWM signal at boot			OK
6	x	x	connected to SPI flash			x
7	x	x	connected to SPI flash			x
8	x	x	connected to SPI flash			x
9	x	x	connected to SPI flash			x
10	x	x	connected to SPI flash			x
11	x	x	connected to SPI flash			x
12	OK	OK	boot fail if pulled high	ADC15	5	OK
13	OK	OK		ADC14	4	OK
14	OK	OK	outputs PWM signal at boot	ADC16	6	OK
15	OK	OK	outputs PWM signal at boot	ADC13	3	OK
16	OK	OK				OK
17	OK	OK				OK
18	OK	OK				OK
19	OK	OK				OK
21	OK	OK				OK
22	OK	OK				OK
23	OK	OK				OK
25	OK	OK		ADC18		OK
26	OK	OK		ADC19		OK
27	OK	OK		ADC17	7	OK
32	OK	OK		ADC4	9	OK
33	OK	OK		ADC5	8	OK
34	OK	x	input only	ADC6		x
35	OK	x	input only	ADC7		x
36	OK	x	input only	ADC0		x
39	OK	x	input only	ADC3		x