

Lab PIR

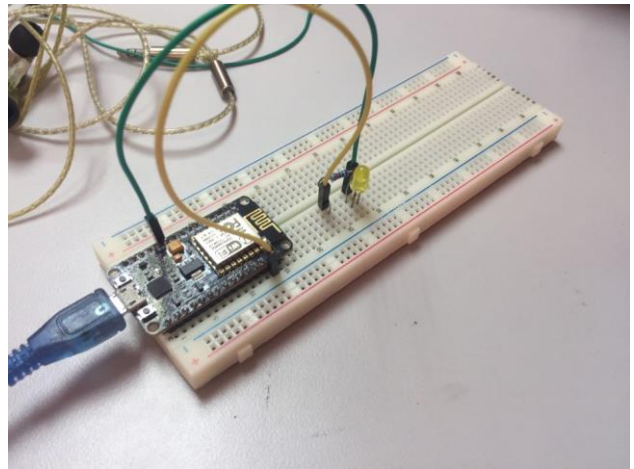
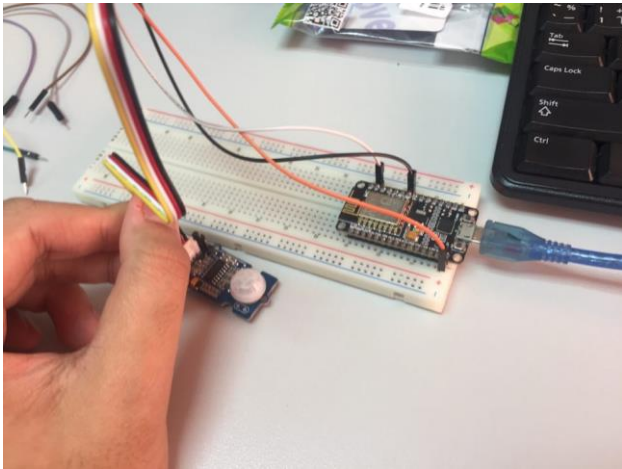
หลักการทำงานของ PIR และ Cloud MQTT


PIR คือ อุปกรณ์ Sensor ชนิดหนึ่งที่ใช้ตรวจจับคลื่นรังสี Infrared ที่แผ่จาก มนุษย์ หรือ สัตว์ ที่มีการเคลื่อนไหว ทำให้มีการนำเอา PIR มาประยุกต์ใช้งานกันเป็นอย่างมากใช้เพื่อตรวจจับการเคลื่อนไหวของสิ่งมีชีวิต หรือ ตรวจจับการบุกรุกในงานรักษาความปลอดภัย

MQTT คือช่องทางสื่อสารสำหรับโปรแกรมชนิดหนึ่งที่ทำงานลักษณะแบบเดียวกับ Cloud แต่มันตรงที่ตัว MQTTจะทำงานด้าน รับ-ส่ง ข้อมูลเพียงอย่างเดียว ข้อดีคือสามารถข้ามแพลตฟอร์ม ในการเขียนโปรแกรมได้โดยมี ลักษณะการเรียกใช้งานไม่ต่างกันมาก จึงช่วยลดเวลาในการพัฒนาโปรเจกต์ได้เยอะพอสมควร ในการใช้งาน MQTT เราสามารถเลือกใช้งานในรูปแบบ Local Network หรือ Cloud ก็ได้ ตามความเหมาะสมของลักษณะโปรเจกต์

Cloud MQTT คือบริการ MQTT Service บน Cloud Service เป็นตัวกลางในการสื่อสารระหว่าง อุปกรณ์ IoT โดยใช้ Protocol MQTT ในการส่งข้อมูล

รูปการต่อวงจร และ Cloud MQTT



 **CloudMQTT** IoT

5706021612072@fitm.kmutnb.ac.th

DETAILS USERS BRIDGES AMAZON KINESIS STREAM LOG WEBSOCKET UI

Details

Server m11.cloudmqtt.com

User vssyxoyj Restart

Password jYOX1aH4jAyZ


Port 17928

SSL Port 27928

Websockets Port (TLS only) 37928

Connection limit 10

Active Plan



Cute Cat

Upgrade Instance

MENU

- Home
- Plans
- Documentation
- Blog
- About

MORE

- Status
- Terms of Service
- Program Policies
- Privacy Policy
- Security Policy
- Imprint

CloudMQTT

Contact Support

Open 24 hours a day, 7 days a week

© Copyright 2011-2016 CloudMQTT.

Chat now

Code

ผังส่ง

```
#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const char* ssid = "itfitm";

const char* password = "jYOX1aM4jAyZ";

const char* mqtt_server = "m11.cloudmqtt.com";

char msg[50];

WiFiClient espClient;

PubSubClient client(espClient);

void setup(){

    Serial.begin(115200);

    pinMode(D1, INPUT);

    setup_wifi();

    client.setServer(mqtt_server, 17928); //port ใน mqtt

    client.setCallback(callback);

}
```

```
void setup_wifi() {

    delay(10);

    // We start by connecting to a WiFi network

    Serial.println();

    Serial.print("Connecting to ");

    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {

        delay(500);

        Serial.print(".");

    }

    Serial.println("");

    Serial.println("WiFi connected");

    Serial.println("IP address: ");

    Serial.println(WiFi.localIP());

}

void callback(char* topic, byte* payload, unsigned int length) {

    Serial.print("Message arrived [");

    Serial.print(topic);
```

```

    Serial.print("] ");
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");

        // Attempt to connect
        if (client.connect("pir", "vssyxoyj", "jYOX1aM4jAyZ")) { //
topic,username,password
            Serial.println("connected");

            // Once connected, publish an announcement...
            // client.publish("iot", "Start");

            // ... and resubscribe
            //client.subscribe("Node2");

            // client.publish("/checkPIR", "Hi");
        } else {
            Serial.print("failed, rc=");

            Serial.print(client.state());

            Serial.println(" try again in 5 seconds");

            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

```

```
void loop(){

    int value= digitalRead(D1);

    //Serial.println(value);

    // if (value == HIGH)

    // {

    //   Serial.println("yes");

    //   //delay(1000) ;

    // }

    //

    // else

    // {

    //   Serial.println("no");

    // }

}

if (!client.connected()) {

    reconnect();

}

client.loop();

// char gas[50];

// float sensor_volt;

// float sensorValue;
```

```
// sensorValue = analogRead(A0);

// sensor_volt = sensorValue / 1024 * 5.0;

delay(1000);

// itoa(sensor_volt,gas,10);

// snprintf (msg,75,value);

//msg = value;

//String msg = String(value, DEC);

//Serial.println(msg);

if (value == HIGH)

{

    client.publish("/checkPIR", "1");

    Serial.println("yes");

    //delay(1000) ;

}

else

{

    client.publish("/checkPIR", "0");

    Serial.println("no");

}

// client.publish("/checkPIR", msg);

}
```

ผังรีบ

```
#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const char* ssid = "itfitm";

const char* password = "";


#define mqtt_server "m11.cloudmqtt.com"

#define mqtt_port 17928

#define mqtt_user "vssyxoyj"

#define mqtt_password "jYOX1aM4jAyZ"

float oldTemp = 0.0;

WiFiClient espClient;

PubSubClient client(espClient);

void setup() {

    pinMode(D1, OUTPUT);

    Serial.begin(115200);

    // lcd.setCursor(2, 0);

    // lcd.print("nuttapun");

    delay(10);


    Serial.println();

    Serial.print("Connecting to ");

    Serial.println(ssid);
```



```
WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}
```

```
Serial.println("");  
Serial.println("WiFi connected");  
Serial.println("IP address: ");  
Serial.println(WiFi.localIP());
```

```
client.setServer(mqtt_server, mqtt_port);  
client.setCallback(callback);  
}
```

```
void loop() {  
    if (!client.connected()) {  
        Serial.print("Attempting MQTT connection...");  
        if (client.connect("iotsub", mqtt_user, mqtt_password)) {  
            Serial.println("connected");  
            client.subscribe("/checkPIR");  
        } else {
```

```
    Serial.print("failed, rc=");  
  
    Serial.print(client.state());  
  
    Serial.println(" try again in 5 seconds");  
  
    delay(5000);  
  
    return;  
}  
  
}  
  
client.loop();  
}
```

```
void callback(char* topic, byte* payload, unsigned int length) {  
  
    //Serial.print("Message arrived [");  
  
    //Serial.print(topic);  
  
    String msg = "";  
  
    String to = "";  
  
    int i = 0;  
  
    while (i < length) msg += (char)payload[i++];  
  
    // Serial.println(msg);  
  
    to = topic;  
  
    // Serial.print(to);  
  
    if (to == "/checkPIR") {  
  
        Serial.println(msg);  
  
        if (msg.toFloat() == 1) {  
  
            digitalWrite(D1, HIGH);  
  

```

```
    } else {  
        digitalWrite(D1, LOW);  
    }  
}  
}
```