

แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต
IoT Approaches to Manufacturing System

ชื่อ-สกุล : B6304577 นายภาณุพงศ์ แคนอินทร์

4/4. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_201 – Web Control 2 LED

- อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 2 ดวง
- https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzukuLbMxV3pOHY4YIPuLEz8-ZzTOX2VhWxcH2QjLGk

← → ↻ Not secure | 192.168.43.237/led1off

LED Status

LED1-Off, LED2-Off

LED1 On
LED2 On

LED1 Off
LED2 Off

< Test Code >

```
#include <WiFiManager.h> // https://github.com/tzapu/WiFiManager
#include <WiFi.h>

WiFiServer server(80);

#define LED1 18
#define LED2 19

void setup() {
  // WiFi.mode(WIFI_STA); // explicitly set mode, esp defaults to STA+AP
  // it is a good practice to make sure your code sets wifi mode how you want it.

  // put your setup code here, to run once:
  Serial.begin(115200);

  //WiFiManager, Local initialization. Once its business is done, there is no need to keep it around
  WiFiManager wm;

  // reset settings - wipe stored credentials for testing
  // these are stored by the esp library
  // wm.resetSettings();

  // Automatically connect using saved credentials,
  // if connection fails, it starts an access point with the specified name ( "AutoConnectAP"),
  // if empty will auto generate SSID, if password is blank it will be anonymous AP (wm.autoConnect())
  // then goes into a blocking loop awaiting configuration and will return success result

  bool res;
  // res = wm.autoConnect(); // auto generated AP name from chipid
  // res = wm.autoConnect("AutoConnectAP"); // anonymous ap
  res = wm.autoConnect("ESP32AutoConnectAP", "12345678"); // password protected ap

  if (!res) {
    Serial.println("Failed to connect");
    // ESP.restart();
  }
}
```

```

    }
    else {
        //if you get here you have connected to the WiFi
        Serial.println("connected...yeey :)");
        Serial.println(res);
    }
    server.begin();

    pinMode(LED1, OUTPUT); // set the LED pin mode
    pinMode(LED2, OUTPUT);
}

bool LED1_Status, LED2_Status, LED3_Status, LED4_Status = LOW;
void loop() {
    // put your main code here, to run repeatedly:
    WiFiClient client = server.available();
    digitalWrite(LED1, LED1_Status);
    digitalWrite(LED2, LED2_Status);
    if (client) {
        Serial.println("New Client.");
        String currentLine = "";
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                Serial.write(c);
                if (c == '\n') {
                    if (currentLine.length() == 0) {
                        client.println("HTTP/1.1 200 OK");
                        client.println("Content-type:text/html");
                        client.println();
                        client.println("<html>");
                        client.println("<body>");
                        client.println("<h1>LED Status</h1>");

                        client.println("<p>");
                        if (LED1_Status == HIGH & LED2_Status == HIGH)
                            client.println("LED1-On,LED2-On");
                        else if (LED1_Status == HIGH & LED2_Status == LOW)
                            client.println("LED1-On,LED2-Off");
                        else if (LED1_Status == LOW & LED2_Status == HIGH)
                            client.println("LED1-Off,LED2-On");
                        else
                            client.println("LED1-Off,LED2-Off");

                        client.println("<p>");
                        client.println("<h1>LED Control</h1>");
                        client.println("<p>");
                        client.println("<a href=\\\"/led1on\\\"><button style = \\\"background-color: #f44336;\\\">LED1
On</button></a>");
                        client.println("<a href=\\\"/led2on\\\"><button style = \\\"background-color: #f44336;\\\">LED2
On</button></a>");
                        client.println("</p>");
                        client.println("<a href=\\\"/led1off\\\"><button style = \\\"background-color: #008CBA;\\\">LED1
Off</button></a>");
                        client.println("<a href=\\\"/led2off\\\"><button style = \\\"background-color: #008CBA;\\\">LED2
Off</button></a>");

                        client.println("<body>");
                        client.println("<html>");
                        break;
                    } else {
                        currentLine += c;
                    }
                } else if (c != '\r') {
                    currentLine += c;
                }
            }
            if (currentLine.endsWith("GET /led1on")) LED1_Status = HIGH;
            if (currentLine.endsWith("GET /led1off")) LED1_Status = LOW;

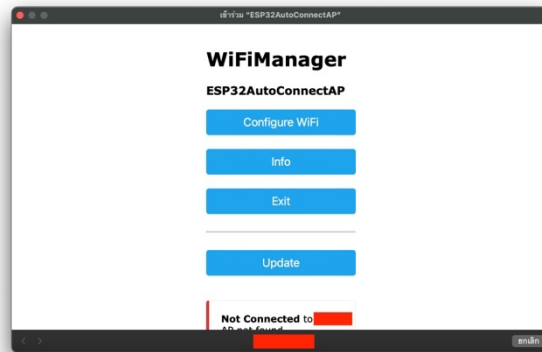
```

```

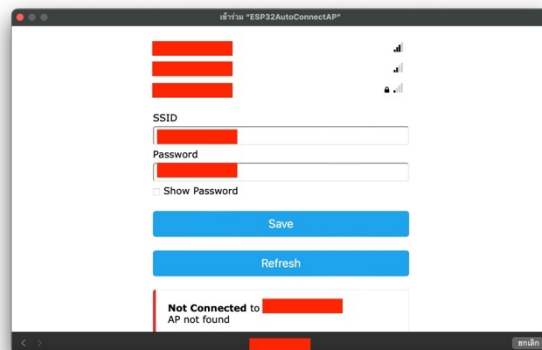
        if (currentLine.endsWith("GET /led2on")) LED2_Status = HIGH;
        if (currentLine.endsWith("GET /led2off")) LED2_Status = LOW;
    }
}
client.stop();
Serial.println("Client Disconnected.");
}
}

```

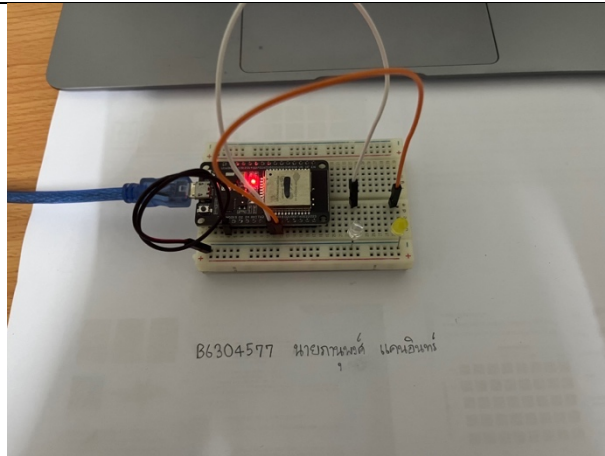
รูปการต่อวงจร – 1



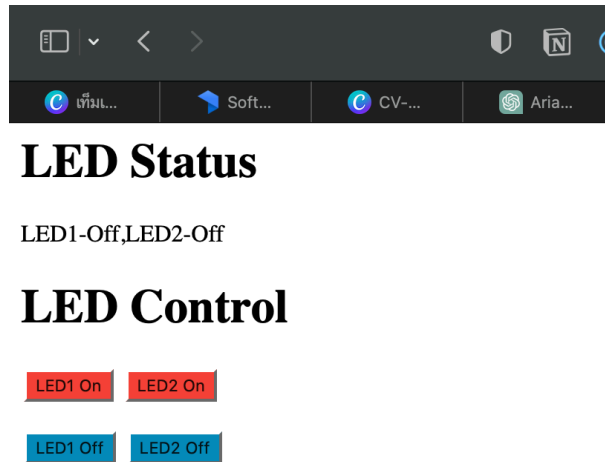
รูปการต่อวงจร – 2



รูปการต่อวงจร – 3



หน้าจอ Web Control



Quiz_202 – Web Control 4 LED and Monitor Humid/Temperature

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
- อยากมีกกด Link ไปที่หน้า FB ของตัวเอง

← → ↻ ⓘ Not secure | 192.168.43.237

The ESP-32 Update web page without refresh

LED1 ON

LED2 ON

LED3 ON

LED4 ON

LED1 OFF

LED2 OFF

LED3 OFF

LED4 OFF

State of [LED1, LED2, LED3, LED4] is >> ON, OFF, OFF, ON

DHT-22 sensor : Temp = 28.10 C, Humidity = 43.90 %

[By Wichai Srisuruk](#)

< Test Code >

ส่วนที่ 1 Quiz_202.ino

```
#include <WiFiManager.h> // https://github.com/tzapu/WiFiManager
#include <WiFiClient.h>
#include <WebServer.h>
#include <DHT.h> //https://www.arduino-libraries.info/libraries/dht-sensor-library
#include "index.h"

#define DHT_SENSOR_PIN 15 // ESP32 pin GPIO15 connected to DHT22 sensor
#define DHT_SENSOR_TYPE DHT22
#define LED1 18
#define LED2 19
#define LED3 22
#define LED4 23

DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);
WebServer server(80);

String ledState1 = "NA";
String ledState2 = "NA";
String ledState3 = "NA";
String ledState4 = "NA";

void handleRoot() {
  String s = MAIN_page;
  server.send(200, "text/html", s);
}

void handleADC() {
  // read humidity
  float humi = dht_sensor.readHumidity();
  // read temperature in Celsius
  float tempC = dht_sensor.readTemperature();
  // read temperature in Fahrenheit
  float tempF = dht_sensor.readTemperature(true);

  String tmpValue = "Temp = ";
  tmpValue += String(tempC) + " C, Humidity = ";
```

```

    tmpValue += String(humi) + " %";
    server.send(200, "text/plain", tmpValue);
}

void handleLED() {
    String t_state = server.arg("LEDstate"); //Refer xhttp.open("GET", "setLED?LEDstate="+led, true);
    Serial.println(t_state);
    if (t_state == "11") {
        digitalWrite(LED1, HIGH);
        ledState1 = "ON";
    }
    if (t_state == "10") {
        digitalWrite(LED1, LOW);
        ledState1 = "OFF";
    }
    if (t_state == "21") {
        digitalWrite(LED2, HIGH);
        ledState2 = "ON";
    }
    if (t_state == "20") {
        digitalWrite(LED2, LOW);
        ledState2 = "OFF";
    }
    if (t_state == "31") {
        digitalWrite(LED3, HIGH);
        ledState3 = "ON";
    }
    if (t_state == "30") {
        digitalWrite(LED3, LOW);
        ledState3 = "OFF";
    }
    if (t_state == "41") {
        digitalWrite(LED4, HIGH);
        ledState4 = "ON";
    }
    if (t_state == "40") {
        digitalWrite(LED4, LOW);
        ledState4 = "OFF";
    }
    server.send(200, "text/plain", ledState1 + ", " + ledState2 + ", " + ledState3 + ", " + ledState4);
}

void setup() {

    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
    dht_sensor.begin(); // initialize the DHT sensor

    // WiFi.mode(WIFI_STA); // explicitly set mode, esp defaults to STA+AP
    // it is a good practice to make sure your code sets wifi mode how you want it.

    // put your setup code here, to run once:
    Serial.begin(115200);

    //WiFiManager, Local initialization. Once its business is done, there is no need to keep it around
    WiFiManager wm;

    // reset settings - wipe stored credentials for testing
    // these are stored by the esp library
    // wm.resetSettings();

    // Automatically connect using saved credentials,
    // if connection fails, it starts an access point with the specified name ( "AutoConnectAP"),
    // if empty will auto generate SSID, if password is blank it will be anonymous AP (wm.autoConnect())
    // then goes into a blocking loop awaiting configuration and will return success result

    bool res;
    // res = wm.autoConnect(); // auto generated AP name from chipid

```

```
// res = wm.autoConnect("AutoConnectAP"); // anonymous ap
res = wm.autoConnect("ESP32AutoConnectAP", "12345678"); // password protected ap

if (!res) {
  Serial.println("Failed to connect");
  // ESP.restart();
}
else {
  //if you get here you have connected to the WiFi
  Serial.println("connected...yeey :)");
  Serial.println(res);
}
server.begin();
server.on("/", handleRoot);
server.on("/setLED", handleLED);
server.on("/readADC", handleADC);
Serial.println("HTTP server started");

}

bool LED1_Status, LED2_Status, LED3_Status, LED4_Status = LOW;
void loop() {
  // put your main code here, to run repeatedly:
  server.handleClient();
}

```

ส่วนที่ 2 index.h

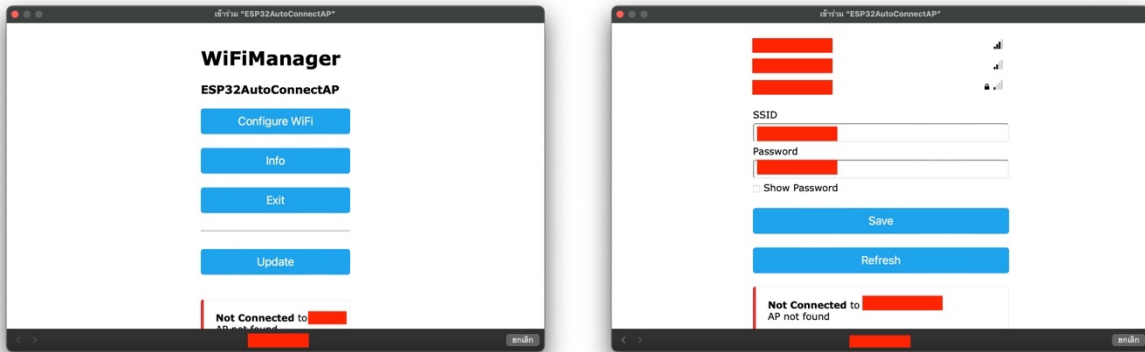
```
const char MAIN_page[] PROGMEM = R"====(
<!DOCTYPE html>
<html>
<body>
  <div id="demo">
    <h1>The ESP-32 Update web page without refresh</h1>
    <button type="button" onclick="sendData(11)" style="background: rgb(202, 60, 60); height: 40px; width: 100px">LED1 ON</button>
    <button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60); height: 40px; width: 100px">LED2 ON</button>
    <button type="button" onclick="sendData(31)" style="background: rgb(202, 60, 60); height: 40px; width: 100px">LED3 ON</button>
    <button type="button" onclick="sendData(41)" style="background: rgb(202, 60, 60); height: 40px; width: 100px">LED4 ON</button><br><br>
    <button type="button" onclick="sendData(10)" style="background: rgb(100,116,255); height: 40px; width: 100px">LED1 OFF</button>
    <button type="button" onclick="sendData(20)" style="background: rgb(100,116,255); height: 40px; width: 100px">LED2 OFF</button>
    <button type="button" onclick="sendData(30)" style="background: rgb(100,116,255); height: 40px; width: 100px">LED3 OFF</button>
    <button type="button" onclick="sendData(40)" style="background: rgb(100,116,255); height: 40px; width: 100px">LED4 OFF</button><br><br>
    State of [LED1, LED2, LED3, LED4] is >> <span id="LEDState">NA</span><br>
  </div>
  <div>
    <br>DHT-22 sensor : <span id="ADCValue">0</span><br>
  </div>
  <script>
    function sendData(led) {
      var xhttp = new XMLHttpRequest();
      xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
          document.getElementById("LEDState").innerHTML =
            this.responseText;
        }
      };
      xhttp.open("GET", "setLED?LEDstate="+led, true);
      xhttp.send();
    }
  </script>
)====
```

```

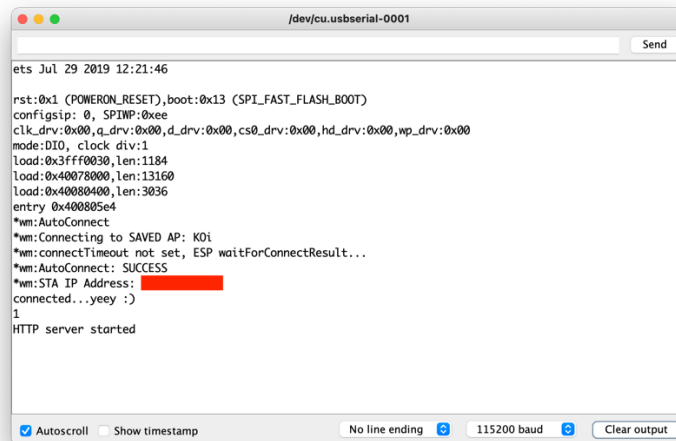
setInterval(function() {
// Call a function repetatively with 2 Second interval
getData();
}, 2000);
function getData() {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("ADCValue").innerHTML =
this.responseText;
}
};
xhttp.open("GET", "readADC", true);
xhttp.send();
}
</script>
<br><a href="https://www.facebook.com/profile.php?id=100006222361707">By Panupong Kaenin</a>
<br><a href="https://github.com/panupongKanin">Github By Panupong Kaenin</a>
</body>
</html>
)=====";

```

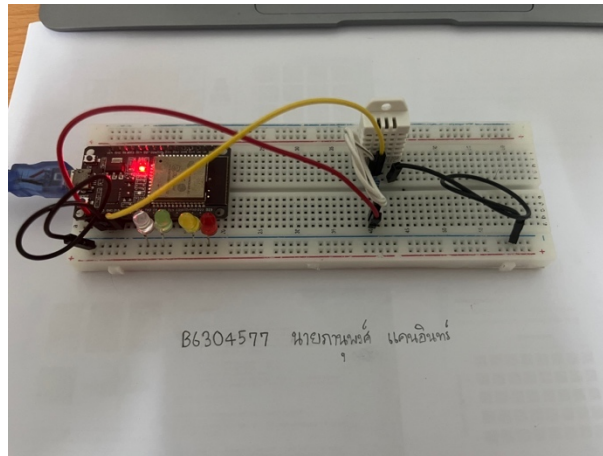
รูปการต่อวงจร – 1



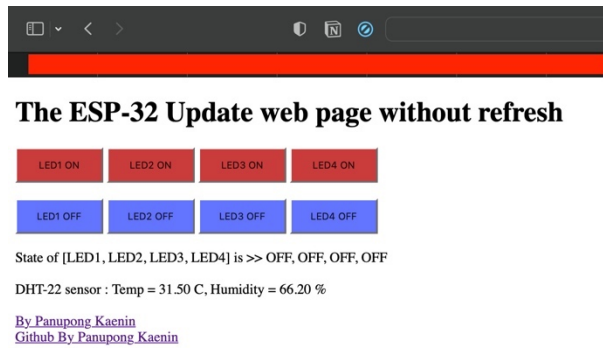
รูปการต่อวงจร – 2



รูปการต่อวงจร – 3



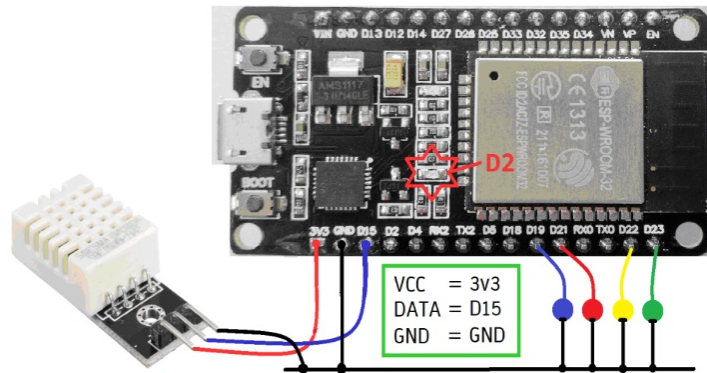
หน้าจอ Web Control



Quiz_203 – Publish

- อ่านค่า DHT-22 แล้วส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
- ควบคุมการแสดงผลให้ 4 LED แสดงผลตามข้อกำหนดดังนี้

* ○ ○ ○ (Blink)	หากการอ่านค่าแล้วเป็น null, หรือไม่มีเซ็นเซอร์
● ○ ○ ○	ช่วงของอุณหภูมิ (-∞, 24)
● ● ○ ○	ช่วงของอุณหภูมิ [24,26)
● ● ● ○	ช่วงของอุณหภูมิ [26,28)
● ● ● ●	ช่วงของอุณหภูมิ [28,30)
* * * * (Blink)	ช่วงของอุณหภูมิ [30,∞)



< Test Code >

```
#include <WiFiManager.h> // https://github.com/tzapu/WiFiManager
#include <PubSubClient.h>
#include <WiFi.h>
#include <DHT.h> //https://www.arduino-libraries.info/libraries/dht-sensor-library
#define DHT_SENSOR_PIN 15 // ESP32 pin GPIO21 connected to DHT22 sensor
#define DHT_SENSOR_TYPE DHT22

DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);

const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "HomeArm1234";

WiFiClient espClient;
PubSubClient client(espClient);

long lastMsg = 0;
char msg[50];
#define LED1 18
#define LED2 19
#define LED3 22
#define LED4 23

void LED_state(float temp)
{ if (temp < 24)
  { digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
  }
  else if (temp >= 24 && temp < 26)
  { digitalWrite(LED1, HIGH);
```

```

    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
}
else if (tempp >= 26 && tempp < 28)
{ digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, LOW);
}
else if (tempp >= 28 && tempp < 30)
{ digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, HIGH);
}
else if (tempp >= 30)
{ digitalWrite(LED1, HIGH); digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH); digitalWrite(LED4, HIGH); delay(250);
  digitalWrite(LED1, LOW); digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW); digitalWrite(LED4, LOW); delay(250);
}
else
{ digitalWrite(LED1, HIGH); delay(250);
  digitalWrite(LED1, LOW); delay(250);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);
}
}

void reconnect()
{ while (!client.connected()) // Loop until we're reconnected
  { Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32 Client-";
    clientId += String(random(0xffff), HEX); // Create a random client ID
    if (client.connect(clientId.c_str())) // Attempt to connect
    { Serial.println("connected"); // Once connected, publish an announcement...
      client.publish(topic1, "Hello"); // ... and resubscribe
      client.subscribe(topic1);
    } else
    { Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {

  dht_sensor.begin(); // initialize the DHT sensor
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);

  // WiFi.mode(WIFI_STA); // explicitly set mode, esp defaults to STA+AP
  // it is a good practice to make sure your code sets wifi mode how you want it.

  // put your setup code here, to run once:
  Serial.begin(115200);

  //WiFiManager, Local initialization. Once its business is done, there is no need to keep it around
  WiFiManager wm;

  // reset settings - wipe stored credentials for testing
  // these are stored by the esp library
  // wm.resetSettings();

```

```

// Automatically connect using saved credentials,
// if connection fails, it starts an access point with the specified name ( "AutoConnectAP"),
// if empty will auto generate SSID, if password is blank it will be anonymous AP (wm.autoConnect())
// then goes into a blocking loop awaiting configuration and will return success result

bool res;
// res = wm.autoConnect(); // auto generated AP name from chipid
// res = wm.autoConnect("AutoConnectAP"); // anonymous ap
res = wm.autoConnect("ESP32AutoConnectAP", "12345678"); // password protected ap

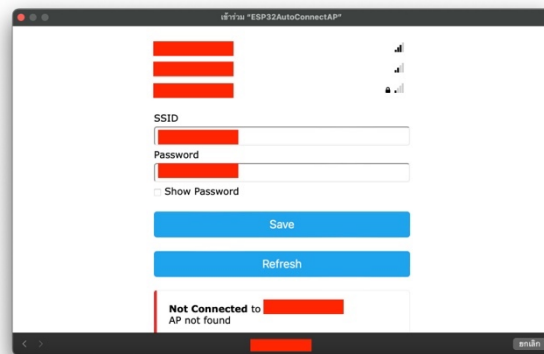
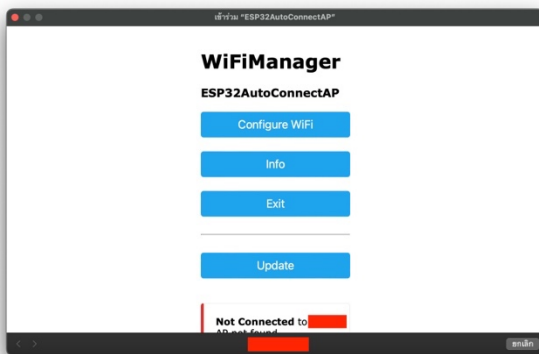
if (!res) {
    Serial.println("Failed to connect");
    // ESP.restart();
}
else {
    //if you get here you have connected to the WiFi
    Serial.println("connected...yeey :)");
    Serial.println(res);
}
client.setServer(mqtt_server, 1883);
}

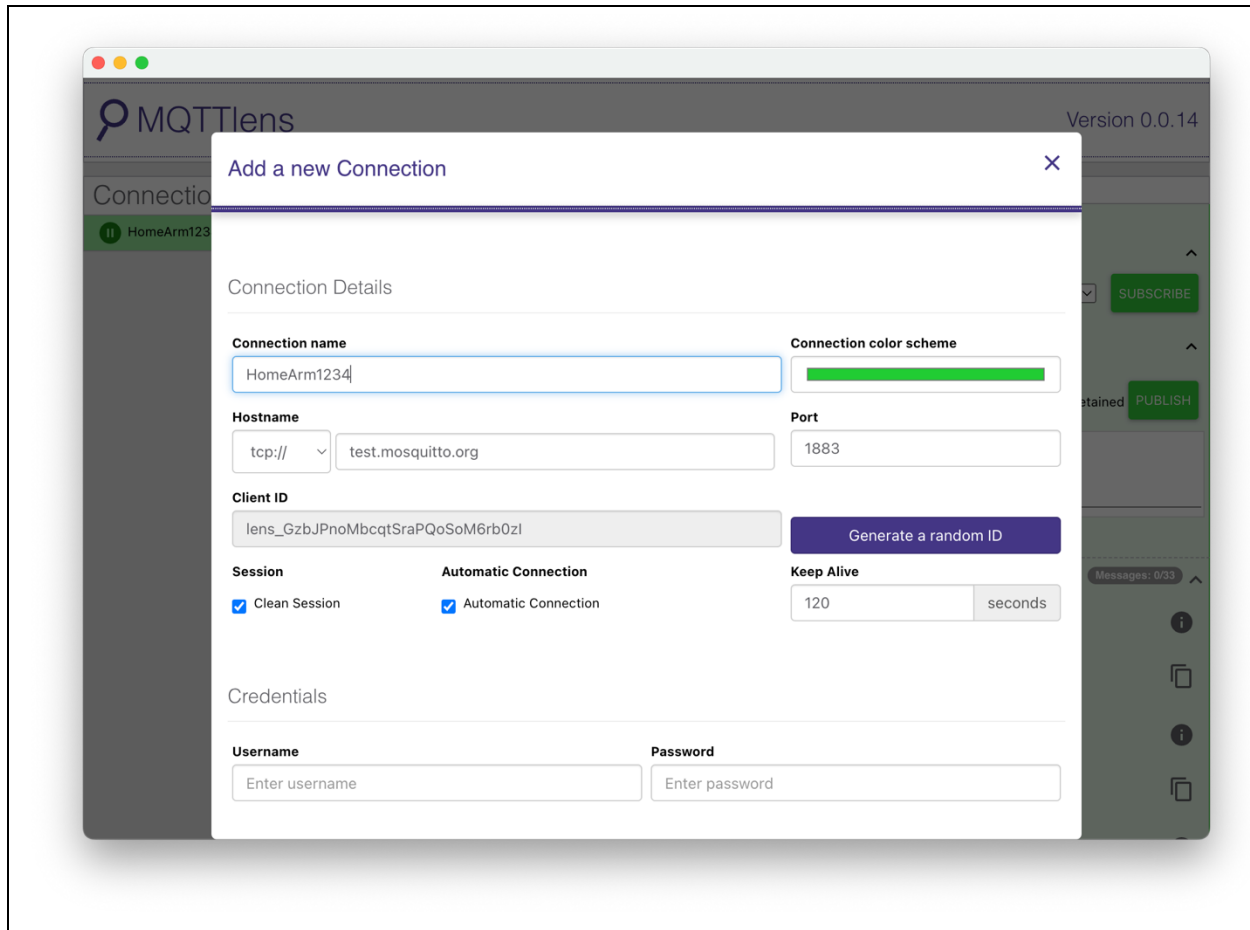
void loop() {
    // put your main code here, to run repeatedly:
    if (!client.connected()) reconnect();
    client.loop();
    long now = millis();
    // read humidity
    float humi = dht_sensor.readHumidity();
    // read temperature in Celsius
    float tempC = dht_sensor.readTemperature();
    // read temperature in Fahrenheit
    float tempF = dht_sensor.readTemperature(true);

    LED_state(tempC);
    if (now - lastMsg > 5000)
    { lastMsg = now;
      sprintf (msg, 75, "Temp: %.2f'C Humid: %.2f%%", tempC, humi);
      Serial.print("Publish message: ");
      Serial.println(msg);
      client.publish(topic1, msg);
    }
}

```

รูปการต่อวงจร – 1





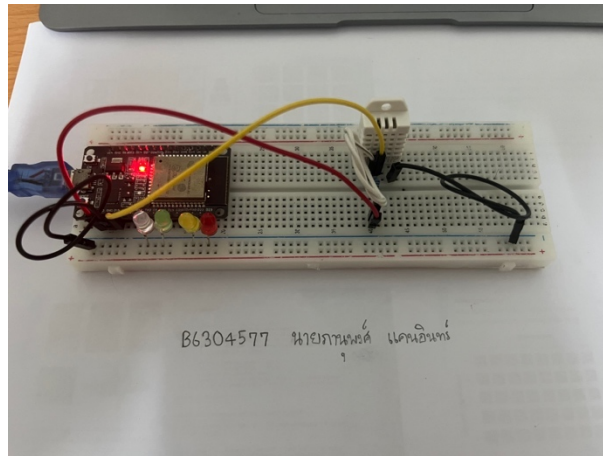
รูปการต่อวงจร - 2

```

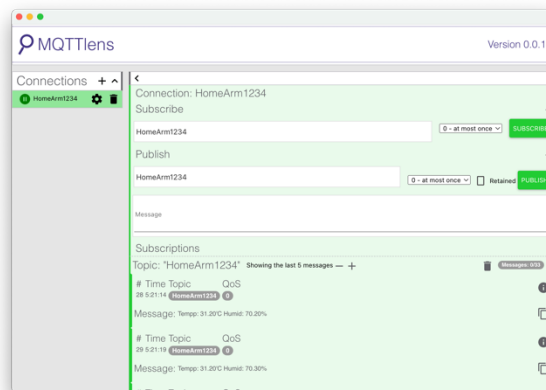
/dev/cu.usbserial-0001
Send
ets Jul 29 2019 12:21:46
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1184
load:0x40078000,len:13160
load:0x40080400,len:3036
entry 0x400805e4
*wm:AutoConnect
*wm:Connecting to SAVED AP: 
*wm:connectTimeout not set, ESP waitForConnectResult...
*wm:AutoConnect: SUCCESS
*wm:STA IP Address: 
connected...yeey :)
1
Attempting MQTT connection...connected
Publish message: Temp: 31.40'C Humid: 60.00%
Publish message: Temp: 31.40'C Humid: 91.40%
Publish message: Temp: 31.40'C Humid: 91.10%
Publish message: Temp: 31.40'C Humid: 91.30%
Publish message: Temp: 31.40'C Humid: 91.00%
Autoscroll Show timestamp No line ending 115200 baud Clear output

```

รูปการต่อวงจร – 3

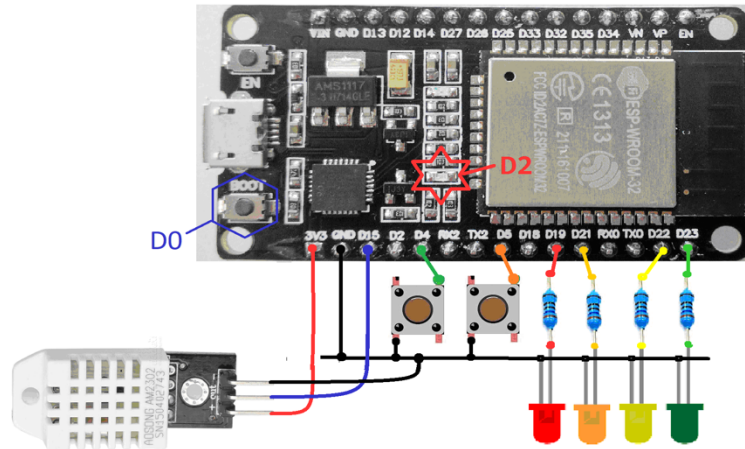


หน้าจอ MQTT Lens



Quiz_204 – Publish and Subscribe

- อ่านค่า DHT-22 แล้วส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
- ควบคุมการปิดเปิด 4 LED
- รับค่าสวิตช์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm



< Test Code >

```
#include <WiFiManager.h> // https://github.com/tzapu/WiFiManager
#include <PubSubClient.h>
#include <WiFi.h>
#include <DHT.h> //https://www.arduino-libraries.info/libraries/dht-sensor-library
#define DHT_SENSOR_PIN 15 // ESP32 pin GPIO21 connected to DHT22 sensor
#define DHT_SENSOR_TYPE DHT22

DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);

const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "HomeArm1234";

WiFiClient espClient;
PubSubClient client(espClient);

long lastMsg = 0;
char msg[50];
#define LED1 18
#define LED2 19
#define LED3 22
#define LED4 23
#define SW1 4
#define SW2 5

void callback(char* topic, byte* payload, unsigned int length)
{
  char myPayload[50];
  Serial.print("Message arrived [");
  Serial.print(topic1);
  Serial.print("] ");
  for (int i = 0; i < length; i++)
  {
    Serial.print((char)payload[i]);
    myPayload[i] = payload[i];
    myPayload[i + 1] = '\0'; // End of String
  }
}
```

```

Serial.print("\n ---> "); Serial.println(myPayload);
myPayload[4] = '\0'; // String less than 4 characters
if ((String)myPayload == "ON")
{
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, HIGH);
}
if ((String)myPayload == "ON1") {
  digitalWrite(LED1, HIGH);
}
if ((String)myPayload == "ON2") {
  digitalWrite(LED2, HIGH);
}
if ((String)myPayload == "ON3") {
  digitalWrite(LED3, HIGH);
}
if ((String)myPayload == "ON4") {
  digitalWrite(LED4, HIGH);
}
if ((String)myPayload == "OFF")
{
  digitalWrite(LED1, LOW); digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW); digitalWrite(LED4, LOW);
}
if ((String)myPayload == "OFF1") {
  digitalWrite(LED1, LOW);
}
if ((String)myPayload == "OFF2") {
  digitalWrite(LED2, LOW);
}
if ((String)myPayload == "OFF3") {
  digitalWrite(LED3, LOW);
}
if ((String)myPayload == "OFF4") {
  digitalWrite(LED4, LOW);
}
}

void reconnect()
{
  while (!client.connected()) // Loop until we're reconnected
  {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32 Client-";
    clientId += String(random(0xffff), HEX); // Create a random client ID
    if (client.connect(clientId.c_str())) // Attempt to connect
    {
      Serial.println("connected"); // Once connected, publish an announcement...
      client.publish(topic1, "Hello"); // ... and resubscribe
      client.subscribe(topic1);
    } else
    {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  dht_sensor.begin(); // initialize the DHT sensor
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(SW1, INPUT_PULLUP);
  pinMode(SW2, INPUT_PULLUP);

  // WiFi.mode(WIFI_STA); // explicitly set mode, esp defaults to STA+AP
  // it is a good practice to make sure your code sets wifi mode how you want it.

```



```

// put your setup code here, to run once:
Serial.begin(115200);

//WiFiManager, Local initialization. Once its business is done, there is no need to keep it around
WiFiManager wm;

// reset settings - wipe stored credentials for testing
// these are stored by the esp library
// wm.resetSettings();

// Automatically connect using saved credentials,
// if connection fails, it starts an access point with the specified name ( "AutoConnectAP"),
// if empty will auto generate SSID, if password is blank it will be anonymous AP (wm.autoConnect())
// then goes into a blocking loop awaiting configuration and will return success result

bool res;
// res = wm.autoConnect(); // auto generated AP name from chipid
// res = wm.autoConnect("AutoConnectAP"); // anonymous ap
res = wm.autoConnect("ESP32AutoConnectAP", "12345678"); // password protected ap

if (!res) {
  Serial.println("Failed to connect");
  // ESP.restart();
}
else {
  //if you get here you have connected to the WiFi
  Serial.println("connected...yeey :)");
  Serial.println(res);
}
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (!client.connected()) reconnect();
  { client.loop();
    if (digitalRead(SW1) == 0)
    { client.loop();
      snprintf (msg, 75, "Overheat Alarm!!!");
      Serial.print("Publish message: ");
      Serial.println(msg);
      client.publish(topic1, msg);
      while (digitalRead(SW1) == 0);
      delay(100);
    }
    if (digitalRead(SW2) == 0)
    { client.loop();
      snprintf (msg, 75, "Intruders Alarm!!!");
      Serial.print("Publish message: ");
      Serial.println(msg);
      client.publish(topic1, msg);
      while (digitalRead(SW2) == 0);
      delay(100);
    }
  }
  long now = millis();
  if (now - lastMsg > 5000)
  { lastMsg = now;
    // read humidity
    float humi = dht_sensor.readHumidity();
    // read temperature in Celsius
    float tempC = dht_sensor.readTemperature();
    // read temperature in Fahrenheit
    float tempF = dht_sensor.readTemperature(true);
    snprintf (msg, 75, "Tempp: %.2f'C Humid: %.2f%", tempC, humi);
    Serial.print("Publish message: ");
  }
}

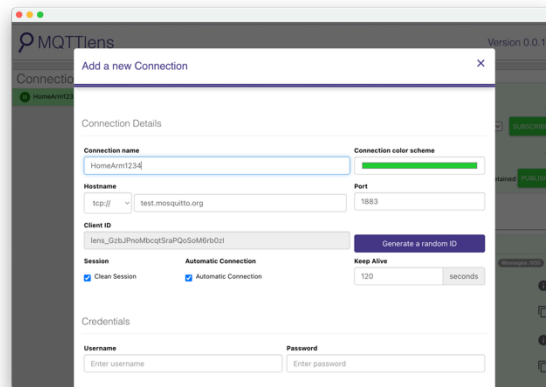
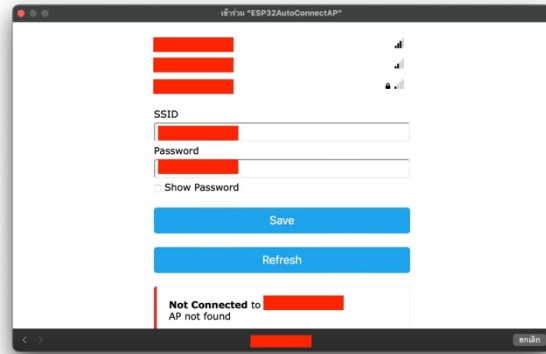
```

```

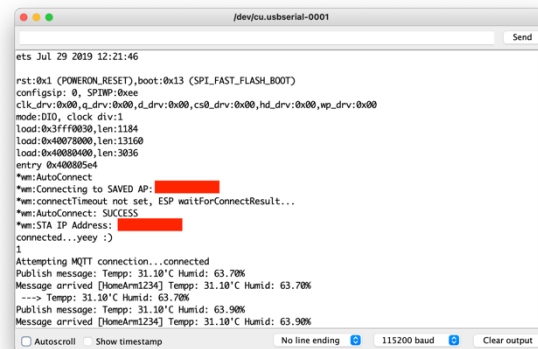
Serial.println(msg);
client.publish(topic1, msg);
}
}

```

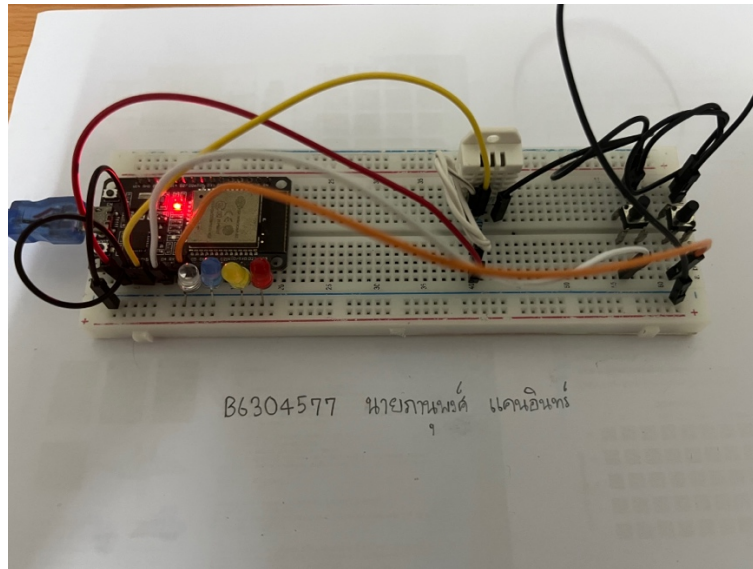
รูปการต่อวงจร – 1



รูปการต่อวงจร – 2



รูปการต่อวงจร - 3



หน้าจอ MQTT Lens

