

## แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

### IoT Approaches to Manufacturing System

ชื่อ-สกุล : นางสาวเกษราภรณ์ เพชรนอก

#### 4/4. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

#### Quiz\_201 – Web Control 2 LED

- อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 2 ดวง
- [https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ\\_gRgDWmREmznuknLbMxV3pOHY4YIPuLEz8-ZzTOX2VhWxcH2QjLGk](https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmznuknLbMxV3pOHY4YIPuLEz8-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 <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "index.h"
#define testLED1 18
#define testLED2 19
//SSID and Password of your WiFi router
const char* ssid = "B6216023";
const char* password = "123456789";
WebServer server(80); //Server on port 80
String ledState1 = "NA";
String ledState2 = "NA";
void handleRoot() {
  String s = MAIN_page; //Read HTML contents
  server.send(200, "text/html", s); //Send web page
}
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(testLED1, HIGH); ledState1 = "ON"; } //Feedback
parameter
if (t_state == "10") { digitalWrite(testLED1, LOW); ledState1 = "OFF"; } //Feedback
parameter
if (t_state == "21") { digitalWrite(testLED2, HIGH); ledState2 = "ON"; } //Feedback
parameter
if (t_state == "20") { digitalWrite(testLED2, LOW); ledState2 = "OFF"; } //Feedback
parameter
server.send(200, "text/plain", ledState1+", "+ledState2); //Send web page
}
void setup(void) {
  Serial.begin(115200);
  pinMode(testLED1, OUTPUT);
  pinMode(testLED2, OUTPUT);
  Serial.print("\n\nConnect to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.print("\nConnected "); Serial.println(ssid);
  Serial.print("IP address: "); Serial.println(WiFi.localIP());
  server.on("/", handleRoot);
  server.on("/setLED", handleLED);
  server.begin();
  Serial.println("HTTP server started");
}
void loop(void) {
  server.handleClient(); //Handle client requests
}

```

```

// index.h
const char MAIN_page[] PROGMEM = R"====(
<!DOCTYPE html>
<html>
<body>
<div id="demo">

```

```

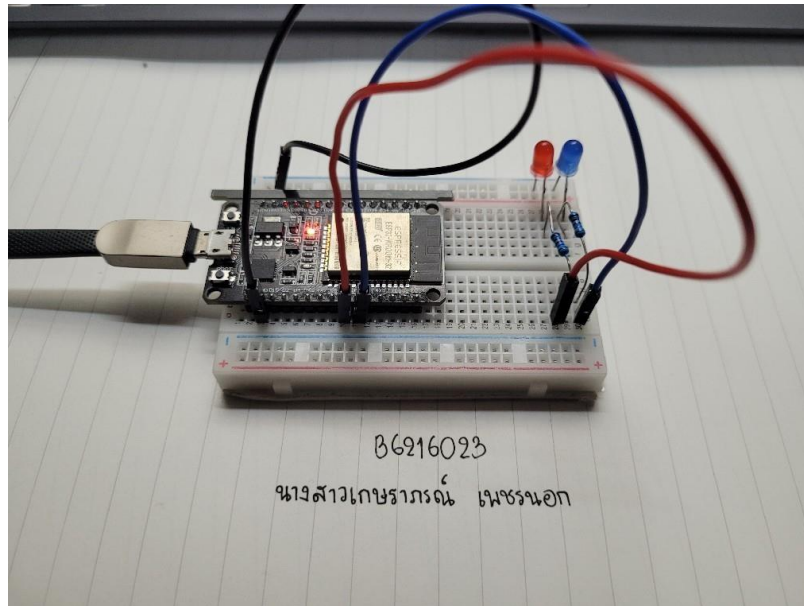
<h1>LED Status</h1>
<button type="button" onclick="sendData(11)" style="background: rgb(202, 60, 60);">LED1 ON</button>
<button type="button" onclick="sendData(10)" style="background: rgb(100,116,255);">LED1 OFF</button><br><br>
<button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60);">LED2 ON</button>
<button type="button" onclick="sendData(20)" style="background: rgb(100,116,255);">LED2 OFF</button><br><br>
State of [LED1, LED2] is >> <span id="LEDState">NA</span><br>
</div>
<div>

</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();
}
setInterval(function() {
// Call a function repetatively with 2 Second interval
getData();
}, 2000); //2000mSeconds update rate
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);

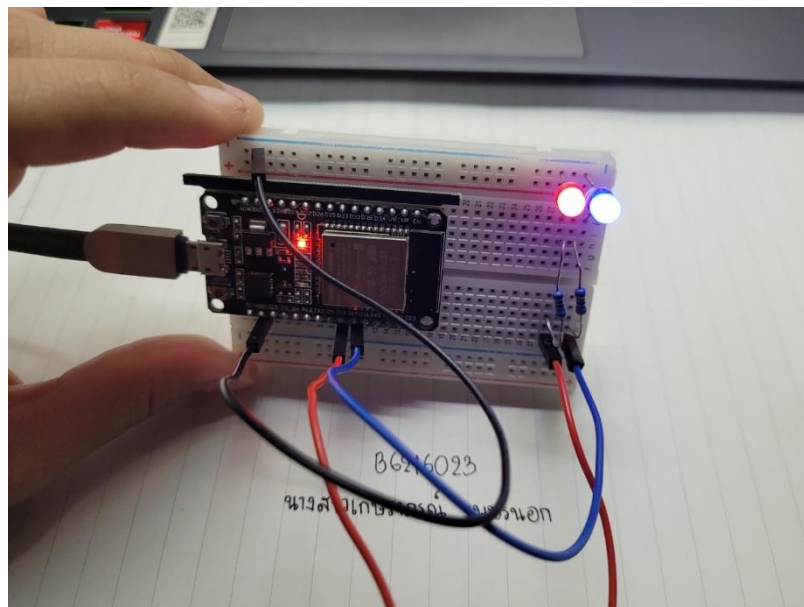
```

```
xhttp.send();
}
</script>
</body>
</html>
)=====";
```

รูปการต่อวงจร - 1



รูปการต่อวงจร - 2



หน้าจอ Web Control

# LED Status

LED1 ON

LED1 OFF

LED2 ON

LED2 OFF

State of [LED1, LED2] is >> ON, ON

## Quiz\_202 – Web Control 4 LED and Monitor Humid/Temperature

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
- อยากมีกด 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 >

```

#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "DHTesp.h"
#include "index.h" //Our HTML webpage contents with javascripts
#define DHT_Pin 4
#define testLED1 18
#define testLED2 19
//SSID and Password of your WiFi router
const char* ssid = "B6216023";
const char* password = "123456789";
WebServer server(80); //Server on port 80
DHTesp dht;
String ledState1 = "NA";
String ledState2 = "NA";
//=====
// This routine is executed when you open its IP in browser
//=====
void handleRoot() {

```

```

String s = MAIN_page; //Read HTML contents
server.send(200, "text/html", s); //Send web page
}

void handleADC() {
  float h = dht.getHumidity();
  float t = dht.getTemperature();
  String tmpValue = "Temp = ";
  tmpValue += String(t) + " C, Humidity = ";
  tmpValue += String(h) + " %";
  server.send(200, "text/plain", tmpValue); //Send value to client ajax request
}

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(testLED1, HIGH); ledState1 = "ON"; } //Feedback
  parameter
  if (t_state == "10") { digitalWrite(testLED1, LOW); ledState1 = "OFF"; } //Feedback
  parameter
  if (t_state == "21") { digitalWrite(testLED2, HIGH); ledState2 = "ON"; } //Feedback
  parameter
  if (t_state == "20") { digitalWrite(testLED2, LOW); ledState2 = "OFF"; } //Feedback
  parameter
  server.send(200, "text/plain", ledState1+", "+ledState2); //Send web page
}

void setup(void) {
  Serial.begin(115200);
  dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22
  pinMode(testLED1, OUTPUT);
  pinMode(testLED2, OUTPUT);
  Serial.print("\n\nConnect to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.print("\nConnected "); Serial.println(ssid);
  Serial.print("IP address: "); Serial.println(WiFi.localIP());
  server.on("/", handleRoot);
}

```

```

server.on("/setLED", handleLED);
server.on("/readADC", handleADC);
server.begin();
Serial.println("HTTP server started");
}
void loop(void) {
  server.handleClient(); //Handle client requests
}

```

```

// 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);">LED1 ON</button>
    <button type="button" onclick="sendData(10)" style="background: rgb(100,116,255);">LED1 OFF</button><br><br>
    <button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60);">LED2 ON</button>
    <button type="button" onclick="sendData(20)" style="background: rgb(100,116,255);">LED2 OFF</button><br><br>
    State of [LED1, LED2] 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>

```



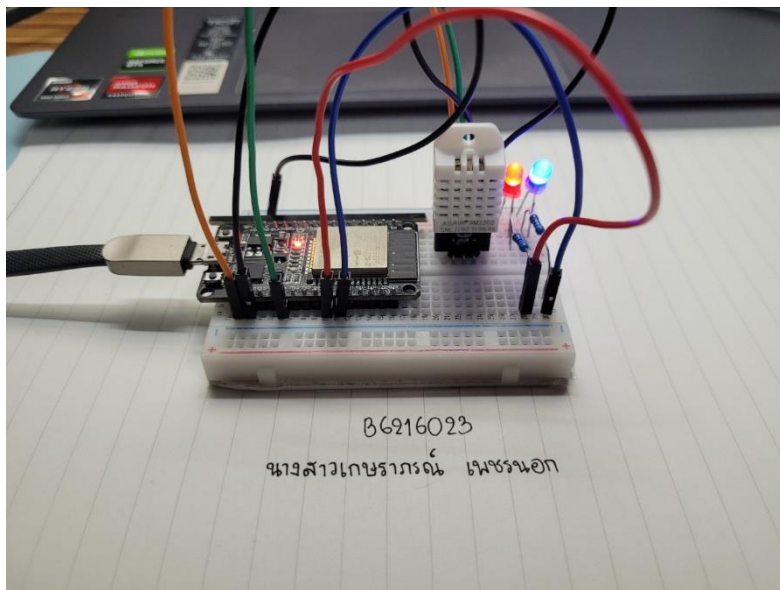
```

}
setInterval(function() {
// Call a function repetatively with 2 Second interval
getData();
}, 2000); //2000mSeconds update rate
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>

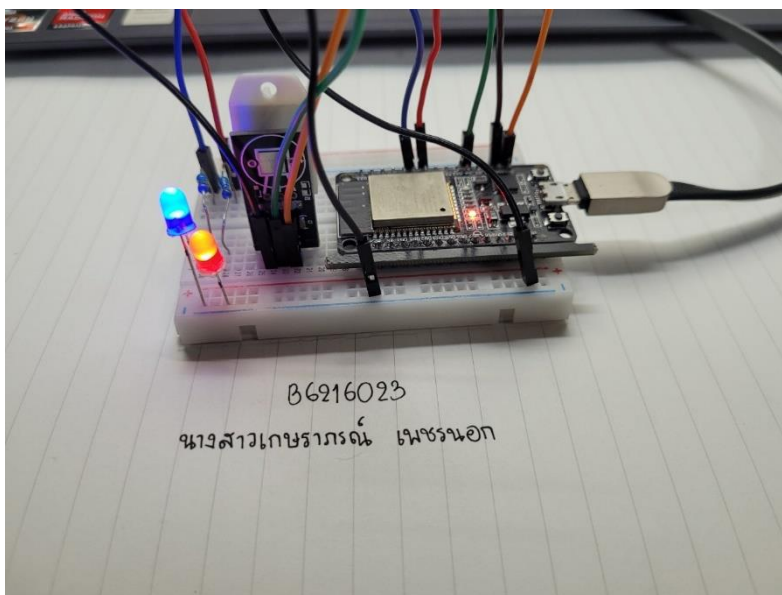
<br><a href="https://www.facebook.com/profile.php?id=100000723050860">By
B6216023 Kassarapron Pechnok</a>
</body>
</html>
)=====";

```

### รูปการต่อวงจร - 1



## รูปการต่อวงจร – 2



## หน้าจอ Web Control

# The ESP-32 Update web page without refresh

LED1 ON LED1 OFF

LED2 ON LED2 OFF

State of [LED1, LED2] is >> ON, ON

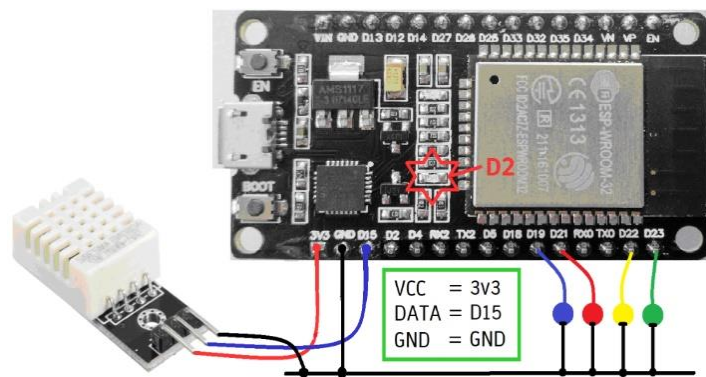
DHT-22 sensor : Temp = 28.50 C, Humidity = 75.10 %

By [B6216023 Kassarapron Pechnok](#)

## Quiz\_203 – Publish

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

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



&lt; Test Code &gt;

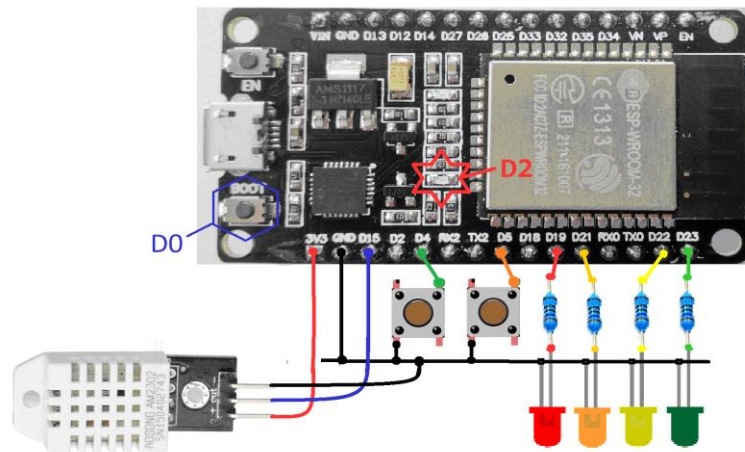
รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

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 >

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

หน้าจอ MQTT Lens