|  |
| --- |
| **แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต**  **IoT Approaches to Manufacturing System** |
| **ขื่อ-สกุล : นายธนพ กาศักดิ์** |

**4/4. คำถามท้ายบทเพื่อทดสอบความเข้าใจ**

**Quiz\_201 – Web Control 2 LED**

* อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 2 ดวง
* <https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzuknLbMxV3pOHy4YIPuLEz8-ZzTOX2VhWxcH2QjLGk>

|  |
| --- |
| **2p** |
| **< Test Code >**  #include <WiFi.h>  const char\* ssid = "iPhoneOhm";  const char\* password = "2444666668888888";  int pinTest = 2;  int pinTest2 = 19;  WiFiServer server(80);  void setup() {  Serial.begin(115200);  pinMode(pinTest, OUTPUT);  pinMode(pinTest2, OUTPUT); // set the LED pin mode  delay(10);  Serial.print("\n\nConnecting 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());  server.begin();  }  int value = 0;  bool LED1\_Status = LOW;  bool LED2\_Status = LOW;  void loop() {  digitalWrite(pinTest, LED1\_Status);  digitalWrite(pinTest2, LED2\_Status);  WiFiClient client = server.available(); // listen for incoming clients  if (client) { // if you get a client,  Serial.println("New Client."); // print a message out the serial port  String currentLine = ""; // make a String to hold incoming data from the client  while (client.connected()) { // loop while the client's connected  if (client.available()) { // if there's bytes to read from the client,  char c = client.read(); // read a byte, then  Serial.write(c); // print it out the serial monitor  if (c == '\n') { // if the byte is a newline character  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("<h1>LED2 Status</h1>");  client.println("<p>");  if (LED1\_Status == HIGH) {  client.println("LED1-On");  } else {  client.println("LED1-Off");  }  if (LED2\_Status == HIGH) {  client.println("LED2-On");  } else {  client.println("LED2-Off");  }  client.println("<p>");  //client.println("<a href=\"/ledon\"><button>LED On</button></a>");  client.println("<a href=\"/LED1-On\"><button style = \"background-color: #f44336;\">LED1On</button></a>");  client.println("<a href=\"/LED2-On\"><button style = \"background-color: #f44336;\">LED2On</button></a>");  client.println("</p>");  //client.println("<a href=\"/ledoff\"><button>LED Off</button></a>");  client.println("<a href=\"/LED1-Off\"><button style = \"background-color: #008CBA;\">LED1Off</button></a>");  client.println("<a href=\"/LED2-Off\"><button style = \"background-color: #008CBA;\">LED2Off</button></a>");  client.println("<body>");  client.println("<html>");  break;  } else {  currentLine = "";  }  } else if (c != '\r') {  currentLine += c;  }  if (currentLine.endsWith("GET /LED1-On")) LED1\_Status = HIGH;  if (currentLine.endsWith("GET /LED2-On")) LED2\_Status = HIGH;  if (currentLine.endsWith("GET /LED1-Off")) LED1\_Status = LOW;  if (currentLine.endsWith("GET /LED2-Off")) LED2\_Status = LOW;  }  }  client.stop(); // close the connection:  Serial.println("Client Disconnected.");  }  } |
| **รูปการต่อวงจร – 1** |
| **รูปการต่อวงจร – 2** |
| **หน้าจอ Web Control** |

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

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

|  |
| --- |
| **XXX** |
| < 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  #define testLED3 22  #define testLED4 23  //SSID and Password of your WiFi router  const char\* ssid = "iPhoneOhm";  const char\* password = "2444666668888888";  WebServer server(80); //Server on port 80  DHTesp dht;  String ledState1 = "OFF";  String ledState2 = "OFF";  String ledState3 = "OFF";  String ledState4 = "OFF";  //===============================================================  // 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/plane", 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); //Feedback parameter  ledState1 = "ON";  }  if (t\_state == "10") {  digitalWrite(testLED1, LOW); //Feedback parameter  ledState1 = "OFF";  }  if (t\_state == "21") {  digitalWrite(testLED2, HIGH); //Feedback parameter  ledState2 = "ON";  }  if (t\_state == "20") {  digitalWrite(testLED2, LOW); //Feedback parameter  ledState2 = "OFF";  }  if (t\_state == "31") {  digitalWrite(testLED3, HIGH); //Feedback parameter  ledState3 = "ON";  }  if (t\_state == "30") {  digitalWrite(testLED3, LOW); //Feedback parameter  ledState3 = "OFF";  }  if (t\_state == "41") {  digitalWrite(testLED4, HIGH); //Feedback parameter  ledState4 = "ON";  }  if (t\_state == "40") {  digitalWrite(testLED4, LOW); //Feedback parameter  ledState4 = "OFF";  }  server.send(200, "text/plane", ledState1 + ", " + ledState2 + ", " + ledState3 + ", " + ledState4); //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);  pinMode(testLED3, OUTPUT);  pinMode(testLED4, 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  } |
| **รูปการต่อวงจร – 1** |
| **รูปการต่อวงจร – 2**  **A picture containing electronics, electrical wiring, cable, electrical supply  Description automatically generated** |
| **หน้าจอ Web Control**  **A screenshot of a computer  Description automatically generated with low confidence** |

**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 <WiFi.h>  #include <Wire.h>  #include <PubSubClient.h>  #include "DHTesp.h"  DHTesp dht;  #define PinLED1 18  #define PinLED2 19  #define PinLED3 22  #define PinLED4 23  #define DHT22\_Pin 4  float h, t;  int blinkStatus = 1;  int LED\_PinArray[] = {PinLED1, PinLED2, PinLED3, PinLED4};  int LED\_StsArray[] = {0, 0, 0, 0};  const char\* ssid = "iPhoneOhm";  const char\* password = "2444666668888888";  const char\* mqtt\_server = "test.mosquitto.org";  const char\* topic1 = "bearish";  String ledState1 = "NA";  WiFiClient espClient;  PubSubClient client(espClient);  long lastMsg = 0;  char msg[50];  int value = 0;  void setup\_wifi() {  delay(10);  Serial.println();  Serial.print("Connecting to ");  Serial.println(ssid);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  delay(500); Serial.print(".");  }  randomSeed(micros());  Serial.println("");  Serial.println("WiFi connected");  Serial.println("IP address: ");  Serial.println(WiFi.localIP());  }  void reconnect()  { while (!client.connected()) // Loop until we're reconnected  { Serial.print("Attempting MQTT connection...");  String clientId = "ESP8266Client-";  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 World Pk007"); // ... and resubscribe  client.subscribe(topic1);  } else  { Serial.print("failed, rc=");  Serial.print(client.state());  Serial.println(" try again in 5 seconds");  delay(5000);  }  }  }  void LEDShowStatus(void) {  if (isnan(t)) {  blinkStatus = 1 - blinkStatus;  LED\_StsArray[0] = 1;  LED\_StsArray[1] = 0;  LED\_StsArray[2] = 0;  LED\_StsArray[3] = 0;  }  if (t < 27) {  blinkStatus = 1;  LED\_StsArray[0] = 1;  LED\_StsArray[1] = 0;  LED\_StsArray[2] = 0;  LED\_StsArray[3] = 0;  }  if (t >= 27) {  blinkStatus = 1 - blinkStatus;  LED\_StsArray[0] = 1;  LED\_StsArray[1] = 1;  LED\_StsArray[2] = 1;  LED\_StsArray[3] = 1;  }  LED\_StsArray[1] = 1;  LED\_StsArray[2] = 1;  LED\_StsArray[3] = 1;  for (int i = 0; i < 4; i++)  digitalWrite(LED\_PinArray[i], LED\_StsArray[i] & blinkStatus);  }  void setup()  { Serial.begin(115200);  setup\_wifi();  //Wire.begin(22, 23);  client.setServer(mqtt\_server, 1883);  dht.setup(DHT22\_Pin, DHTesp::DHT22);  for (int i = 0; i < 4; i++) {  pinMode(LED\_PinArray[i], OUTPUT);  }  }  void loop()  {  if (!client.connected()) reconnect();  client.loop();  long now = millis();  if (now - lastMsg > 5000)  { lastMsg = now;  ++value;  //float t = s.readTempC();  //float h = s.readHumidity();  delay(dht.getMinimumSamplingPeriod());  h = dht.getHumidity();  t = dht.getTemperature();  sprintf (msg, "TempC: %.2f C, Humidity: %.2f %%", t, h);  Serial.print("Publish message: ");  Serial.println(msg);  client.publish(topic1, msg);  }  LEDShowStatus(); delay(250);  LEDShowStatus(); delay(250);  LEDShowStatus(); delay(250);  LEDShowStatus(); delay(250);  LEDShowStatus(); delay(250);  LEDShowStatus(); delay(250);  }  **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);width:100px;height:30px">LED1 ON</button>  <button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60);width:100px;height:30px">LED2 ON</button>  <button type="button" onclick="sendData(31)" style="background: rgb(202, 60, 60);width:100px;height:30px">LED3 ON</button>  <button type="button" onclick="sendData(41)" style="background: rgb(202, 60, 60);width:100px;height:30px">LED4 ON</button><br><br>  <button type="button" onclick="sendData(10)" style="background: rgb(100,116,255);width:100px;height:30px">LED1 OFF</button>  <button type="button" onclick="sendData(20)" style="background: rgb(100,116,255);width:100px;height:30px">LED2 OFF</button>  <button type="button" onclick="sendData(30)" style="background: rgb(100,116,255);width:100px;height:30px">LED3 OFF</button>  <button type="button" onclick="sendData(40)" style="background: rgb(100,116,255);width:100px;height:30px">LED4 OFF</button><br><br>  State of [LED1, LED2, LED3, LED4] is >> <span id="LEDState">/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();  }  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/tanapom.kasak/">By Tanapon kasak B6332235</a>  </body>  </html>  )====="; |
| **รูปการต่อวงจร – 1**  **A picture containing text, electrical supply, cable, electrical wiring  Description automatically generated** |
| **รูปการต่อวงจร – 2** |
| **หน้าจอ Serial Monitor**  **A screenshot of a computer  Description automatically generated with medium confidence**  **หน้าจอ MQTT Lens**  **A screenshot of a computer  Description automatically generated with medium confidence** |

**Quiz\_204 – Publish and Subscribe**

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

|  |
| --- |
|  |
| **< Test Code >**  #include <WiFi.h>  #include <Wire.h>  #include <PubSubClient.h>  #include "DHTesp.h"  DHTesp dht;  #define testLED1 18  #define testLED2 19  #define testLED3 22  #define testLED4 23  #define DHT22\_Pin 15  const char\* ssid = "iPhoneOhm";  const char\* password = "2444666668888888";  const char\* mqtt\_server = "test.mosquitto.org";  const char\* topic1 = "bearish";  String ledState1 = "NA";  int pushButton1 = 4;  int pushButton2 = 5;  WiFiClient espClient;  PubSubClient client(espClient);  long lastMsg = 0;  char msg[50];  int value = 0;  void setup\_wifi() {  delay(10);  Serial.println();  Serial.print("Connecting to ");  Serial.println(ssid);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  delay(500); Serial.print(".");  }  randomSeed(micros());  Serial.println("");  Serial.println("WiFi connected");  Serial.println("IP address: ");  Serial.println(WiFi.localIP());  pinMode(testLED1, OUTPUT);  pinMode(testLED2, OUTPUT);  pinMode(testLED3, OUTPUT);  pinMode(testLED4, OUTPUT);  }  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 lessthan 4 Charector  if ((String)myPayLoad == "ON1") digitalWrite(testLED1, HIGH);  if ((String)myPayLoad == "OFF1") digitalWrite(testLED1, LOW);  if ((String)myPayLoad == "ON2") digitalWrite(testLED2, HIGH);  if ((String)myPayLoad == "OFF2") digitalWrite(testLED2, LOW);  if ((String)myPayLoad == "ON3") digitalWrite(testLED3, HIGH);  if ((String)myPayLoad == "OFF3") digitalWrite(testLED3, LOW);  if ((String)myPayLoad == "ON4") digitalWrite(testLED4, HIGH);  if ((String)myPayLoad == "OFF4") digitalWrite(testLED4, LOW);  }  void reconnect()  { while (!client.connected()) // Loop until we're reconnected  { Serial.print("Attempting MQTT connection...");  String clientId = "ESP8266Client-";  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 World Pk007"); // ... 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()  { Serial.begin(115200);  setup\_wifi();  dht.setup(DHT22\_Pin, DHTesp::DHT22);  pinMode(pushButton1, INPUT\_PULLUP);  pinMode(pushButton2, INPUT\_PULLUP);  client.setServer(mqtt\_server, 1883);  client.setCallback(callback);  pinMode(testLED1, OUTPUT);  pinMode(testLED2, OUTPUT);  pinMode(testLED3, OUTPUT);  pinMode(testLED4, OUTPUT);  }  void loop()  {  if (!client.connected()) reconnect();  client.loop();  long now = millis();  if (now - lastMsg > 5000)  { lastMsg = now;  ++value;  float h = dht.getHumidity();  float t = dht.getTemperature();  sprintf (msg, "TempC: %.2f C, Humidity: %.2f %%", t, h);  Serial.print("Publish message: ");  Serial.println(msg);  client.publish(topic1, msg);  }  if (digitalRead(pushButton1) == 0) {  sprintf (msg, "Overheat Alarm");  Serial.println(msg);  client.publish(topic1, msg);  delay(500);  }  if (digitalRead(pushButton2) == 0) {  sprintf (msg, "Intruders Alarm");  Serial.println(msg);  client.publish(topic1, msg);  delay(500);  }  } |
| **รูปการต่อวงจร – 1** |
| **รูปการต่อวงจร – 2** |
| **หน้าจอ MQTT Lens**  **A screenshot of a computer  Description automatically generated with medium confidence** |