|  |
| --- |
| **การใช้งาน ThingsBoard IoTs Platform เพื่อสร้างและจัดการระบบอัฉริยะ**  **ThingsBoard IoTs Platform for smart system** |
| **ขื่อ-สกุล : วราสิริ ลิ้มประเสริฐ B6214005** |

**6/6 -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ**

**Quiz\_101 – กดติด กดดับ 2 ชุด**

* หากต้องการให้ใช้ 1 สวิตซ์ ควบคุม 1 LED แบบกดติด-กดดับ จำนวน 2 วงจรจะต่อวงจรและเขียนโปรแกรมอย่างไร {SW-D22 -- LED-D19, SW-D23 -- LED-D18}

|  |
| --- |
|  |
| **#define pushButton1 22**  **#define pushButton2 23**  **#define LEDPin1 18**  **#define LEDPin2 19**  **int buttonState1 = 0;**  **int buttonState2 = 0;**  **void setup() {**  **Serial.begin(115200);**  **pinMode(pushButton1, INPUT\_PULLUP);**  **pinMode(pushButton2, INPUT\_PULLUP);**  **pinMode(LEDPin1, OUTPUT);**  **pinMode(LEDPin2, OUTPUT);**  **}**  **void loop() {**  **if (digitalRead(pushButton1) == LOW) {**  **delay(20);**  **buttonState1 = 1 - buttonState1;**  **digitalWrite(LEDPin1, buttonState1);**  **while (digitalRead(pushButton1) == LOW);**  **delay(20);**  **}**  **if (digitalRead(pushButton2) == LOW) {**  **delay(20);**  **buttonState2 = 1 - buttonState2;**  **digitalWrite(LEDPin2, buttonState2);**  **while (digitalRead(pushButton2) == LOW);**  **delay(20);**  **}**  **}** |
|  |
|  |

**Quiz\_102 – Web Control 4 LED and Monitor Humid/Temperature**

* เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
* อยากมีกด Link ไปที่หน้า FB ของตัวเอง
* <https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzuknLbMxV3pOHy4YIPuLEz8-ZzTOX2VhWxcH2QjLGk>

|  |
| --- |
|  |
| **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 = "V2036";**  **const char\* password = "fnafchica";**  **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**  **}** | |
| **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/chi.sweethome.50/">By Varasiri Limprasert B6214005</a>**  **</body>**  **</html>**  **)=====";** | |
|  |
|  |

**Quiz\_103 – Pub/Sub Data from (DHT22 + 4 LED + 2 Switch)**

* อ่านค่า DHT-22 แล้วส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
* กำหนดให้ใช้ mqtt.eclipse.org เป็น Broker
* ควบคุมการปิดเปิด 4 LED
* รับค่าสวิตซ์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm

|  |  |
| --- | --- |
|  | แนะนำวิธีใช้โปรแกรม Eclipse – วิธีดาวน์โหลดโปรแกรมซอฟต์แวร์ |
| **#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 = "V2036";**  **const char\* password = "fnafchica";**  **const char\* mqtt\_server = "test.mosquitto.org";**  **const char\* topic1 = "bearish";**  **String ledState1 = "NA";**  **int pushButton1 = 2;**  **int pushButton2 = 4;**  **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);**  **}**  **}** | |
|  | |
|  | |
|  | |

**Quiz\_104 – Blynk and LINE from (DHT22 + 4 LED + 2 Switch)**

* ควบคุมการปิดเปิด 4 LED
* อ่านค่า DHT-22 แล้วส่งไปยัง Blynk ทุกๆ 5 วินาที
* บันทึกค่าไปยัง Google Sheet
* หากอุณหภูมิเกิน 28’C ให้แจ้งไปยัง LINE
* รับค่าสวิตซ์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm ไปยัง LINE

|  |  |
| --- | --- |
|  | C:\Users\Pk007_20210701NB\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F6B283DD.tmp |
| **#define BLYNK\_PRINT Serial**  **#include <WiFi.h>**  **#include <HTTPClient.h>**  **#include <WiFiClient.h>**  **#include <BlynkSimpleEsp32.h>**  **#include "DHTesp.h"**  **#define DHT22\_Pin 15**  **#define sw1 2**  **#define sw2 4**  **#define WebHooksKey "oXSQX-hS7mc2o1blAA3UlubXBXN2WIrMlIheoCkvYQI"**  **#define WebHooksEventName "Test\_Key"**  **char auth[] = "Y1ccpnuLjmwpjmQ1n\_ZqSVxraOe88oHp";**  **char ssid[] = "V2036";**  **char pass[] = "fnafchica";**  **DHTesp dht;**  **WidgetLED LED1(V2);**  **WidgetLED LED2(V3);**  **BlynkTimer timer;**  **void setup() {**  **Serial.begin(115200);**  **dht.setup(DHT22\_Pin, DHTesp::DHT22); // Connect DHT sensor to GPIO 15**  **pinMode(sw1, INPUT\_PULLDOWN);**  **pinMode(sw2, INPUT\_PULLDOWN);**  **Blynk.begin(auth, ssid, pass);**  **timer.setInterval(1000L, myTimerEvent);**  **}**  **void myTimerEvent() {**  **float humidity = dht.getHumidity();**  **float temperature = dht.getTemperature();**  **Blynk.virtualWrite(V0, temperature);**  **Blynk.virtualWrite(V1, humidity);**  **if (digitalRead(sw1)) LED1.on();**  **else LED1.off();**  **if (digitalRead(sw2)) LED2.on();**  **else LED2.off();**  **Serial.print(" Temp('C) >> "); Serial.print(temperature, 1);**  **Serial.print(", Humidity(%) >> "); Serial.println(humidity, 1);**  **}**  **void loop()**  **{**  **Blynk.run();**  **if (digitalRead(sw1) == LOW) {**  **String serverName = "http://maker.ifttt.com/trigger/" +**  **String(WebHooksEventName) + "/with/key/" + String(WebHooksKey);**  **String httpRequestData = "value1=" + String("Door Open Alarm");**  **Serial.println("Server Name :" + serverName);**  **Serial.println("json httpRequestData :" + httpRequestData);**  **if (WiFi.status() == WL\_CONNECTED) {**  **HTTPClient http;**  **http.begin(serverName);**  **http.addHeader("Content-Type", "application/x-www-form-urlencoded");**  **int httpResponseCode = http.POST(httpRequestData);**  **Serial.print("HTTP Response code: ");**  **Serial.println(httpResponseCode);**  **http.end();**  **if (httpResponseCode == 200)**  **Serial.println("Successfully sent");**  **else**  **Serial.println("Failed!");**  **}**  **else {**  **Serial.println("WiFi Disconnected");**  **}**  **}**  **if (digitalRead(sw2) == LOW) {**  **String serverName = "http://maker.ifttt.com/trigger/" +**  **String(WebHooksEventName) + "/with/key/" + String(WebHooksKey);**  **String httpRequestData = "value1=" + String("Intruders Alarm");**  **Serial.println("Server Name :" + serverName);**  **Serial.println("json httpRequestData :" + httpRequestData);**  **if (WiFi.status() == WL\_CONNECTED) {**  **HTTPClient http;**  **http.begin(serverName);**  **http.addHeader("Content-Type", "application/x-www-form-urlencoded");**  **int httpResponseCode = http.POST(httpRequestData);**  **Serial.print("HTTP Response code: ");**  **Serial.println(httpResponseCode);**  **http.end();**  **if (httpResponseCode == 200)**  **Serial.println("Successfully sent");**  **else**  **Serial.println("Failed!");**  **}**  **else {**  **Serial.println("WiFi Disconnected");**  **}**  **timer.run(); // running timer every 250ms**  **}**  **}** | |
|  | |
|  | |
|  | |