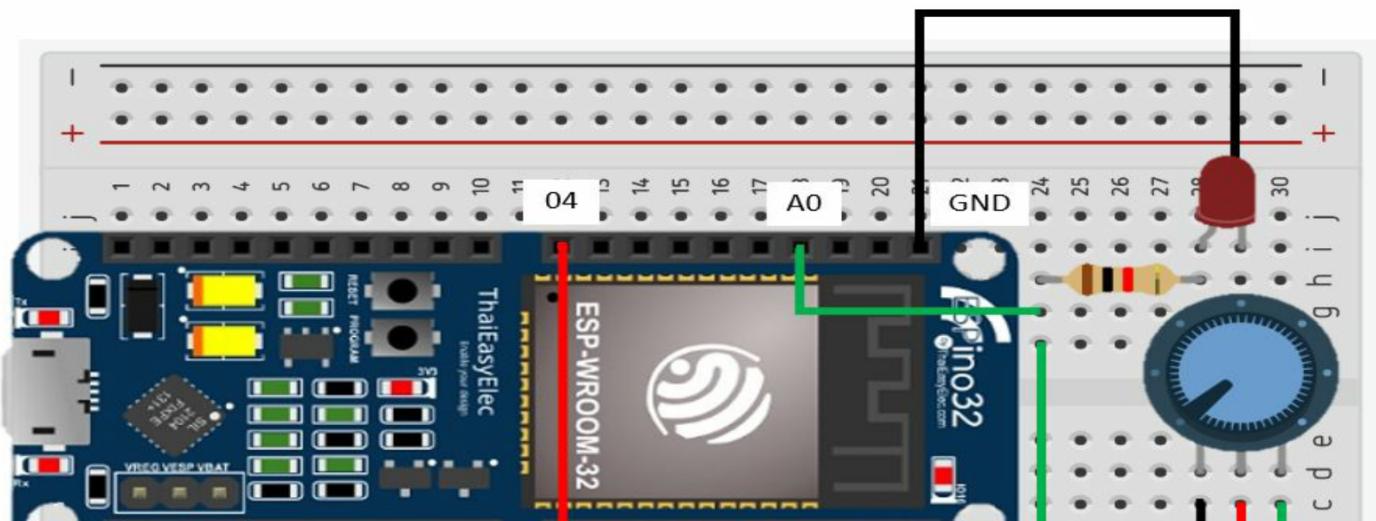


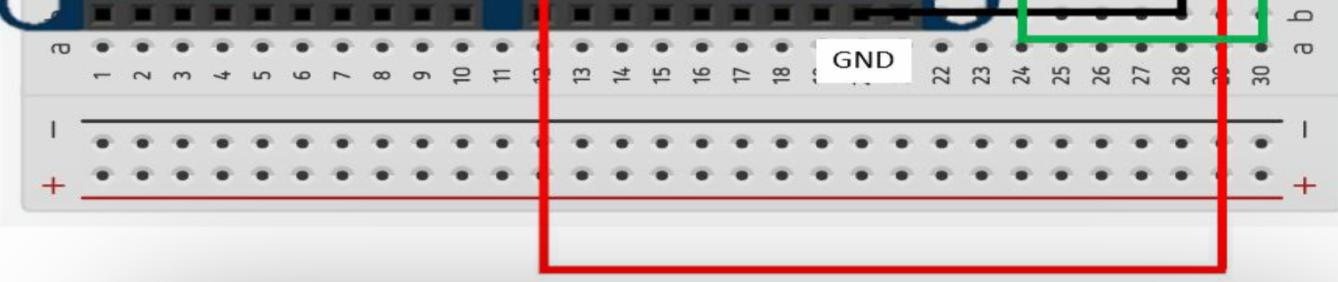


Lap 5.1

1 Lap 5.1 ไฟดาวน์ไลท์ควบคุมผ่านตัวต้านทานปรับค่าได้

```
2 int rotary = 4;           // สร้างตัวแปรชื่อ rotary สำหรับจ่ายไฟไปยัง potentiometer เป็น GPIO 04
3 const byte sig = A0;       // สร้างตัวแปรชื่อ sig สำหรับกำหนดขาที่รับสัญญาณจาก SIG เป็นขา Analog 0
4 int value = 0;             // สร้างตัวแปรชื่อ value เก็บค่าที่ได้จาก sig โดยจะรับมาเป็นตัวเลข
5 float voltageValue = 0;    // สร้างตัวแปรชื่อ voltageValue เก็บค่ากำลังไฟฟ้าที่ได้จาก SIG เป็นตัวเลขแบบจุดทศนิยม มีหน่วยเป็น
6 วัตต์
7 void setup() {
8   Serial.begin(115200);
9   analogReadResolution(10); // กำหนดระดับค่าที่อ่านได้จากขา A0
10  pinMode(rotary, OUTPUT); // กำหนดขา GPIO 04 ให้เป็นขาลักษณะขาออก
11 }
12
13 void loop() {
14   digitalWrite(rotary, HIGH); // เชื่อมค่า 1 ส่งออกจาก GPIO 04
15   value = analogRead(sig); // กำหนดค่าในตัวแปร value เป็นค่าจากการอ่านค่า analog จากขา SIG
16   voltageValue = value * (3.3 / 1024); // กำหนดค่ากำลังไฟฟ้า โดยคำนวณจาก 3.3/1024
17   // โค้ดด้านล่างนี้ใช้ในการแสดงค่าจากเซ็นเซอร์และค่ากำลังไฟที่คำนวณได้
18   Serial.print("sensor value = ");
19   Serial.print(value);
20   Serial.print(" voltage value = ");
21   Serial.println(voltageValue);
22   if (voltageValue > 2.00){ // สร้างเงื่อนไข หากกำลังไฟมากกว่า 2.00 วัตต์
23     Serial.println("LED is on"); // ใช้ทำการแสดงข้อความผ่าน serial monitor ว่า "LED is on"
24   }
25   else{
26     Serial.println("LED is off"); // หาก ไม่ เป็นเท็จ ใช้ทำการแสดงข้อความผ่าน serial monitor ว่า "LED is off"
27   }
28   delay(500); // กำหนดตัวแปรหน่วงเวลาไว้ 0.5 วินาที
29 }
```





Lap 5.2

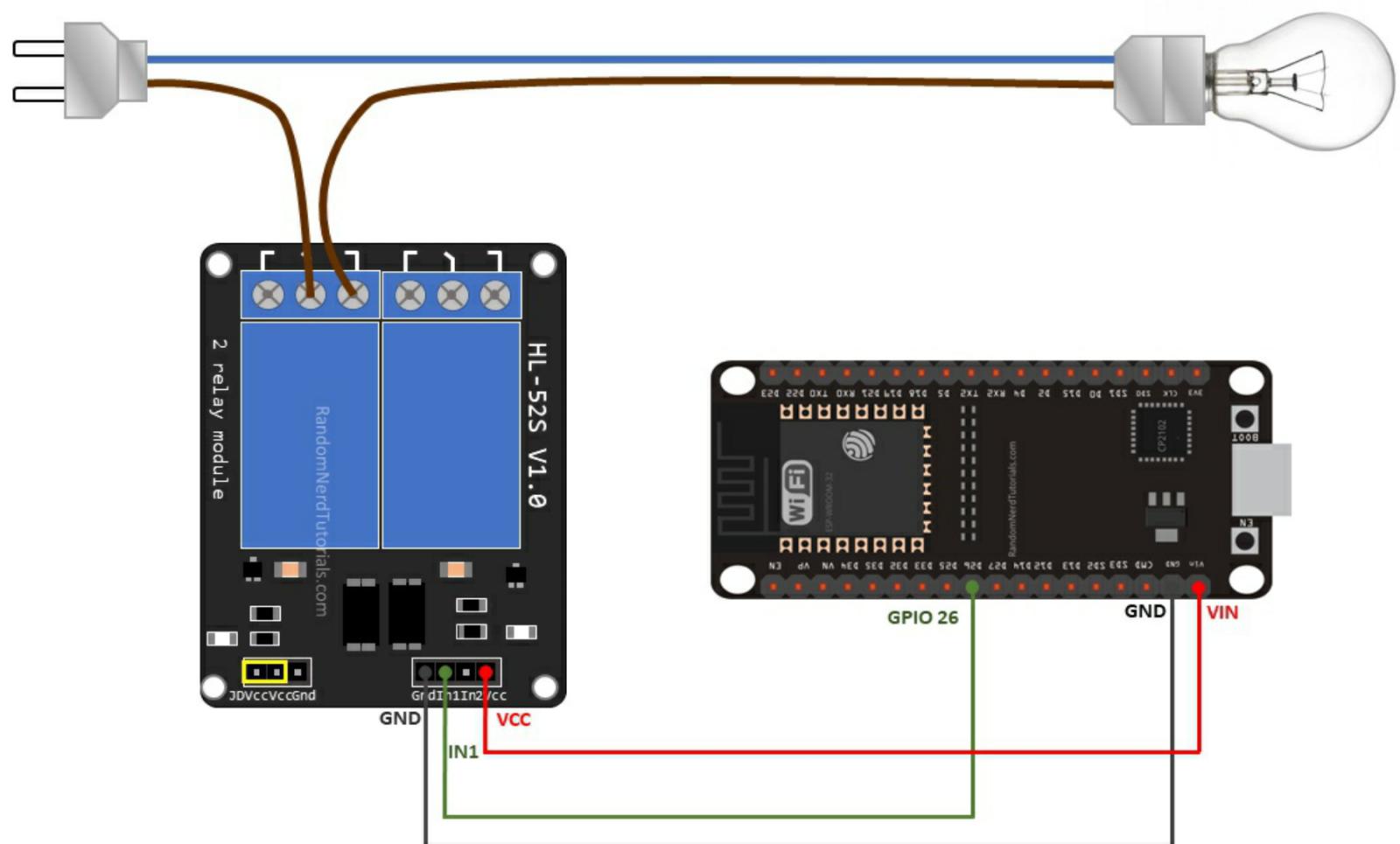
1 Lap 5.2 การใช้งาน Relay ควบคุมเครื่องใช้ไฟฟ้าภายในบ้าน

```
2 const int pingPin = A19;
3 int inPin = A18;
4 int Relay1 = 4;
5
6 void setup() {
7     pinMode(Relay1, OUTPUT);
8     digitalWrite(Relay1, HIGH);
9     Serial.begin(9600);
10 }
11
12 void loop() {
13     long duration, cm;
14     pinMode(pingPin, OUTPUT);
15     digitalWrite(pingPin, LOW);
16     delayMicroseconds(2);
17     digitalWrite(pingPin, HIGH);
18     delayMicroseconds(5);
19     digitalWrite(pingPin, LOW);
20     pinMode(inPin, INPUT);
21     duration = pulseIn(inPin, HIGH);
22     cm = microsecondsToCentimeters(duration);
23     Serial.print(cm);
24     Serial.print("cm");
25     Serial.println();
26     if (cm < 20) {
27         digitalWrite(Relay1, HIGH);
```

```

28 }
29 else {
30   digitalWrite(Relay1, LOW);
31 }
32 delay(500);
33 }
34
35 long microsecondsToCentimeters(long microseconds){
36 // The speed of sound is 340 m/s or 29 microseconds per centimeter.
37 // The ping travels out and back, so to find the distance of the
38 // object we take half of the distance travelled.
39 return microseconds / 29 / 2;
40 }

```





```
1 Lap 7.1 ส่งค่าขึ้น Firebase
2 //ไฟล์นี้ก็ต้องการส่งค่า int และ float ขึ้นไปยัง Firebase
3 #include <Firebase_ESP_Client.h> // import ไลบรารีสำหรับ Firebase
4 // การทำงานของ token และ payload ที่จะเป็นสำหรับ firebase
5 #include "addons	TokenNameHelper.h"
6 #include "addons/RTDBHelper.h"
7 #include "DHT.h"
8
9 #define WiFi_SSID "Firsttt"           // ชื่อ wifi hotspot
10 #define WiFi_PASSWORD "Firsttt28" // รหัส wifi hotspot
11 #define API_KEY "AIzaSyCX58G57yq2i729z8aO9vU0m648e7sYRDM" // API KEY ที่ได้จาก Firebase
12 #define DATABASE_URL "https://mytemp-351c8.firebaseio.southeast1.firebaseio.database.app/"
13 // ระบุลิงค์ Realtime Database
14 #define DHTPIN 33
15 #define DHTTYPE DHT22
16
17 DHT dht(DHTPIN, DHTTYPE);
18 // ตัวแปรที่จะเป็นสำหรับใช้งาน Firebase ในการเชื่อมต่อและการส่งค่า
19 FirebaseData fbd;
20 FirebaseAuth auth;
21 FirebaseConfig config;
22 // ตัวแปรกำหนดค่าสมมติในการส่งเพื่อการทดสอบ
23 unsigned long sendDataPrevMillis = 0;
24 int count = 0;
25 bool signupOK = false;
26
27 void setup() {
28   Serial.begin(115200);
29   Serial.println(F("DHTxx test!"));
30   dht.begin();
31   // การเชื่อมต่อ WiFi
32   WiFi.begin(WiFi_SSID, WiFi_PASSWORD);
33   Serial.print("connecting to ");
34   Serial.print(WiFi_SSID);
35   while (WiFi.status() != WL_CONNECTED){
36     Serial.print(".");
37     delay(1000);
38   }
39   Serial.println();
40   Serial.print("Connected with IP: ");
41   Serial.println(WiFi.localIP());
```

```

42 Serial.println();
43 config.api_key = API_KEY; // นำ API-KEY ไปใส่คำสั่ง config
44 config.database_url = DATABASE_URL; // // นำ URL ของ Realtime database ไปใส่คำสั่ง config
45 // ตรวจสอบการเชื่อมต่อ ทำการเชื่อมต่อโดยใช้ค่าจาก config และ auth
46 if (Firebase.signUp(&config, &auth, "", "")){
47   Serial.println("ok"); // แสดงคำว่า "ok" หากเชื่อมต่อสำเร็จ
48   signupOK = true; // หากเชื่อมต่อสำเร็จ ให้ทำการระบุค่า "จริง" ให้กับตัวแปรนี้
49 }
50 else{
51   Serial.printf("%s\n", config.signer.signupError.message.c_str());
52 }
53 config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h
54 Firebase.begin(&config, &auth);
55 Firebase.reconnectWIFI(true);
56 Serial.println(F("DHTxx test!"));
57 dht.begin();
58 }
59
60 void loop() {
61 delay(2000);
62 // Reading temperature or humidity takes about 250 milliseconds!
63 // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
64 float h = dht.readHumidity();
65 // Read temperature as Celsius (the default)
66 float t = dht.readTemperature();
67 // Read temperature as Fahrenheit (isFahrenheit = true)
68 float f = dht.readTemperature(true);
69 // Check if any reads failed and exit early (to try again).
70 if (isnan(h) || isnan(t) || isnan(f)) {
71   Serial.println(F("Failed to read from DHT sensor!"));
72   return;
73 }
74 // Compute heat index in Fahrenheit (the default)
75 float hif = dht.computeHeatIndex(f, h);
76 // Compute heat index in Celsius (isFahrenheit = false)
77 float hic = dht.computeHeatIndex(t, h, false);
78
79 Serial.print(F("Humidity: "));
80 Serial.print(h);
81 Serial.print(F("% Temperature: "));
82 Serial.print(t);
83 Serial.print(F(" C "));
84 Serial.print(f);
85 Serial.print(F(" F Heat index: "));
86 Serial.print(hic);

```

```

87 Serial.print(F(" C "));
88 Serial.print(hf);
89 Serial.println(F(" F"));
90 // មួយនេះឱ្យ ការចាត់ឲ្យ signupOK ប៊ូនទេ និង Firebase សាមរភ័ត្ន ឱ្យប្រើប្រាស់នៅក្នុងកើតឡើង និងការប្រើប្រាស់នៅក្នុងកើតឡើង
91 if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis ==
92 0)){
93     sendDataPrevMillis = millis();
94     // សំគាល់ int ទូទៅពី count ដោយប្រើប្រាស់ path test/int
95     if (Firebase.RTDB.setInt(&fbdo, "sensor/Fahrenheit", hf)){
96         Serial.println("PASSED"); // ឱ្យប្រើប្រាស់នៅក្នុងកើតឡើងថា "PASSED"
97         Serial.println("PATH: " + fbdo.dataPath()); // ឱ្យប្រើប្រាស់នៅក្នុងកើតឡើងថា path នៃទូទៅ
98     }
99     else {
100        Serial.println("FAILED"); // ឱ្យប្រើប្រាស់នៅក្នុងកើតឡើងថា "FAILED"
101        Serial.println("REASON: " + fbdo.errorReason()); // ឱ្យប្រើប្រាស់នៅក្នុងកើតឡើងថា មិនអាចបញ្ចប់ទូទៅបាន
102    }
103    count++; // ឱ្យលើកឡើងចុះឱ្យចុះថា count មានតុលាការ 1
104    // សំគាល់ float ទូទៅពី count ដោយប្រើប្រាស់ path test/float
105    if (Firebase.RTDB.setFloat(&fbdo, "sensor/Celsius", hio)){
106        Serial.println("PASSED");
107        Serial.println("PATH: " + fbdo.dataPath());
108        Serial.println("TYPE: " + fbdo.dataType());
109    }
110    else {
111        Serial.println("FAILED");
112        Serial.println("REASON: " + fbdo.errorReason());
113    }
114 }
115 }

```

≡ MyTemp ▾

Realtime Database



<https://mytemp-351c8-default-rtbd.firebaseio.com>



<https://mytemp-351c8-default-rtbd.firebaseio.app>

```
sensor
  └── Celsius: -3.94441
      └── Fahrenheit: 24.9
test
  └── float: 58.01
      └── int: 159
```



Lap 7.2

```
1 Lap 7.2 ໃນຕັ້ງເຄື່ອງ
2 #include <WiFi.h>
3 #include <HTTPClient.h>
4
5 const char* ssid = "GoJiraSan";
6 const char* pwd = "Rr0971189116";
7
8 float esp_temp = 25.5;
9 float esp_humid = 78;
10
11 const char* serverName = "https://iotnarathon.000webhostapp.com/data_receive.php";
12 String esp_api_key = "1234567899";
13
14 void setup() {
15     Serial.begin(115200);
16     WiFi.begin(ssid, pwd);
17     Serial.println("ກຳລັງເຂື້ອມຕ່ວ");
18     while(WiFi.status() != WL_CONNECTED) {
19         delay(500);
20         Serial.print(".");
21     }
22     Serial.println("");
23     Serial.print("ເຂື້ອມຕ່ວ Wi-Fi ແລ້ວ IP Address ຕີ້ວ: ");
24     Serial.println(WiFi.localIP());
25 }
26
27 void loop() {
```

```

28 HttpClient http;
29 http.begin(serverName);
30 http.addHeader("Content-Type", "application/x-www-form-urlencoded");
31 // https://iotghia.000webhostapp.com/data_receive.php?
32     api_key_value=1234567899&temp=25.5&humid=70
33 String httpRequestData = "esp_api_key=" + esp_api_key + "&esp_temp=" + esp_temp + "&esp_humid="
34     + esp_humid;
35 int httpRequestCode = http.POST(httpRequestData);
36 Serial.print(httpRequestData);
37 http.end();
38 delay(5000);
39 }

```



Lap 7.2 000host : data_receive.php

```

1 Lap 7.2 000host : data_receive.php
2 <?php
3     include 'con_db.php'
4 ?>
5 <?php
6 $api_key_value = "1234567899";
7 $api_key = $temp = $humid = "";
8 if($_SERVER["REQUEST_METHOD"] == "POST"){
9     $api_key = esp_input($_POST["esp_api_key"]);
10    if($api_key == $api_key_value){
11        $temp = esp_input($_POST["esp_temp"]);
12        $humid = esp_input($_POST["esp_humid"]);
13        $sql = "INSERT INTO data (temp, humid) VALUES ('" . $temp . "' , '" . $humid . "')";
14        if($conn->query($sql) == TRUE){
15            echo "data successfully inserted!";
16        }
17    } else{
18        echo "data not inserted: " . "<br>" . $conn->error;
19    }
20    $conn->close();

```

```

21     }
22 else{
23     echo "wrong API-KEY";
24 }
25 }
26 else{
27     echo "no data come via POST";
28 }
29
30 function esp_input($data){
31     $data = trim($data);
32     $data = stripslashes($data);
33     $data = htmlspecialchars($data);
34     return $data;
35 }
36 ?>

```



Lap 7.2 000host : con_db.php

```

1 Lap 7.2 000host : con_db.php
2 <?php
3 /* นำข้อมูลมาใส่โดยหาได้จาก Database Manager ใน 000webhost*/
4 $servername = "localhost";           // คือหาได้จาก DB Host
5 $username = "id20422652_admin";    // หาได้จาก DB User
6 $password = "Giter_172544";        // รหัสผ่านตอนล็อค Database
7 $dbname = "id20422652_iotwithterlap7"; // หาได้จาก DB Name
8 // ส่วนนี้จะเป็นการเชื่อมต่อฐานข้อมูล
9 $conn = new mysqli($servername, $username, $password, $dbname); // ตัวแปรเก็บค่าคำลั่งในการเชื่อมต่อ
   // ฐานข้อมูล
10 // สร้างเงื่อนไขเพื่อตรวจสอบการเชื่อมต่อ โดยหากเชื่อมต่อสำเร็จ จะขึ้นข้อความว่า "connected"
11 if ($conn->connect_error){
12     die("cannot connect to database" . $conn->connect_error);
13 }
14 else{
15     echo "connected!";

```

```
16 }
17 ?>
```

The screenshot shows the phpMyAdmin interface connected to a MySQL database named 'id20422652_iotwithterap7'. The 'data' table is selected, displaying 24 rows of data. The columns are 'data_id', 'temp', 'humid', and 'time_stamp'. The data shows temperature (temp) and humidity (humid) values ranging from 25.5 to 79, and time stamps from March 9, 2023, to March 16, 2023.

data_id	temp	humid	time_stamp
1	25.5	70	2023-03-09 07:51:33.529831
2	25.5	70	2023-03-09 09:17:36.137761
3	25.5	79	2023-03-16 08:44:22.557773
4	25.5	79	2023-03-16 08:44:32.189456
5	25.5	79	2023-03-16 08:44:40.550320
6	25.5	79	2023-03-16 08:44:48.989684
7	25.5	79	2023-03-16 08:44:57.151550
8	25.5	79	2023-03-16 08:45:05.791499
9	25.5	79	2023-03-16 08:45:14.597212
10	25.5	79	2023-03-16 08:45:22.770375
11	25.5	79	2023-03-16 08:46:42.236826

Lap 8 : RTOS

```
1 Lap 8 : RTOS (Real-time Operating System)
2 #define ledPIN1 4
3 #define ledPIN2 13
4 #define ledPIN3 2
5 #define ledPIN4 12
6
7 void setup() {
8     /*สร้าง task ด้วยคำสั่ง xTaskCreate() มี parameter ดังนี้
9     xTaskCreate( pvTaskCode, pcName, usStackDepth, pvParameters, uxPriority, pxCreatedTask)*/
10    xTaskCreate(&first_task, "first_task", 1024, NULL, 5, NULL); // สร้าง task ที่ 1
11    xTaskCreate(&second_task, "second_task", 1024, NULL, 5, NULL); // สร้าง task ที่ 2
12    xTaskCreate(&third_task, "third_task", 1024, NULL, 5, NULL); // สร้าง task ที่ 1
13    xTaskCreate(&fourth_task, "fourth_task", 1024, NULL, 5, NULL); // สร้าง task ที่ 2
14 }
15 // พิมพ์ชื่อ loop ไม่ได้ใช้งาน เนื่องจากพิมพ์ชื่อแต่ละงานจะทำงานแบบวนลูปอยู่แล้ว
16 void loop() {
17 }
```

```
17 }
18 //สร้างฟังก์ชันการทำงานสำหรับ task ที่ 1
19 void first_task(void *pvParameter){      // ไม่ส่งค่า parameter ได้ๆเข้าฟังก์ชัน
20   pinMode(ledPIN1,OUTPUT);
21   while(true) {
22     digitalWrite(ledPIN1, HIGH);
23     vTaskDelay(100 / portTICK_RATE_MS); // คำสั่ง vTaskDelay สำหรับกำหนดเวลาให้กับการทำงานแบบ
24     digitalWrite(ledPIN1, LOW);
25     vTaskDelay(100 / portTICK_RATE_MS);
26   }
27 }
28 //สร้างฟังก์ชันการทำงานสำหรับ task ที่ 2
29 void second_task(void *pvParameter){
30   pinMode(ledPIN2,OUTPUT);
31   while(true) {
32     digitalWrite(ledPIN2, HIGH);
33     vTaskDelay(500 / portTICK_RATE_MS);
34     digitalWrite(ledPIN2, LOW);
35     vTaskDelay(500 / portTICK_RATE_MS);
36   }
37 }
38 //สร้างฟังก์ชันการทำงานสำหรับ task ที่ 3
39 void third_task(void *pvParameter){
40   pinMode(ledPIN3,OUTPUT);
41   while(true) {
42     digitalWrite(ledPIN3, HIGH);
43     vTaskDelay(500 / portTICK_RATE_MS);
44     digitalWrite(ledPIN3, LOW);
45     vTaskDelay(500 / portTICK_RATE_MS);
46   }
47 }
48 //สร้างฟังก์ชันการทำงานสำหรับ task ที่ 4
49 void fourth_task(void *pvParameter){
50   pinMode(ledPIN4,OUTPUT);
51   while(true) {
52     digitalWrite(ledPIN4, HIGH);
53     vTaskDelay(500 / portTICK_RATE_MS);
54     digitalWrite(ledPIN4, LOW);
55     vTaskDelay(500 / portTICK_RATE_MS);
56   }
57 }
```



Lap 9 : RFID

```
1 Lap 9 : RFID
2 //Libraries
3 #include <SPI.h>
4 #include <MFRC522.h>
5
6 #define SS_PIN 21
7 #define RST_PIN 0
8 #define LED_PIN 15
9 //Variables
10 byte uidPICC[4] = {0, 0, 0, 0};
11 MFRC522::MIFARE_Key key;
12 MFRC522 rfid = MFRC522(SS_PIN, RST_PIN);
13
14 void setup() {
15     //เริ่มต้นการทำงานของ Serial
16     Serial.begin(115200);
17     Serial.println(F("เริ่มการทำงาน"));
18     //เริ่มต้นการทำงานของ RFID และการส่งข้อมูลผ่าน Serial
19     SPI.begin();
20     rfid.PCD_Init();
21     // แสดงเวอร์ชันของ firmware
22     Serial.print(F("Reader :"));
23     rfid.PCD_DumpVersionToSerial();
```

```
24 }
25
26 void loop() {
27     readRFID();
28 }
29
30 void readRFID(void ) { // พึงก์ชั้นสำหรับการอ่าน tag
31     /// อ่าน (Read) RFID card
32     for (byte i = 0; i < 6; i++) {
33         key.keyByte[i] = 0xFF;
34     }
35     // เชื่อมต่อหากพบ tag
36     if ( ! rfid.PICC_IsNewCardPresent())
37         return;
38     // เชื่อมต่อหากตรวจสอบ tag id
39     if ( ! rfid.PICC_ReadCardSerial())
40         return;
41     // บันทึกค่า tag id
42     for (byte i = 0; i < 4; i++) {
43         nuidPICC[i] = rfid.uid.uidByte[i];
44     }
45     // แสดง tag id ผ่าน serial monitor
46     Serial.print(F("หมายเลข RFID แบบฐานสิบ: "));
47     printDec(rfid.uid.uidByte, rfid.uid.size);
48     Serial.println();
49     // หยุดการทำงาน
50     rfid.PICC_HaltA();
51     // หยุดการเข้ารหัส
52     rfid.PCD_StopCrypto1();
53 }
```

```
55 //พังก์ชัน ในการนำค่า ID ที่เป็นเลขฐาน 10 ส่งผ่านช่องทาง
56 void printDec(byte *buffer, byte bufferSize) {
57     int IP_blue[4] = {227, 106, 238, 43};
58     pinMode(LED_PIN,OUTPUT);
59     for (byte i = 0; i < bufferSize; i++) {
60         Serial.print(buffer[i] < 0x10 ? " 0" : " ");
61         Serial.print(buffer[i], DEC);
62     }
63     for (int x = 0; x < 4; x++){
64         if (buffer[x] != IP_blue[x]){
65             digitalWrite(LED_PIN, HIGH);
66             Serial.print("ไฟติด");
67         }
68     else{
69         digitalWrite(LED_PIN, LOW);
70         Serial.print("ไฟดับ");
71     }
72 }
73 }
```



```
2 #define BLYNK_PRINT Serial
3 #include <ESP8266WiFi.h>
4 #include <BlynkSimpleEsp8266.h>
5 #include "DHT.h"
6 #define BLYNK_TEMPLATE_ID "TMPL6x605X7Jd"
7 #define BLYNK_TEMPLATE_NAME "Narathon"
8 #define BLYNK_AUTH_TOKEN "18m3UdIT4PGMnP5-_sc-pfBSsuJXn9PE"
9 DHT dht;
10 // You should get Auth Token in the Blynk App.
11 // Go to the Project Settings (nut icon).
12 char auth[] = "18m3UdIT4PGMnP5-_sc-pfBSsuJXn9PE";
13 // Your WiFi credentials.
14 // Set password to "" for open networks.
15 char ssid[] = "Firsttt";
16 char pass[] = "Firsttt28";
17
18 void setup(){
19 // Debug console
20 Serial.begin(9600);
21 Blynk.begin(auth, ssid, pass);
22 // You can also specify server:
23 //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
24 //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
25 Serial.println();
26 Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");
27 dht.setup(2); // data pin 2
28 }
29
30 void loop(){
31 delay(dht.getMinimumSamplingPeriod());
32 float humidity = dht.getHumidity(); // ดึงค่าความชื้น
33 float temperature = dht.getTemperature(); // ดึงค่าอุณหภูมิ
```

```

33     float temperature = dht.getTemperature(); // ດົງຄາວຸນໜຸ່ມ
34     Serial.print(dht.getStatusString());
35     Serial.print("\t");
36     Serial.print(humidity, 1);
37     Serial.print("\t\t");
38     Serial.print(temperature, 1);
39     Serial.print("\t\t");
40     Serial.println(dht.toFahrenheit(temperature), 1);
41     Blynk.run();
42     delay(100);
43     Blynk.virtualWrite(V5, temperature);
44     Blynk.virtualWrite(V6, humidity);
45 }

```

The screenshot shows the Blynk app interface. On the left, there's a sidebar with various icons: a green circle with a white 'B', a magnifying glass, a building, a grid, a house, a mail icon, a megaphone, a gear, and a person. The main area has a header 'My organization - 4887NA'. Below it, there's a search bar and a list showing '1 Device' named 'Narathon'. The main content area is a detailed view of the device 'Narathon'. It shows the device icon, name 'Narathon', status 'Offline', and organization 'My organization - 4887NA'. There are tabs for 'Dashboard', 'Timeline', 'Device Info', 'Metadata', and 'Actions Log', with 'Dashboard' being the active tab. Below the tabs is a time selector with options: 'Latest' (highlighted in green), 'Last Hour', '6 Hours', '1 Day', '1 Week', '1 Month' (with a red notification badge), '3 Months' (with a red notification badge), and 'Custom'. At the bottom, there are two circular gauge charts: one for 'temperature' (value 0) and one for 'humidity' (value 0). The bottom right corner of the screen shows the text 'Region: sgp1 Privacy Policy'.

