

### วัตถุประสงค์

- เพื่ออธิบายการทำงานของ ESP32 ในการวัดฝุ่นและความชื้น ซึ่งจะแสดงผลทาง website ใช้งานบนอุปกรณ์ laptop, pc, smartphone และอุปกรณ์อื่นๆที่สามารถเชื่อมต่อ wifi ได้ โดยเชื่อมต่อที่อยู่ของ website ในการใช้งานและแจ้งเตือนออนไลน์

### เทคโนโลยีที่ใช้

**Arduino IDE** ซอฟต์แวร์ที่ใช้ในการพัฒนางานสำหรับบอร์ด Arduino นั่นคือโปรแกรมที่เรียกว่า Arduino IDE ในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของ ระบบการพัฒนาหรือตัวช่วยต่างๆที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้ เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่างๆเร็วมากขึ้น

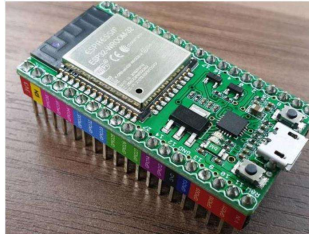


**C Programming Language** คือ ภาษาคอมพิวเตอร์ใช้สำหรับพัฒนาโปรแกรมทั่วไป ถูกพัฒนาครั้งแรกเพื่อใช้เป็นภาษาสำหรับพัฒนาระบบปฏิบัติการยูนิกซ์ ( Unix Operating System) แทนภาษา assembly ซึ่งเป็นภาษาระดับต่ำที่สามารถกระทำในระบบฮาร์ดแวร์ได้ด้วยความเร็ว

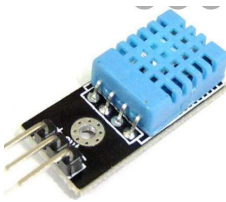


### อุปกรณ์ที่ใช้

- espressif esp32-wroom-32 NODE32 LITE



- DHT11 or DHT22 Temperature and Humidity Sensor

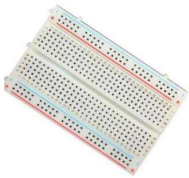


use DHT 11

-4.7k Ohm Resistor



-Breadboard

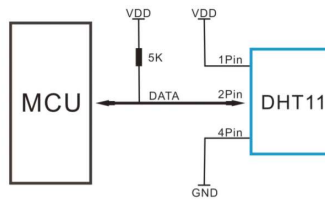


-Jumper wires

### Data of DHT11 Temperature and Humidity Sensor

Sensor นี้การรวมระบบทำได้ง่ายและรวดเร็ว มีขนาดเล็ก ใช้พลังงานต่ำและสูงมากถึง 20 ของ  
ตัวพลังงาน มี 4-pin

Parameters	Conditions	Minimum	Typical	Maximum
<b>Humidity</b>				
Resolution		1%RH	1%RH	1%RH
Repeatability			8 Bit	
Accuracy	25 °C		± 1%RH	
	0-50 °C		± 4%RH	
Interchangeability	Fully Interchangeable			
Measurement Range	0 °C	30%RH		90%RH
	25 °C	20%RH		90%RH
	50 °C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25 °C, 1m/s Air	6 S	10 S	15 S
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
<b>Temperature</b>				
Resolution		1 °C	1 °C	1 °C
Repeatability		8 Bit	8 Bit	8 Bit
Accuracy		± 1 °C	± 1 °C	± 2 °C
Measurement Range		0 °C		50 °C
Response Time (Seconds)	1/e(63%)	6 S		30 S



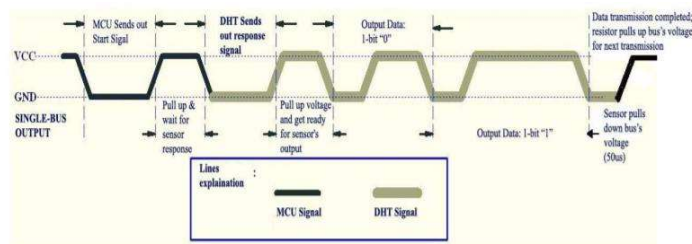
แหล่งจ่ายไฟของ DHT11 คือ 3-5.5V DC ค่าสั่งให้ sensor ภายใน 1 s เพื่อผ่านสถานะไม่เสถียร สามารถเพิ่มตัวเก็บประจุค่า 100nF ระหว่าง VDD และ GND สำหรับการกรองพลังงาน ในรูปแบบข้อมูลบัสเดียวใช้สำหรับการสื่อสารและการซิงโครไนซ์ระหว่าง MCU และsensor DHT11 กระบวนการสื่อสารหนึ่งกระบวนการใช้เวลาประมาณ 4ms การส่งข้อมูลที่สมบูรณ์คือ 40 บิตและ Sensor ส่งข้อมูลบิตที่สูงขึ้นก่อนรูปแบบข้อมูล

ข้อมูลที่ส่ง 40 บิตคือ

ข้อมูล RH รวม 8 บิต + ข้อมูล RH ทศนิยม 8 บิต + ข้อมูล T รวม 8 บิต + ทศนิยม 8 บิต Tข้อมูล + ผลรวมเช็ค 8 บิต

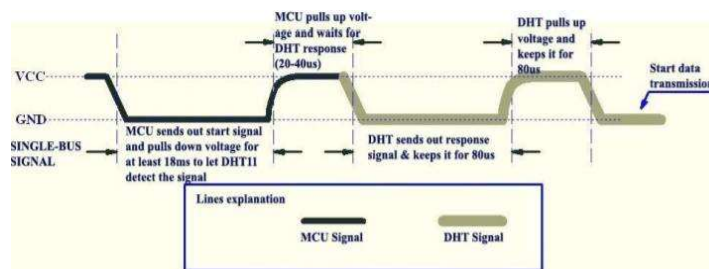
### กระบวนการสื่อสารโดยรวม DHT11 Temperature and Humidity Sensor

เมื่อ MCU ส่งสัญญาณเริ่มต้น DHT11 จะเปลี่ยนจากโหมดสิ้นเปลืองพลังงานต่ำเป็นโหมดวิ่ง รอให้ MCU เสร็จสิ้นสัญญาณเริ่มต้น เมื่อเสร็จแล้ว DHT11 จะส่งสัญญาณตอบสนองของข้อมูล 40 บิตที่มีข้อมูล ความชื้นสัมพัทธ์และอุณหภูมิถึง ผู้ใช้สามารถเลือกที่จะรวบรวมข้อมูลบางส่วนโดยไม่มีสัญญาณเริ่มต้นจาก MCU, DHT11 จะไม่ให้สัญญาณตอบสนองต่อ MCU เมื่อรวบรวมข้อมูลแล้ว DHT11 จะเปลี่ยนเป็นโหมดกินไฟต่ำกว่าจะได้รับสัญญาณเริ่มต้นจาก MCU อีกครั้ง



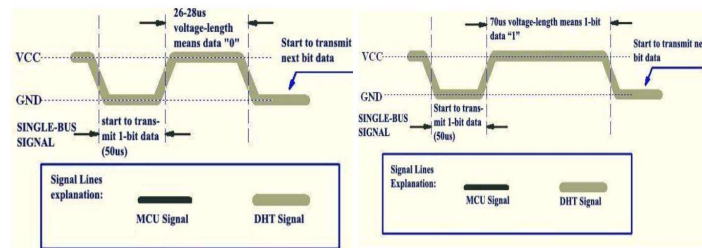
### MCU ส่งสัญญาณเริ่มต้นไปยัง DHT 11

ข้อมูลสถานะว่างบัสเดี่ยวจะอยู่ที่ระดับไฟฟ้าแรงสูง เมื่อถึงการสื่อสารระหว่าง MCU และ DHT11 โปรแกรมของ MCU จะตั้งค่าระดับแรงดัน Data Single-bus จากสูงไปต่ำและกระบวนการนี้ต้องใช้เวลอย่างน้อย 18 ms เพื่อให้แน่ใจว่า DHT11 ตรวจพบสัญญาณของ MCU จากนั้น MCU จะดึงแรงดันไฟฟ้าขึ้นมาและรอ 20-40us สำหรับการตอบสนองของ DHT 11



### การตอบสนอง DHT 11 ต่อ MCU

เมื่อ DHT ตรวจพบสัญญาณเริ่มต้น มันจะส่งสัญญาณตอบสนองระดับแรงดันต่ำ ซึ่งกินเวลา 80us จากนั้นโปรแกรมของ DHT จะตั้งค่า Data Single-bus voltage level จากต่ำไปสูง และเก็บไว้ 80us สำหรับการเตรียม DHT11 สำหรับการส่งข้อมูลเมื่อ DATA Single-Bus อยู่ที่ระดับแรงดันไฟฟ้าต่ำ แสดงว่า DHT11 กำลังส่งการตอบสนองสัญญาณ. เมื่อ DHT11 ส่งสัญญาณตอบสนอง มันจะดึงแรงดันไฟฟ้าขึ้นมาและเก็บไว้ที่ 80us และเตรียมรับส่งข้อมูลเมื่อ DHT11 กำลังส่งข้อมูลไปยัง MCU ข้อมูลทุกบิตจะเริ่มด้วยระดับแรงดันไฟฟ้าต่ำ 50us และความยาวของสัญญาณไฟฟ้าแรงสูงต่อไปนี้กำหนดว่าบิตข้อมูลเป็น "0" หรือ "1"





DATA "0"

DATA "1"

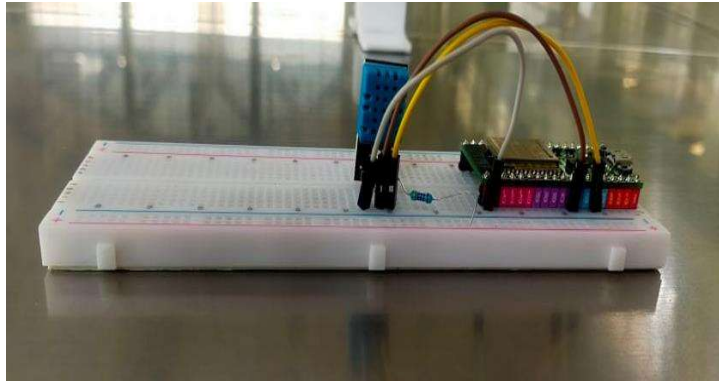
หากสัญญาณตอบสนองจาก DHT อยู่ที่ระดับแรงดันสูงเสมอ แสดงว่า DHT ไม่ตอบสนองอย่างถูกต้องและเมื่อตรวจสอบการเชื่อมต่อ เมื่อส่งข้อมูลบิตสุดท้าย DHT11 ดึงระดับแรงดันไฟฟ้าลงและเก็บไว้ที่ 50us จากนั้นแรงดันบัสเดี่ยวจะเป็นตัวกำหนดขึ้นเพื่อตั้งค่ากลับเป็นสถานะว่าง

### DHT11 vs DHT22 Temperature and Humidity Sensor

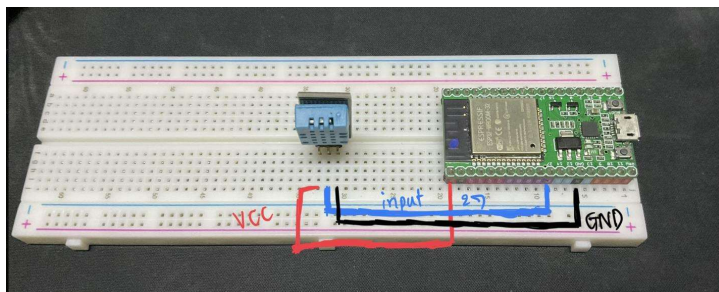
	
<b>DHT11</b>	<b>DHT22</b>
0 - 50°C / ± 2°C	Temperature Range -40 - 125 °C / ± 0.5 °C
20 - 80% / ± 5%	Humidity Range 0 - 100 % / ± 2-5%
1Hz (one reading every second)	Sampling Rate 0.5 Hz (one reading every two seconds)
15.5mm x 12mm x 5.5mm	Body Size 15.1mm x 25mm x 7.7mm
3 - 5V	Operating Voltage 3 - 5V
2.5mA	Max Current During Measuring 2.5mA

### อธิบายการต่อวงจร Hardware wiring

ต่อวงจรจริง



circuit diagram



## อธิบายการทำงาน

การทำงานในส่วนโปรแกรมใน arduino

```

int setsec = 30;
int count = 0;
#include <AntoIO.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#define LINE_TOKEN "EX2DQ60RUCB3qMzRYVELHC6FuZqVp7YeicVbVav5b2b"

const char *ssid = "Edok5555";
const char *pass = "0910395891";
const char *user = "nattalesrunner";
const char *token = "IUFaCcA9WUMJHir2q51CSELotKUgLnKdmIZ126e";
const char *thing = "myserver";

/* create AntoIO instance */
AntoIO anto(user, token, thing);

#include "DHT.h"

#define DHTPIN 27

#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)
DHT dht(DHTPIN, DHTTYPE);

```

เริ่มแรกจะกำหนด setsec เวลาในการส่งข้อความทางไลน์ count คือจะนับไปเรื่อยๆในการ check ตรวจสอบค่า จะ import library AntoIO เป็น library ที่ใช้เชื่อมต่อเว็บ , library WiFi ,DHT คือ ตัว module ในการวัดอุณหภูมิและความชื้น จะกำหนด ค่า Line token กำหนด ssid และ password ของ wifi มีชื่อ user และ thing ของเว็บ AntoIO กำหนดใช้ pin ที่ 27 ในการใช้งาน กำหนด DHT11 ใน การใช้งาน

```

void setup() {

  Serial.begin(9600);
  Serial.print("\nTrying to connect ");
  Serial.print(ssid);
  Serial.print("\nTrying to connect ");
  Serial.print(ssid);
  Serial.println("...");

  while (!anto.wifi.begin(ssid, pass));
  Serial.println("\nConnected, trying to connect to broker...");

  while (!anto.mqtt.connect(user, token, true));
  Serial.println("\nConnected");
  LINE_Notify("ฉฉฉฉฉ");
  delay(1000);
  dht.begin();

}

```

ตรวจสอบการเชื่อมต่อ wifi ของ wifi ที่สร้างขึ้นมา



```

void loop() {
  anto.mqtt.loop();
  // Wait a few seconds between measurements.
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%   Temperature: "));
  Serial.print(t);
  Serial.print(F("°C   "));
  Serial.print(f);

  Serial.print(F("°F   Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C   "));
  Serial.print(hif);
  Serial.println(F("°F"));
  anto.mqtt.pub("temp", t);
  anto.mqtt.pub("humid", h);
  count += 1;
  Serial.println("                                count = " + String(count));
  delay(1000);

  if ( count == setsec )
  {
    LINE_Notify("อุณหภูมิ:" + String(t, 1) + " องศา  ");
    Serial.println("    Temp Line !!!");
    delay(1000);
    LINE_Notify("ความชื้นอากาศ:" + String(h, 1) + " เปอร์เซ็นต์  ");
    Serial.println("    Humid Line !!!");
    delay(1000);

    count = 0;
  }
}

```

เว็บจะรับค่าจาก sensor ที่ค่าอุณหภูมิและความชื้นถ้าเวลา = setsec ที่ตั้งไว้จะส่งข้อมูลไปทางไลน์

```

void messageReceived(String topic, String payload, char * bytes, unsigned int length) {
  Serial.print("incoming: ");
  Serial.print(topic);
  Serial.print(" - ");
  Serial.print(payload);
  Serial.println();
}

```

รับข้อมูลจากเว็บไซต์มาที่ board

```

bool LINE_Notify(String message) {
  WiFiClientSecure client;

  if (!client.connect("notify-api.line.me", 443)) {
    Serial.println("connection failed");
    return false;
  }

  String payload = "message=" + message;
  String req = "";
  req += "POST /api/notify HTTP/1.1\r\n";
  req += "Host: notify-api.line.me\r\n";
  req += "Authorization: Bearer " + String(LINE_TOKEN) + "\r\n";
  req += "User-Agent: ESP32\r\n";
  req += "Content-Type: application/x-www-form-urlencoded\r\n";
  req += "Content-Length: " + String(payload.length()) + "\r\n";
  req += "\r\n";
  req += payload;
  client.print(req);

  delay(20);

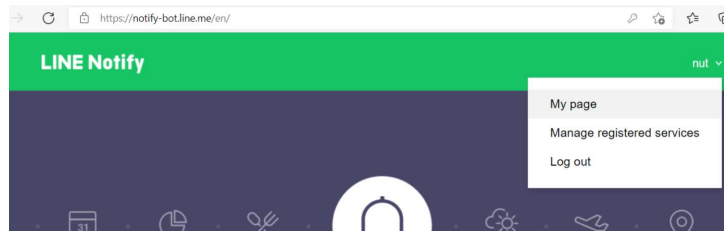
  long timeOut = millis() + 1000; // 30000
  Serial.println("timeOut = " + String(timeOut));
  while (client.connected() && timeOut > millis()) {
    if (client.available()) {
      String str = client.readString();
      Serial.print(str);
    }
    delay(10);
  }
  return timeOut > millis();
}

```

จะใช้ api ของ line ในการส่งค่าและรับค่า โดยใช้ส่งข้อความไป

## ขั้นตอนการออกแบบและการทำงาน

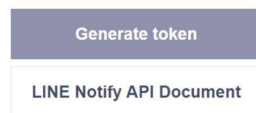
ขั้นตอนการขอ token line ไปที่เว็บไซต์ดังภาพ



กด my page แล้วเลือก generate token

## Generate access token (For developers)

By using personal access tokens, you can configure notifications without having to add a web service.



ตั้งชื่อและเลือกแบบ 1 on 1


**Generate token** ✕


Please enter a token name to be displayed before each notification.


**ตั้งชื่อ**


Select a chat to send notifications to.


🔍 Search by group name

 **1-on-1 chat with LINE Notify**

 100% // KoL

 CPE ทีม 1 rox

 CPEอมากมีเก็บร้

 DWG-Fri 621(150310)

Note: Revealing your personal access token can allow a third party to obtain the names of your connected chats as well as your profile name.

**Generate token**

เอาตัว user ไปใช้งาน

ACCOUNT INFORMATION

ACCOUNT INFORMATION

FULL NAME

ณัฐ บุญมี \*

NICKNAME

TELEPHONE NUMBER

0946514255 \*

USERNAME (CAN NOT EDIT)

nattalesrunner

EMAIL (CAN NOT EDIT)

nut.bun@ku.th

TIMEZONE OFFSET (FOR DATE DISPLAY) \*

คัดลอกถึง token ของตัวผู้ใช้ที่สร้างเพื่อไปใช้ใน code program

KEY (CASE-SENSITIVE)

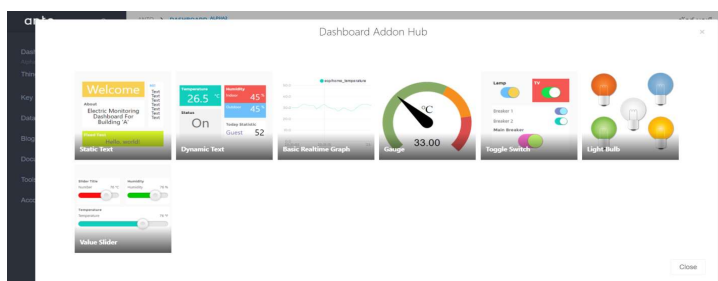
IUFaCcA9WUMJHir2

เอาชื่อ thing ไปใช้

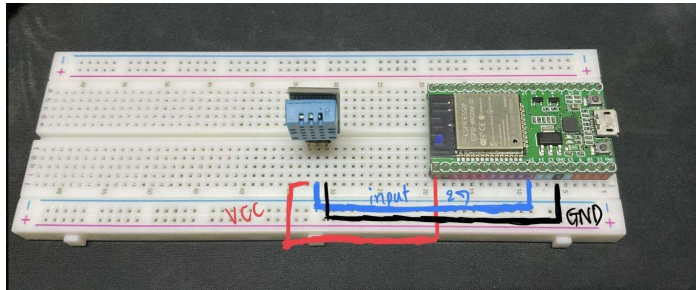
### THING LIST

NAME	DESCRIPTION
myserver	myserver

ตัวเลือกในการออกแบบหน้าเว็บที่สร้างซึ่งสามารถออกแบบเองได้



### วิธีการใช้งาน



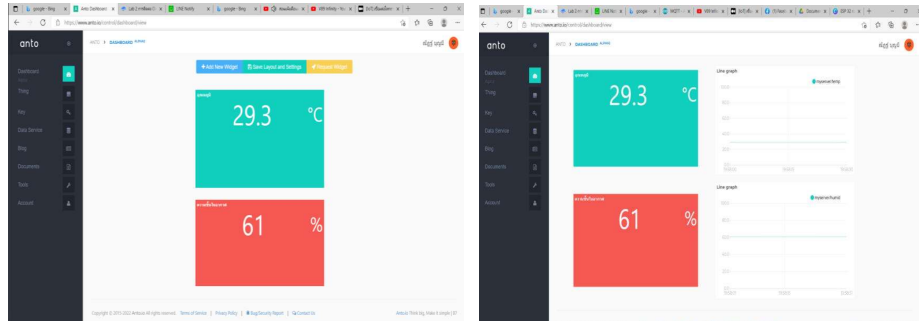
+

### แหล่งจ่ายไฟ



or

## สรุปผลการใช้งาน



แสดงผลข้อมูล อุณหภูมิและความชื้นในอากาศในอากาศทางเว็บ ทางไลน์จะแจ้งเตือนตามเวลาที่ set ไว้ใน code โปรแกรม