

บทที่ 4

ผลการดำเนินงาน

ผู้ศึกษาได้ดำเนินการศึกษาเพื่อพัฒนาระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ของมหาวิทยาลัยราชภัฏเชียงใหม่ ตามขั้นตอนที่ได้กำหนดไว้ตามลำดับ สามารถนำเสนอผลการวิจัยออกเป็น 4 ส่วน ดังนี้

ส่วนที่ 1 ผลการศึกษาระดับความขึ้นและอุณหภูมิของห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ ของมหาวิทยาลัยราชภัฏเชียงใหม่

ส่วนที่ 2 ผลการวิเคราะห์ความคิดเห็น โดยการสนทนากลุ่ม (Focus Group) จำนวน 10 คน เกี่ยวกับการใช้ระบบตรวจสอบสภาพแวดล้อมในห้องบริหาร ภายในมหาวิทยาลัยราชภัฏเชียงใหม่ การใช้แอปพลิเคชันไลน์ในการจัดการระบบเครือข่ายคอมพิวเตอร์ และเกี่ยวกับแนวทางการแก้ปัญหา กระบวนการทำงานของระบบตรวจสอบสภาพแวดล้อมในห้องบริหาร โดยผ่านแอปพลิเคชันไลน์ โดยสรุปความคิดเห็นในประเด็นต่าง ๆ

ส่วนที่ 3 ผลการพัฒนาระบบระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ของมหาวิทยาลัยราชภัฏเชียงใหม่

ส่วนที่ 4 ผลการประเมินระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ของมหาวิทยาลัยราชภัฏเชียงใหม่

ส่วนที่ 1 ผลการศึกษาระดับความขึ้นและอุณหภูมิของห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ ของมหาวิทยาลัยราชภัฏเชียงใหม่

จากการศึกษาอุณหภูมิและความขึ้นในห้องบริการเครื่องแม่ข่ายที่เกิดจากกรณีไฟฟ้าดับ พบว่า ผลการแจ้งเตือนการเกิดความผิดปกติในห้องบริการคอมพิวเตอร์แม่ข่ายมีความแม่นยำ โดยจากการติดตั้งอุปกรณ์ตรวจสอบอุณหภูมิและความขึ้น มีการเก็บข้อมูลตั้งแต่วันที่ 1 มิ.ย. 2561 ถึง วันที่ 31 ส.ค. 2561 พบว่ามีการแจ้งเตือนมายังผู้ดูแลระบบเมื่อมีความผิดปกติเกิดขึ้นตามค่าที่ตั้งไว้จำนวน 372 ครั้ง ทั้งนี้มีความคลาดเคลื่อนจำนวน 18 ครั้ง คิดเป็นร้อยละ 4.84 อยู่ในเกณฑ์ที่ผู้ดูแล

ระบบยอมรับได้ โดยไม่มีอุปกรณ์คอมพิวเตอร์แผ่ขยายเสียหายจากกรณีไฟฟ้าดับ อุณหภูมิและความชื้นที่เปลี่ยนแปลง

ส่วนที่ 2 ผลการวิเคราะห์ความคิดเห็น โดยการสนทนากลุ่ม (Focus Group)

จากการสรุปผลจากแบบสอบถามความคิดเห็น โดยการสนทนากลุ่ม (Focus Group) เกี่ยวกับการใช้ระบบตรวจสอบสภาพแวดล้อมในห้องบริหาร ภายในมหาวิทยาลัยราชภัฏเชียงใหม่ การใช้แอปพลิเคชันไลน์ในการจัดการระบบเครือข่ายคอมพิวเตอร์ และเกี่ยวกับแนวทางการแก้ปัญหา กระบวนการทำงานของระบบตรวจสอบสภาพแวดล้อมในห้องบริหาร ผ่านแอปพลิเคชันไลน์ โดยสรุปความคิดเห็นในประเด็นต่าง ๆ ดังต่อไปนี้

ผลการสนทนากลุ่ม ในประเด็น “มหาวิทยาลัยของท่านมีการนำระบบตรวจสอบสภาพแวดล้อมมาใช้ในห้องบริหาร หรือไม่ ? อย่างไร” พบว่า ในมหาวิทยาลัยราชภัฏเชียงใหม่ยังไม่มี การนำระบบตรวจจับอุณหภูมิ ความชื้น หรือตรวจสอบสภาพแวดล้อม อื่น ๆ มาใช้ภายในหน่วยงาน หรือคณะต่าง ๆ ซึ่งจะเป็นประโยชน์อย่างมาก ทั้งในคณะหรือหน่วยงานที่มีความจำเป็นในการใช้งาน อุปกรณ์ต่าง ๆ เป็นเวลานาน หรือภายในห้องแลปที่มีเครื่องที่ต้องเปิดใช้ตลอดเวลา

ผลการสนทนากลุ่ม ในประเด็น “หากมีระบบตรวจสอบสภาพแวดล้อมในห้องบริหารภายใน หน่วยงาน จะทำให้กระบวนการทำงานของท่านมีประสิทธิภาพมากขึ้นหรือไม่ อย่างไร” พบว่า ผู้เข้าร่วมสนทนา ส่วนใหญ่มีความคิดเห็นตรงกันว่า เมื่อหน่วยงานมีระบบตรวจสอบสภาพแวดล้อมใน ห้องบริหารจะทำให้บุคลากรที่ดูแลห้องบริการสามารถดูแล หรือรับทราบปัญหาของสภาพแวดล้อม ภายในห้องบริการได้ตลอดเวลา ทำให้สามารถแก้ปัญหาได้ทัน่วงที่

ผลการสนทนากลุ่ม ในประเด็น “ท่านใช้แอปพลิเคชันไลน์ ในการสื่อสารลักษณะใด รูปแบบใดบ้าง” พบว่า อันดับหนึ่งผู้เข้าร่วมการสนทนาใช้แอปพลิเคชันไลน์ ในการติดต่อกับเพื่อน ครอบครัว ติดต่อเรื่องงานหรือเป็นช่องทางในการซื้อของ รองลงมาคือการใช้แอปพลิเคชันไลน์ในการอ่านข่าว และการใช้แอปพลิเคชันไลน์ในการเล่นเกมน

ผลการสนทนากลุ่ม ในประเด็น “ท่านคิดว่าการใช้แอปพลิเคชันไลน์ในตรวจสอบ สภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT มี ประสิทธิภาพมากน้อยอย่างไร และให้ประโยชน์อย่างไรกับการสื่อสาร” พบว่า ผู้เข้าสนทนามีความ

คิดเห็นว่า ปัจจุบันเทคโนโลยีเข้ามามีส่วนร่วมในทุก ๆ ขั้นตอน แม้กระทั่งการทำงาน เทคโนโลยีก็ต้องมีส่วนที่จะมาปรับใช้ในการทำงานมากที่สุด และแนวคิด IoT ที่ย่อมาจาก Internet of Things ซึ่งเป็นส่วนหนึ่งที่จะช่วยให้อุปกรณ์รอบตัวของเรานั้น สามารถทำงานเชื่อมต่อและควบคุมได้อย่างอัจฉริยะ เสมือนว่าอุปกรณ์สามารถเชื่อมต่อและพูดคุยกันเอง ทำให้การใช้ชีวิตประจำวันของเราสะดวกสบายและปลอดภัยมากขึ้น การจะประยุกต์ IoT มากปรับใช้ในลักษณะของงานย่อมทำให้งานนั้นมีประสิทธิภาพมากยิ่งขึ้น ก่อให้เกิดความสะดวกสบาย ความรวดเร็ว และง่ายมากขึ้นอีกด้วย

ผลการสนทนากลุ่ม ในประเด็น “หากมีการใช้ระบบตรวจสอบสภาพแวดล้อมในห้องบริหาร และจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ในกระบวนการทำงานในหน่วยงานของท่าน ท่านคิดว่าจะพบปัญหาอะไรบ้าง และสามารถปรับปรุง แก้ปัญหานั้นอย่างไร ” พบว่า ผู้เข้าสนทนามีความคิดเห็นว่า การติดตั้งอุปกรณ์วัดอุณหภูมิและความชื้นนั้น อาจจะไม่เที่ยงตรงในบางครั้งเนื่องจาก อุณหภูมิของห้องนั้นขึ้นอยู่กับเครื่องวัดเพียงเครื่องเดียว หรือวัดเพียงแค่อุปกรณ์เดียว ดังนั้นวิธีแก้ไขคือ ให้ติดตั้งอุปกรณ์ในจุดสำคัญหลายจุด เพื่อกระจายการวัดอุณหภูมิได้หลายจุด

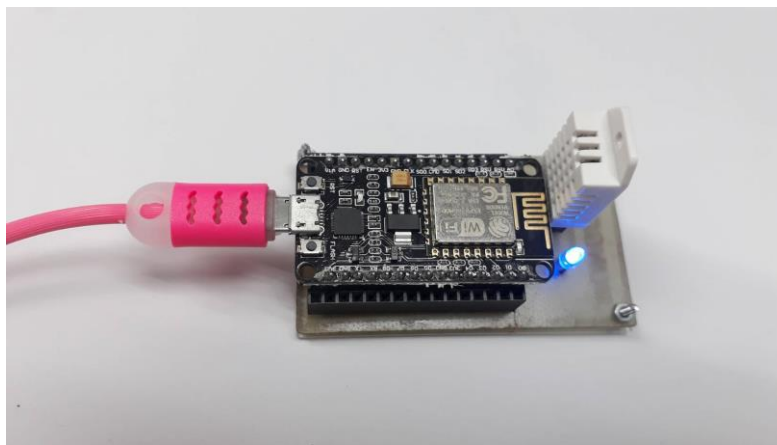
ผลการสนทนากลุ่ม ในประเด็น “ท่านมีข้อเสนอแนะอย่างไร หากมีการพัฒนาระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ภายในมหาวิทยาลัยราชภัฏเชียงใหม่” พบว่าผู้เข้าสนทนามีข้อเสนอแนะแรกคือ ควรพัฒนาระบบที่สามารถใช้งานได้จริงและเป็นประโยชน์ต่อการทำงาน ข้อที่สองคือ เมื่อพัฒนาระบบขึ้นมาแล้วควรมีการปรับปรุงอย่างสม่ำเสมอ เพื่อให้ผลที่ได้มีความแม่นยำและถูกต้อง และข้อสุดท้ายคือ ควรมีการสำรวจความต้องการของหน่วยงาน ความคุ้มค่า คุ้มค่า และผลที่จะได้รับจากการพัฒนาระบบนี้ขึ้นมาด้วย

ส่วนที่ 3 ผลการพัฒนาระบบ

จากการผลการวิเคราะห์ความคิดเห็น โดยการสนทนากลุ่ม (Focus Group) ข้างต้น ผู้วิจัยจึงได้ทำการออกแบบระบบจากความคิดเห็นที่ได้สอบถามมาจากผู้เกี่ยวข้อง และจัดทำระบบโดยมีขั้นตอนดังต่อไปนี้

1. ประกอบชิ้นส่วนอุปกรณ์เครื่องวัดอุณหภูมิ

ทำการสร้างและประกอบโมดูลวัดอุณหภูมิ (DHT22 Temperature and Humidity Sensor Module) ที่เป็นโมดูลวัดอุณหภูมิและความชื้นที่สามารถวัดอุณหภูมิบริเวณในห้อง ร่วมกับ MCU ESP8266 ดังภาพที่ 4.1 ซึ่งเป็นแพลตฟอร์มในการสร้างงานวิจัย ซึ่งตัวบอร์ดประเภทนี้ประกอบไปด้วย ซอฟต์แวร์ และโมดูล WiFi ที่ช่วยให้สามารถเชื่อมต่อเข้ากับอินเทอร์เน็ตได้ โดยใช้ port serial และใช้การเขียนโปรแกรมคำสั่งในการควบคุมการทำงานของบอร์ด เมื่อได้ทำการประกอบเรียบร้อยแล้ว และทำการเขียนโปรแกรมควบคุมในขั้นตอนต่อไปของการสร้างเครื่องวัดอุณหภูมิห้องขนาดเล็กแจ้งเตือนผ่าน Line application



ภาพที่ 4.1 การเชื่อมต่อ ESP8266 เซนเซอร์วัดอุณหภูมิ DHT22



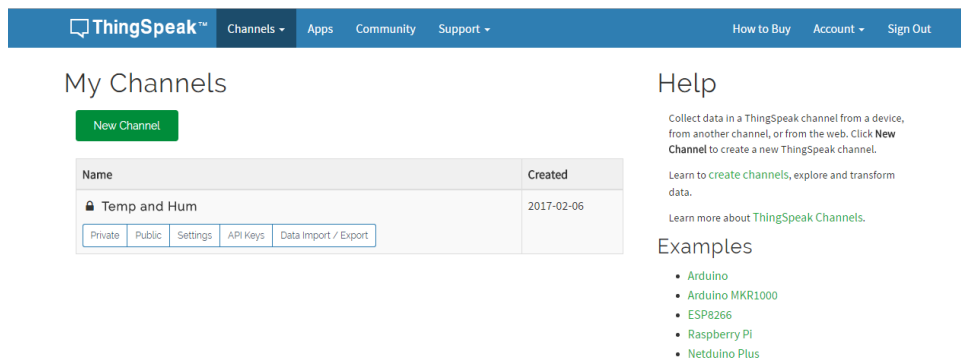
ภาพที่ 4.2 การติดตั้ง ESP8266 และเซนเซอร์วัดอุณหภูมิ DHT22 ผ่านเครื่องควบคุม

2. สร้างแอปพลิเคชันการบันทึกอุณหภูมิและความชื้นในอากาศจาก Thing Speak

Thing Speak เป็น open source Internet of Things (IoT) Platform และ API ทำการเก็บรับค่าจากเซ็นเซอร์ผ่านจัดเก็บและดึงข้อมูลจากสิ่งต่าง ๆ โดยใช้โปรโตคอล HTTP ผ่านทางอินเทอร์เน็ตหรือผ่าน Local Area Network Thing Speak ช่วยให้สามารถสร้างแอปพลิเคชันการบันทึกข้อมูลเซ็นเซอร์ แอปพลิเคชันการติดตามตำแหน่งและเครือข่ายทางสังคมของสิ่งที่มีการอัปเดตสถานะได้ โดยมีขั้นตอนการสร้างแอปพลิเคชันการบันทึกอุณหภูมิและความชื้นในอากาศจาก Thing Speak ดังนี้

ขั้นตอนที่ 1 สร้าง Channel

1) เริ่มจาก Login เข้าสู่ระบบ แล้ว กดที่ปุ่ม New Channel



ภาพที่ 4.3 การสร้าง Channel ใน Thing Speak

2) จากนั้น ตั้งค่าช่องตามต้องการ (ข้อมูลหลักๆ ที่ต้องการ ดังนี้)

- Name ชื่อช่อง Channel
- Description คำอธิบาย
- Field 1 - Field 8 คือ ชื่อของตาราง thingspeak.com ยอมให้สร้างได้ 8 กราฟ
- Make Public ต้องการให้แสดงเป็น สาธารณะหรือไม่

ขั้นตอนที่ 2 เพิ่มรายละเอียดข้อมูลภายใน Channel

ThingSpeak™ Channels Apps Community Support How to Buy Account Sign Out

New Channel

Name:

Description:

Field 1: ☒

Field 2: ☒

Field 3: ☐

Field 4: ☐

Field 5: ☐

Field 6: ☐

Field 7: ☐

Field 8: ☐

Metadata:

Tags:
(Tags are comma separated)

Make Public: ☐

URL:

Elevation:

Show Location: ☐

Latitude:

Longitude:

Show Video: ☐

* YouTube ☐

Vimeo ☐

Video ID:

Show Status: ☐

Save Channel

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Latitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the latitude of the city of London is 51.5072.
- Longitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
- Elevation:** Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
- Make Public:** If you want to make the channel publicly available, check this box.
- URL:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Video ID:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.

Using the Channel

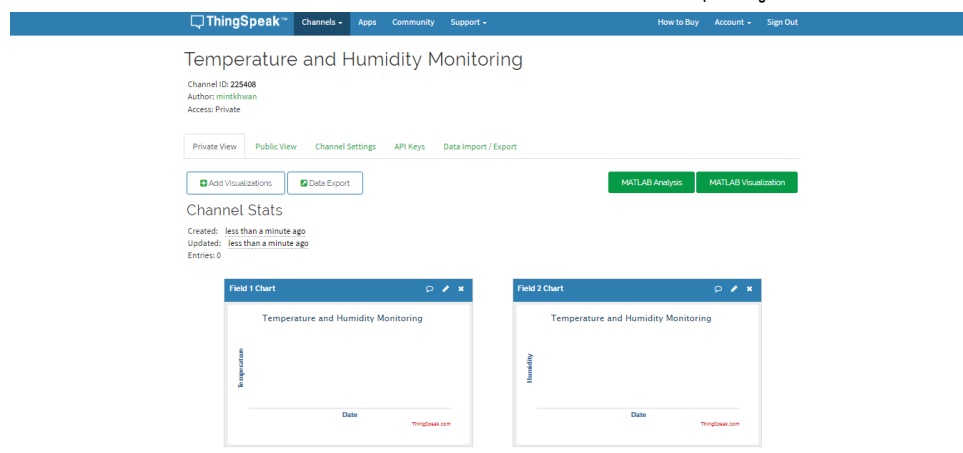
You can get data into a channel from a device, website, or another ThingSpeak channel. You can then visualize data and transform it using [ThingSpeak Apps](#).

See [Tutorial: ThingSpeak and MATLAB](#) for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

[Learn More](#)

ภาพที่ 4.4 เพิ่มรายละเอียดข้อมูลภายใน Channel ใน Thing Speak

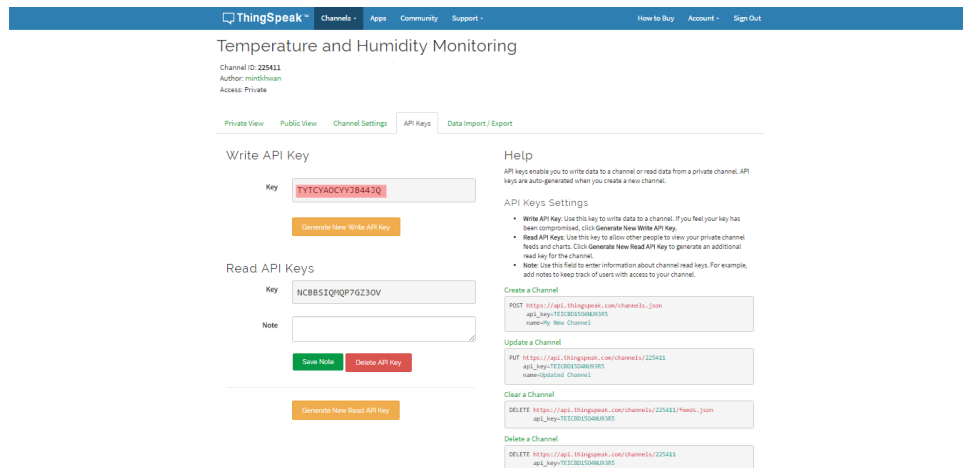
ขั้นตอนที่ 3 เมื่อสร้าง Chanel จะมีส่วนที่ใช้ในการแสดงผลค่าอุณหภูมิ ความชื้น



ภาพที่ 4.5 ส่วนที่ใช้ในการแสดงผลค่าอุณหภูมิ ความชื้น

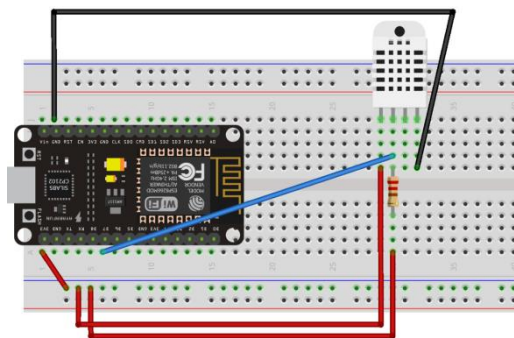
ขั้นตอนที่ 4 API Keys ส่วนที่ใช้ส่งข้อมูลจาก DHT22 มายัง Thing Speak

เมื่อสร้างช่อง Channel ให้คลิกไปยังช่องที่สร้างขึ้นแล้วกด ที่แถบ API Key เก็บค่า Write API Key ให้นำไปแก้ไข Code ของ Nodemcu esp8266



ภาพที่ 4.6 API Keys ส่วนที่ใช้ส่งข้อมูลจาก DHT22 มายัง Thing Speak

ขั้นตอนที่ 5 ประกอบวงจรให้ Sensor DHT22 ทำการรับค่า Temperature และ Humidity



ภาพที่ 4.7 ประกอบวงจรให้ Sensor DHT22 ทำการรับค่า Temperature และ Humidity

ขั้นตอนที่ 6 เขียน code เพื่อให้แสดงค่า Temperature และ Humidity

โดยมีรายละเอียด code ดังนี้

```
#include <ESP8266WiFi.h>
```

```
#include "DHT.h"
```

```
#define DHTPIN D4 // ขา Out ของ Sensor ต่อเข้าขา D4 ของ Esp8266
```

```
//เลือกชนิดของ Sensor
```

```

#define DHTTYPE DHT11    //DHT 11

//#define DHTTYPE DHT22  // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21  // DHT 21 (AM2301)

DHT dht(DHTPIN, DHTTYPE);

const char* ssid    = "SSID Wifi";  // SSID Wifi

const char* password = "Password Wifi"; // Password Wifi

const char* host = "api.thingspeak.com"; // Host ของ thingspeak ไม่ต้องแก้ไข

const char* api  = "RRHS37ETW76RFAWB"; //API Key ที่เราจำไว้ในขั้นตอนเมื่อครั้ง

void setup() {

  Serial.begin(9600);

  delay(10);

  // We start by connecting to a WiFi network

  Serial.println();

  Serial.println();

  Serial.print("Connecting 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());

  Serial.println("DHTxx test!");

  dht.begin();

```

```

}

int value = 0;

void loop() {
  delay(5000);
  ++value;
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");
  Serial.print("Heat index: ");
  Serial.print(hic);
  Serial.print(" *C ");
  Serial.print(hif);
  Serial.println(" *F");
}

```

```

Serial.print("connecting to ");
Serial.println(host);

// Use WiFiClient class to create TCP connections
WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
}

// We now create a URI for the request
String url = "/update?api_key=";
url += api;
url += "&field1=";
url += t;
url += "&field2=";
url += h;
// เราจะส่งข้อมูล
https://api.thingspeak.com/update?api_key=RRHS37ETW76RFAWB&field1=
(อุณหภูมิ)&field2=(ความชื้น)

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");

delay(10);

```

```
// Read all the lines of the reply from server and print them to Serial

while(client.available()){

    String line = client.readStringUntil("\r");

    Serial.print(line);

}

Serial.println();

Serial.println("closing connection");

}
```

ขั้นตอนที่ 7 สามารถเช็คค่า Temperature และ Humidity จาก Serial Monitor

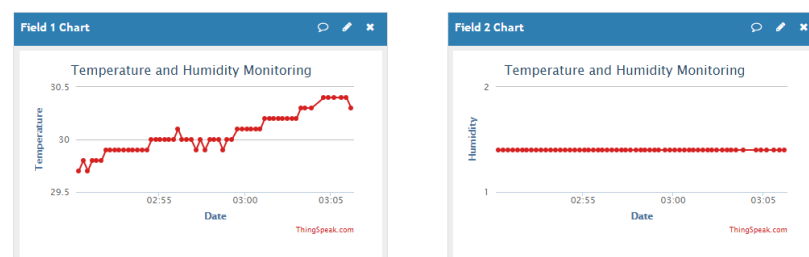


ภาพที่ 4.8 การเช็คค่า Temperature และ Humidity จาก Serial Monitor

ขั้นตอนที่ 8 การแสดงค่า Temperature และ Humidity ในกราฟ บน Cloud Service

Channel Stats

Created: 37 minutes ago
 Updated: less than a minute ago
 Last entry: less than a minute ago
 Entries: 58

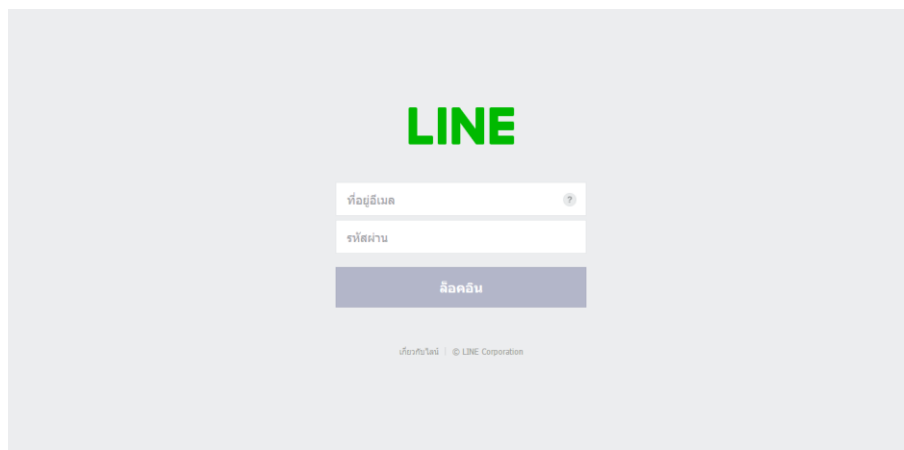


ภาพที่ 4.9 การแสดงค่า Temperature และ Humidity ในกราฟ บน Cloud Service

3. การแจ้งเตือนผ่าน Line notify ใน Line Application

การส่งการแจ้งเตือนค่าอุณหภูมิไปยังผู้ใช้งานไลน์ได้นั้น จะต้องมีการเพิ่มเพื่อนกับ Line notify ในสมาร์ทโฟนก่อน เพื่อที่จะทำให้ Line notify สามารถส่งการแจ้งเตือนข้อมูลได้ แล้วทำการสมัคร Line notify เพื่อเข้าไปรับรหัส Token ของ Line notify ไว้ใช้งานการแจ้งเตือนของ Line group หลังจากนั้นสร้าง Line group

การใช้งาน API ในทุก ๆ บริการ จะมีสิ่งที่เรียกว่า Access Token ไว้สำหรับเป็นรหัสที่ใช้สำหรับเข้าใช้งาน API โดยรหัสนี้จะเป็นข้อความแทนอีเมล และรหัสผ่าน โดยเข้าไปที่หน้าเว็บ <https://notify-bot.line.me/my/> จากนั้นระบบจะให้ทำการล็อกอินด้วย Account LINE โดยกรอกอีเมล และรหัสผ่านที่ได้ตั้งไว้ลงไป

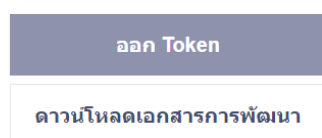


ภาพที่ 4.10 การกรอก User Password เพื่อการใช้งาน

เมื่อล็อกอินสำเร็จแล้วให้เลื่อนลงมาด้านล่างจะพบออก Access Token (สำหรับผู้พัฒนา) ให้กดปุ่ม ออก Token

ออก Access Token (สำหรับผู้พัฒนา)

เมื่อใช้ Access Token แบบบุคคล จะสามารถตั้งค่าการแจ้งเตือนได้โดยไม่ต้องลงทะเบียนกับเว็บเซอร์วิส

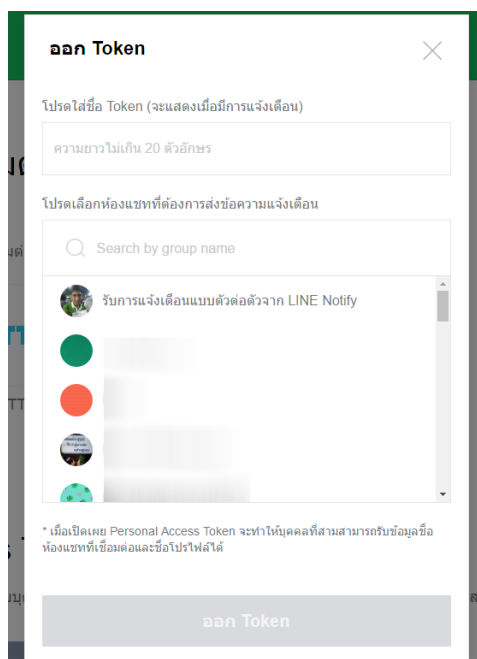


ภาพที่ 4.11 การออก Access Token

เมื่อกดออก Token เรียบร้อยแล้วระบบจะให้กรอกข้อมูลที่จะให้แสดงดังนี้

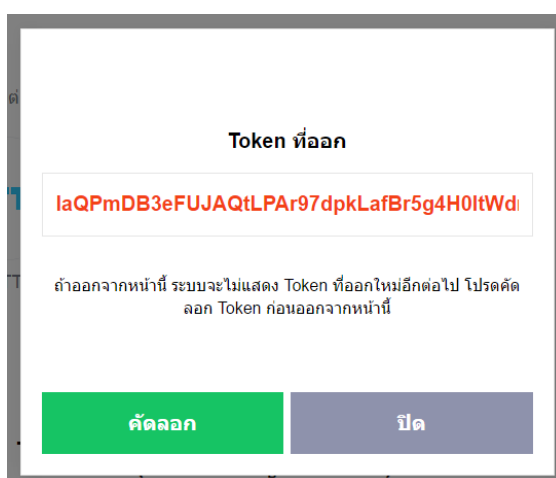
ช่องที่ 1 สามารถกรอกเป็นอะไรก็ได้ และสิ่งที่กรอกนั้นจะติดไปพร้อมกับข้อความเสมอ เช่น หากกรอกว่า Arduino เมื่อใช้ API ส่งข้อความว่า "สวัสดี" ข้อความจะขึ้นว่า "Arduino:สวัสดี"

ช่องที่ 2 จะให้เลือกว่าจะส่งข้อความเข้าไปในกลุ่มไหนหรือส่งให้ตัวเองเท่านั้นเมื่อกรอกครบแล้วให้กดปุ่มออก Token

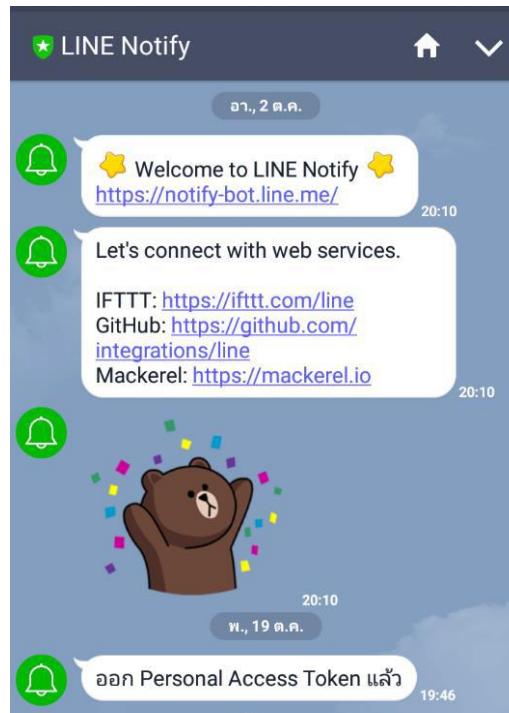


ภาพที่ 4.12 การเลือกส่งข้อมูลที่จะให้แสดง

เมื่อกดปุ่มแล้วจะปรากฏรหัส Token ให้เก็บรหัสนี้ไว้ให้ดีเพราะจะออกให้เพียงครั้งเดียว ส่วนใน LINE ก็จะมีการแจ้งเตือนว่าออก Access Token ใหม่แล้ว



ภาพที่ 4.13 การแสดง Access Token



ภาพที่ 4.14 การแจ้งเตือนว่าออก Access Token สำเร็จ

4. วิธีการใช้งานร่วมกับ NodeMCU ให้นำ Token ที่ได้มาใส่ใน Code บรรทัดที่ระบุไว้ในรูปภาพด้านล่าง แล้วแก้ไขข้อความที่ต้องให้ Line ส่งไปยังกลุ่มที่ต้องการ แล้วอัปโหลดไปยัง Board NodeMCU



ภาพที่ 4.15 วิธีการใช้งานร่วมกับ NodeMCU

5. การแจ้งเตือนผ่านไลน์ ได้ออกแบบเงื่อนไขการแจ้งเตือนไว้ คือ ทุก ๆ หนึ่งชั่วโมง จะแจ้งเตือนค่าอุณหภูมิ และค่าความชื้นสัมพัทธ์เฉลี่ยที่เซนเซอร์ วัดได้ผ่าน Line group โดยใช้ Line notify



ภาพที่ 4.16 ตัวอย่างแสดงการแจ้งเตือน ค่าอุณหภูมิและความชื้นผ่าน Line notify

ส่วนที่ 4 ผลการประเมินระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ของมหาวิทยาลัยราชภัฏเชียงใหม่

ตอนที่ 1 ความคิดเห็นเกี่ยวกับประสิทธิภาพของระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ของมหาวิทยาลัยราชภัฏเชียงใหม่ แบบประเมินระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ของมหาวิทยาลัยราชภัฏเชียงใหม่โดยเก็บข้อมูลจากเจ้าหน้าที่ผู้ทดลองใช้ระบบ จำนวน 10 คน

ตารางที่ 4.1 ผลการวิเคราะห์ประสิทธิภาพของระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ของมหาวิทยาลัยราชภัฏเชียงใหม่

รายการ	ค่าเฉลี่ย	SD	ระดับความคิดเห็น
ด้านการทำงานของระบบ			
1. สามารถทำการแจ้งเตือนได้อย่างรวดเร็ว	4.88	0.65	มากที่สุด
2. การแสดงข้อมูลการแจ้งเตือนอุณหภูมิและความชื้นมีความถูกต้อง	4.67	0.72	มากที่สุด
3. การแสดงข้อมูลแผนภูมิ กราฟอุณหภูมิมีความถูกต้อง	4.58	0.78	มากที่สุด
4. การส่งข้อมูลอุณหภูมิและความชื้นให้ผู้ใช้ทราบผ่าน LINE Application ได้ถูกต้อง	4.75	0.71	มากที่สุด
5. การส่งข้อมูลมีความแน่นอน ถูกต้อง เสียรและเที่ยงตรง	4.60	0.55	มากที่สุด
รวมด้าน	4.70	0.69	มากที่สุด
ด้านความสามารถในการใช้งานของระบบ			
1. ความสะดวกในการใช้งาน	4.65	0.85	มากที่สุด
2. ความเร็วในการส่งข้อมูลอุณหภูมิในแต่ละครั้ง	4.63	0.83	มากที่สุด
3. ความถูกต้องของข้อมูลเมื่อทำการแจ้งเตือนอุณหภูมิและความชื้น	4.68	0.83	มากที่สุด
4. อุปกรณ์มีขนาดที่เหมาะสม และสะดวกเมื่อทำการติดตั้ง	4.45	0.75	มาก

รายการ	ค่าเฉลี่ย	SD	ระดับความคิดเห็น
5. ความเหมาะสมของการแสดงข้อมูลบนหน้าจอสมาร์ทโฟน	4.54	0.71	มากที่สุด
รวมด้าน	4.59	0.74	มากที่สุด

ตารางที่ 4.1 (ต่อ)

รายการ	ค่าเฉลี่ย	SD	ระดับความคิดเห็น
ด้านประโยชน์ที่ได้จากระบบ			
1. ผู้ใช้สามารถตรวจสอบอุณหภูมิและความชื้นผ่าน Line Application ได้	4.78	0.67	มากที่สุด
2. ผู้ใช้สามารถทราบการแจ้งเตือนอุณหภูมิและความชื้นได้อย่างรวดเร็ว	4.86	0.56	มากที่สุด
3. สามารถทราบอุณหภูมิและความชื้นได้ โดยผู้ใช้ไม่ต้องอยู่ที่ห้องที่อุปกรณ์นั้นติดตั้ง	4.85	0.78	มากที่สุด
4. สะดวกต่อการติดตั้ง อุปกรณ์มีความทันสมัยและใช้งานง่าย	4.61	0.74	มากที่สุด
รวมด้าน	4.78	0.84	มากที่สุด
โดยรวม	4.69	0.75	มากที่สุด

จากตารางที่ 4.1 จากการศึกษาความคิดเห็นเกี่ยวกับประสิทธิภาพของระบบตรวจสอบสภาพแวดล้อมในห้องบริหารและจัดการระบบเครือข่ายคอมพิวเตอร์ด้วยเทคโนโลยี IOT ผ่าน Line API ของมหาวิทยาลัยราชภัฏเชียงใหม่ พบว่า โดยรวมผู้ตอบแบบสอบถามมีความคิดเห็นว่าระบบมีประสิทธิภาพมากที่สุด มีค่าเฉลี่ยรวม 4.69 จัดอยู่ในระดับความคิดเห็นมากที่สุด เมื่อวิเคราะห์แยกเป็นรายด้าน พบว่า ทุกด้านมีประสิทธิภาพมากที่สุด โดยด้านที่มีค่าเฉลี่ยมากที่สุด คือ ด้านประโยชน์ที่ได้จากระบบ มีค่าเฉลี่ย 4.78 รองลงมาคือ ด้านการทำงานของระบบ มีค่าเฉลี่ย 4.70 และ ด้านความสามารถในการใช้งานของระบบ มีค่าเฉลี่ย 4.59 ตามลำดับ