บทที่ 3

ขั้นตอนการดำเนินงาน

สำหรับในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงงานปริญญานิพนธ์ซึ่งจะทำให้ทราบถึงการวิเคราะห์และออกแบบแอปพลิเคชันโดยละเอียดว่ามีแนวทางในการดำเนินงานหรือมีขั้นตอนในการทำงานของแอปพลิเคชันอย่างไรบ้าง โดยขั้นตอนในการดำเนินงานมีรายละเอียดดังนี้

3.1 กรอบการพัฒนาระบบ

3.2 ทฤษฎีที่เกี่ยวข้องกับการออกแบบ

3.3 แผนภาพบริบท (Context Diagram)

3.4 แผนภาพการไหลของข้อมูล (Data Flow Diagram)

3.5 Data Store

3.6 External Entity Description

3.7 Data Flow (Data Flow Description and Data Structure of Data Flow)

3.8 คำอธิบายการประมวลผล (Process Description)

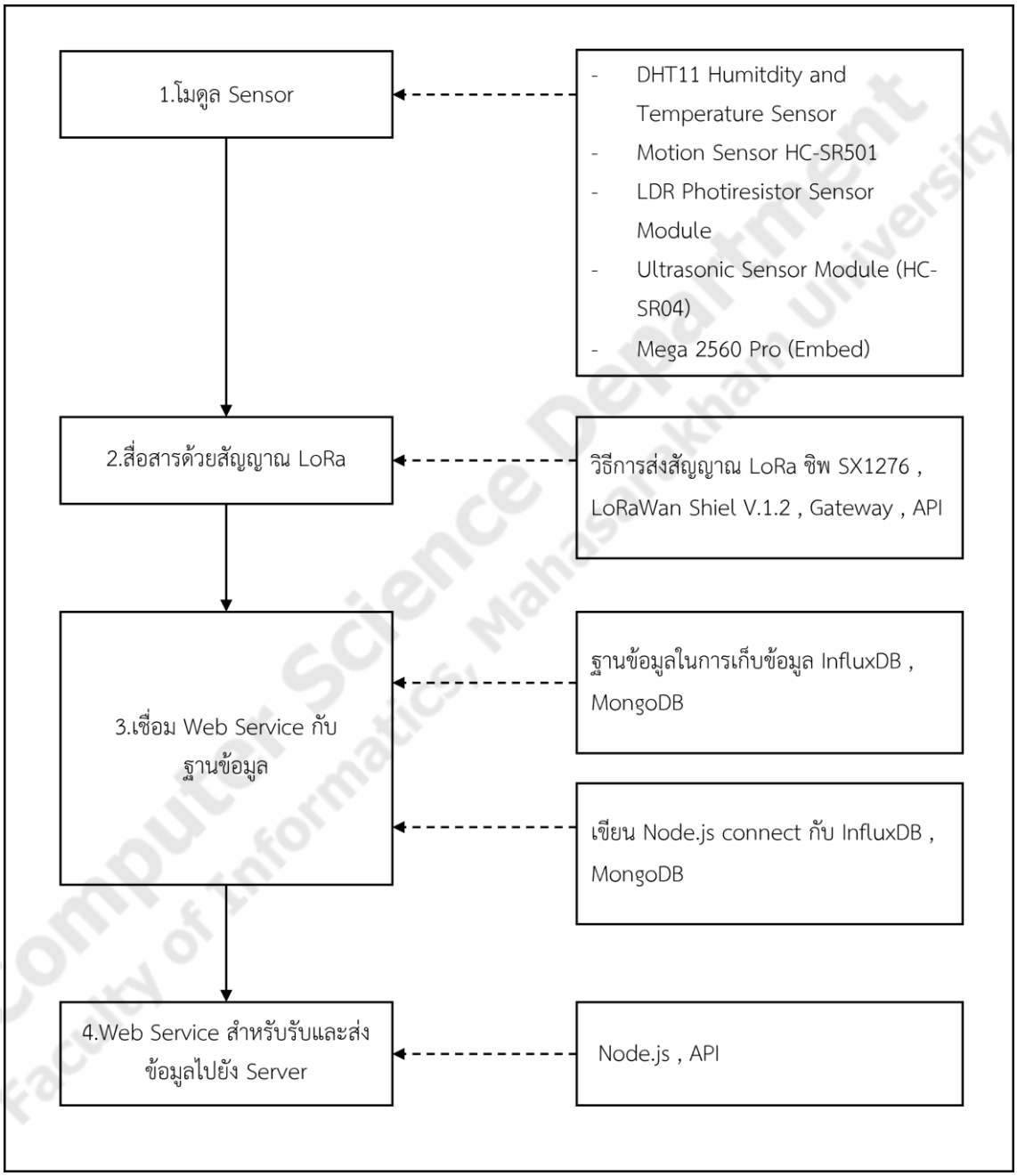
3.9 การออกแบบฐานข้อมูล (Database Design)

3.10 การนำชุดคำสั่ง (API) มาใช้

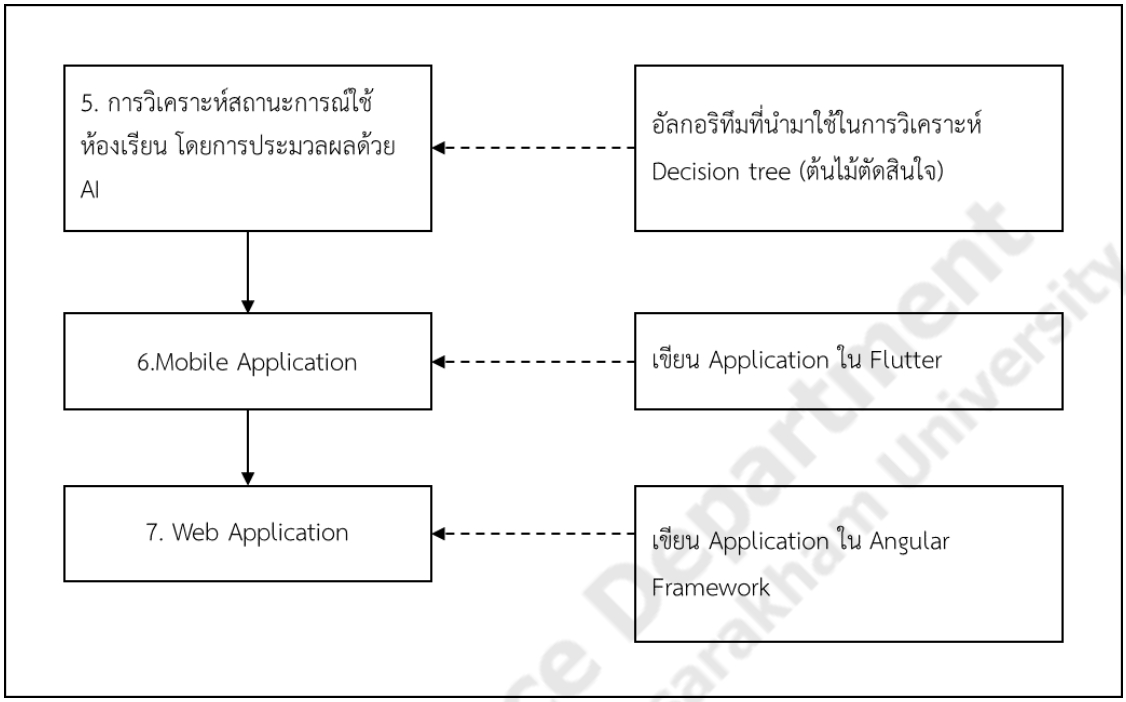
3.11 อัลกอริทึมที่ใช้ในการประมวลผล Decision tree

3.1 กรอบการพัฒนาระบบ

3.1.1 ระบบตรวจสอบการเข้าใช้ห้องเรียนแบบเรียลไทม์ จะมีโมดูลหลักๆ ดังนี้



ภาพประกอบที่ 3.1 กรอบการพัฒนาระบบ



ภาพประกอบที่ 3.1 กรอบการพัฒนาระบบ (ต่อ)

3.1.1 คำอธิบาย

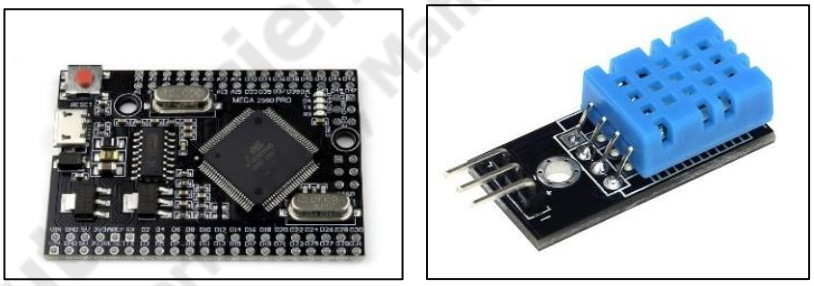
การต่ออุปกรณ์โมดูล มีอุปกรณ์ดังนี้ เซนเซอร์ทั้ง 4 อย่าง ได้แก่ 1. เซนเซอร์วัดอุณหภูมิและความชื้น DHT11 2. เซนเซอร์วัดความสว่างของแสง 3. เซนเซอร์ตรวจจับความเคลื่อนไหว 4. เซนเซอร์อัลตราโซนิก ซึ่งโมดูล Mega 2560 pro จะเป็นตัวความคุมโมดูลทั้ง 4 โมดูล โดยโพรโทรคอลที่ใช้ในการส่งสัญญาณออกไปที่ Gateway จะใช้ LoRa 920 – 925 MHz ด้วยการต่อขาของอุปกรณ์ดังนี้

Mega pro	----------→	LoRa
GND	----------→	GND
3.3 v	----------→	3.3 v
D50	----------→	MISO
D51	----------→	MOSI
D52	----------→	SCK
D53	----------→	NSS
D3	----------→	RST
D4	----------→	D0
D5	----------→	D1
D7	----------→	D2

ภาพประกอบที่ 3.2 การต่อ Module Mega pro กับ Module LoRa

วิธีการต่อเซนเซอร์ทั้ง 4 อย่าง

1. เซนเซอร์วัดอุณหภูมิและความชื้น DHT11 , Mega 2560 pro

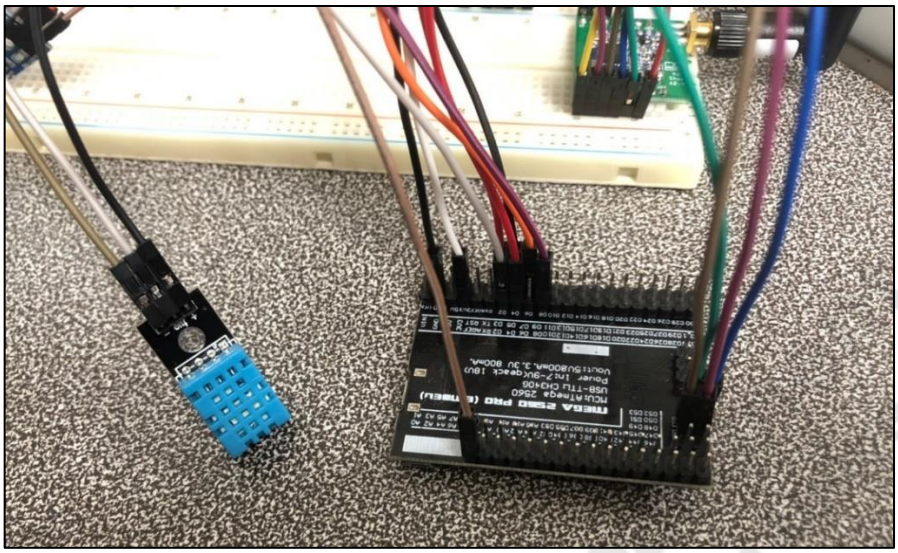


ภาพประกอบที่ 3.3 อุปกรณ์ต่อเซนเซอร์วัดอุณหภูมิ DHT11, Mega 2560 pro

วิธีการต่อ เซนเซอร์วัดอุณหภูมิและความชื้น DHT11 + Mega 2560 pro

DHT11	---→	Mega pro
5V	---→	3.3 V
GND	---→	GND
OUT	---→	D2

ภาพประกอบที่ 3.4 การต่อเซนเซอร์วัดอุณหภูมิและความชื้น



ภาพประกอบที่ 3.5 การต่ออุปกรณ์เซนเซอร์วัดอุณหภูมิและความชื้น (DHT11)

คำสั่งเซนเซอร์วัดอุณหภูมิและความชื้น DHT11

```
#include "DHT.h"
DHT dht;

void setup() {
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");
  dht.setup(2); // data pin 2

}

void loop() {

  delay(dht.getMinimumSamplingPeriod());
  float humidity = dht.getHumidity(); // ดึงค่าความชื้น
  float temperature = dht.getTemperature(); // ดึงค่าอุณหภูมิ

  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1);
  Serial.print("\t\t");
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  Serial.println(dht.toFahrenheit(temperature), 1);
  delay(1000);
}
```

ภาพประกอบที่ 3.6 คำสั้งเซนเซอร์วัดอุณหภูมิและความชื้น

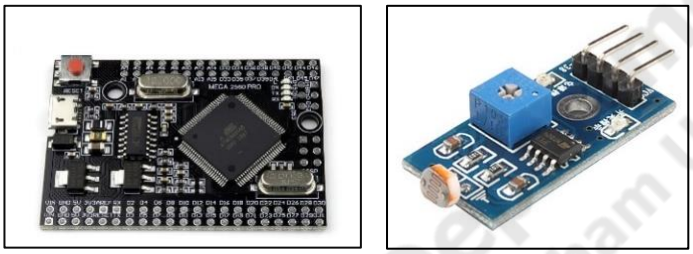
บรรทัดที่ 1 การเรียกใช้ Library วัดอุภณหมิูและความชื้น DHT11

บรรทัดที่ 2 การประกาศตัวแปรเพื่อใช้งาน

บรรทัดที่ 8 การกำหนดเซนเซอร์ DHT ต่อกับขา 2

บรรทัดที่ 12 เป็นคำสั่งในการ แสดงค่าออกมาแบบวนลูป
บรรทัดที่ 15 เป็นคำสั่งดึงค่าความชื้นในอากาศเป็นเปอร์เซ็นต์
บรรทัดที่ 16 เป็นคำสั่งดึงค่าอุณหภูมิในหน่วยองศาเซลเซียส
บรรทัดที่ 18 – 24 เป็นการพิมพ์ค่า ความชื้นในอากาศและอุณหภูมิ
บรรทัดที่ 25 เป็นคำสั่งก่อนอ่านค่าและแสดงผลครั้งถัดไป

2. เซนเซอร์วัดความสว่างของแสง LDR Photoresistor Sensor Module, Mega 2560 pro

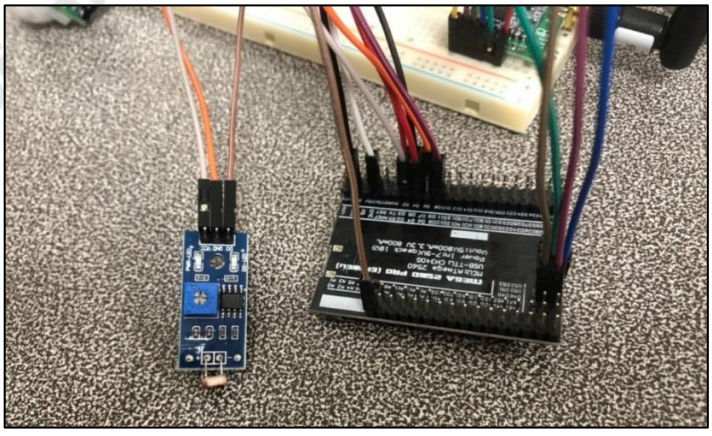


ภาพประกอบที่ 3.7 อุปกรณ์เซนเซอร์วัดความสว่างของแสง LDR, Mega 2560 pro

วิธีการต่อ เซนเซอร์วัดความสว่างของแสง LDR Photoresistor Sensor Module + Mega 2560 pro

LDR	---→	Mega pro
5V	---→	3.3 V
GND	---→	GND
A0	---→	A0
D0	---→	-

ภาพประกอบที่ 3.8 การต่อเซนเซอร์วัดความสว่างของแสง LDR



ภาพประกอบที่ 3.9 การต่อเซนเซอร์วัดความสว่างของแสง (LDR Photoresistor)

คำสั่งเซนเซอร์วัดความสว่างของแสง LDR Photoresistor Sensor Module

```
1  int SensorPin = A0;
2  int val;
3
4  void setup() {
5    pinMode(SensorPin, INPUT);
6    Serial.begin(9600);
7  }
8  void loop() {
9    val = digitalRead(SensorPin);
10   delay(500);
11   if(val == 0){
12    Serial.print(" แสงสว่าง ");   //เปิด
13    Serial.println(val);
14   }else if(val == 1){
15     Serial.print(" ไม่สว่าง"); //ปิด
16     Serial.println(val);
17   }
18 }
```

ภาพประกอบที่ 3.10 คำสั่งเซนเซอร์วัดความสว่างของแสง

บรรทัดที่ 1 เป็นการประกาศตัวแปร ให้ DigitalPin แทนขา Digital ขาที่ A0

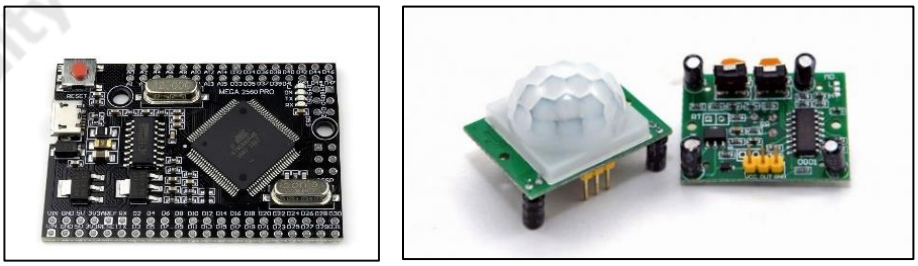
บรรทัดที่ 2 การประกาศตัวแปรเพื่อใช้งาน

บรรทัดที่ 5 เป็นการตั้งค่าพินเป็น input

บรรทัดที่ 9 อ่านค่าสัญญาณ Digital ขา A0 ที่ต่อกับ LDR Photoresistor Sensor Module

บรรทัดที่ 11 – 16 เป็นคำสั่งเช็คเงื่อนไขของค่า Digital ที่มีค่าเป็น 0 และ 1 คือถ้า ค่าเป็น 0 หมายความว่าเซนเซอร์สามารถวัดค่าความสว่างได้ และถ้าเป็น 1 หมายความว่าเซนเซอร์ไม่สามารถวัดค่าความสว่างได้

3. เซ็นเซอร์ตรวจจับความเคลื่อนไหว PIR Motion Sensor (HC-SR501) , Mega 2560 pro

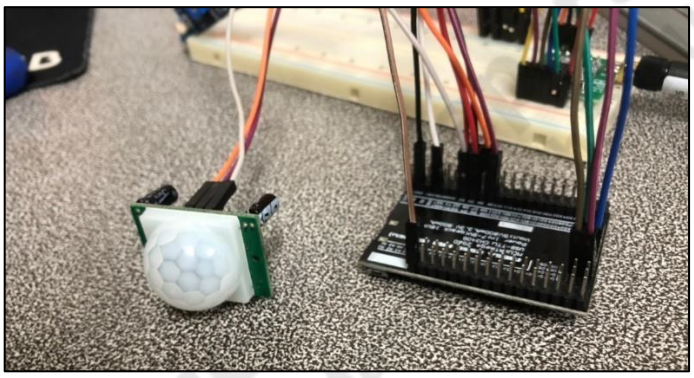


ภาพประกอบที่ 3.11 เซ็นเซอร์ตรวจจับความเคลื่อนไหว HC-SR501 ,Mega pro

วิธีการต่อ เซ็นเซอร์ตรวจจับความเคลื่อนไหว PIR Motion Sensor Detector Module (HC-SR501) + Mega 2560 pro

HC-SR501	---→	Mega pro
5V	---→	3.3 V
GND	---→	GND
OUT	---→	D8

ภาพประกอบที่ 3.12 การต่อเซ็นเซอร์ตรวจจับความเคลื่อนไหว (HC-SR501)



ภาพประกอบที่ 3.13 การต่อเซ็นเซอร์ตรวจจับความเคลื่อนไหว (HC-SR501)

คำสั่งเซ็นเซอร์ตรวจจับความเคลื่อนไหว PIR Motion Sensor (HC-SR501)

```
int digitalPin = 8;
int val = 0;

void setup() {
    pinMode(digitalPin, INPUT); // sets the pin as input
    Serial.begin(9600);
}

void loop() {
    val = digitalRead(digitalPin);
    Serial.print("val : ");
    Serial.println(val);

    if(val == 0){
      Serial.print(" ตราจจับเจอวัตถุ\n");
    }
    else{
      Serial.print(" ไม่พบวัตถุ\n");
    }
    delay(1000);
}
```

ภาพประกอบที่ 3.14 คำสั่งเซนเซอร์ตรวจจับความเคลื่อนไหว (HC-SR501)

บรรทัดที่ 1 เป็นการประกาศตัวแปร ให้ DigitalPin แทนขา Digital ขาที่ A0

บรรทัดที่ 2 การประกาศตัวแปรเพื่อใช้งาน

บรรทัดที่ 5 เป็นการตั้งค่าพินเป็น input

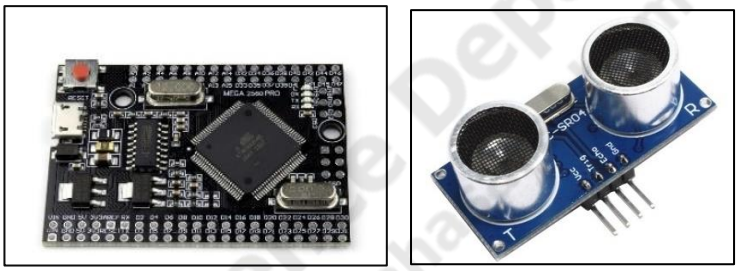
บรรทัดที่ 9 อ่านค่าสัญญาณ Digital ขา 8 ที่ต่อกับ PIR Motion Sensor

บรรทัดที่ 11 พิมพ์ข้อความส่งเข้าคอมพิวเตอร์

บรรทัดที่ 12 พิมพ์ค่าตัวแปร

บรรทัดที่ 14 – 19 เป็นคำสั่งเช็คเงื่อนไขของค่า Digital ที่มีค่าเป็น 0 และ 1 คือถ้า ค่าเป็น 0 หมายความว่าเซนเซอร์สามารถตรวจจับเจอวัตถุได้ และถ้าเป็น 1 หมายความว่าเซนเซอร์ไม่พบวัตถุได้

4. เซนเซอร์อัลตราโซนิก Ultrasonic Sensor Module (HC-SR04) , Mega 2560 pro



ภาพประกอบที่ 3.15 Ultrasonic Sensor Module (HC-SR04) , Mega 2560 pro

วิธีการต่อ เซนเซอร์อัลตราโซนิก Ultrasonic Sensor Module (HC-SR04) , Mega 2560 pro

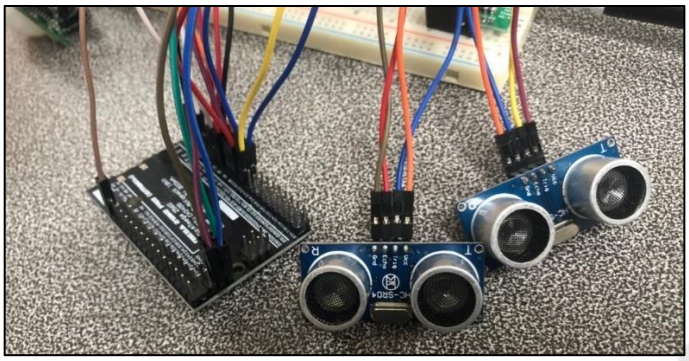
เซนเซอร์อัลตราโซนิกตัวที่ 1

HC-SR04	---→	Mega pro
5V	---→	3.3 V
GND	---→	GND
Trig	---→	D9
Echo	---→	D10

เซนเซอร์อัลตราโซนิกตัวที่ 2

HC-SR04	---→	Mega pro
5V	---→	3.3 V
GND	---→	GND
Trig	---→	D5
Echo	---→	D6

ภาพประกอบที่ 3.16 การต่อ Ultrasonic Sensor Module (HC-SR04)



ภาพประกอบที่ 3.17 การต่อ (Ultrasonic Sensor Module HC-SR04)

คำสั่งเซนเซอร์อัลตราโซนิก Ultrasonic Sensor Module (HC-SR04)

```
1  const int trigPin1 = 9;
2  const int echoPin1 = 10;
3  const int trigPin2 = 5;
4  const int echoPin2 = 6;
5
6  long duration; //เวลา
7  int distance; //ระยะเวลา
8  int count_in = 0;
9  int count_out = 0;
10 int count_sum = 0;
11 long sensorA;
12 long sensorB;
13 int a = 0;
14 int b = 0;
15
16 void setup() {
17   Serial.begin(9600);
18 }
20 void loop() {
21
22   sensorA = getLength(echoPin1, trigPin1); //นอก
23   sensorB = getLength(echoPin2, trigPin2); //ใน
24
25   if(sensorA!=0 && sensorA<sensorB){
26     in(sensorA,sensorB);
27   }
28   if(sensorB!=0 && sensorB<sensorA){
29     out(sensorA,sensorB);
30   }
31   sumPeople(count_in,count_out);
32
33 }
```

ภาพประกอบที่ 3.18 (Ultrasonic Sensor Module HC-SR04)

บรรทัดที่ 1 – 4 เป็นการประกาศขา trig และ echo ทั้งสองเซนเซอร์

บรรทัดที่ 6 – 7 เป็นการประกาศตัวแปรเก็บค่าระยะทาง

บรรทัดที่ 8 – 14 การประกาศตัวแปรเพื่อใช้งาน

บรรทัดที่ 22 – 23 เป็นการเป็นข้อมูลจากเมธอด getLength() มาเก็บที่ตัวแปรทั้งสองแทนเซนเซอร์ทั้งสองตัว โดยจะนำมานับจำนวนคนเข้าออก

บรรทัดที่ 22 – 30 เป็นคำสั่งเช็คเงื่อนไข ของข้อมูลที่มาจากเซนเซอร์ทั้งสองตัว โดยจะมาเช็คในเงื่อนไหนและไปทำงานในเมธอด in() และ out()

บรรทัดที่ 31 เป็นเมธอดที่จะนำ ข้อมูลคนเข้าและคนออกมาทำการนับจำนวนคนภายในห้อง

```
34 long in(long a1,long b1){
35
36     if(a1<=50){
37       if(b1>150){
38         a = 1;
39         count_in +=1;
40         Serial.print("จำนวนคน(เข้า) : ");
41         Serial.println(count_in);
42       }
43     }
44     delay(300);
45     return count_in;
46 }
47 long out(int a2 ,int b2){
48
49      if(b2<=50){
50         if(a2>150){
51           a = 1;
52           count_out +=1;
53           Serial.print("จำนวนคน(ออก) : ");
54           Serial.println(count_out);
55         }
56      }
57      delay(300);
58      return count_out;
59 }
60 int sumPeople(int countIn,int countOut){
61
62       count_sum = countIn - countOut;
63       Serial.print("จำนวนคนภายในห้องเรียน : ");
64       Serial.println(count_sum);
65       delay(1000);
66 }
67 long getLength(int echo , int trig){
68   pinMode(trig, OUTPUT);
69   digitalWrite(trig, LOW);
70   delayMicroseconds(2);
71
72   digitalWrite(trig, HIGH);
73   delayMicroseconds(10);
74   digitalWrite(trig, LOW);
75   pinMode(echo, INPUT);
76
77   duration = pulseIn(echo, HIGH);
78   distance = duration * 0.034/2; //คำนวณเป็น centimeters
79   delay(100);
80   return distance;
81 }
```

ภาพประกอบที่ 3.18 (Ultrasonic Sensor Module HC-SR04) (ต่อ)

บรรทัดที่ 34 - 46 เป็นการทำงานของการนับคนเข้า

บรรทัดที่ 47 – 59 เป็นการทำงานของการนับคนออก

บรรทัดที่ 60 – 66 เป็นเมธดในการนับจำนวนภายในห้อง โดยจะนำค่าของคนเข้าห้องและค่าของของคนออกจากห้อง มาทำการลบกันและจะได้จำนวนคนที่อยู่ในห้องในปัจจุบัน

บรรทัดที่ 68 – 75 เป็นการตั้งค่าพิน ให้กับขา trig และ echo

บรรทัดที่ 77 อ่าน Echo Pin แล้ว pulseIn() จะคืนค่าระยะเวลา (ความยาวของพัลส์) เป็นไมโครวินาที

บรรทัดที่ 78 ทำการคำนวณระยะทาง

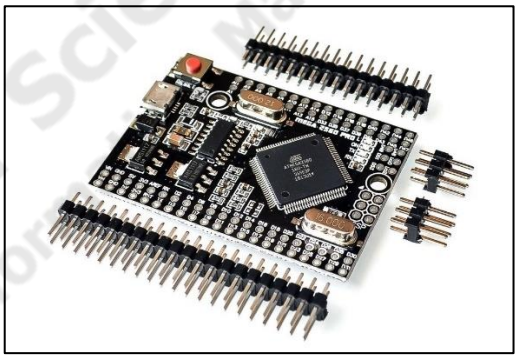
บรรทัดที่ 79 เป็นคำสั่งก่อนอ่านค่าและแสดงผลครั้งถัดไป

บรรทัดที่ 80 ส่งค่าย้อนกลับเพื่อใช้งาน

3.2 ทฤษฎีที่เกี่ยวข้องกับการออกแบบ

3.2.1 ฮาร์ดแวร์ (Hardware)

(1) Mega 2560 pro (Embed) CH340G



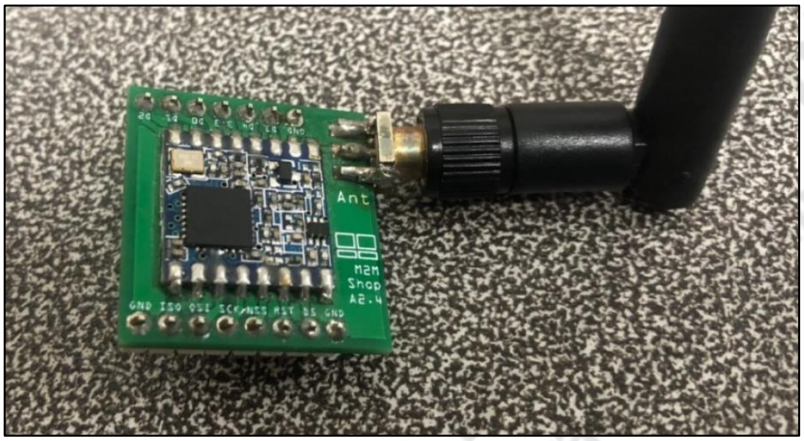
ภาพประกอบที่ 3.19 Mega 2560 pro (Embed) CH340G

บอร์ด Mega Pro Embed CH340G / ATmega2560 [18] ใช้ไมโครคอนโทรลเลอร์ ATmega2560 และ adapter USB-UART CH340 บอร์ดนี้เข้ากันได้กับ Arduino Mega 2560 บอร์ดมีขนาดกะทัดรัด 38X55 mm. และเป็นโซลูชั่นสำหรับการพัฒนาโครงการโดยใช้ ATmega2560 ใช้ชิปดั้งเดิมและมีเรโซเนเตอร์ควอทซ์คุณภาพสูง 16 MHz

- แรงดันใช้งาน 5V
- แรงดันไฟฟ้าขาเข้าที่แนะนำ 7-12V
- แรงดันไฟฟ้าขาเข้าที่จำกัด 6-20V
- ขนาดของหน่วยความจำ 256kb

- ประเภท หรือ ขนาดของ RAM ของข้อมูล 8Kb
- ประเภท หรือ ขนาดของ ROM ของข้อมูล 4Kb
- อุณหภูมิในการทำงาน –40C° / +85C°

(2) SX1276 LoRa module



ภาพประกอบที่ 3.20 SX1276 LoRa module

LoRa module [19] สามารถใช้กับย่านความถี่ 920-925 MHz ได้

- มีการปกป้องที่ดีเยี่ยม
- โปรแกรมที่อัตราบิตสูงถึง 300 kbps
- ความไวสูงสามารถลงไปที่ -148 เดซิเบลมิลลิวัตต์
- กระแสไฟ RX ต่ำที่ 9.9 มิลลิแอมแปร์ และ ความจำรีจีสเตอร์ 200 นาโนแอมแปร์
- ซินธิไซเซอร์แบบครบวงจรที่มีความละเอียด 61 Hz
- การปรับ FSK, GFSK, MSK, GMSK, LoRa และ OOK
- ส่วนนำการตรวจจับ
- ช่วงไดนามิกของ RSSI ที่ 127 เดซิเบล

3.2.2 ซอฟต์แวร์และเครื่องมือ (Software and Tool)

3.2.2.1 การติดตั้ง MongoDB บนเครื่อง

1) ทำการติดตั้ง MongoDB
2) ติดตั้ง MongoDB Compass (เป็นโปรแกรมที่เอาไว้จัดการกับ MongoDB จะมีความคล้ายกับ PHP MyAdmin ของ mySQL)

3.2.2.2 การเชื่อมต่อ MongoDB ด้วย express

คำสั่งในการติดตั้ง express

- npm install express –save

```
const express = require('express')
const app = express()
```

ภาพประกอบที่ 3.21 คำสั่งในการติดตั้ง express

3.2.2.3 การเชื่อมต่อกับ MongoDB โดยใช้ Library mongoose

1) คำสั่งในการติดตั้ง mongoose

- npm i mongoose

2) แล้วใส่ code + connect ตามด้วย host ชื่อ database

```
const mongoose = require('mongoose');
```

ภาพประกอบที่ 3.22 ใช้งาน mongoose ขั้นที่ 1

```
mongoose.connect('mongodb://Roomsystem:admin01@202.28.34.197:27017/Roomsystem',
{useNewUrlParser:true,useUnifiedTopology:true,useCreateIndex:true});
```

ภาพประกอบที่ 3.23 ใช้งาน mongoose ขั้นที่ 2

3.2.2.4 การเชื่อมต่อ database users ของ MongoDB

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema

const productschema = new Schema({
    email :{ type : String,required: true},
    password :{type:String},
    username : {type:String,required: true},
    Image :{type: String}
},
{collection: 'users'}
)

const ProductModel = mongoose.model('testSchema', productschema)
module.exports = ProductModel
```

ภาพประกอบที่ 3.24 การเชื่อมต่อ database users ของ MongoDB

3.2.2.5 การใช้งาน InfluxDB

1. สามารถเรียกใช้งาน Influx ได้โดยไม่ต้องระบุ Part เพื่อเชื่อมต่อ เนื่องจากเครื่องมือจะตรวจหาการติดตั้งในเครื่องของ InfluxDB โดยอัตโนมัติ

```
$ influx
```

ภาพประกอบที่ 3.25 เริ่มต้นใช้งานฐานข้อมูลในการใช้งานเรียกใช้คำสั่งของ Influx

2. InfluxDB ไม่มีฐานข้อมูลเป็นค่าเริ่มต้นดังนั้นจะต้องสร้างฐานข้อมูลขึ้นมาเองการสร้างฐานข้อมูลใน InfluxDB นั้นง่ายและสามารถทำได้โดยใช้คำสั่ง “CREATE DATABASE <DBname>” สำหรับตัวอย่างจะสร้างข้อมูลที่ชื่อว่า room

```
> CREATE DATABASE room
```

ภาพประกอบที่ 3.26 สร้างบานข้อมูลขึ้นมา

3. การที่จะเริ่มแก้ไขฐานข้อมูลใหม่จะต้องบอก CLI ให้ใช้คำสั่ง “USE” เพื่อเริ่มต้นใช้งานด้วยการรันคำสั่งต่อไปนี้

```
> use room
```

ภาพประกอบที่ 3.27 ใช้คำสั่ง ues เพื่อเริ่มต้นใช้งานด้วยการรัน

4. ต่อไปจะเป็นการเพิ่มข้อมูลเข้าไปในฐานข้อมูล สำหรับฐานข้อมูลตัวอย่างนี้จะเก็บค่า แสง ความเคลื่อนไหว อุณหภูมิ

```
> Insert labroom,LDR=1,Motion=1,Humidity=27 value=20,value=20,value=40
```

ภาพประกอบที่ 3.28 เพิ่มข้อมูลเข้าไปในฐานข้อมูล

5. ขณะนี้มีข้อมูลตัวอย่างที่สามารถแสดงวิธีค้นหาข้อมูลโดยใช้ “SELECT” ด้วยคำสั่งต่อไปนี้

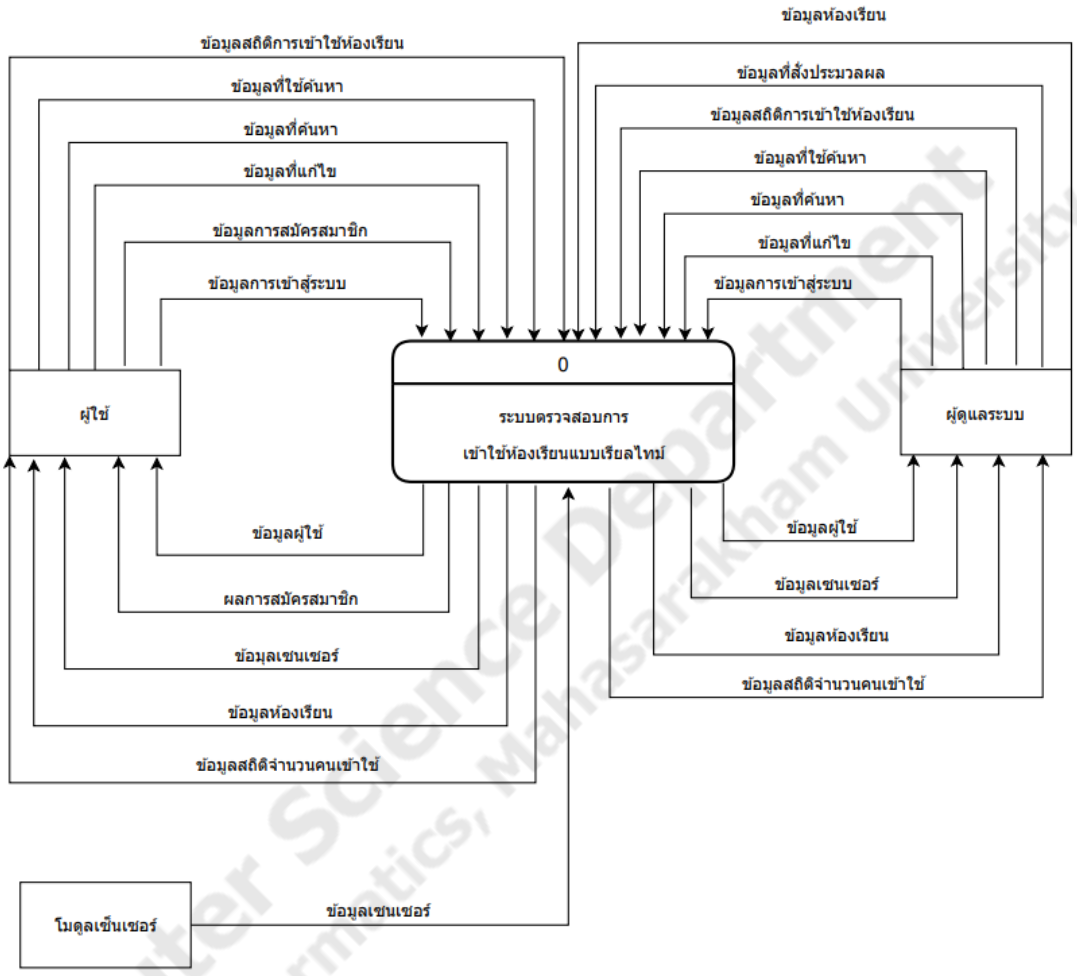
```
> Select * from labroom
```

ภาพประกอบที่ 3.29 แสดงวิธีค้นหาข้อมูลด้วยคำสั่ง Select

```
> select * from labroom
name: labroom
time                Humidity LDR Motion humidity value
----                -------- --- ------ -------- -----
1623834480142564469 27       1   1                40
1623834564646081096          1   0      28        40
1623834587877143851          0   0      25        40
1623834653950313368          1   1      25        40
>
```

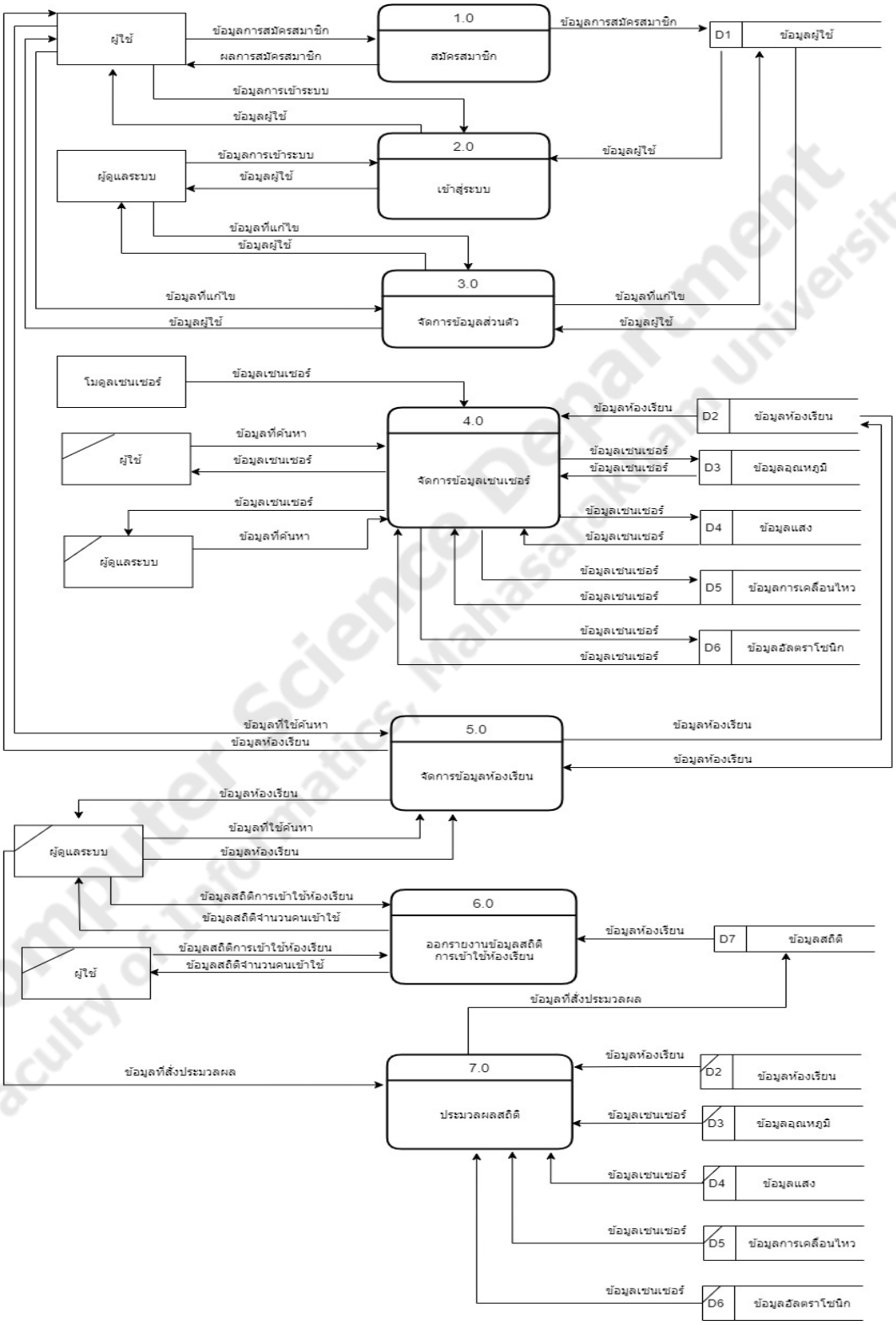
ภาพประกอบที่ 3.30 แสดงค่าที่เพิ่มข้อมูลลงใน database

3.3 แผนภาพบริบท (Context Diagram)

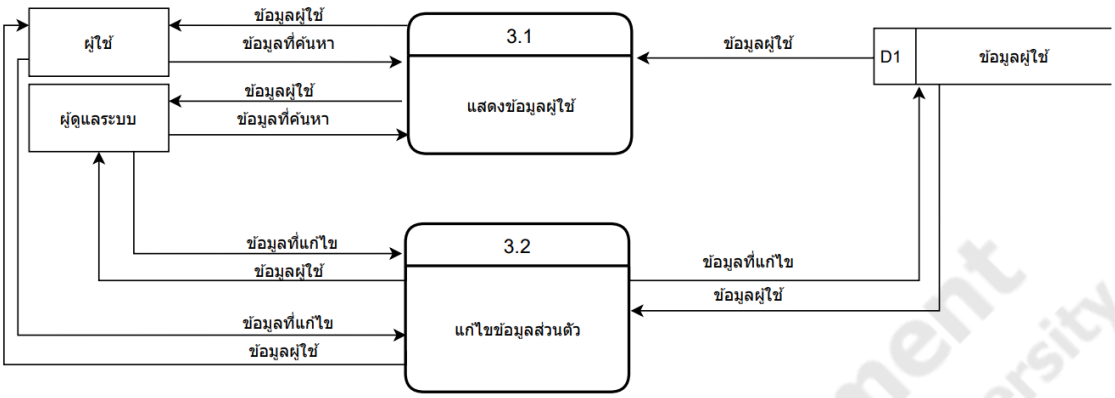


ภาพประกอบที่ 3.31 แผนภาพบริบท (Context Diagram)

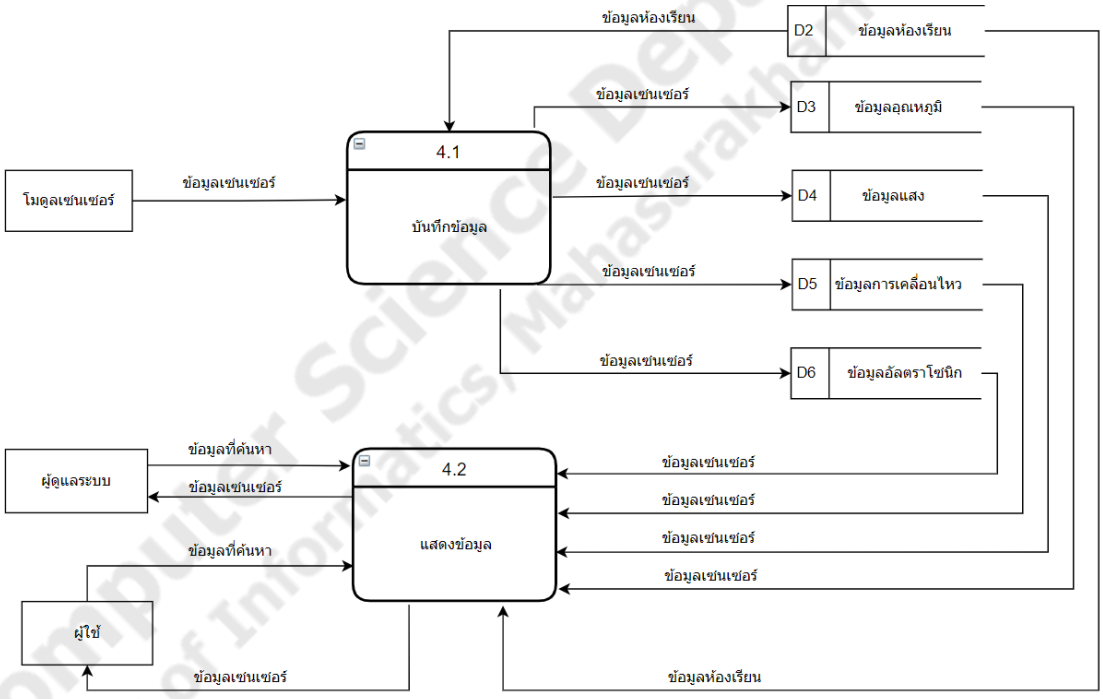
3.4 แผนภาพการไหลของข้อมูล (Data Flow Diagram)



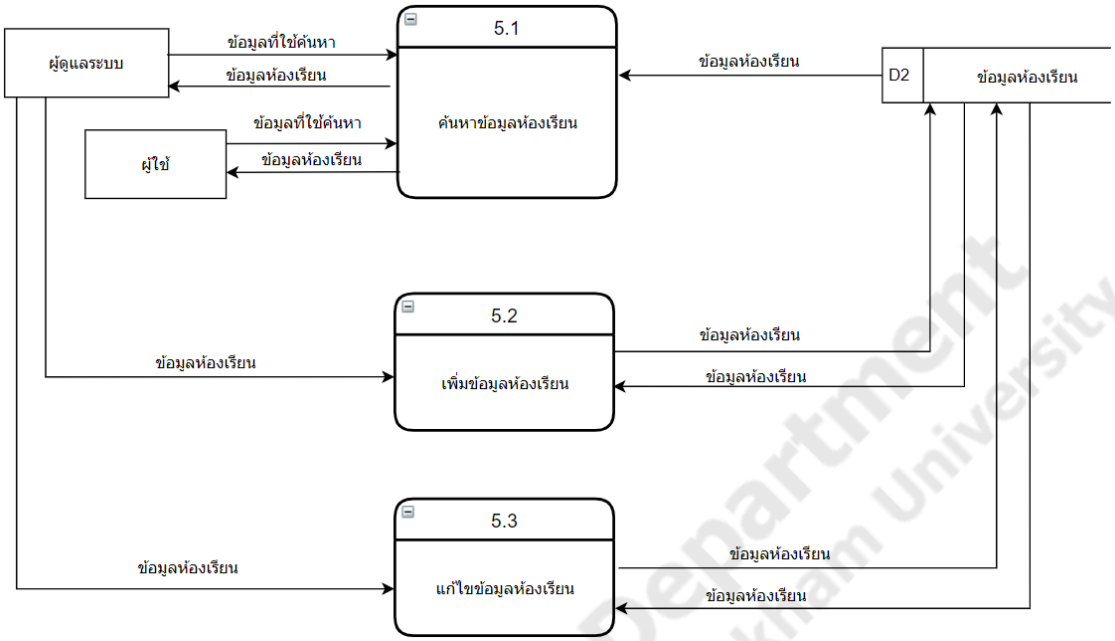
ภาพประกอบที่ 3.32 Data flow diagram level 1



ภาพประกอบที่ 3.33 Data flow diagram level 2 Process 3



ภาพประกอบที่ 3.34 Data flow diagram level 2 Process 4



ภาพประกอบที่ 3.35 Data flow diagram level 2 Process 5

3.5 Data Store

ตารางที่ 3.1 Data Store Description

ID	Data store	Description	Data Structure
D1	ข้อมูลผู้ใช้	เก็บรายละเอียดของผู้ใช้ระบบ	Email+ Password+ ชื่อผู้ใช้+ รูปภาพผู้ใช้
D2	ข้อมูลห้องเรียน	เก็บรายละเอียดของข้อมูลห้องเรียน	ชื่อห้องเรียน
D3	ข้อมูลอุณหภูมิ	เก็บรายละเอียดของข้อมูลอุณหภูมิ	ค่าของเซนเซอร์อุณหภูมิ+ วันเวลาที่ตรวจจับ
D4	ข้อมูลแสง	เก็บรายละเอียดของข้อมูลแสง	ค่าของเซนเซอร์แสง+ วันเวลาที่ตรวจจับ
D5	ข้อมูลการเคลื่อนไหว	เก็บรายละเอียดของข้อมูลการเคลื่อนไหว	ค่าของเซนเซอร์ตรวจจับการเคลื่อนไหว + วันเวลาที่ตรวจจับ
D6	ข้อมูลอัลตราโซนิก	เก็บรายละเอียดของข้อมูลอัลตราโซนิก	ค่าของเซนเซอร์อัลลตราโซนิก + วันเวลาที่ตรวจจับ

ตารางที่ 3.1 Data store Description (ต่อ)

ID	Data store	Description	Data Structure
D7	ข้อมูลสถิติ	เก็บรายละเอียดของข้อมูลสถิติ	ค่าของเซนเซอร์อัลลตราโซนิก + ค่าสถิติ

3.6 External Entity Description

ตารางที่ 3.2 External Entity Description

Name	Descriptions	Input Data Flow	Output Data Flow
ผู้ใช้	ผู้ใช้งานระบบ	-ผลการสมัครสมาชิก -ข้อมูลผู้ใช้ -ข้อมูลเซนเซอร์ -ข้อมูลห้องเรียน -ข้อมูลสถิติจำนวนคนเข้าใช้	-ข้อมูลการสมัครสมาชิก -ข้อมูลการเข้าระบบ -ข้อมูลที่แก้ไข -ข้อมูลที่ค้นหา -ข้อมูลที่ใช้ค้นหา -ข้อมูลสถิติการเข้าใช้ห้องเรียน
ผู้ดูแลระบบ	ผู้ใช้งานประเภทผู้ดูแลระบบ	-ข้อมูลผู้ใช้ -ข้อมูลเซนเซอร์ -ข้อมูลห้องเรียน -ข้อมูลสถิติจำนวนคนเข้าใช้	-ข้อมูลการเข้าระบบ -ข้อมูลที่แก้ไข -ข้อมูที่ค้นหา -ข้อมูลที่ใช้ค้นหา -ข้อมูลสถิติการเข้าใช้ห้องเรียน -ข้อมูลห้องเรียน -ข้อมูลที่สั่งประมวลผล
โมดูลเซนเซอร์	อุปกรณ์การตรวจวัดค่าต่างๆ อย่างตรวจวัดค่าแสง, อุณหภูมิ,การเคลื่อนไหว,อัลตราโซนิก		-ข้อมูลเซนเซอร์

3.7 Data Flow (Data Flow Description and Data Structure of Data Flow)

ตารางที่ 3.3 Data Flow (Data Flow Description and Data Structure of Data Flow)

Name	Description	Source	Destination	Data Structure
ข้อมูลการสมัครสมาชิก	รายละเอียดข้อมูลที่ผู้ใช้ต้องกรอกเพื่อเป็นสมาชิกของระบบ	ผู้ใช้	Process 1.0 สมัครสมาชิก	อีเมล + รหัสผ่าน + ชื่อผู้ใช้ + รูปภาพ
		Process 1.0 สมัครสมาชิก	D1 ข้อมูลผู้ใช้	
ข้อมูลการเข้าระบบ	รายละเอียดการเข้าระบบ	ผู้ใช้/ผู้ดูแลระบบ	Process 2.0 เข้าสู่ระบบ	ชื่อผู้ใช้ + รหัสผ่าน
		D1 ข้อมูลผู้ใช้	Process 2.0 เข้าสู่ระบบ	
ข้อมูลผู้ใช้	รายละเอียดข้อมูลส่วนตัวของผู้ใช้ระบบ	D1 ข้อมูลผู้ใช้	Process 3.1 แสดงข้อมูลผู้ใช้ Process 3.2 แก้ไขข้อมูลส่วนตัว	อีเมล + รหัสผ่าน + ชื่อผู้ใช้ + รูปภาพ
		Process 3.1 แสดงข้อมูลผู้ใช้ Process 3.2 แก้ไขข้อมูลส่วนตัว	ผู้ใช้/ผู้ดูแลระบบ	
ข้อมูลที่แก้ไข	รายละเอียดการแก้ไขข้อมูลส่วนตัวของผู้ใช้ระบบ	ผู้ใช้/ผู้ดูแลระบบ	Process 3.2 แก้ไขข้อมูลส่วนตัว	อีเมล + ชื่อผู้ใช้ + รูปภาพ
		Process 3.2 แก้ไขข้อมูลส่วนตัว	D1 ข้อมูลผู้ใช้	
ข้อมูลเซนเซอร์	รายละเอียดข้อมูลเซนเซอร์	โมดูลเซนเซอร์	Process 4.1 บันทึกข้อมูล	ค่าของเซนเซอร์อุณหภูมิ + ค่าของเซนเซอร์แสง +

ตารางที่ 3.3 Data Flow (Data Flow Description and Data Structure of Data Flow) (ต่อ)

Name	Description	Source	Destination	Data Structure
				ค่าของเซนเซอร์การเคลื่อนไหว + ค่าของเซนเซอร์อัลตราโซนิก + วันเวลาที่ตรวจจับ + ชื่อของเซนเซอร์
		Process 4.1 บันทึกข้อมูล	D3 ข้อมูลอุณหภูมิ D4 ข้อมูลแสง D5 ข้อมูลการเคลื่อนไหว D6 ข้อมูลอัลตราโซนิก	
		D3 ข้อมูลอุณหภูมิ D4 ข้อมูลแสง D5 ข้อมูลการเคลื่อนไหว D6 ข้อมูลอัลตราโซนิก	Process 4.2 แสดงข้อมูล Process 7.0 ประมวลผลสถิติ	
		Process 4.2 แสดงข้อมูล	ผู้ใช้/ผู้ดูแลระบบ	
ข้อมูลที่ค้นหา	รายละเอียดการค้นหาข้อมูลเซนเซอร์ที่ผู้ใช้ระบบต้องการค้นหา	ผู้ใช้	Process 4.2 แสดงข้อมูล	สถานะห้องเรียน + จำนวนคนที่เข้าใช้งาน
		ผู้ดูแลระบบ	Process 4.2 แสดงข้อมูล	เซนเซอร์อุณหภูมิ + เซนเซอร์แสง + เซนเซอร์การเคลื่อนไหว + เซนเซอร์อัลตราโซนิก + สถานะห้องเรียน + จำนวนคนที่เข้าใช้งาน

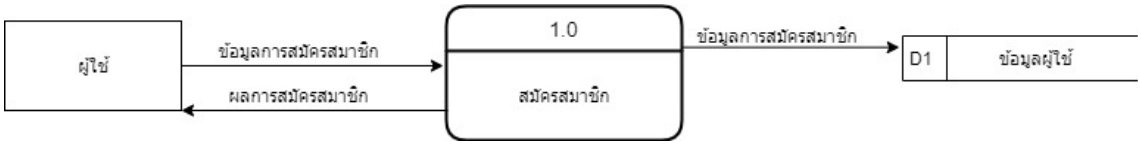
ตารางที่ 3.3 Data Flow (Data Flow Description and Data Structure of Data Flow) (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลที่ใช้ค้นหา	รายละเอียดการค้นหาข้อมูลห้องเรียนที่ผู้ใช้ระบบต้องการค้นหา	ผู้ใช้/ผู้ดูแลระบบ	Process 5.1 ค้นหาข้อมูลห้องเรียน	หมายเลขห้องเรียน
ข้อมูลห้องเรียน	รายละเอียดข้อมูลเซนเซอร์	D2 ข้อมูลห้องเรียน	Process 4.1 บันทึกข้อมูล Process 4.2 แสดงข้อมูล Process 5.1 ค้นหาข้อมูลห้องเรียน Process 5.2 เพิ่มข้อมูลห้องเรียน Process 5.3 แก้ไขข้อมูลห้องเรียน Process 7.0 ประมวลผลสถิติ	รหัสห้องเรียน+หมายเลขห้องเรียน
		Process 5.1 ค้นหาข้อมูลห้องเรียน	ผู้ใช้/ผู้ดูแลระบบ	
		Process 5.2 เพิ่มข้อมูลห้องเรียน	D2 ข้อมูลห้องเรียน	
		Process 5.3 แก้ไขข้อมูลห้องเรียน		

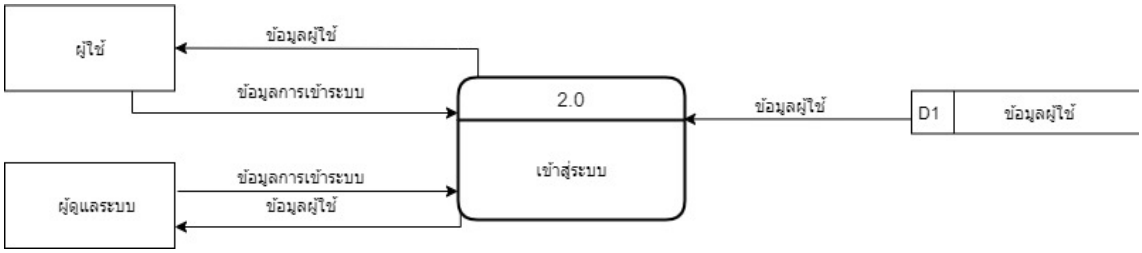
ตารางที่ 3.3 Data Flow (Data Flow Description and Data Structure of Data Flow) (ต่อ)

Name	Description	Source	Destination	Data Structure
		ผู้ดูแลระบบ	Process 5.2 เพิ่มข้อมูลห้องเรียน Process 5.3 แก้ไขข้อมูลห้องเรียน	
		D7 ข้อมูลสถิติ	Process 6.0 ออกรายงานข้อมูลสถิติการเข้าใช้ห้องเรียน	
ข้อมูลสถิติการเข้าใช้ห้องเรียน	รายละเอียดข้อมูลสถิติการเข้าใช้ห้องเรียน	ผู้ใช้/ผู้ดูแลระบบ	Process 6.0 ออกรายงานข้อมูลสถิติการเข้าใช้ห้องเรียน	ชื่อห้องเรียน + ข้อมูลสถิติ
ข้อมูลสถิติจำนวนคนเข้าใช้	รายละเอียดข้อมูลสถิติจำนวนคนเข้าใช้ห้องเรียน	Process 6.0 ออกรายงานข้อมูลสถิติการเข้าใช้ห้องเรียน	ผู้ใช้/ผู้ดูแลระบบ	ชื่อห้องเรียน + ข้อมูลสถิติ
ข้อมูลที่สั่งประมวลผล	ข้อมูลที่สั่งประมวลผลสถิติ	ผู้ดูแลระบบ	Process 7.0 ประมวลผลสถิติ	สถานะการประมวลผล
		Process 7.0 ประมวลผลสถิติ	D7 ข้อมูลสถิติ	

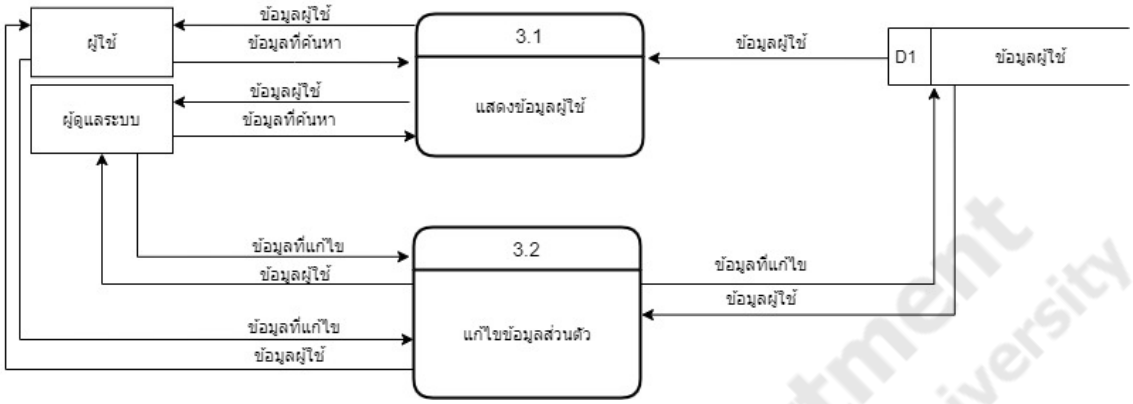
3.8 คำอธิบายการประมวลผล (Process Description)



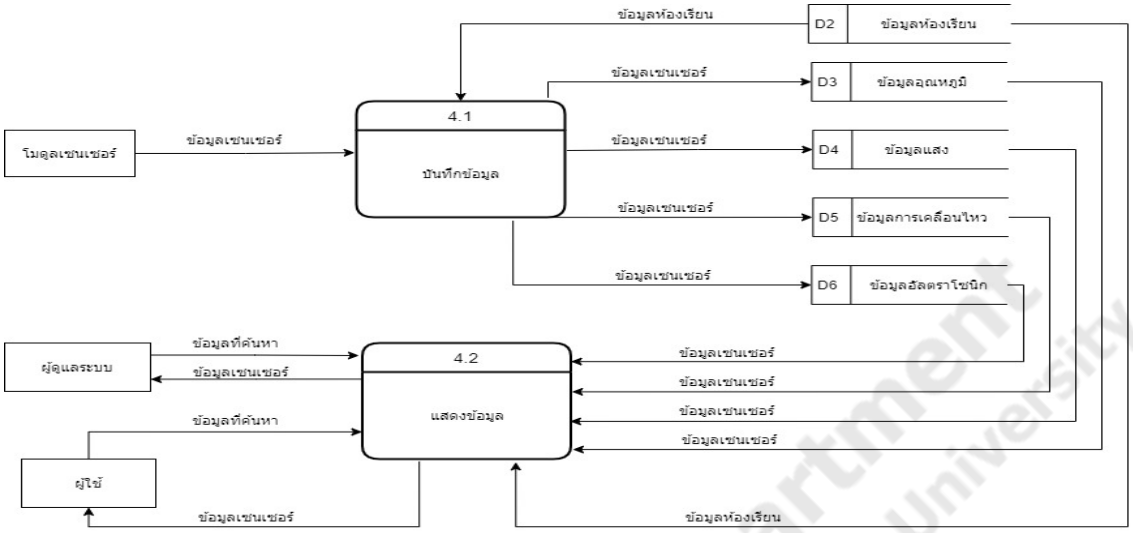
ID	1
Name	สมัครสมาชิก
Description	การสมัครสมาชิกเพื่อขอเข้าใช้ระบบ
Input Data Flow	-ข้อมูลการสมัครสมาชิก
Output Data Flow	-ผลการสมัครสมาชิก
Process Description	เริ่มต้น 1.รับข้อมูลการสมัครสมาชิกจากผู้ใช้ระบบ 2. ตรวจสอบรายละเอียดข้อมูลการสมัครสมาชิก ถ้า (ถูกต้องและครบถ้วน) บันทึกคำขอลงข้อมูลผู้ใช้ ถ้าไม่ (ถูกต้องแลละครบถ้วน) แจ้งเตือนว่าข้อมูลผิดพลาด กรุณาตรวจสอบข้อมูลให้ถูกต้อง จบการทำงาน



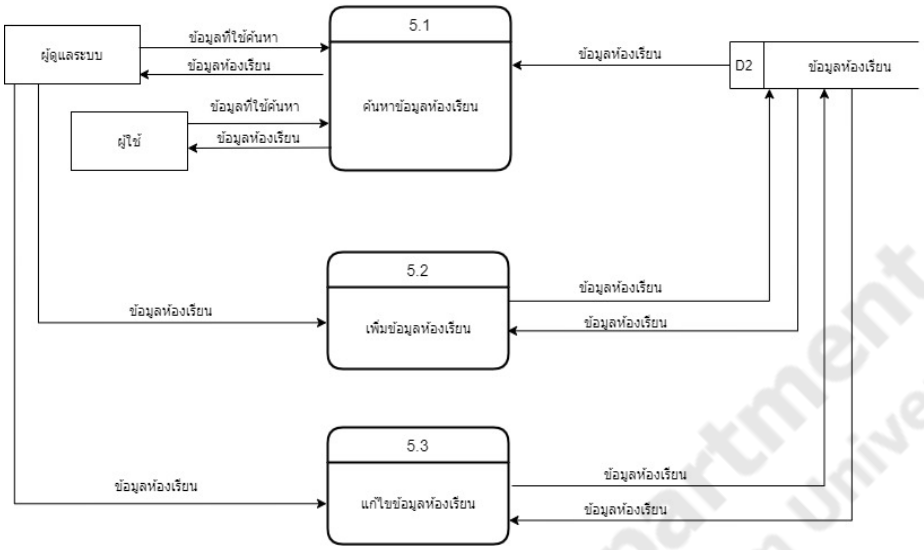
ID	2
Name	เข้าสู่ระบบ
Description	การเข้าสู่ระบบตรวจสอบการเข้าใช้ห้องเรียนแบบเรียลไทม์ของผู้ดูแลระบบและผู้ใช้
Input Data Flow	-ข้อมูลการเข้าระบบ
Output Data Flow	-ข้อมูลผู้ใช้
Process Description	เริ่ม 1. รับข้อมูลการเข้าระบบของผู้ใช้หรือผู้ดูแลระบบ 2. ตรวจสอบข้อมูลการเข้าระบบกับฐานข้อมูลผู้ใช้ ถ้า (ถูกต้องและครบถ้วน) ทำการเข้าสู่ระบบ ถ้าไม่ (ถูกต้องและครบถ้วน) แจ้งเตือนว่าข้อมูลผิดพลาด กรุณาตรวจสอบข้อมูลให้ถูกต้อง จบการทำงาน



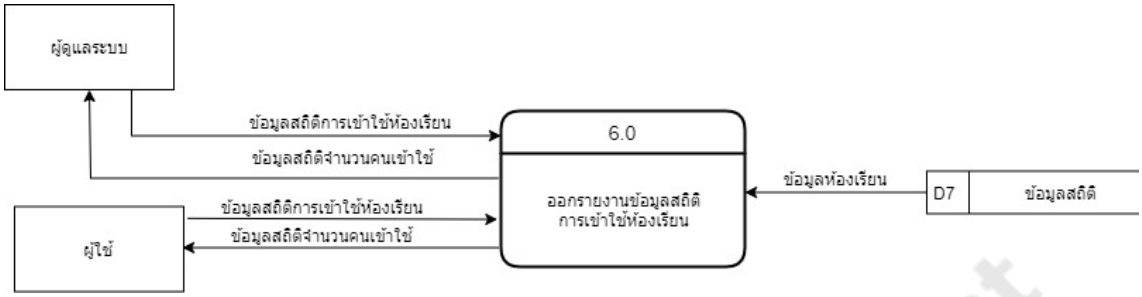
ID	3
Name	จัดการข้อมูลส่วนตัว
Description	การแสดงข้อมูลผู้ใช้ ,การแก้ไขข้อมูลส่วนตัว
Input Data Flow	-ข้อมูลที่ค้นหา -ข้อมูลที่แก้ไข
Output Data Flow	-สถานะการจัดการผู้ใช้ระบบ -ข้อมูลผู้ใช้
Process Description	เริ่ม 1. รับข้อมูลผู้ใช้ระบบ 2. ตรวจสอบรายละเอียดข้อมูลผู้ใช้ระบบ ถ้า (ถูกต้องและครบถ้วน) แสดงสถานะการจัดการผู้ใช้ระบบ บันทึกข้อมูลผู้ใช้ระบบลงในฐานข้อมูลผู้ใช้ ถ้าไม่ (ถูกต้องและครบถ้วน) แจ้งเตือนว่าข้อมูลผิดพลาด ไม่สามารถทำรายการได้ จบการทำงาน



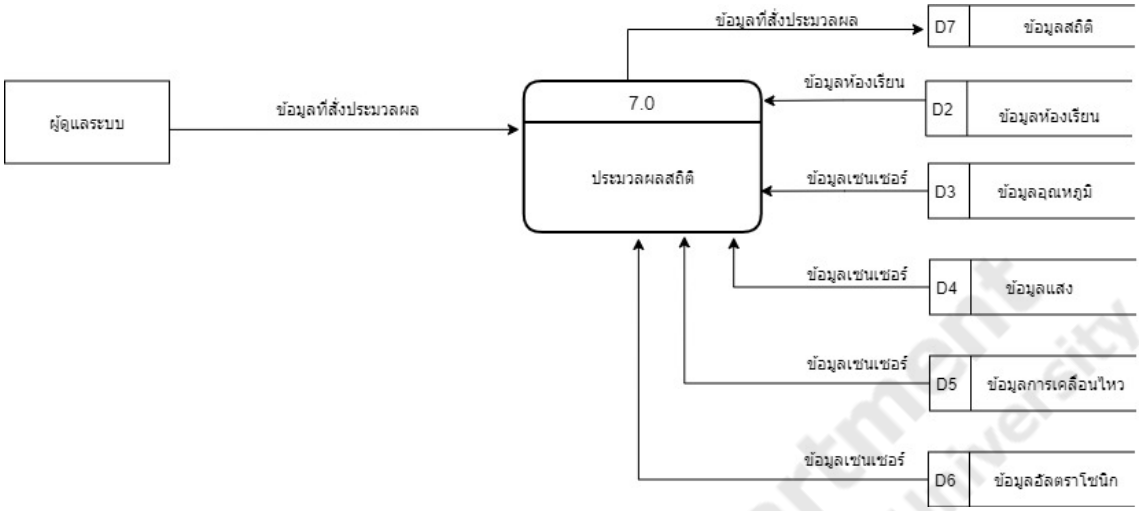
ID	4
Name	จัดการข้อมูลเซนเซอร์
Description	การบันทึกข้อมูล,แสดงข้อมูล
Input Data Flow	-ข้อมูลเซนเซอร์ -ข้อมูลที่ค้นหา
Output Data Flow	-ข้อมูลเซนเซอร์ -ข้อมูลห้องเรียน
Process Description	เริ่ม 1. รับข้อมูลที่ค้นหาที่ต้องการจะค้นหา 2. ตรวจสอบว่ามีข้อมูลครบถ้วนถูกต้องหรือไม่ ถ้า (มีข้อมูล) แสดงข้อมูลห้องเรียน แสดงข้อมูลของเซนเซอร์ของห้องเรียนนั้น ถ้าไม่ (มีข้อมูล) แสดงข้อความเตือน ไม่พบข้อมูลที่ค้นหา จบการทำงาน



ID	5
Name	จัดการข้อมูลห้องเรียน
Description	การค้นหาข้อมูลห้องเรียน,การเพิ่มข้อมูลห้องเรียน,การแก้ไขข้อมูลห้องเรียน
Input Data Flow	-ข้อมูลที่ใช้ค้นหา -ข้อมูลห้องเรียน
Output Data Flow	-ข้อมูลห้องเรียน
Process Description	เริ่ม 1. รับข้อมูลที่ใช้ค้นหาที่ต้องการจะค้นหา 2. ตรวจสอบว่ามีข้อมูลครบถ้วนถูกต้องหรือไม่ ถ้า (มีข้อมูล) แสดงข้อมูลของห้องเรียน ถ้าไม่ (มีข้อมูล) แสดงข้อความเตือน ไม่พบข้อมูลที่ค้นหา 3. รับข้อมูลห้องเรียน 4. ตรวจสอบรายละเอียดข้อมูลห้องเรียน ถ้า (ถูกต้องและครบถ้วน) สถานการณ์จัดการรห้องเรียน ถ้าไม่ (ถูกต้องและครบถ้วน) แจ้งเตือนว่าเกิดข้อผิดพลาด กรุณาทำรายการใหม่ 5. บันทึกการจัดการข้อมูลห้องงเรียน จบการทำงาน



ID	6
Name	ออกรายงานข้อมูลสถิติการเข้าใช้ห้องเรียน
Description	การออกรายงานข้อมูลสถิติการเข้าใช้ห้องเรียน
Input Data Flow	-ข้อมูลสถิติการเข้าใช้ห้องเรียน
Output Data Flow	-ข้อมูลสถิติจำนวนคนเข้าใช้ -ข้อมูลห้องเรียน
Process Description	เริ่ม 1. รับข้อมูลสถิติการเข้าใช้ห้องเรียน 2. ตรวจสอบว่ามีข้อมูลครบถ้วนถูกต้องหรือไม่ ถ้า (มีข้อมูล) แสดงข้อมูลสถิติของห้องเรียน ถ้าไม่ (มีข้อมูล) แสดงข้อความเตือน ไม่พบข้อมูล จบการทำงาน



ID	7
Name	ประมวลผลสถิติ
Description	การประมวลผลสถิติ
Input Data Flow	-ข้อมูลที่สั่งประมวลผล
Output Data Flow	-ข้อมูลสถิติ
Process Description	เริ่ม 1. รับข้อมูลที่สั่งประมวลผล 2. ตรวจสอบว่ามีข้อมูลครบถ้วนถูกต้องหรือไม่ ถ้า (มีข้อมูล) บันทึกลงในฐานข้อมูลสถิติของห้องเรียน แสดงข้อมูลสถิติของห้องเรียน ถ้าไม่ (มีข้อมูล) แสดงข้อความเตือนเกิดข้อผิดพลาด กรุณาทำรายการใหม่ จบการทำงาน

3.9 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.4 ข้อมูลผู้ใช้

Attribute	Type	Description	Example
Email	String	ที่อยู่อีเมล	61011212030@msu.ac.th
Password	String	รหัสผ่านเพื่อเข้าสู่ระบบ	5994471abb01112afcc18...
Username	String	ชื่อที่ใช้งาน	ชนิตา หล่มศักดิ์
Image	String	รูปภาพผู้ใช้เก็บเป็น Base64	4AAQSkZJRgABAAKL21...

```
{
        “Email”: “61011212030@ msu.ac.th”,
        “Password”: “12345”,
        “Username”: “ชนิตา หล่มศักดิ์”,
        “Image”: “petch030.jpg”
}
```

ภาพประกอบที่ 3.36 ตัวอย่างการเก็บข้อมูลผู้ใช้ระบบ

ตารางที่ 3.5 ข้อมูลห้องเรียน

Attribute	Type	Description	Example
Idroom	String	รหัสของห้องเรียน	015
Room_number	String	ชื่อห้องเรียน	IT-505
Status	String	สถานะห้องเรียน	blank
temperature	Number	ค่าอุณหภูมิ	25
motion	Number	ค่าความเคลื่อนไหว	0
luminance	Number	ค่าความสว่าง	0

```
{
        “Idroom”: “015”,
        “Room_number”: “IT-505”,
        “status”: “blank”,
        “temperature”: 0,
        “motion”: 0,
        “luminance”: 0
}
```

ภาพประกอบที่ 3.37 ตัวอย่างการเก็บข้อมูลห้องเรียน

ตารางที่ 3.6 ข้อมูลที่ admin ที่เก็บค่าห้อง

Attribute	Type	Description	Example
Idroom	String	รหัสของห้องเรียน	015
Room_number	String	ชื่อห้องเรียน	IT-505
date	String	เดือน วัน ปี	7/13/2021
time	String	เวลา	9:00-9:30 pm
label	String	แสดงสถานะห้อง	blank

```
{
        “Idroom”: “015”,
        “Room_number”: “IT-505”,
        “date”: “7/13/2021”,
        “time”: “9:00-9:30pm”,
        “label”: “blank”
}
```

ภาพประกอบที่ 3.38 ตัวอย่างการเก็บข้อมูลของ admin

3.10 การนำชุดคำสั่ง (API) มาใช้

3.10.1 การนำชุดคำสั่ง (API) มาใช้ ในฝั่ง front-end

- หน้าสำหรับล็อคอิน จะมีการเรียกใช้ API และประกาศตัวแปรดังนี้

```
1  import { Component, OnInit } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Router } from '@angular/router';
4  import { NgbModal, ModalDismissReasons } from '@ng-bootstrap/ng-bootstrap';
5
6  @Component({
7    selector: 'app-login',
8    templateUrl: './login.component.html',
9    styleUrls: ['./login.component.css']
10 })
11 export class LoginComponent implements OnInit {
12
13   email: any;
14   i_username
15   i_password: any;
16   value
17   id
18   pass
19   closeResult: string;
20   hide = true;
21   mineID
22   url = "./assets/138-1388270_transparent-user-png-icon.png"
23
24   constructor(private http: HttpClient, private router: Router, private modalService: NgbModal) {
```

ภาพประกอบที่ 3.39 Add plugin และ ประกาศตัวแปรหน้าล็อคอิน

บรรทัดที่ 2 import API HttpClient จาก @angular/common/http เพื่อใช้งาน HttpClient

บรรทัดที่ 3 import API Router จาก @augular/router เพื่อใช้งาน Router

บรรทัดที่ 4 import NgbModal กับ ModalDismissReasons จาก @ng-bootstrap/ng-bootstrap
เพื่อใช้งาน NgbModal และ ModalDismissReasons

บรรทัดที่ 13 สร้างตัวแปรสำหรับรับ email ที่เข้ามา

บรรทัดที่ 14 สร้างตัวแปรสำหรับรับ username ที่เข้ามา

บรรทัดที่ 15 สร้างตัวแปรสำหรับรับ password ที่เข้ามา

บรรทัดที่ 16 สร้างตัวแปรสำหรับรับ value ที่เข้ามา

บรรทัดที่ 17 สร้างตัวแปรสำหรับรับ username ที่เข้ามา

บรรทัดที่ 18 สร้างตัวแปรสำหรับรับ password ที่เข้ามา

บรรทัดที่ 19 สร้างตัวแปรสำหรับรับ closeResuit ที่เข้ามา เป็นString

บรรทัดที่ 20 สร้างตัวแปรสำหรับรับ hide ที่เข้ามา เป็นBoolean

บรรทัดที่ 21 สร้างตัวแปรสำหรับรับ ID ของผู้ใช้ที่เข้ามา

บรรทัดที่ 22 สร้างตัวแปรสำหรับรับ url

บรรทัดที่ 24 สร้างตัวแปรสำหรับเรียกใช้ API ภายในเมธอด constructor

- เมธอดสำหรับส่ง username และ password เข้าไปเช็คข้อมูลใน database

```
26    login() {
27
28      let json = { username: this.i_username, password: this.i_password };
29      this.http.post('http://localhost:9000/login', json, { observe: 'response' })
30        .subscribe((response: any) => {
31          console.log(response)
32          if (response) {
33            console.log(response.body)
34            this.value = response.body.data
35            this.mineID = response.body._id
36            sessionStorage.setItem('token', this.value);
37            sessionStorage.getItem('token')
38            sessionStorage.setItem('mineID', this.mineID);
39            console.log(response.status)
40            if (response.status == 200) {
41              console.log('Welcome')
42              this.router.navigateByUrl('/home')
43            } else {
44              console.log('Error')
45            }
```

ภาพประกอบที่ 3.40 เรียกใช้ service login

```
46            } else {
47              console.log('Login fail')
48            }
49          }, error => {
50            console.log('Error!', error)
51          })
52
53        sessionStorage.login = "Login";
54        if (typeof (Storage) !== "undefined") {
55          if (sessionStorage.clickcountLogin) {
56            sessionStorage.clickcountLogin = Number(sessionStorage.clickcountLogin) + 1;
57            console.log("Creating a success session...");
58          }
59          else {
60            sessionStorage.clickcountLogin = 1;
61            console.log("Start creating sessions...");
62          }
63          sessionStorage.getItem("result") + sessionStorage.clickcountLogin;
64        }
65        else {
66          sessionStorage.getItem("result");
67        }
68        console.log('session count : ' + sessionStorage.clickcountLogin);
69      }
```

ภาพประกอบที่ 3.36 เรียกใช้ service login (ต่อ)

บรรทัดที่ 26 สร้างเมธอด login

บรรทัดที่ 28 สร้างตัวแปรมาเก็บค่าข้อมูลที่ได้จากหน้า Interface ใส่ลงใน body เพื่อไปเช็คใน database

บรรทัดที่ 29-30 เรียกใช้ web service login

บรรทัดที่ 31 แสดงค่าข้อมูลผลลัพธ์ออกมา

บรรทัดที่ 32 สร้างเงื่อนไขถ้าเจอข้อมูล

บรรทัดที่ 33 แสดงค่าข้อมูลออกมาเป็น json

บรรทัดที่ 34 นำ response.body.data มาเก็บไว้ที่ตัวแปร value

บรรทัดที่ 35 นำ response.body._id มาเก็บไว้ที่ตัวแปร mineID

บรรทัดที่ 36 ส่งค่า value ไปบน google chrome โดยการสร้าง key ชื่อ token เพื่อเรียกใช้

บรรทัดที่ 37 เรียกใช้ token

บรรทัดที่ 38 ส่งค่า mineID ไปบน google chrome โดยการสร้าง key ชื่อ mineID เพื่อเรียกใช้

บรรทัดที่ 39 แสดงค่า status

บรรทัดที่ 40-44 สร้างเงื่อนไขถ้าหาก status มีค่าเท่ากับ 200

บรรทัดที่ 46-47 ถ้าไม่อยู่ในเงื่อนไขให้แสดงข้อความ “Login fail”

บรรทัดที่ 49-50 ถ้าเข้าเงื่อนไข error ให้แสดงข้อความ “Error!”

บรรทัดที่ 53 สร้าง sessionstorage login ที่มี key ชื่อ login

บรรทัดที่ 54-68 สร้างเงื่อนไขการกำหนดค่า count ในหน้า login

- เมธอดสำหรับการสมัครโดยการใช้ msu account สำหรับส่ง id และ password เข้าไปเช็คข้อมูลใน database

```
70  msu() {
71    let json = { id: this.id, password: this.pass }
72
73    this.http.post('http://202.28.34.197/csapis/authentication/reg', json)
74      .subscribe((response: any) => {
75        console.log(response);
76        if (response) {
77          let json = { email: this.id+"@msu.ac.th", password: this.pass, username: this.id, Image: this.url };
78          this.http.post('http://localhost:9000/register/msu', json)
79            .subscribe((response: any) => {
80              console.log(response);
81              this.http.get('http://localhost:9000/username/' + this.id).subscribe((response: any) => {
82                console.log(response);
83                this.mineID = response[0]._id
84                console.log(this.mineID);
85                sessionStorage.setItem('token', this.value);
86                sessionStorage.getItem('token')
87                sessionStorage.setItem('mineID', this.mineID);
88                this.router.navigateByUrl('/home')
89
90              })
91            }, error => {
92              console.log('Error', error);
93
94            })
95        } else {
96          console.log('No response');
97
98        }
99      }, error => {
100       console.log('Error', error);
```

ภาพประกอบที่ 3.41 เรียกใช้ service authentication reg ผ่าน account msu , service register msu และ service username

บรรทัดที่ 70 สร้างเมธอด msu

บรรทัดที่ 71 สร้างตัวแปรมาเก็บค่าข้อมูลที่ได้จากหน้า Interface ใส่ลงใน body เพื่อไปเช็คใน database

บรรทัดที่ 73-74 เรียกใช้ web service ของ authentication reg

บรรทัดที่ 75 แสดงข้อมูลผลลัพธ์

บรรทัดที่ 76 สร้างเงื่อนไขถ้าพบข้อมูล

บรรทัดที่ 77 สร้างตัวแปรมาเก็บค่าข้อมูลที่ได้จากหน้า Interface ใส่ลงใน body เพื่อไปเพิ่มข้อมูลลงใน database

บรรทัดที่ 78-79 เรียกใช้ web service register msu

บรรทัดที่ 80 แสดงข้อมูลผลลัพธ์

บรรทัดที่ 81 เรียกใช้ web service username โดยระบุ id

บรรทัดที่ 82 แสดงข้อมูลผลลัพธ์

บรรทัดที่ 83 นำ response[0]._id มาเก็บไว้ที่ตัวแปร mineID

บรรทัดที่ 84 แสดงค่า mineID ซึ่งก็คือค่า id ของ user

บรรทัดที่ 85 ส่งค่า value ไปบน google chrome โดยการสร้าง key ชื่อ token เพื่อเรียกใช้

บรรทัดที่ 86 เรียกใช้ token

บรรทัดที่ 87 ส่งค่า mineID ไปบน google chrome โดยการสร้าง key ชื่อ mineID เพื่อเรียกใช้

บรรทัดที่ 88 ลิงค์หน้าไปที่หน้า home

บรรทัดที่ 91-92 ถ้าเข้าเงื่อนไข error ในweb service register msu ให้แสดงข้อความ “Error”

บรรทัดที่ 95-96 ถ้าไม่อยู่ในเงื่อนไข แสดงข้อความ “No response”

บรรทัดที่ 99 -100 ถ้าเข้าเงื่อนไข error ใน web service ของ msu account ให้แสดงข้อความ “Error”

- หน้าสำหรับสมัครสมาชิก จะมีการเรียกใช้ API และประกาศตัวแปรดังนี้

```
1   import { Component, OnInit } from '@angular/core';
2   import { HttpClient } from '@angular/common/http';
3   import { Router } from '@angular/router';
4
5   @Component({
6     selector: 'app-register',
7     templateUrl: './register.component.html',
8     styleUrls: ['./register.component.css']
9   })
10  export class RegisterComponent implements OnInit {
11
12    i_email: any;
13    i_username: any;
14    i_password: any;
15    image: any;
16    urls = "./assets/138-1388270_transparent-user-png-icon.png"
17
18    constructor(private router: Router, private http: HttpClient) {
```

ภาพประกอบที่ 3.42 Add plugin และ ประกาศตัวแปรหน้าสมัครสมาชิก

บรรทัดที่ 2 import API HttpClient จาก @angular/common/http เพื่อใช้งาน HttpClient

บรรทัดที่ 3 import API Router จาก @augular/router เพื่อใช้งาน Router

บรรทัดที่ 12 สร้างตัวแปรสำหรับรับค่า email ที่เข้ามา

บรรทัดที่ 13 สร้างตัวแปรสำหรับรับค่า username ที่เข้ามา

บรรทัดที่ 14 สร้างตัวแปรสำหรับรับค่า password ที่เข้ามา

บรรทัดที่ 15 สร้างตัวแปรสำหรับรับค่า image ที่เข้ามา

บรรทัดที่ 16 สร้างตัวแปรสำหรับรับค่า url ของรูปภาพ

บรรทัดที่ 18 สร้างตัวแปรสำหรับเรียกใช้ API ภายในเมธอด constructor

- เมธอดสำหรับส่งข้อมูลที่รับเข้ามาเก็บภายใน database

```
22  submit() {
23    let json = { email: this.i_email, password: this.i_password, username: this.i_username, Image: this.urls };
24    console.log(JSON.stringify(json));
25    this.http.post('http://localhost:9000/register', json,{ observe: 'response' })
26      .subscribe((response: any) => {
27
28        if (response) {
29          console.log(response);
30          console.log(response.status);
31          if(response.status == 200){
32            console.log('status: OK');
33            this.router.navigateByUrl('/login');
34          }else{
35            console.log('not register');
36
37          }
38        } else {
39          console.log('Status : failed')
40
41        }
42      }, error => {
43        console.log('Error! ',error);
44      });
```

ภาพประกอบที่ 3.43 เรียกใช้ service register

บรรทัดที่ 22 สร้างเมธอด submit

บรรทัดที่ 23 สร้างตัวแปรมาเก็บเพื่อนำข้อมูลที่รับเข้ามาจากหน้า Interface ใส่ลงใน body เพื่อนำไปเพิ่มลงบน database

บนนทัดที่ 24 แสดงค่าออกมาในรูปแบบสตริง json

บรรทัดที่ 25-26 เรียกใช้ web service register ในการสมัครสมาชิก

บรรทัดที่ 28 สร้างเงื่อนไขถ้าพบข้อมูล

บรรทัดที่ 29 แสดงข้อมูล

บรรทัดที่ 30 แสดงข้อมูล status

บรรทัดที่ 31-35 สร้างเงื่อนไขถ้าหาก status มีค่าเท่ากับ 200

บรรทัดที่ 38-39 ถ้าไม่อยู่ในเงื่อนไข แสดงข้อความ “Status : failed”

บรรทัดที่ 42-43 ถ้าเข้าเงื่อนไข error ให้แสดงข้อความ “Error!”

- หน้าสำหรับแก้ไขข้อมูลของผู้ใช้จะมีการเรียกใช้ API และประกาศตัวแปรดังนี้

```
2  import { HttpClient } from '@angular/common/http';
3  import { onMainContentChange } from '../animations/animations';
4  import { SidenavService } from '../services/sidenav.service';
5  import {NgbModal, ModalDismissReasons} from '@ng-bootstrap/ng-bootstrap';
6
7  @Component({
8    selector: 'app-profiles',
9    templateUrl: './profiles.component.html',
10   styleUrls: ['./profiles.component.css'],
11   animations: [onMainContentChange,]
12 })
13 export class ProfilesComponent implements OnInit{
14
15   mineID
16   username
17   img
18   email
19   public onSideNavChange: boolean;
20   closeResult = ''
21   base64: string | ArrayBuffer;
22   filename: string;
23
24   constructor(private http: HttpClient,
25     private _sidenavService: SidenavService,private modalService: NgbModal
```

ภาพประกอบที่ 3.44 Add plugin และ ประกาศตัวแปรหน้าแก้ไขข้อมูลผู้ใช้

บรรทัดที่ 15 สร้างตัวแปรสำหรับรับ id ของ user ที่เข้ามา

บรรทัดที่ 16 สร้างตัวแปรสำหรับรับ username ที่เข้ามา

บรรทัดที่ 17 สร้างตัวแปรสำหรับรับ image ที่เข้ามา

บรรทัดที่ 18 สร้างตัวแปรสำหรับรับ email ที่เข้ามา

บรรทัดที่ 19 สร้างตัวแปรสำหรับเก็บค่า onsidenavchange

บรรทัดที่ 20 สร้างตัวแปรสำหรับเก็บค่า closeResult ที่เป็นสตริง

บรรทัดที่ 21 สร้างตัวแปรสำหรับเก็บค่า base64 ที่เป็นข้อมูลของ image ที่เข้ามา

บรรทัดที่ 22 สร้างตัวแปรสำหรับรับ filename ที่เข้ามา

บรรทัดที่ 24-25 สร้างตัวแปรสำหรับเรียกใช้ API ภายในเมธอด constructor

- หน้าสำหรับแสดงข้อมูลของ user ใน database

```
27      this.mineID = sessionStorage.getItem("mineID")
28
29      this._sidenavService.sideNavState$.subscribe(res => {
30        console.log(res)
31        this.onSideNavChange = res;
32      })
33
34      this.http.get('http://localhost:9000/user/' + this.mineID)
35        .subscribe((res: any) => {
36          if (res) {
37            this.username = res.username
38            this.email = res.email
39            this.img = res.Image
40          }
41        })
```

ภาพประกอบที่ 3.45 เรียกใช้ session id และ service user

บรรทัดที่ 27 เรียกใช้ sessionstorage id ของ user นี้

บรรทัดที่ 29 เรียกใช้ sidenavservice.sidenavstate

บรรทัดที่ 30 แสดงข้อมูลที่เป็น Boolean ออกมา

บรรทัดที่ 31 นำค่า res มาเก็บไว้ที่ตัวแปร onsidenavchange เก็บค่า Boolean

บรรทัดที่ 34-35 เรียกใช้ webservice user

บรรทัดที่ 36 สร้างเงื่อนไขถ้าพบข้อมูล

บรรทัดที่ 37 นำ res.username มาเก็บไว้ที่ตัวแปร username

บรรทัดที่ 38 นำ res.email มาเก็บไว้ที่ตัวแปร email

บรรทัดที่ 39 นำ res.image มาเก็บไว้ที่ตัวแปร img

- เมธอดสำหรับเลือกไฟล์รูปภาพจากแกลลอรี่

```
60    getFile(target: EventTarget) {
61
62      let files = (target as HTMLInputElement).files;
63      if (files != null) {
64        let file = files[0]
65        this.filename = file?.name
66        console.log(this.filename)
67        let reader = new FileReader()
68        reader.readAsDataURL(files[0])
69        reader.onload = (event:any) => {
70          this.img=event.target.result;
71          this.base64 = reader.result
72          let json = {
73            base64: this.base64
74          }
75        }
76        console.log('file ok');
77      } else {
78        console.log('No file');
79      }
80    }
```

ภาพประกอบที่ 3.46 คำสั่งเลือกไฟล์รูปภาพจากแกลลอรี่

บรรทัดที่ 60 สร้างเมธอด getfile

บรรทัดที่ 62 สร้างตัวแปร files ไว้เก็บค่า (target as htmlinputelement).files

บรรทัดที่ 63 สร้างเงื่อนไข ถ้า file ไม่เท่ากับ null

บรรทัดที่ 64 สร้างตัวแปร file มาเก็บค่าของ files[0]

บรรทัดที่ 65 นำ file?.name มาเก็บไว้ใน filename

บรรทัดที่ 66 แสดงข้อมูลของ filename ออกมา

บรรทัดที่ 67 สร้าง object file reader สำหรับอ่านไฟล์ใน input

บรรทัดที่ 68 เป็นคำสั่งอ่านรูปภาพ

บรรทัดที่ 69 onload เป็นคำสั่งให้ทำงานในการโหลดไฟล์

บรรทัดที่ 70 สร้างตัวแปรมาเก็บ event.taget.result

บรรทัดที่ 71 นำ reader.result มาเก็บไว้ที่ตัวแปร base64

บรรทัดที่ 72-73 นำค่า base64 มาเก็บเป็น json

บรรทัดที่ 76 แสดงข้อความ “file ok”

บรรทัดที่ 77-78 หากไม่อยู่ในเงื่อนไข ให้แสดงข้อความ “No file”

- เมธอดสำหรับบันทึกข้อมูลการแก้ไขข้อมูลของ user

```
82  save() {
83    let json = { email: this.email, username: this.username, Image:this.img };
84    this.http.post('http://localhost:9000/user/'+this.mineID, json)
85      .subscribe((response: any) => {
86        console.log(response);
87        if (response) {
88          console.log(this.email);
89          console.log(this.username);
90
91          window.location.reload();
92        } else {
93          console.log('Status : failed')
94        }
95      }, error => {
96        console.log('Error! ',error);
97      });
```

ภาพประกอบที่ 3.47 เรียกใช้ service user โดยระบุ id

บรรทัดที่ 82 สร้างเมธอด save

บรรทัดที่ 83 สร้างตัวแปรมาเก็บเพื่อนำข้อมูลที่รับเข้ามาจากหน้า Interface ใส่ลงใน body เพื่อนำไปอัพเดทบน database

บรรทัดที่ 84-85 เรียกใช้ webservice user โดยระบุ id ของ user

บรรทัดที่ 86 แสดงข้อมูล response ออกมา

บรรทัดที่ 87 สร้างเงื่อนไขถ้าพบข้อมูล

บรรทัดที่ 88 แสดง email

บรรทัดที่ 89 แสดง username

บรรทัดที่ 91 คำสั่ง reload หน้าเว็บ

บรรทัดที่ 92-93 หากไม่อยู่เงื่อนไขให้ทำการแสดงข้อความ “Status : failed”

บรรทัดที่ 95-96 ถ้าเข้าเงื่อนไข error ให้แสดงข้อความ “Error!”

3.10.2 การนำชุดคำสั่ง (API) มาใช้ ในฝั่ง Back-end

- เป็นการสมัครสมาชิกลงใน table users

```
71  app.post('/register',  async (req, res) => {
72      const errors = validationResult(req)
73      const { email, password: plainTextPassword, username, Image } = req.body
74
75      if (!errors.isEmpty()) {
76          return res.send(signupTemplet({ errors }))
77      }
78      if (!email || typeof email !== 'string') {
79          return res.status(401).json({ error: 'Invalid email' })
80      }
81      if (!username || typeof username !== 'string') {
82          return res.status(402).json({ error: 'Invalid username' })
83      }
84
85      if (!plainTextPassword || typeof plainTextPassword !== 'string') {
86          return res.status(403).json({ error: 'Invalid password' })
87      }
88      if (!Image || typeof Image !== 'string') {
89          return res.status(404).json({ error: 'Invalid Image' })
90      }
91

92      if (plainTextPassword.length < 5) {
93          return res.status(201).json({
94              data: 'Password error',
95              error: 'Password too small. Should be atleast 5 character'
96          })
97      }
98      const password = await bcrypt.hash(plainTextPassword, 10)
99      try {
100         await User.create({
101             email,
102             password,
103             username,
104             Image
105
106         })
107     } catch (error) {
108
109         return res.status(400).json()
110     }
111     res.status(200).json({ email, username, Image, message: "Data saved successfully." })
```

ภาพประกอบที่ 3.48 sevice สมัครสมาชิก

บรรทัดที่ 71 เรียกใช้แบบ post() มีชื่อเส้นทาง คือ register

บรรทัดที่ 72 validationResult สำหรับเช็คว่ามี error ในการตรวจสอบหรือไม่

บรรทัดที่ 73 การประกาศตัวแปรไว้เก็บ request ที่ส่งเข้ามาจากหน้าเว็บหรือ tool ที่ใช้ส่งข้อมูลเข้ามา

บรรทัดที่ 75-76 เงื่อนไขคำสั่งเช็คว่ามี error เกิดขึ้นหรือไม่ ก็ให้ส่ง response

บรรทัดที่ 78-79 เงื่อนไขคำสั่งเช็ค email แล้วส่ง response ค่า status error

บรรทัดที่ 81-82 เงื่อนไขคำสั่งเช็ค username แล้วส่ง response ค่า status error

บรรทัดที่ 85-86 เงื่อนไขคำสั่งเช็ค password แล้วส่ง response ค่า status error

บรรทัดที่ 88-89 เงื่อนไขคำสั่งเช็ค image แล้วส่ง response ค่า status error

บรรทัดที่ 92 เงื่อนไขเช็คจำนวน password ที่รับเข้ามา

บรรทัดที่ 93-95 ส่งค่า status “201” และข้อความที่เป็น json

บรรทัดที่ 98 ทำการ hash password 10 รอบ

บรรทัดที่ 99-104 สร้างข้อมูลลง database

บรรทัดที่ 107 เงื่อนไข error

บรรทัดที่ 109 ส่งค่า response status “400”

บรรทัดที่ 111 ส่งค่า response status “200” และข้อมูล json

- เป็นการเข้าสู่ระบบโดย กรอก email และ password

```
46  app.post('/login', async (req, res,) => {
47      const { username, password } = req.body
48      const user = await User.findOne({ username }).lean()
49      if (!user) {
50          return res.status(400).json({ status: 'error',
51          error: 'Invaild username/password' })
52      }
53      bcrypt.compare(password, user.password).then((result) => {
54          // ทำแบบอนาคต
55          if (result) {
56              const token = jwt.sign({
57                  id: user._id,
58                  username: user.username
59              }, JWT_SECRET)
60
61              return res.status(200).json({ _id: user._id, data: token })
62
63          } else {
64              return res.status(400).json({ error: 'Invaild username/password' })
65          }
66      });
67  })
```

ภาพประกอบที่ 3.49 sevice เข้าสู่ระบบด้วย email และ password

บรรทัดที่ 46 เรียกใช้แบบ post() มีชื่อเส้นทาง คือ login

บรรทัดที่ 47 การประกาศตัวแปรไว้เก็บ request ที่ส่งเข้ามาจากหน้าเว็บหรือ tool ที่ใช้ส่งข้อมูลเข้ามา

บรรทัดที่ 48 ประกาศตัวแปร username ที่อยู่ใน database มาเก็บไว้

บรรทัดที่ 49-51 เงื่อนไข user แล้วส่งข้อมูลค่า response status “400” และข้อมูล json

บรรทัดที่ 53 คำสั่งเปรียบเทียบรหัสผ่านที่ hash ไว้และ รหัสที่รับเข้ามา แล้วส่งค่า result ออกมา

บรรทัดที่ 55 เงื่อนไข ถ้าพบ result

บรรทัดที่ 56-58 สร้างตัวแปร token มาเก็บค่า jwt ที่มีข้อมูล id และ username ของผู้ใช้

บรรทัดที่ 59 key ของ jwt

บรรทัดที่ 61 ทำการส่งค่า response status “200” และข้อมูลที่เป็น json

บรรทัดที่ 63-64 ทำการส่งค่า response status “400” และข้อมูล error ที่เป็น json

- เป็นการ select ข้อมูล users ออกมา

```
134   app.get('/userall', async (req, res) => {
135       var uses = await User.find({});
136       res.json(uses)
137   })
```

ภาพประกอบที่ 3.50 service select user

บรรทัดที่ 134 เรียกใช้แบบ get() มีชื่อเส้นทาง คือ userall

บรรทัดที่ 135 เป็นคำสั่งแสดงค่าข้อมูลของuser ออกมาทั้งหมด

บรรทัดที่ 136 ส่งข้อมูลออกมาเป็น json

- เป็นการ select ข้อมูล email ของ users ออกมา

```
153   app.get('/useremail/:email', async (req, res) => {
154       try {
155           var person = await User.find({ email: req.params.email}).exec()
156           console.log(req.params.email);
157           res.send(person)
158       } catch (error) {
159           res.status(500).send(error)
160       }
161   })
```

ภาพประกอบที่ 3.51 service select user โดยระบุ email

บรรทัดที่ 153 เรียกใช้แบบ get() มีชื่อเส้นทาง คือ useremail และมีตัวแปล คือ email (email คือ อีเมล์ของผู้ใช้)

บรรทัดที่ 154-157 เป็นคำสั่งแสดงค่าข้อมูลโดยระบุ email ทำการแสดง email และส่งข้อมูลออกมา

บรรทัดที่ 158-159 เงื่อนไข error ส่ง status “500”

- เป็นการ select ข้อมูล username ของ users ออกมา

```
163  app.get('/username/:username', async (req, res) => {
164      try {
165          var person = await User.find({ username:  req.params.username}).exec()
166          res.send(person)
167      } catch (error) {
168          res.status(500).send(error)
169      }
170  })
```

ภาพประกอบที่ 3.52 service select username โดยระบุ username

บรรทัดที่ 163 เรียกใช้แบบ get() มีชื่อเส้นทาง คือ username และมีตัวแปล คือ username (username คือชื่อผู้ใช้)

บรรทัดที่ 164-166 เป็นคำสั่งแสดงค่าข้อมูลโดยระบุ username และส่งข้อมูลออกมา

บรรทัดที่ 167-168 เงื่อนไข error ส่ง status “500”

- เป็นการ select ข้อมูล id ของ users ออกมา

```
183  app.get('/user/:_id', async (req, res) => {
184      try {
185          var person = await User.findById(req.params._id).exec()
186          res.send(person)
187      } catch (error) {
188          res.status(500).send(error)
189      }
190  })
```

ภาพประกอบที่ 3.53 service select user โดยระบุ id

บรรทัดที่ 183 เรียกใช้แบบ get() มีชื่อเส้นทาง คือ user และมีตัวแปล คือ _id (_id คือรหัสของผู้ใช้)

บรรทัดที่ 184-186 เป็นคำสั่งแสดงค่าข้อมูลโดยระบุ id และส่งข้อมูลออกมา

บรรทัดที่ 187-188 เงื่อนไข error ส่ง status “500”

- เป็นการ update ข้อมูลของ users

```
192  app.post('/user/:_id', function (req, res) {
193
194      User.findByIdAndUpdate(
195          req.params._id,
196          {
197              email: req.body.email,
198              username: req.body.username,
199              Image: req.body.Image
200          },
201          {
202              new: true
203          },
204          (err, data) => {
205              if (err) {
206                  res.json({
207                      success: false,
208                      message: err
209                  })
210              } else if (!data) {
211                  res.json({
212                      success: false,
213                      message: "Not Found"
214                  })
215              } else {
216                  res.json({
217                      success: true,
218                      data: data
219                  })
220              }
221          }
222      )
223  })
```

ภาพประกอบที่ 3.54 service update

บรรทัดที่ 192 เรียกใช้แบบ post() มีชื่อเส้นทาง คือ user และมีตัวแปล คือ _id (_id คือรหัสของผู้ใช้)

บรรทัดที่ 194 ทำหน้าที่สร้าง user ใหม่

บรรทัดที่ 195 id ของผู้ใช้ที่รับเข้ามาจากหน้าเว็บหรือ tool ที่ใช้ส่งข้อมูลเข้ามา

บรรทัด 197-199 ใช้ข้อมูลที่อยู่ใน request body

บรรทัดที่ 202 ข้อมูลที่ใส่เข้ามา true

บรรทัดที่ 204 เป็นฟังก์ชันที่คอยรับข้อมูลหลักจากที่สั่งให้ฟังก์ชันนั้น ทำงานเสร็จแล้ว
บรรทัดที่ 205 เงื่อนไข error
บรรทัดที่ 206-208 ทำการส่งค่า response ข้อมูล json
บรรทัดที่ 210 เงื่อนไข data
บรรทัดที่ 211-213 ทำการส่งค่า response ข้อมูล json
บรรทัดที่ 215-218 ถ้าไม่อยู่เงื่อนไข ทำการส่ง response ข้อมูล json

3.10.3 การนำชุดคำสั่ง (API) ที่ใช้ใน Mega 2560 pro

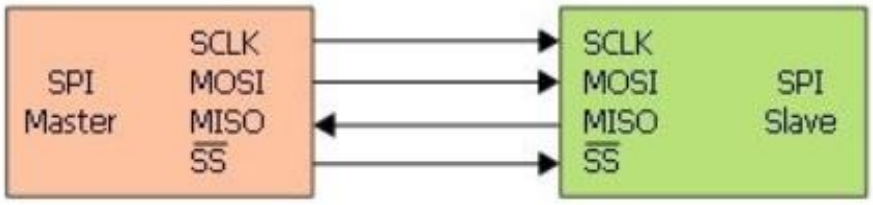
API ที่ใช้ในการควบคุม LoRa Node ส่งข้อมูลไปเก็บยัง database ได้โดยผ่าน Gateway คือ LMIC-Arduino-AS923-upper-master ที่สามารถแก้ไขโค้ดให้ส่งได้หลายคลื่นความถี่ ทำให้สะดวกในการใช้งาน

```
1 #include <lmic.h>
2 #include <hal/hal.h>
3 #include <SPI.h>
```

ภาพประกอบที่ 3.55 การเรียกใช้ Library ของ LMIC-Arduino-AS923-upper-master

บรรทัดที่ 1 เรียกไฟล์ limc.h ใน Libraries ของ LMIC-Arduino-AS923-upper-master
บรรทัดที่ 2 เรียกไฟล์ hal และไฟล์ hal ใน Libraries ของ LMIC-Arduino-AS923-upper-master
บรรทัดที่ 3 SPI หรือ Serial Peripheral Interface เป็นวิธีการสื่อสารแบบซิงโครนัสรูปแบบหนึ่ง โดย SPI ทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็น MASTER ในขณะที่อีกตัวหนึ่งทำหน้าที่เป็น SLAVE และส่งข้อมูลในโหมด Full-duplex นั่นหมายความว่า สัญญาณสามารถส่งหากันได้ระหว่าง MASTER และ SLAVE ได้อย่างต่อเนื่อง ในการสื่อสารแบบ SPI นี้ ไม่ได้มาตรฐานกำหนดตายตัว ว่าข้อมูลที่ส่งหากันต้องอยู่ในรูปแบบหรือ format แบบไหน เป็นการคิด protocol การสื่อสารกันเอาเอง อุปกรณ์ที่ยังคงมีการใช้การสื่อสารแบบ SPI ได้ดังนี้

- การแปลงข้อมูลจาก Analog to Digital หรือจาก Digital to Analog
- การติดต่อกับหน่วยความจำ EEPROM และ FLASH
- โมดูลนาฬิกาดิจิตอล หรือ Real Time Clock : RTC
- เซ็นเซอร์ จำพวก Temperature sensor , Pressure sensor อื่น ๆ เช่น signal mixer , Potentiometer , LCD controller , USART , CAN controller , USB controller , Amplifier



ภาพประกอบที่ 3.56 การเชื่อต่อการสื่อสารแบบ SIP ระหว่าง Master – Slave

การเชื่อมต่อการสื่อสารแบบ SIP ระหว่างอุปกรณ์ Master – Slave โดยมีสายสัญญาณ 4 เส้น ประกอบด้วย

- SCK (Serial Clock) ใช้ส่งสัญญาณนาฬิกาจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave เพื่อกำหนดจังหวะการรับส่งข้อมูล
- MOSI (Master Out Slave In) ใช้ส่งข้อมูลจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave
- MISO (Master In Slave Out) ใช้รับข้อมูลจากอุปกรณ์ Slave
- SS/CS (Slave Select/Chip Select) ใช่ส่งสัญญาณ Low ไปยังอุปกรณ์ Slave ที่ต้องการรับส่งข้อมูล

```
8  static const PROGMEM u1_t NWKSKEY[16] = { [redacted]
9                                           0x52, 0x2c, 0x7b, 0xe4, 0xbb, 0x46, 0x28, 0x44, 0x1e };
10 static const u1_t PROGMEM APPSKEY[16] = { 0x[redacted]
11                                          0x29, 0x39, 0xbb, 0x6c, 0xa4, 0x7d, 0xd6, 0xb9, 0xae };
12 static const u4_t DEVADDR = 0[redacted] ;
```

ภาพประกอบที่ 3.57 การกำหนด Key ให้กับ End Device

บรรทัดที่ 8 – 9 กำหนด Network Session Key ที่ได้จากเว็บ the thingsnetwork ให้กับ LoRa Networktracker เพื่อส่งข้อมูลไปแสดงใน the thing network

บรรทัดที่ 10-11 กำหนด App Session Key ที่ได้จากเว็บ the thing network ให้กับ LoRa Networktracker เพื่อส่งข้อมูลไปแสดงใน the thing network

บรรทัดที่ 12 กำหนด Device Address ที่ได้จากเว็บ the thing network ให้กับ LoRa Networktracker เพื่อส่งข้อมูลไปแสดงใน the thingsnetwork

```
16 const unsigned TX_INTERVAL = 60;
17
18 // Pin mapping
19 const lmic_pinmap lmic_pins = {
20   .nss = 53,
21   .rxtx = LMIC_UNUSED_PIN,
22   .rst = LMIC_UNUSED_PIN,
23   .dio = {4, 5, 7 },
24 };
```

ภาพประกอบที่ 3.58 การกำหนด เวลาในการส่งข้อมูลและกำหนด pin ให้กับ LoRa

บรรทัดที่ 16 กำหนดเวลาในการส่งข้อมูลจาก LoRa Network Tracker ไปยัง Gateway มีหน่วยเป็นวินาที

บรรทัดที่ 28 – 33 เซ็ตพินให้กับขาของ Mega 2560 Pro ให้ตรงกับที่ต่อเอาไว้ในอุปกรณ์

```
30 int LDRPin = A0;
31 int valueLDR = 0;
```

ภาพประกอบที่ 3.59 คำสั่งเซนเซอร์วัดความสว่างของแสงกำหนดตัวแปร LoRa

บรรทัดที่ 30 – 31 การประกาศคำสั่งของเซนเซอร์วัดความสว่างของแสง

```
35 int MotionPin = 8;
36 int valueMotion = 0;
```

ภาพประกอบที่ 3.60 คำสั่งเซนเซอร์ตรวจจับความเคลื่อนไหวกำหนด LoRa

บรรทัดที่ 35 – 36 การประกาศคำสั่งของเซนเซอร์ตรวจจับความเคลื่อนไหว

```
40 #include "DHT.h"
41 DHT dht;
42 #define DHTPIN 2
43 float valuetemp;
```

ภาพประกอบที่ 3.61 คำสั่งเซนเซอร์วัดอุณหภูมิและความชื้นกำหนดตัวแปร LoRa

บรรทัดที่ 40 การเรียกใช้ Library วัดอุภณหมิูและความชื้น DHT11

บรรทัดที่ 41 การประกาศตัวแปรเพื่อใช้งาน

บรรทัดที่ 42 เชื่อมต่อกับเซ็นเซอร์ Analog Pin

บรรทัดที่ 43 การประกาศคำสั่งเซนเซอร์วัดอุณหภูมิและความชื้นกำหนดตัวแปร

```
112 void do_send(osjob_t* j) {
113   // Check if there is not a current TX/RX job running
114   if (LMIC.opmode & OP_TXRXPEND) {
115     Serial.println(F("OP_TXRXPEND, not sending"));
116   }
117   else {
118 
119     valueLDR = digitalRead(LDRPin);
120     valueMotion = digitalRead(MotionPin);
121 
122     delay(dht.getMinimumSamplingPeriod());
123     valuetemp = dht.getTemperature();
124 
125     payload = "{\"Luminance\":" + String(valueLDR) +
126               ",\" Motion\":" + String(valueMotion) +
127               ",\" Temperature(C)\":" + String(valuetemp, 2) +
128               "}}";
129 
130     byte payloadByte[payload.length()];
131     payload.toCharArray(payloadByte, payload.length());
132     Serial.println(payload);
133     LMIC_setTxData2(1, payloadByte, sizeof(payloadByte) - 1, 0);
134   }
135 }
```

ภาพประกอบที่ 3.62 เมธอดส่งข้อมูลไปยัง Gateway

บรรทัดที่ 112 เมธอดที่ใช้งานในการส่งข้อมูลจาก LoRa Network tracker ไปยัง Gateway

บรรทัดที่ 115 เช็คการทำงานของ pin RX และ TX หากมีการทำงานจะไม่สามารถส่งข้อมูลได้

บรรทัดที่ 119 อ่านค่าสัญญาณ Digital ขา A0 ที่ต่อกับ LDR Photoresistor Sensor Module

บรรทัดที่ 120 อ่านค่าสัญญาณ Digital ขา D8 ที่ต่อกับ PIR Motion Sensor (HC-SR501)

บรรทัดที่ 123 เป็นคำสั่งดึงค่าความชื้นในอากาศเป็นเปอร์เซ็นต์

บรรทัดที่ 125 – 128 เป็นการนำข้อความที่รับเซนเซอร์ มาต่อกันให้อยู่ในตัวแปรที่มีชื่อ payload

บรรทัดที่ 130 กำหนดตัวแปรชนิด byte ชื่อ payloadByte ให้มีขนาดเท่ากับข้อความในตัวแปร payload ที่เก็บ ข้อมูลที่ได้รับจากเซนเซอร์ทุกตัว

บรรทัดที่ 133 ส่งข้อมูลจากตัวแปร payloadByte และหาขนาดของข้อความ เพื่อส่งไปที่ Gateway

```
148 void setup() {
149   Serial.begin(9600);
150   Serial.println(F("Starting"));
151 
152   /*------------------------- LDR(แสง) ----------------------------------------------------*/
153   pinMode(LDRPin, INPUT);
154   /*---------------------------------------------------------------------------------------*/
155 
156   /*------------------------- Motion(เคลื่อนไหว) ------------------------------------------*/
157   pinMode(MotionPin, INPUT);
158   /*---------------------------------------------------------------------------------------*/
159 
160   /*------------------------- DHT11(อุณหภูมิ) ---------------------------------------------*/
161   dht.setup(DHTPIN);
162   /*---------------------------------------------------------------------------------------*/
163 #ifdef VCC_ENABLE
164   // For Pinoccio Scout boards
165   pinMode(VCC_ENABLE, OUTPUT);
166   digitalWrite(VCC_ENABLE, HIGH);
167   delay(1000);
168 #endif
```

ภาพประกอบที่ 3.63 เมธอด setup

บรรทัดที่ 148 เมธอด setup คือ คือเมธอดที่ใช่ตั้งค่าเริ่มต้นให้กับ Arduino

บรรทัดที่ 165 กำหนดให้ขา vcc หรือขา 5V เป็น Output หรือ การกำหนดให้ไฟออกนั่นเอง

บรรทัดที่ 166 สั่งให้ขา vcc หรือขา 5V จ่ายไฟออก

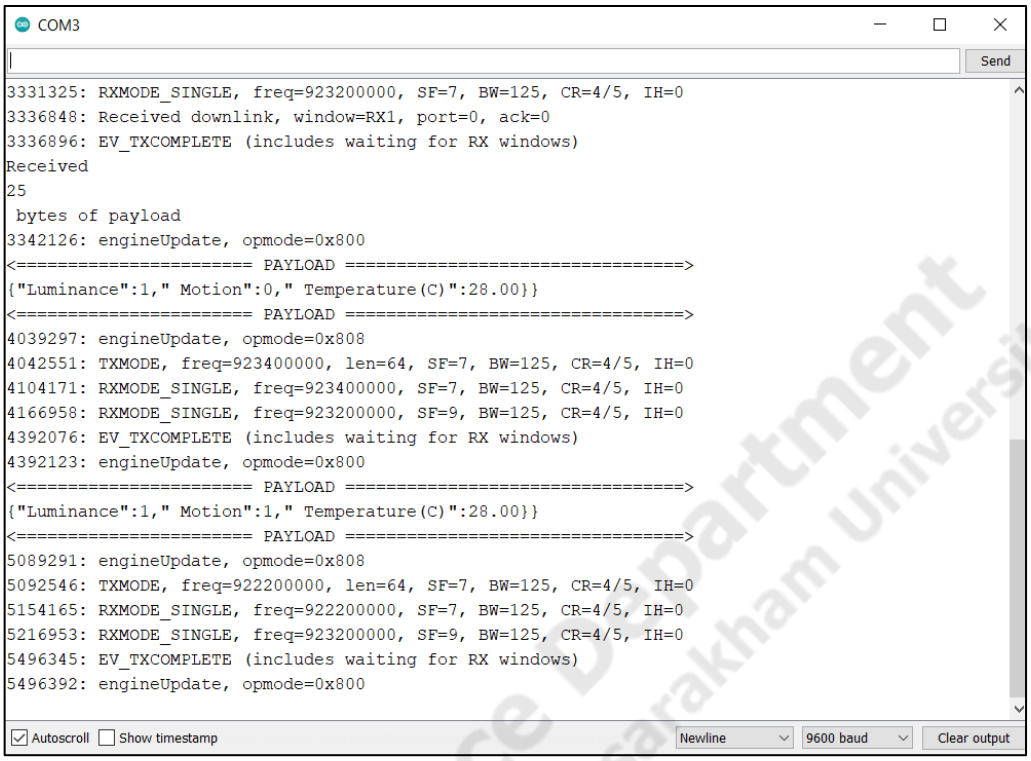
```
181 #if defined(CFG_eu868)
182     LMIC_setupChannel(0, 923200000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);
183 
184 #endif
185   LMIC_setLinkCheckMode(0);
186 
187   LMIC.dn2Dr = DR_SF9;
188   LMIC_setDrTxpow(DR_SF7, 14);
189   do_send(&sendjob);
190 }
```

ภาพประกอบที่ 3.64 เมธอด loop

บรรทัดที่ 181 เช็คว่าเป็นบอร์ดที่ต้องการหรือไม่ ถ้าใช่ให้ทำการส่งข้อมูลออกไปที่ความถี่ หรือ Frequency 923.2 MHz และ Spreading Factor เท่ากับ 7 ในโปรเจคนี้คือ CFG_eu868

บรรทัดที่ 188 กำหนดให้ Spreading Factor เท่ากับ 7

บรรทัดที่ 189 ส่งข้อความออกไปที่ Gateway



ภาพประกอบที่ 3.65 ตัวอย่างข้อมูลจาด โปรแกรมควบคุม End Device

3.11 อัลกอริทึมที่ใช้ในการประมวลผล Decision tree

Decision tree learning [20] หรือ การเรียนรู้ต้นไม้ตัดสินใจนั้นใน machine learning เป็นโมเดลทางคณิตศาสตร์ที่จะใช้ทำนายประเภทของวัตถุ โดยการพิจารณาจากลักษณะของวัตถุและใช้ต้นไม้ตัดสินใจในการสร้างโมเดลที่พยากรณ์ได้ ซึ่งจะเชื่อมโยงข้อมูลสังเกตการณ์เข้ากับข้อมูลปลายทาง การทำงานของต้นไม้การตัดสินใจจะทำการจัดกลุ่ม (classify) ชุดข้อมูลนำเข้าในแต่ละกรณี (Instance) แต่ละบัพ (node) ของต้นไม้การตัดสินใจคือตัวแปร (attribute) ต่างๆของชุดข้อมูล

ต้นไม้ตัดสินใจเป็นอัลกอริทึมที่ถูกนำมาใช้ในหลายๆด้าน ในด้านของธุรกิจก็นำมาช่วยในการตัดสินใจเพื่อช่วยในการสร้างแผนงาน และเป็นพื้นฐานหนึ่งในการทำเหมืองข้อมูล และยังสามารถนำมาใช้ในการทำนายหุ้นได้อีกด้วย decision tree learning สามารถนำมาใช้ได้ในหลายๆ ด้าน และก็เป็นอีกหนึ่งอัลกอริทึมที่ได้รับความนิยม

ตารางที่ 3.7 ตัวอย่างข้อมูลในภายห้อง

ครั้ง	แสง	เคลื่อนไหว	อุณหภูมิ	สถานะ
1	มืด	ไม่มีความเคลื่อนไหว	ร้อน	ว่าง
2	แสงน้อย	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
3	แสงมาก	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
4	แสงจ้า	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
5	แสงมาก	มีความเคลื่อนไหว	ร้อน	ว่าง
6	แสงปานกลาง	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
7	แสงปานกลาง	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
8	แสงน้อย	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง
9	มืด	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง
10	มืด	ไม่มีความเคลื่อนไหว	ร้อน	ว่าง
11	แสงมาก	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
12	แสงจ้า	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
13	มืด	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง
14	แสงปานกลาง	ไม่มีความเคลื่อนไหว	ร้อน	ว่าง
15	แสงน้อย	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง

จากข้อมูลใน ตารางที่ 3.7 ประกอบไปด้วย 3 แอตทริบิวต์ คือ

- **แสง** : ประกอบด้วย 5 ค่า คือ มืด , แสงน้อย , แสงปานกลาง , แสงมาก , แสงจ้า
- **เคลื่อนไหว** : ประกอบด้วย 2 ค่า คือ มีความเคลื่อนไหว , ไม่มีความเคลื่อนไหว
- **อุณหภูมิ** : ประกอบด้วย 5 ค่า คือ เย็น , ปานกลาง , อบอุ่น , ร้อน , ร้อนมาก

และ 1 คลาส

- **สถานะ** : ประกอบด้วย 2 สถานะ คือ ว่าง , ไม่ว่าง

ตารางที่ 3.8 ตารางความถี่

สถานะ	
ว่าง	ไม่ว่าง
8	7

จาก ตารางที่ 3.8 การแยกความถี่ของสถานะ แบ่งตามสถานะ จากนั้นทำการหาค่าเฉลี่ยของสถานะ ทั้งสองสถานะค่าเฉลี่ย ว่าง = 0.5333, ไม่ว่าง = 0.4667

ในขั้นตอนตอนไป คำนวณค่า Entropy สำหรับคอลัมน์ สถานะ ดังสมการที่ (3.1)

$$\text{Entropy}(c1) = -\text{p}(c1)\log \text{p}(c1) \tag{3.1}$$

จะได้ดังนี้

$$\text{Entropy}(\text{สถานะ}) = \text{E}(8,7)$$

$$= -\left(\tfrac{8}{15}\log_2 \tfrac{8}{15}\right) - \left(\tfrac{7}{15}\log_2 \tfrac{7}{15}\right)$$
$$= -(0.5333 \log_2 0.5333) - (0.4667 \log_2 0.4667)$$
$$= 0.996791$$

ภาพประกอบที่ 3.66 ค่าเอนโทรปีของสถานะ

ขั้นตอนต่อไปสำหรับแอตทริบิวต์อีก 3 แอตทริบิวต์ ต้องคำนวณแอนโทรปีแยกแต่ละแอตทริบิวต์

- E (สถานะ, แสง)
- E (สถานะ, เคลื่อนไหว)
- E (สถานะ, อุณหภูมิ)

โดยการคำนวณเอนโทรปีสำหรับสองตัวแปรจะใช้สมการสูตรดังนี้

$$\text{Entropy}(S,T) = \sum_{C\in T} \text{P}(c)\,\text{E}(c) \tag{3.2}$$

การคำนวณควรสร้างตารางความถี่สำหรับสองตัวแปรเพื่อง่ายในการคำนวณ สามารถสร้างได้ดังนี้

- **E (สถานะ, แสง)**

ตารางที่ 3.9 ตารางความถี่สำหรับแสง

		สถานะ (30)		
		ว่าง	ไม่ว่าง	
แสง	มืด	4	0	4
	แสงน้อย	2	1	3
	แสงปานกลาง	1	2	3
	แสงมาก	1	2	3
	แสงจ้า	0	2	2

จาก ตารางที่ 3.9 มาทำการหาค่าเอนโทรปีของแต่ละช่วงที่อยู่ในสถานะแสง สามารถหาได้ดังนี้

E (4,0) :

$$E(\text{มืด}) = E(4,0)$$
$$= -\left(\tfrac{4}{4}\log_2\tfrac{4}{4}\right) - \left(\tfrac{0}{4}\log_2\tfrac{0}{4}\right)$$
$$= -(1\ \log_2 1) - (0\ \log_2 0)$$
$$= 0$$

ภาพประกอบที่ 3.67 ค่าเอนโทรปีของ E (มืด)

E (2,1) :

$$E(\text{แสงน้อย}) = E(2,1)$$
$$= -\left(\tfrac{2}{3}\log_2\tfrac{2}{3}\right) - \left(\tfrac{1}{3}\log_2\tfrac{1}{3}\right)$$
$$= -(0.6667\log_2 0.6667) - (0.3333\log_2 0.3333)$$
$$= 0.91826$$

ภาพประกอบที่ 3.68 ค่าเอนโทรปีของ E (แสงน้อย)

E (1,2) :

$$E(\text{ปานกลาง}) = E(1,2)$$
$$= -\left(\tfrac{1}{3}\log_2\tfrac{1}{3}\right) - \left(\tfrac{2}{3}\log_2\tfrac{2}{3}\right)$$
$$= -(0.3333\log_2 0.3333) - (0.6667\log_2 0.6667)$$
$$= 0.91826$$

ภาพประกอบที่ 3.69 ค่าเอนโทรปีของ E (แสงปานกลาง)

E (1,2) :

$$E(\text{แสงมาก}) = E(1,2)$$
$$= -\left(\tfrac{1}{3}\log_2\tfrac{1}{3}\right) - \left(\tfrac{2}{3}\log_2\tfrac{2}{3}\right)$$
$$= -(0.3333\log_2 0.3333) - (0.6667\log_2 0.6667)$$
$$= 0.91826$$

ภาพประกอบที่ 3.70 ค่าเอนโทรปีของ E (แสงมาก)

E (0,2) :

$$E(\text{แสงจ้า}) = E(0,2)$$
$$= -\left(\tfrac{0}{2}\log_2\tfrac{0}{2}\right) - \left(\tfrac{2}{2}\log_2\tfrac{2}{2}\right)$$
$$= -(0\log_2 0) - (1\log_2 1)$$
$$= 0$$

ภาพประกอบที่ 3.71 ค่าเอนโทรปีของ E (แสงจ้า)

เมื่อทำการคำนวณค่าเอนโทรปีของแต่ละช่วงของค่าแสงเรียบร้อย จากนั้นมาคำนวณค่าเอนโทรปีของแสงในสมการที่ (3.2) ได้ดังนี้

$$E(\text{สถานะ, แสง}) = P(\text{มืด})E(4,0) + P(\text{แสงน้อย})E(2,1) + P(\text{แสงปานกลาง})E(1,2) + P(\text{แสงมาก})E(1,2) + P(\text{แสงจ้า})E(0,2)$$

$$E(\text{สถานะ, แสง}) = \frac{4}{15}E(4,0) + \frac{3}{15}E(2,1) + \frac{3}{15}E(1,2) + \frac{3}{15}E(1,2) + \frac{2}{15}E(0,2)$$

$$= \frac{4}{15}(0.0) + \frac{3}{15}(0.91826) + \frac{3}{15}(0.91826) + \frac{3}{15}(0.91826) + \frac{2}{15}(0.0)$$

$$= (0.2667 * 0.0) + (0.2 * 0.91826) + (0.2 * 0.91826) + (0.2 * 0.91826) + (0.1333 * 0.0)$$

$$= 0.550956$$

ภาพประกอบที่ 3.72 ค่าเอนโทรปีของแสง

เช่นเดียวกับการคำนวณค่าเอนโทรปีของแสงครั้งก่อน เหลือ 2 แอตทริบิวต์ E (สถานะ, เคลื่อนไหว) E (สถานะ, อุณหภูมิ) จากนั้นทำการคำนวณและใช้สมการหรือสูตรเดียวกับการคำนวณค่าเอนโทรปีของแสงและแยกตารางความถี่สำหรับแต่ละช่วงเช่นกัน

- **E (สถานะ, เคลื่อนไหว)**

ตารางที่ 3.10 ตารางความถี่สำหรับเคลื่อนไหว

		สถานะ (30)		
		ว่าง	ไม่ว่าง	
เคลื่อนไหว	มีความเคลื่อนไหว	7	0	7
	ไม่มีความเคลื่อนไหว	7	1	8

จากตารางที่ 3.10 การหาค่าเอนโทรปีของแต่ละช่วงที่อยู่ในสถานะเคลื่อนไหว จะได้ค่าเอนโทรปีของแต่ละช่วงดังนี้

Entropy (เคลื่อนไหว, มีความเคลื่อนไหว) = 0

Entropy (เคลื่อนไหว, มีความเคลื่อนไหว) = 0.543564

$$E(\text{สถานะ, เคลื่อนไหว}) = P(\text{มีความเคลื่อนไหว})E(7,0) + P(\text{ไม่มีความเคลื่อนไหว})E(7,1)$$

$$E(\text{สถานะ, เคลื่อนไหว}) = \frac{7}{15} E(7,0) + \frac{8}{15} E(7,1)$$

$$= \frac{7}{15} (0.0) + \frac{8}{15} (0.543564)$$

$$= (0.2 * 0.0) + (0.5333 * 0.543564)$$

$$= 0.289883$$

ภาพประกอบที่ 3.73 ค่าเอนโทรปีของเคลื่อนไหว

- **E (สถานะ, อุณหภูมิ)**

ตารางที่ 3.11 ตารางความถี่สำหรับอุณหภูมิ

		สถานะ (30)		
		ว่าง	ไม่ว่าง	
อุณหภูมิ	เย็น	0	0	0
	ปานกลาง	0	4	4
	อบอุ่น	4	3	7
	ร้อน	4	0	4
	ร้อนมาก	0	0	0

จากตารางที่ 3.11 การหาค่าเอนโทรปีของแต่ละช่วงที่อยู่ในสถานะอุณหภูมิ จะได้ค่าเอนโทรปีของแต่ละช่วงดังนี้

Entropy (อุณหภูมิ, เย็น) = 0

Entropy (อุณหภูมิ, ปานกลาง) = 0

Entropy (อุณหภูมิ, อบอุ่น) = 0.985227

Entropy (อุณหภูมิ, ร้อน) = 0

Entropy (อุณหภูมิ, ร้อนมาก) = 0

$$E(\text{สถานะ, อุณหภูมิ}) = P(\text{เย็น})E(0,0) + P(\text{ปานกลาง})E(0,4) + P(\text{อบอุ่น})E(4,3) + P(\text{ร้อน})E(4,0) + P(\text{ร้อนมาก})E(0,0)$$

$$E(\text{สถานะ, อุณหภูมิ}) = \frac{0}{15}E(0,0) + \frac{4}{15}E(0,4) + \frac{7}{15}E(4,3) + \frac{4}{15}E(4,0) + \frac{0}{15}E(0,0)$$

$$= \frac{0}{15}(0.0) + \frac{4}{15}(0.0) + \frac{7}{15}(0.985227) + \frac{4}{15}(0.0) + \frac{0}{15}(0.0)$$

$$= (0 * 0.0) + (0.2667 * 0.0) + (0.4667 * 0.985227) + (0.2667 * 0.0) + (0 * 0.0)$$

$$= 0.4598054$$

ภาพประกอบที่ 3.74 ค่าเอนโทรปีของอุณหภูมิ

ดังนั้นค่าเอนโทรปีทั้ง 4 แอตทริบิวต์มีดังนี้

1) E (สถานะ, แสง) = 0.550956
2) E (สถานะ, เคลื่อนไหว) = 0.289883
3) E (สถานะ, อุณหภูมิ) = 0.4598054

เมื่อได้ค่าเอนโทรปีของแต่ละแอตทริบิวต์มาแล้ว จากนั้นทำการหาค่า Information Gain ของแต่ละแอตทริบิวต์ โดยหาได้จากจากสมการ ดังนี้

$$\text{Gain}(S,T) = \text{Entropy}(S) - \text{Entropy}(S,T) \qquad (3.3)$$

1. E (สถานะ, แสง) = 0.550956

Gain (สถานะ, แสง) = Entropy (สถานะ) – Entropy (สถานะ, แสง)
= 0.996791 - 0.550956
= 0.445835

2. E (สถานะ, เคลื่อนไหว) = 0.289883

Gain (สถานะ, เคลื่อนไหว) = Entropy (สถานะ) – Entropy (สถานะ, เคลื่อนไหว)
= 0.996791 - 0.289883
= 0.706908

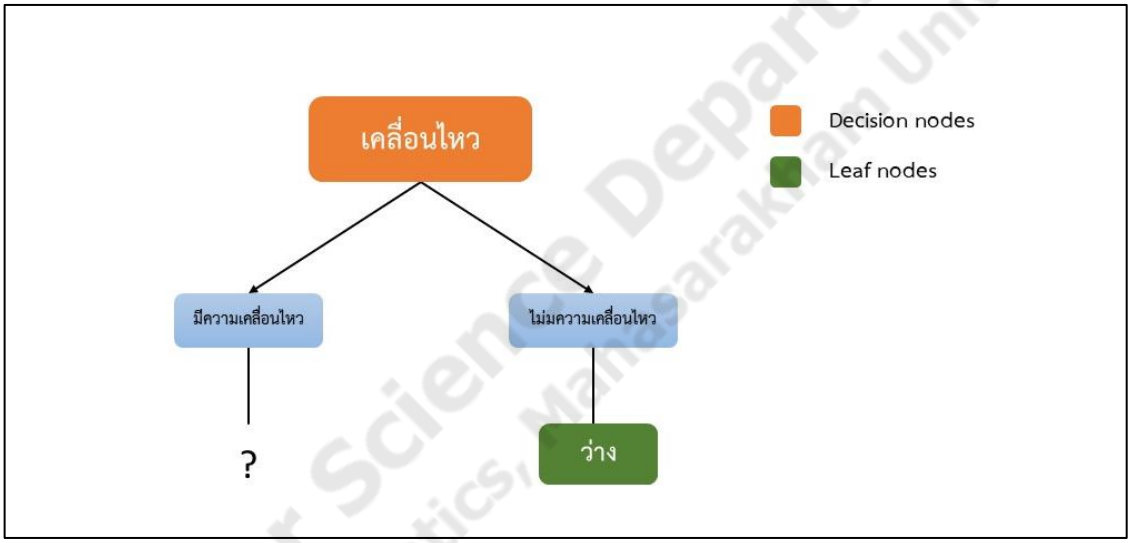
3. E (สถานะ, อุณหภูมิ) = 0.4598054

Gain (สถานะ, อุณหภูมิ) = Entropy (สถานะ) – Entropy (สถานะ, อุณหภูมิ)
= 0.996791 - 0.4598054
= 0.5367856

ขั้นตอนต่อไปเริ่มทำการแยกแผนภูมิการตัดสินใจ

เมื่อได้ข้อมูลค่า Information Gain ของทุกแอตทริบิวต์ ซึ่งเกณฑ์ที่ช่วยตัดสินในการเลือก root node คือ ทดลองเลือกแอตทริบิวต์ แต่ละตัวมาทำหน้าที่เป็น root node แล้วหาค่า Information Gain ซึ่งเป็นค่าที่ใช้บอกว่า แอตทริบิวต์ ที่ทำหน้าที่เป็น root node สามารถจำแนกข้อมูลได้ดีมากน้อยเพียงใด โดยจะเลือกแอตทริบิวต์ ที่ใช้ค่า Information Gain สูงสุดเป็น root node

จากการคำนวณ ข้อมูลที่ได้ค่า Information Gain สูงสุดคือ เคลื่อนไหว ดังนั้นการแยกแผนภูมิการตัดสินใจจะมีลักษณะดังนี้



ภาพประกอบที่ 3.75 โครงสร้างการตัดสินใจหลังจากแยกครั้งแรก

เนื่องจากแอตทริบิวต์ เคลื่อนไหว ยังไม่สามารถจัดกลุ่มข้อมูลให้เป็นคลาสเดียวกันทั้งหมดโดยยังมี ค่าเคลื่อนไหว = มีความเคลื่อนไหว ยังมีคลาส ว่าง และไม่ว่างคละกันอยู่ *(เคลื่อนไหว = มีความเคลื่อนไหว จัดกลุ่มข้อมูลที่เป็นคลาส ว่าง จำนวน 1 ฟิลด์ และคลาส ไม่ว่าง จำนวน 7 ฟิลด์)* จึงต้องสร้าง decision tree ต่อไป โดยพิจารณาเลือกแอตทริบิวต์ ที่จะมาเป็นโหนดในระดับที่ 2 ต่อจาก root node แต่ในกรณี เคลื่อนไหว = ไม่มีความเคลื่อนไหว ไม่จำเป็นต้องสร้างโหนดเพิ่มเติม เนื่องจากสามารถจัดกลุ่มข้อมูลที่เป็นคลาส ว่าง ได้ทั้งหมด

เนื่องจากง่ายต่อการเรียงข้อมูลและง่ายต่อการจัดกลุ่มข้อมูล จึงทำการแยกตารางต้นฉบับเพื่อสร้างตารางย่อยโดยใช้ค่าเคลื่อนไหวเป็นเกณฑ์ในการแยกตารางย่อย ดังนั้นจะได้ตารางย่อยดังนี้

ตารางที่ 3.12 ตารางการแบ่งตารางย่อยของค่าเคลื่อนไหวตามช่วงไม่มีความเคลื่อนไหว

แสง	เคลื่อนไหว	อุณหภูมิ	สถานะ
มืด	ไม่มีความเคลื่อนไหว	ร้อน	ว่าง
แสงน้อย	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง
มืด	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง
มืด	ไม่มีความเคลื่อนไหว	ร้อน	ว่าง
มืด	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง
แสงปานกลาง	ไม่มีความเคลื่อนไหว	ร้อน	ว่าง
แสงน้อย	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง

ตารางที่ 3.13 ตารางการแบ่งตารางย่อยของค่าเคลื่อนไหวตามช่วงมีความเคลื่อนไหว

แสง	เคลื่อนไหว	อุณหภูมิ	สถานะ
แสงปานกลาง	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงปานกลาง	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
แสงมาก	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงจ้า	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
แสงน้อย	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงมาก	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
แสงจ้า	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงมาก	มีความเคลื่อนไหว	ร้อน	ว่าง

จากตารางที่ 3.13 จะเห็นได้ว่า การแบ่งตารางย่อยของค่าเคลื่อนไหวตามช่วงมีความเคลื่อนไหวยังไม่สามารถจัดกลุ่มข้อมูลให้เป็นคลาสเดียวกันทั้งหมด จึงจะใช้ช่วงของมีความเคลื่อนไหวมาทำการสร้างโหนดในในระดับที่ 2 ต่อจาก root node โดยแอตทริบิวต์ที่จะใช้ในการคำนวณที่ใช้สร้างโหนดต่อไปมีดังนี้ แอตทริบิวต์ของแสง และแอตทริบิวต์ของอุณหภูมิ โดยทั้งสองแอตทริบิวต์จะคำนวณหาค่า Entropy และหาค่า Information Gain เช่นเดียวกับการคำนวณหาค่าเพื่อสร้าง root nodes โดยจะใช้การคำนวณและสมการเดียวกัน แต่การคำนวณโหนดระดับที่ 2 ต่างจากการคำนวณ root nodes ใน

ระดับที่ 2 จะดูชุดข้อมูลภายในกลุ่มสถานะเคลื่อนไหว ในช่วงของ มีความเคลื่อนไหวเท่านั้น หรือดูได้จาก ตารางที่ 3.13

ต่อไปจะต้องคำนวณแอนโทรปีแยกแต่ละครั้งสำหรับแอตทริบิวต์ 2 แอตทริบิวต์ที่เหลือ ได้ดังนี้

- $\mathbf{E}_{\text{แสง}}$ (สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว)
- $\mathbf{E}_{\text{อุณหภูมิ}}$ (สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว)

โดยจะแยกตารางความถี่ค่าเคลื่อนไหวตามช่วงมีความเคลื่อนไหว สามารถแยกได้ดังนี้

- **$\mathbf{E}_{\text{แสง}}$ (สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว)**

ตารางที่ 3.14 ตารางความถี่ของแสงตามช่วงมีความเคลื่อนไหว

		(สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว) (8)		
		ว่าง	ไม่ว่าง	
แสง	มืด	0	0	0
	แสงน้อย	0	1	1
	แสงปานกลาง	0	2	2
	แสงมาก	1	2	3
	แสงจ้า	0	2	2

จากตารางที่ 3.14 มาทำการหาค่าเอนโทรปีของแต่ละช่วงที่อยู่ในสถานะแสง โดยจะใช้สมการ (3.1) สามารถคำนวณได้ดังนี้

E (0,0) :

$$E(\text{มืด}) = E(0,0)$$
$$= -\left(\tfrac{0}{0}\log_2\tfrac{0}{0}\right) - \left(\tfrac{0}{0}\log_2\tfrac{0}{0}\right)$$
$$= -(0\ \log_2 0) - (0\ \log_2 0)$$
$$= 0$$

ภาพประกอบที่ 3.76 ค่าเอนโทรปีของ E (มืด) ตามช่วงมีความเคลื่อนไหว

E (0,1) :

$$E(\text{แสงน้อย}) = E(0,1)$$
$$= -\left(\tfrac{0}{1}\log_2\tfrac{0}{1}\right) - \left(\tfrac{1}{1}\log_2\tfrac{1}{1}\right)$$
$$= -(0\ \log_2 0) - (1\ \log_2 1)$$
$$= 0$$

ภาพประกอบที่ 3.77 ค่าเอนโทรปีของ E (แสงน้อย) ตามช่วงมีความเคลื่อนไหว

E (0,2) :

$$E(\text{ปานกลาง}) = E(0,2)$$
$$= -\left(\tfrac{0}{2}\log_2\tfrac{0}{2}\right) - \left(\tfrac{2}{2}\log_2\tfrac{2}{2}\right)$$
$$= -(0\ \log_2 0) - (1\ \log_2 1)$$
$$= 0$$

ภาพประกอบที่ 3.78 ค่าเอนโทรปีของ E (แสงปานกลาง) ตามช่วงมีความเคลื่อนไหว

E (1,2) :

$$E(\text{แสงมาก}) = E(1,2)$$
$$= -\left(\tfrac{1}{3}\log_2\tfrac{1}{3}\right) - \left(\tfrac{2}{3}\log_2\tfrac{2}{3}\right)$$
$$= -(0.3333\ \log_2 0.3333) - (0.6667\log_2 0.6667)$$
$$= 0.91826$$

ภาพประกอบที่ 3.79 ค่าเอนโทรปีของ E (แสงมาก) ตามช่วงมีความเคลื่อนไหว

E (0,2) :

$$E(\text{แสงจ้า}) = E(0,2)$$
$$= -\left(\tfrac{0}{2}\log_2\tfrac{0}{2}\right) - \left(\tfrac{2}{2}\log_2\tfrac{2}{2}\right)$$
$$= -(0\ \log_2 0) - (1\ \log_2 1)$$
$$= 0$$

ภาพประกอบที่ 3.80 ค่าเอนโทรปีของ E (แสงจ้า) ตามช่วงมีความเคลื่อนไหว

เมื่อทำการคำนวณค่าเอนโทรปีของแต่ละช่วงของค่าแสงตามช่วงมีความเคลื่อนไหวเรียบร้อย จากนั้นมาคำนวนค่าเอนโทรปีของแสงในสมการที่ (3.2)

$$E_{\text{แสง}}(\text{สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว}) = P(\text{มืด})E(0,0) + P(\text{แสงน้อย})E(0,1) + P(\text{แสงปานกลาง})E(0,2) + P(\text{แสงมาก})E(1,2) + P(\text{แสงจ้า})E(0,2)$$
$$E(\text{สถานะ, แสง}) = \tfrac{4}{8}E(0,0) + \tfrac{3}{8}E(0,1) + \tfrac{3}{8}E(0,2) + \tfrac{3}{8}E(1,2) + \tfrac{2}{8}E(0,2)$$
$$= \tfrac{0}{8}(0.0) + \tfrac{1}{8}(0.0) + \tfrac{2}{8}(0.0) + \tfrac{3}{8}(0.91826) + \tfrac{2}{8}(0.0)$$
$$= (0 * 0.0) + (0.125 * 0.0) + (0.25 * 0.0) + (0.375 * 0.91826) + (0.125 * 0.0)$$
$$= 0.344347$$

ภาพประกอบที่ 3.81 ค่าเอนโทรปีของแสง ตามช่วงมีความเคลื่อนไหว

ในการหาค่าเอนโทรปีจาก ภาพประกอบที่ 3.81 ค่าที่นำมาหารนั้นจะเป็นค่าของชุดข้อมูลที่อยู่ในช่วงมีความเคลื่อนไหว โดยจะมีชุดข้อมูลทั้งหมด 8 ครั้ง จะดูได้จาก ตารางที่ 3.13 และจะเหลืออีก 1 แอตทริบิวต์ คืออุณหภูมิ ดังนั้นการคำนวณและใช้สมการหรือสูตรเดียวกับการคำนวณค่าเอนโทรปีของแสง ตามช่วงมีความเคลื่อนไหว จะหาค่าและคำนวณเหมือนกัน

- $\boldsymbol{E}_{\text{อุณหภูมิ}}$ **(สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว)**

ตารางที่ 3.15 ตารางความถี่ของอุณหภูมิตามช่วงมีความเคลื่อนไหว

		(สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว) (8)		
		ว่าง	ไม่ว่าง	
อุณหภูมิ	เย็น	0	0	0
	ปานกลาง	0	4	4
	อบอุ่น	0	3	3
	ร้อน	1	0	1
	ร้อนมาก	0	0	0

จากตารางที่ 3.15 การหาค่าเอนโทรปีของแต่ละช่วงที่อยู่ในสถานะอุณหภูมิตามช่วงมีความเคลื่อนไหว จะได้ค่าเอนโทรปีของแต่ละช่วงดังนี้

Entropy (อุณหภูมิ, เย็น) = 0

Entropy (อุณหภูมิ, ปานกลาง) = 0

Entropy (อุณหภูมิ, อบอุ่น) = 0

Entropy (อุณหภูมิ, ร้อน) = 0

Entropy (อุณหภูมิ, ร้อนมาก) = 0

$$\boldsymbol{E}_{\text{อุณหภูมิ}}\text{ (สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว)} = P(\text{เย็น})E(0,0) + P(\text{ปานกลาง})E(0,4) + P(\text{อบอุ่น})E(0,3) + P(\text{ร้อน})E(1,0) + P(\text{ร้อนมาก})E(0,0)$$

$$E(\text{สถานะ, อุณหภูมิ}) = \frac{0}{8}E(0,0) + \frac{4}{8}E(0,4) + \frac{3}{8}E(0,3) + \frac{1}{8}E(1,0) + \frac{0}{8}E(0,0)$$

$$= \frac{0}{8}(0.0) + \frac{4}{8}(0.0) + \frac{3}{8}(0.0) + \frac{1}{8}(0.0) + \frac{0}{8}(0.0)$$

$$= (0 * 0.0) + (0.5 * 0.0) + (0.375 * 0.0) + (0.125 * 0.0) + (0 * 0.0)$$

$$= 0$$

ภาพประกอบที่ 3.82 ค่าเอนโทรปีของอุณหภูมิ ตามช่วงมีความเคลื่อนไหว

ดังนั้นค่าเอนโทรปีที่ตามช่วงมีความเคลื่อนไหวทั้ง 2 แอตทริบิวต์มีดังนี้

1) $\mathbf{E}_{\text{แสง}}$ (สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว) = 0.344347
2) $\mathbf{E}_{\text{อุณหภูมิ}}$ (สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว) = 0

เมื่อได้ค่าเอนโทรปีของแต่ละแอตทริบิวต์มาแล้ว จากนั้นทำการหาค่า Information Gain ของแต่ละแอตทริบิวต์ โดยหาได้จากจากสมการ ดังนี้

1. $\mathbf{E}_{\text{แสง}}$ (สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว) = 0.344347

Gain (สถานะ, แสง) = **Entropy** (เคลื่อนไหว = มีความเคลื่อนไหว) – $\mathbf{Entropy}_{\text{แสง}}$ ((เคลื่อนไหว = มีความเคลื่อนไหว)

= 0.543564 - 0.344347

= 0.199217

2. $\mathbf{E}_{\text{อุณหภูมิ}}$ (สถานะ, เคลื่อนไหว = มีความเคลื่อนไหว) = 0

Gain (สถานะ, อุณหภูมิ) = **Entropy** (เคลื่อนไหว = มีความเคลื่อนไหว) – $\mathbf{Entropy}_{\text{อุณหภูมิ}}$ (เคลื่อนไหว = มีความเคลื่อนไหว)

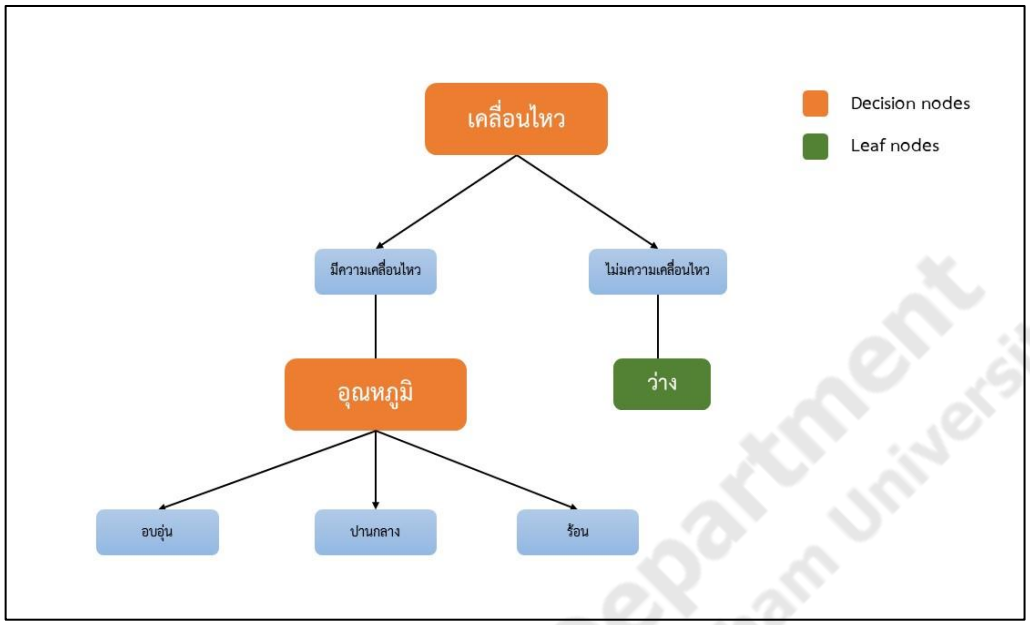
= 0.543564 - 0

= 0.543564

ในการหาค่า Gain ค่าเอนโทรปีที่ในการลบนั้นจะใช้ค่าเอนโทรปีของช่วงมีความเคลื่อนไหวมาทำการคำนวณ โดยมีค่าดังนี้ Entropy (เคลื่อนไหว, มีความเคลื่อนไหว) = 0.543564

ขั้นตอนต่อไปเริ่มทำการแยกแผนภูมิการตัดสินใจ

เมื่อคำนวณหาค่า Information Gain ของแต่ละแอตทริบิวต์เรียบร้อย เกณฑ์ในการเลือกโหนดที่จะสร้างโหนดในระดับที่ 2 ก็จะเลือกเหมือนเกณฑ์การเลือก root nodes เลือกแอตทริบิวต์ที่มีค่า Information Gain ที่มากที่สุด โดยแอตทริบิวต์ที่เลือกจะสร้างโหนดระดับที่ 2 คือ แอตทริบิวต์อุณหภูมิ โดยมีค่า Information Gain = 0.543564 เมื่อเทียบกับแอตทริบิวต์ที่เหลือ และเมื่อเลือกแอตทริบิวต์ที่จะสร้างในโหนดระดับที่ 2 ดังนั้นการแยกแผนภูมิการตัดสินใจจะมีลักษณะดังนี้



ภาพประกอบที่ 3.83 โครงสร้างการตัดสินใจโดยใช้แอตทริบิวต์อุณหภูมิ

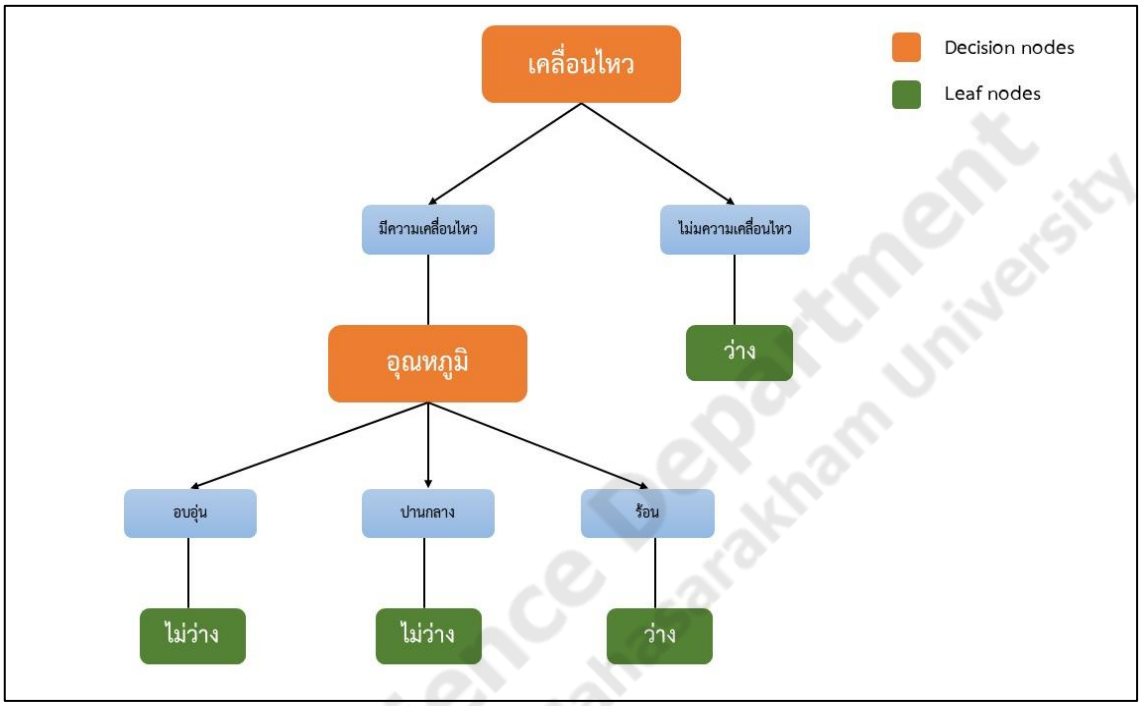
ในส่วนนี้จะเห็นได้ว่าการจัดกลุ่มข้อมูลของแอตทริบิวต์อุณหภูมิจะอยู่ในคลาสเดียวกัน (*อุณหภูมิ = อบอุ่น จัดกลุ่มข้อมูลที่เป็นคลาส ว่าง จำนวน 0 ฟิลด์ และคลาส ไม่ว่าง จำนวน 4 ฟิลด์ อุณหภูมิ = ปานกลาง จัดกลุ่มข้อมูลที่เป็นคลาส ว่าง จำนวน 0 ฟิลด์ และคลาส ไม่ว่าง จำนวน 3 ฟิลด์ อุณหภูมิ = ร้อน จัดกลุ่มข้อมูลที่เป็นคลาส ว่าง จำนวน 1 ฟิลด์ และคลาส ไม่ว่าง จำนวน 0 ฟิลด์*) ซึ่งถ้าในกรณีไม่จำเป็นต้องสร้างโหนดเพิ่มเติม เนื่องจากสามารถจัดกลุ่มข้อมูลที่เป็นคลาส ว่าง และ ไม่ว่าง ได้ทั้งหมด

เนื่องจากง่ายต่อการเรียงข้อมูลและง่ายต่อการจัดกลุ่มข้อมูล จึงทำการแยกตารางต้นฉบับเพื่อสร้างตารางย่อยให้เห็นในแอตทริบิวต์อุณภูมิ ดังนั้นจะได้ตารางย่อยดังนี้

ตารางที่ 3.16 ตารางการแบ่งตารางย่อยของค่าอุณหภูมิตามช่วงมีความเคลื่อนไหว

แสง	เคลื่อนไหว	อุณหภูมิ	สถานะ
แสงปานกลาง	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
แสงจ้า	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
แสงมาก	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
แสงปานกลาง	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงมาก	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงน้อย	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงจ้า	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงมาก	มีความเคลื่อนไหว	ร้อน	ว่าง

จากตารางที่ 3.16 จะเห็นการแบ่งกลุ่มได้อย่างชัดเจน ว่าแอตทริบิวต์อุณภูมิ สามารถจัดกลุ่มให้อยู่ในกลุ่มเดียวกันได้ทั้งหมด ดังงนั้นการแยกแผนภูมิการตัดสินใจจะมีลักษณะดังนี้



ภาพประกอบที่ 3.84 Final Decision Tree

ต้นไม้ตัดสินใจ สามารถแปลง Classification rules ได้ดังนี้

rule 1 : if (เคลื่อนไหว = มีความเคลื่อนไหว) AND (อุณหภูมิ = ปานกลาง) THEN สถานะ = ไม่ว่าง

rule 2 : if (เคลื่อนไหว = มีความเคลื่อนไหว) AND (อุณหภูมิ = อบอุ่น) THEN สถานะ = ไม่ว่าง

rule 3 : if (เคลื่อนไหว = มีความเคลื่อนไหว) AND (อุณหภูมิ = ร้อน) THEN สถานะ = ว่าง

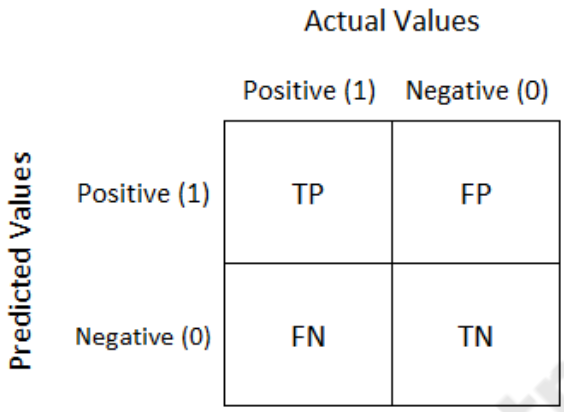
rule 4 : if (เคลื่อนไหว = ไม่มีความเคลื่อนไหว) THEN สถานะ = ว่าง

3.12 การทำนายและการประเมินผล

วัดประสิทธิภาพ Model จาก Confusion Matrix [21]

การที่จะนำ Model ไปใช้งานจริงได้นั้น จำเป็นต้องมีการวัดประสิทธิภาพ Model ก่อนว่า Model นั้นมีประสิทธิภาพเพียงพอที่จะนำมาพัฒนา หรือ นำไปใช้งานด้านต่างๆ ซึ่งการวัดประสิทธิภาพนั้นส่วนใหญ่จะวัดค่าจากใน Table ข้อมูลที่มี (Confusion Matrix)

Confusion Matrix คือตารางสำคัญในการวัดความสามารถของ machine learning ในการแก้ปัญหา classification



ภาพประกอบที่ 3.85 ตัวอย่างตาราง Confusion Matrix ขนาด 2x2

True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” และ มีค่าเป็น “ จริง ”

True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” และ มีค่า “ ไม่จริง ”

False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” แต่ มีค่าเป็น “ไม่จริง”

False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” แต่ มีค่าเป็น “จริง”

ตัววัดที่นิยมใช้กันในงานวิจัยและการทำงานต่างๆ อยู่ 3 ค่า และสมการ คือ

1. Accuracy เป็นการวัดความถูกต้องของ Model โดยพิจารณารวมทุกคลาส

$$\frac{TP + TN}{TP + TN + FP + FN}$$

ภาพประกอบที่ 3.86 สมการหาค่า Accuracy

2. Recall เป็นการวัดความถูกต้องของ Model โดยพิจารณาแยกทีละคลาส

$$\frac{TP}{TP + FN}$$

ภาพประกอบที่ 3.87 สมการหาค่า Recall

3. Precision เป็นการวัดความแม่นยำของข้อมูล โดยพิจารณาแยกทีละคลาส

$$\frac{TP}{TP + FP}$$

ภาพประกอบที่ 3.88 สมการหาค่า Precision

อีกหนึ่งวิธีที่นำมาใช้คือ F1-Score

F1-Score คือค่าเฉลี่ยแบบ harmonic mean ระหว่าง precision และ recall สร้าง F1 ขึ้นมาเพื่อเป็น single metric ที่วัดความสามารถของโมเดล

$$F1 = 2 * \left(\frac{precision * recall}{precision + recall}\right)$$

ภาพประกอบที่ 3.89 สมการ F1-Score

จาก ตารางที่ 3.17 ตัวอย่างชุดข้อมูล และกฎที่ได้ดังนี้

กฎข้อที่ 1 : if (เคลื่อนไหว = มีความเคลื่อนไหว) AND (อุณหภูมิ = ปานกลาง) THEN สถานะ = ไม่ว่าง

กฎข้อที่ 2 : if (เคลื่อนไหว = มีความเคลื่อนไหว) AND (อุณหภูมิ = อบอุ่น) THEN สถานะ = ไม่ว่าง

กฎข้อที่ 3 : if (เคลื่อนไหว = มีความเคลื่อนไหว) AND (อุณหภูมิ = ร้อน) THEN สถานะ = ว่าง

กฎข้อที่ 4 : if (เคลื่อนไหว = ไม่มีความเคลื่อนไหว) THEN สถานะ = ว่าง

ตารางที่ 3.17 ตัวอย่างชุดข้อมูลทดสอบ

แสง	เคลื่อนไหว	อุณหภูมิ	สถานะ
มืด	ไม่มีความเคลื่อนไหว	ร้อน	ว่าง
แสงน้อย	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงมาก	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
แสงจ้า	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงมาก	มีความเคลื่อนไหว	ร้อน	ว่าง
แสงปานกลาง	มีความเคลื่อนไหว	ปานกลาง	ไม่ว่าง
แสงปานกลาง	มีความเคลื่อนไหว	อบอุ่น	ไม่ว่าง
แสงน้อย	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง
มืด	ไม่มีความเคลื่อนไหว	อบอุ่น	ว่าง
มืด	มีความเคลื่อนไหว	อบอุ่น	ว่าง

ตารางที่ 3.18 ตารางผล Confusion Matrix

		Actual Values	
		C1 = ว่าง	**C2 = ไม่ว่าง**
Predicted Values	**C1 = ว่าง**	TP = 4	FP = 0
	C2 = ไม่ว่าง	FN = 1	TN = 5

โดยที่ :: C1 คือ ห้องเรียนว่าง

C2 คือ ห้องเรียนไม่ว่าง

TP คือ จำนวนข้อมูลทอดสอบที่ทำนายถูกต้องว่าห้องเรียนว่าง

FP คือ จำนวนข้อมูลทดสอบที่ทำนายว่าห้องเรียนว่างแต่ความเป็นจริงห้องเรียนไม่ว่าง

FN คือ จำนวนข้มูลทดสอบที่ทำนายว่าห้องเรียนไม่ว่างแต่ความเป้นจริงห้องเรียนว่าง

TN คือ จำนวนข้อมูลทดสอบที่ทำนายถูกต้องว่าห้องเรียนไม่ว่าง

การหาค่า Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$= \frac{4+5}{4+0+1+5}$$

$$= \frac{9}{10}$$

$$= 0.9$$

ดังนั้นค่าความถูกต้อง (Accuracy) คือ 0.9 หรือ 90%

การหาค่า Precision

$$Precision_{c1} = \frac{TP}{TP+FN}$$

$$= \frac{4}{4+1}$$

$$= \frac{4}{5}$$

$$= 0.8$$

$$Precision_{c2} = \frac{TN}{TN+FP}$$

$$= \frac{5}{5+0}$$

$$= \frac{5}{5}$$

$$= 1$$

ดังนั้น ค่าความแม่นยำ (Precision) คลาส C1 มีค่าเท่ากับ 0.8 หรือ 80%
ค่าความแม่นยำ (Precision) คลาส C2 มีค่าเท่ากับ 1.00 หรือ 100%
แสดงว่า ค่าความแม่นยำในการทำนายคลาส C2 มีค่ามากที่สุด

การหาค่า Recall

$$Recall_{c1} = \frac{TP}{TP+FP}$$

$$= \frac{4}{4+0}$$

$$= \frac{4}{4}$$

$$= 1$$

$$Recall_{c2} = \frac{TN}{TN+FN}$$

$$= \frac{5}{5+1}$$

$$= \frac{5}{6}$$

$$= 0.83$$

ดังนั้น ค่าระลึก (Recall) คลาส C1 มีค่าเท่ากับ 1.00 หรือ 100%

ค่าระลึก (Recall) คลาส C2 มีค่าเท่ากับ 0.83 หรือ 83%

แสดงว่า ค่าระลึกในการทำนายคลาส C1 มีค่ามากที่สุด

การหาค่า F1-Score

$$F1 - Score_{c1} = \frac{2 \times (precision_{c1} \times recall_{c1})}{precision_{c1} + recall_{c1}}$$

$$= \frac{2 \times (0.8 \times 1)}{(0.8+1)}$$

$$= 0.89$$

$$F1 - Score_{c2} = \frac{2 \times (precision_{c2} \times recall_{c2})}{precision_{c2} + recall_{c2}}$$

$$= \frac{2 \times (1 \times 0.83)}{(1+0.83)}$$

$$= 0.9$$

ดังนั้น ค่า (F1-Score) คลาส C1 มีค่าเท่ากับ 0.89 หรือ 89%

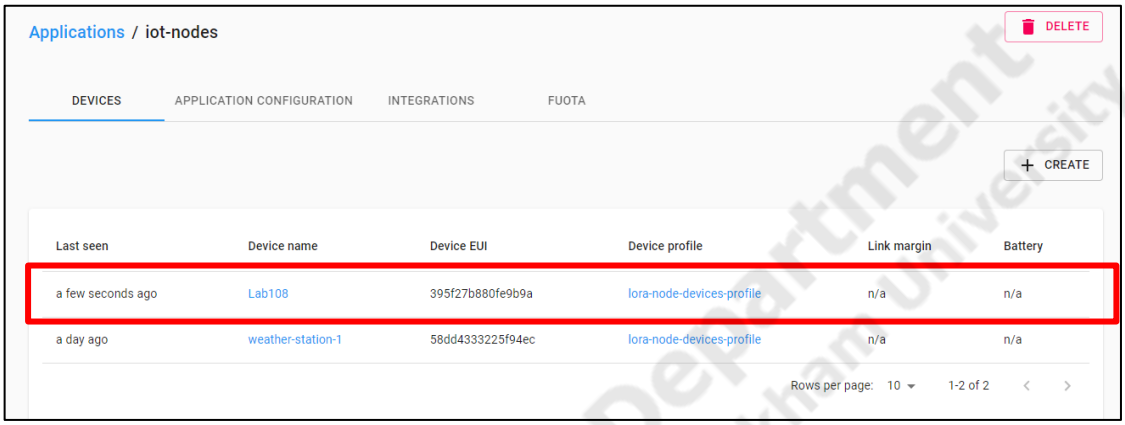
ค่า (F1-Score) คลาส C2 มีค่าเท่ากับ 0.9 หรือ 90%

แสดงว่า ค่า (F1-Score) ในการทำนายคลาส C2 มีค่ามากที่สุด

จากค่าการวัดประสิทธิภาพโมเดล ในข้อมูล 10 ข้อมูลมาทำการทำนายโดยสามารถทำนายได้ว่า C2 นั้นมีค่า F1-Score มากที่สุด

3.13 การเพิ่มอุปกรณ์ใหม่

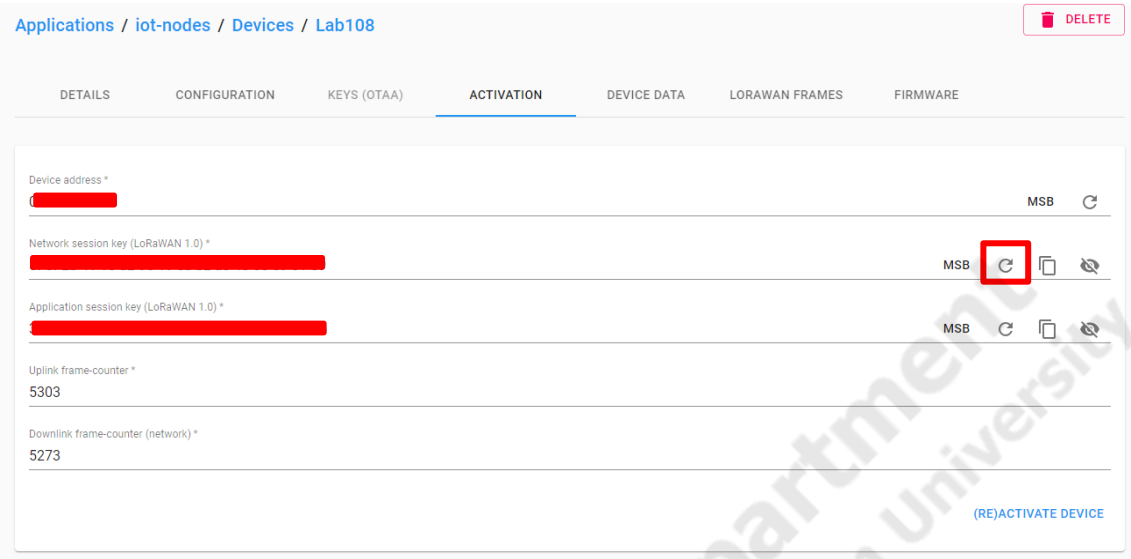
3.13.1 การเพิ่ม Device sensor ตัวใหม่ใน Chirp Stack วิธีการเพิ่ม Node ทั้งใน Chirp Stack หรือ TheThingsNetwork จะเก็บไว้ในเมนูที่เรียกว่า Application



ภาพประกอบที่ 3.90 การเพิ่มอุปกรณ์ใหม่

เมื่อ create Device เสร็จ ให้คลิก Device ที่เราสร้างขึ้น เพื่อเข้าไปเพิ่มเติมข้อมูล แท็ปแรกที่ต้องเพิ่มคือ Activation ให้ป้อน Device Address, Network Session Key, Application

Session Key ถ้าค่าเหล่านี้เราไม่มีสามารถคลิกเครื่องหมายลูกศรวงกลมด้านหลังให้ระบบสร้างขึ้นใหม่ แล้วคลิก Reactivate Device



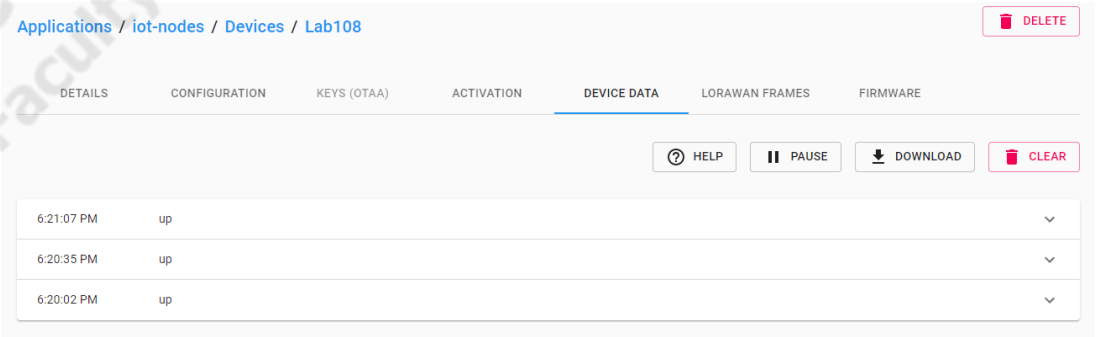
ภาพประกอบที่ 3.91 กรอกข้อมูลเพิ่มเติม

3.13.2 ชื่อ Sensor Node เป็นชื่อ Lab108 ให้นำค่า Device Address, Network Session Key, Application Session Key ไปใส่ในโปรแกรม

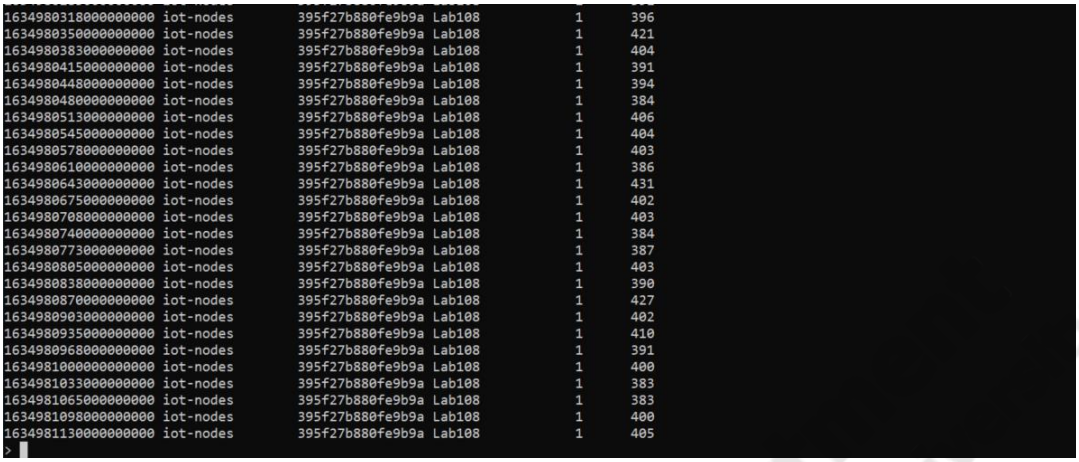
```
2
3 static const PROGMEM u1_t NWKSKEY[16] = { 0
4                                           0x19, 0xea, 0xb2, 0xa0, 0x48, 0x08, 0x59, 0x61, 0x59 };
5 static const u1_t PROGMEM APPSKEY[16] = { 0
6                                          0xa9, 0x11, 0x35, 0xd3, 0x85, 0xab, 0xa0, 0x78, 0xc6 };
7 static const u4_t DEVADDR = 0       ; // <-- Change this address for every node!
```

ภาพประกอบที่ 3.92 เพิ่มค่าลงในโปรแกรม

3.13.3 ถ้า Node Device Lab108 เปิดและส่งข้อมูลอยู่จะเห็นข้อมูล Up ขึ้น โดยแสดง Packet ละบรรทัด



ภาพประกอบที่ 3.93 ข้อมูลส่งมายัง server ได้สำเร็จ



```
1634980318000000000 iot-nodes        395f27b880fe9b9a Lab108          1     396
1634980350000000000 iot-nodes        395f27b880fe9b9a Lab108          1     421
1634980383000000000 iot-nodes        395f27b880fe9b9a Lab108          1     404
1634980415000000000 iot-nodes        395f27b880fe9b9a Lab108          1     391
1634980448000000000 iot-nodes        395f27b880fe9b9a Lab108          1     394
1634980480000000000 iot-nodes        395f27b880fe9b9a Lab108          1     384
1634980513000000000 iot-nodes        395f27b880fe9b9a Lab108          1     406
1634980545000000000 iot-nodes        395f27b880fe9b9a Lab108          1     404
1634980578000000000 iot-nodes        395f27b880fe9b9a Lab108          1     403
1634980610000000000 iot-nodes        395f27b880fe9b9a Lab108          1     386
1634980643000000000 iot-nodes        395f27b880fe9b9a Lab108          1     431
1634980675000000000 iot-nodes        395f27b880fe9b9a Lab108          1     402
1634980708000000000 iot-nodes        395f27b880fe9b9a Lab108          1     403
1634980740000000000 iot-nodes        395f27b880fe9b9a Lab108          1     384
1634980773000000000 iot-nodes        395f27b880fe9b9a Lab108          1     387
1634980805000000000 iot-nodes        395f27b880fe9b9a Lab108          1     403
1634980838000000000 iot-nodes        395f27b880fe9b9a Lab108          1     390
1634980870000000000 iot-nodes        395f27b880fe9b9a Lab108          1     427
1634980903000000000 iot-nodes        395f27b880fe9b9a Lab108          1     402
1634980935000000000 iot-nodes        395f27b880fe9b9a Lab108          1     410
1634980968000000000 iot-nodes        395f27b880fe9b9a Lab108          1     391
1634981000000000000 iot-nodes        395f27b880fe9b9a Lab108          1     400
1634981033000000000 iot-nodes        395f27b880fe9b9a Lab108          1     383
1634981065000000000 iot-nodes        395f27b880fe9b9a Lab108          1     383
1634981098000000000 iot-nodes        395f27b880fe9b9a Lab108          1     400
1634981130000000000 iot-nodes        395f27b880fe9b9a Lab108          1     405
>
```

ภาพประกอบที่ 3.94 ข้อมูลลง InfluxDB