# การเฝ้าสังเกตสภาพแวดล้อมห้องเซิร์ฟเวอร์ด้วยอินเทอร์เน็ตประสานสรรพสิ่ง โดยใช้ ESP8266 และ NETPIE

**นางสาวขวัญทิพย์ อิทธิผล**

การศึกษาค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศและการสื่อสาร)
สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช
พ.ศ. 2566

# Monitoring a server room environment with the Internet of Everything using ESP8266 and NETPIE

Miss. KHWANTHIP ITTHIPHOL

An Independent Study Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science (Information and Communication Technology)
School of Science and Technology Sukhothai Thammathirat Open University
2023

**หัวข้อการศึกษาค้นคว้าอิสระ** การเฝ้าสังเกตสภาพแวดล้อมห้องเซิร์ฟเวอร์ด้วยอินเทอร์เน็ตประสานสรรพสิ่งโดยใช้ ESP8266 และ NETPIE

**ชื่อและนามสกุล** นางสาวขวัญทิพย์ อิทธิผล

**แขนงวิชา / วิชาเอก** กลุ่มวิชาเทคโนโลยีสารสนเทศและการสื่อสาร

**สาขาวิชา** วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช

**อาจารย์ที่ปรึกษา** อาจารย์ ดร.เตชค์ฐสิณป์ เพียซ้าย

การศึกษาค้นคว้าอิสระนี้ได้รับความเห็นชอบให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรระดับปริญญาโท เมื่อวันที่ 28 สิงหาคม พ.ศ. 2566

คณะกรรมการสอบการศึกษาค้นคว้าอิสระ

---------- ประธานกรรมการ

(อาจารย์ ดร.เตชค์ฐสิณป์ เพียซ้าย)

---------- กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วีรพล จิรจริต)

---

(ผู้ช่วยศาสตราจารย์ ดร.ภูมิ เจือศิริภักดี)

ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

ชื่อการศึกษาค้นคว้าอิสระ การเฝ้าสังเกตสภาพแวดล้อมห้องเซิร์ฟเวอร์ด้วยอินเทอร์เน็ตประสานสรรพสิ่งโดยใช้ ESP8266 และ NETPIE
ผู้ศึกษา นางสาวขวัญทิพย์ อิทธิผล รหัสนักศึกษา 2649600265
ปริญญา วิทยาศาสตรมหาบัณฑิต
อาจารย์ที่ปรึกษา อาจารย์ ดร.เตชค์ฐสิณป์ เพียซ้าย ปีการศึกษา 2566

# บทคัดย่อ

การศึกษาค้นคว้าอิสระนี้มีวัตถุประสงค์คือ (1) เพื่อประยุกต์อินเทอร์เน็ตประสานสรรพสิ่งตรวจจับสภาพแวดล้อมของห้องเซิร์ฟเวอร์ด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์ม และ (2) เพื่อทดสอบประสิทธิภาพการทำงานของการพัฒนาระบบต้นแบบที่เสนอ

แนวคิดของอินเทอร์เน็ตประสานสรรพสิ่งได้ถูกใช้ในการพัฒนาระบบ โดยใช้อุปกรณ์เซนเซอร์สำหรับ ตรวจจับอุณหภูมิและความชื้นของเครื่องแม่ข่ายและห้องเซิร์ฟเวอร์ ตรวจสอบจับน้ำแอร์หยด และประยุกต์อุปกรณ์รีเลย์เพื่อตรวจจับไฟฟ้าดับและควบคุมการทำงานของอุปกรณ์พัดลมระบายความร้อนจากข้อมูลค่าอุณหภูมิที่ตรวจจับได้ โดยแต่ละอุปกรณ์จะถูกควบคุมการทำงานด้วยอุปกรณ์ ESP8266 และแต่ละข้อมูลที่ถูกตรวจจับได้จะถูกนำเสนอเป็นรายงานแบบแดชบอร์ดด้วย NETPIE แพลตฟอร์ม เพื่อให้ผู้ดูแลระบบสามารถเฝ้าสังเกตได้ ค่าอุณหภูมิของเครื่องแม่ข่ายและห้องเซิร์ฟเวอร์และค่าน้ำหยดจากแอร์ที่สูงกว่าเกณฑ์กำหนดจะถูกส่งเป็นข้อความผ่านบริการแจ้งเตือนระบบไลน์

ผลของการศึกษาพบว่า (1) การประยุกต์อินเทอร์เน็ตประสานสรรพสิ่งของอุปกรณ์เซนเซอร์ อุปกรณ์รีเลย์ อุปกรณ์ ESP8266 ข้อมูลการตรวจจับ กระบวนการตรวจจับ NETPIE แพลตฟอร์ม และผู้ดูแลระบบสามารถทำงานประสานกันเป็นระบบเพื่อเฝ้าสังเกตและตรวจจับค่าสภาพแวดล้อมห้องเซิร์ฟเวอร์ได้ และ (2) ระบบที่พัฒนาเป็นระบบต้นแบบที่มีฟังก์ชันการทำงาน 9 ฟังก์ชัน ได้แก่ ฟังก์ชันตรวจจับไฟฟ้าดับ ฟังก์ชันตรวจจับน้ำแอร์หยด ฟังก์ชันตรวจจับอุณหภูมิของห้อง ฟังก์ชันตรวจจับความชื้นของห้อง ฟังก์ชันตรวจจับอุณหภูมิของโฮสต์ ฟังก์ชันตรวจจับความชื้นของโฮสต์ ฟังก์ชันควบคุมสำหรับพัดลม ฟังก์ชันแจ้งเตือน และฟังก์ชันรายงานสภาพแวดล้อมของห้อง ซึ่งการทดสอบประสิทธิภาพการทำงานของทั้ง 9 ฟังก์ชัน ระบบสามารถทำงานได้อย่างมีประสิทธิภาพตามคำสั่งที่ได้โปรแกรมไว้เป็นอย่างดี

**คำสำคัญ** อินเทอร์เน็ตประสานสรรพสิ่ง อุปกรณ์ ESP8266 NETPIE แพลตฟอร์ม

Independent Study title: “Monitoring a server room environment with the Internet of Everything using ESP8266 and NETPIE”

Author: “Miss. KHWANTHIP ITTHIPHOL”; ID: “2649600265”;

Degree: Masrter of Science

Independent Study Advisor: Dr.Tejtasin Phiasai; Academic year: 2023

# Abstract

The objectives of this independent study were (1) to apply the Internet of Everything to detect an environment of a server room with sensor devices co-operated with the ESP8266 device and the NETPIE platform, and (2) to test the performance of developing a prototype system that propose.

The concept of the Internet of Everything was used to develop the system. The sensor devices were used to detect the temperature and humidity of the host and the server room and to check for dripping water in the air conditioner and the relay devices were applied to detect power outages and used to control the operation of cooling fan devices from the detected temperature data. Each device was controlled by ESP8266 and each detected data was presented as a dashboard report by the NETPIE platform for an administrator to observe. Temperature values of the host and the server room and the value of dripping water from the air conditioner that had values above the specified limit would be sent as a message via the LINE notification service.

The study results found that (1) the application of the Internet to Everything that included the sensor devices, the relay devices, the ESP8266 device, the detection data, the detection processes, the NETPIE platform, and the system administrator could work together as the system to monitor and detect the server room environment and (2) the developed system was a prototype system and had the 9 operating functions that included the detection function of power outages, the detection function of the dripping water of air conditioner, the detection function of room temperature, the detection function of room humidity, the detection function of host temperature, the detection function of host humidity, the control function for cooling fan, the notification function, and the report function of room environment. For the performance testing of all functions, the developed system could work efficiently according to well-programmed instructions.

**Keywords :** Internet of Everything, ESP8266 device, NETPIE platform

# กิตติกรรมประกาศ

การค้นคว้าอิสระครั้งนี้สำเร็จลุล่วงด้วยดี โดยได้รับความกรุณาอย่างยิ่งจากอาจารย์ ดร. เตชค์ฐสิณป์ เพียซ้าย ที่ช่วยแนะนำให้คำปรึกษาตลอดจนการตรวจแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่และติดตามด้วยดีตลอดมา จนการศึกษาค้นคว้าประสบความสำเร็จสมบูรณ์ได้ ขอขอบพระคุณอาจารย์ทุกท่านที่ให้ความรู้และประสบการณ์ใหม่ ๆ เพื่อนำไปเป็นแนวทางในการศึกษาครั้งต่อไป

นางสาวขวัญทิพย์ อิทธิผล

# สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฎ

สารบัญตาราง (ต่อ) ..... ฎ

สารบัญตาราง (ต่อ) ..... ฐ

สารบัญรูปภาพ ..... ฑ

สารบัญรูปภาพ (ต่อ) ..... ฒ

บทที่ 1 บทนำ ..... 1

**1.1 ความเป็นมาและความสำคัญของปัญหา** ..... 1

**1.2 วัตถุประสงค์ในการศึกษา** ..... 2

**1.3 กรอบความคิดทางทฤษฎี** ..... 2

**1.4 ขอบเขตของการศึกษา** ..... 2

**1.5 นิยามศัพท์เฉพาะ** ..... 3

**1.6 ประโยชน์ที่คาดว่าจะได้รับ** ..... 4

บทที่ 2 วรรณกรรมที่เกี่ยวข้อง ..... 5

**2.1 Internet of Things, Internet of Everything, NETPIE** ..... 5

**2.1.1 Internet of Things (IoT)** ..... 5

**2.1.2 Internet of Everything (IoE)** ..... 6

**2.1.3 NETPIE (Network Platform for Internet of Everything)** ..... 6

**2.2 ความสามารถของอุปกรณ์ ESP8266 เซนเซอร์ในการตรวจจับ และอุปกรณ์ในการควบคุมสัญญาณไฟฟ้า** ..... 8

**2.2.1 อุปกรณ์ NodeMCU ESP8266** ..... 8

**2.2.2 DHT22 Temperature and Humidity Sensor Module** ..... 10

**2.2.3 Rain/Water Detection Sensor Module** ..... 10

**2.2.4 Relay Module 5V** ..... 11

**2.3 ระบบการจัดการฐานข้อมูล NoSQL** ..... 12

**2.3.1 Document Based** ..... 12

**2.3.2 Key-Value** ..... 12

**2.3.3 Wide-Column SQL** ..... 12

**2.3.4 Graph** ..... 13

**2.4 เครื่องมือและภาษาคอมพิวเตอร์ในการพัฒนา** ..... 13

**2.4.1 Arduino IDE** ..... 13

**2.4.2 Line Application** ..... 14

**2.4.3 Line Application** ..... 14

**2.5 วรรณกรรมที่เกี่ยวข้อง** ..... 15

**2.5.1 ESP8266** ..... 15

**2.5.2 Arduino Microcontroller** ..... 17

**2.5.3 WeMos D1 Mini Node** ..... 18

บทที่ 3 วิธีดำเนินการศึกษา ..... 19

**3.1 การวิเคราะห์ และพัฒนาระบบการตรวจวัดค่าและระบบการแจ้งเตือนสภาพแวดล้อมห้องคอมพิวเตอร์แม่ข่าย**.....20

**3.2 การพัฒนาระบบการตรวจวัดค่าและระบบการแจ้งเตือน**.....21

**3.3.1 ฮาร์ดแวร์**.....21

**3.2.2 ซอฟต์แวร์**.....23

**3.3 การทดสอบและการประเมินประสิทธิภาพของการทำงานของการพัฒนาระบบต้นแบบจากการประยุกต์การเฝ้าสังเกตด้วยการตรวจวัดค่าสภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์ม**.....29

**3.3.1 ระบบการตรวจจับการทำงานของไฟฟ้า**.....29

**3.3.2 ระบบการตรวจวัดระดับน้ำจากถาดรองน้ำในเครื่องปรับอากาศ**.....29

**3.3.3 ระบบการตรวจวัดค่าอุณหภูมิและความชื้นภายในห้องเซิร์ฟเวอร์และระบบปรับระดับความแรงของพัดลมอัตโนมัติ**.....29

**3.3.4 ระบบการตรวจวัดค่าจากเครื่องคอมพิวเตอร์แม่ข่าย**.....30

**3.3.5 การแสดงผลบนหน้า Dashboard**.....30

บทที่ 4 ผลการวิเคราะห์ข้อมูล.....31

**4.1 การพัฒนาระบบต้นแบบจากการประยุกต์การเฝ้าสังเกตด้วยการตรวจวัดค่าสภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์ม**.....31

**4.2 การประเมินประสิทธิภาพระบบการตรวจวัดค่าและระบบการแจ้งผ่านแอปพลิเคชันไลน์**.....33

บทที่ 5 สรุปการศึกษาค้นคว้าอิสระ อภิปรายผล และข้อเสนอแนะ.....67

**5.1 สรุปการศึกษาค้นคว้าอิสระ**.....67

**5.2 ข้อเสนอแนะ** ..... 68

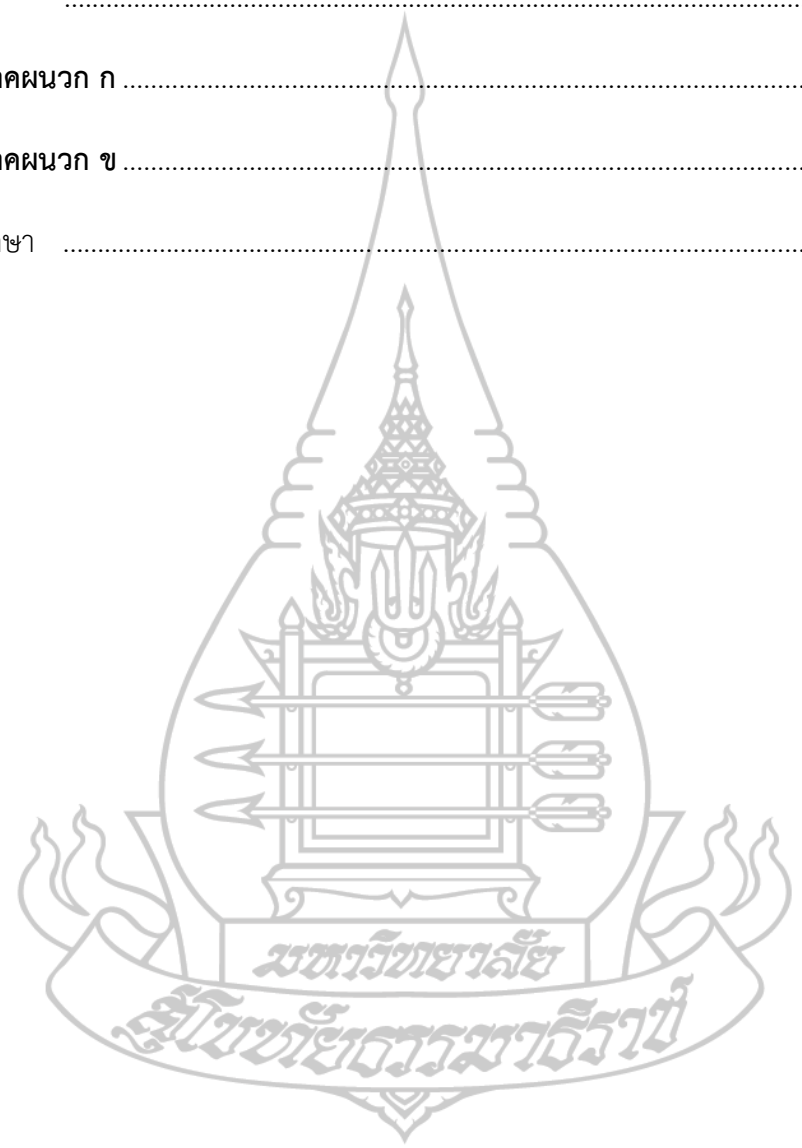
บรรณานุกรม ..... 73

ภาคผนวก ..... 75

**ภาคผนวก ก** ..... 76

**ภาคผนวก ข** ..... 84

ประวัติผู้ศึกษา ..... 86



# สารบัญตาราง

หน้า

ตารางที่ 4.1 เงื่อนไขการทดสอบระบบที่พัฒนา ..... 33
ตารางที่ 4.2 สถานการณ์ทดลองที่ 1ไฟฟ้าและเครื่องปรับอากาศทำงานปกติ ..... 35
ตารางที่ 4.3 การตรวจวัดอุณหภูมิไฟฟ้าและเครื่องปรับอากาศทำงานปกติ ..... 36
ตารางที่ 4.4 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าและเครื่องปรับอากาศทำงานปกติ ..... 37
ตารางที่ 4.5 การทำงานของพัดลมขณะไฟฟ้าและเครื่องปรับอากาศทำงานปกติ ..... 37
ตารางที่ 4.6 สถานการณ์ทดลองที่ 2 ไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน ..... 37
ตารางที่ 4.7 การตรวจวัดอุณหภูมิไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน ..... 38
ตารางที่ 4.8 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน 39
ตารางที่ 4.9 การทำงานของพัดลมขณะไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน ..... 40
ตารางที่ 4.10 สถานการณ์ทดลองที่ 3 ไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุด 40
ตารางที่ 4.11 การตรวจวัดอุณหภูมิไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุด..... 41
ตารางที่ 4.12 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับเป็นเวลา 0 - 10 นาที และ ..... 42
ตารางที่ 4.13 การทำงานของพัดลมขณะไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุด ..... 43
ตารางที่ 4.14 สถานการณ์ทดลองที่ 4 ไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุด ..... 43
ตารางที่ 4.15 การตรวจวัดอุณหภูมิไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุด ... 44
ตารางที่ 4.16 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับเป็นเวลา 10 - 30 นาที และ..... 45
ตารางที่ 4.17 การทำงานของพัดลมขณะไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศ .. 46

# สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 4.18 สถานการณ์ทดลองที่ 5 ไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุด ..... 46

ตารางที่ 4.19 การตรวจวัดอุณหภูมิไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุด ... 47

ตารางที่ 4.20 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับเป็นเวลา 30 - 50 นาที และ..... 48

ตารางที่ 4.21 การทำงานของพัดลมขณะไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศ .. 49

ตารางที่ 4.22 สถานการณ์ทดลองที่ 6 ไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศ... 49

ตารางที่ 4.23 การตรวจวัดอุณหภูมิไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศ..... 50

ตารางที่ 4.24 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับเป็นเวลา 50 - 150 นาที และ..... 51

ตารางที่ 4.25 การทำงานของพัดลมขณะไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศ52

ตารางที่ 4.26 สถานการณ์ทดลองที่ 7 ไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน ..... 52

ตารางที่ 4.27 การตรวจวัดอุณหภูมิเมื่อไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน ..... 53

ตารางที่ 4.28 การแจ้งเตือนผ่านแอปพลิชันไลน์เมื่อไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศ .. 54

ตารางที่ 4.29 การทำงานของพัดลมเมื่อไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน..... 55

ตารางที่ 4.30 สถานการณ์ทดลองที่ 8 ไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติต่ำกว่า 25°C ..... 55

ตารางที่ 4.31 การตรวจวัดไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติต่ำกว่า 25°C และ ..... 56

ตารางที่ 4.32 การแจ้งเตือนผ่านแอปพลิชันไลน์เมื่อไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะ.... 57

ตารางที่ 4.33 ตารางการทำงานของพัดลมเมื่อไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติ .... 58

ตารางที่ 4.34 สถานการณ์ทดลองที่ 9 น้ำหยดจากเครื่องปรับอากาศ..... 58

ตารางที่ 4.35 การตรวจวัดระดับน้ำในสภาวะปกติ และเมื่อมีน้ำหยด..... 59

# สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 4.36 สรุปผลการทดลองการตรวจจับไฟฟ้าดับ อุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์... 60

ตารางที่ 4.37 สรุปผลการทดลองการตรวจจับอุณหภูมิและความชื้นของเครื่องแม่ข่าย ..... 64

ตารางที่ 4.38 สรุปผลการทดลองการตรวจจับน้ำหยดจากเครื่องปรับอากาศ ..... 66

# สารบัญรูปภาพ

หน้า

ภาพที่ 2.1 องค์ประกอบของอินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Everything : IoE)..... 6
ภาพที่ 2.2 คุณสมบัติหลัก ๆ ของ NETPIE 2020 Platform ..... 7
ภาพที่ 2.3 NodeMCU ESP8266 ..... 8
ภาพที่ 2.4 NodeMCU V3 ESP8266 และขาต่าง ๆ ..... 9
ภาพที่ 2.5 DHT22 Temperature and Humidity Sensor Module ..... 10
ภาพที่ 2.6 Rain/Water Detection Sensor Module ..... 10
ภาพที่ 2.7 Relay Module 5V 4 Channel ..... 11
ภาพที่ 2.8 สัญลักษณ์ในวงจรไฟฟ้าของรีเลย์ ..... 11
ภาพที่ 2.9 โปรแกรม Arduino IDE ..... 13
ภาพที่ 2.10 การแจ้งเตือนในส่วนของ Line Notify ..... 14
ภาพที่ 2.11 ตัวอย่างการทำงานของ MQTT ..... 15
ภาพที่ 3.1 แสดงขั้นตอนการดำเนินงานในการพัฒนาระบบการตรวจวัดสภาพแวดล้อมและ ..19
ภาพที่ 3.2 แสดงขั้นตอนพัฒนาระบบการตรวจวัดสภาพแวดล้อมและระบบการแจ้งเตือนของห้อง. 20
ภาพที่ 3.3 บอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 ..... 21
ภาพที่ 3.4 Relay Module 5V 1 Channel ..... 21
ภาพที่ 3.5 Rain/Water Detection Sensor Module ..... 22
ภาพที่ 3.6 DHT22 Temperature and Humidity Sensor Module ..... 22
ภาพที่ 3.7 P2H1588A0 Relay Module 5V 4 Channel ..... 22
ภาพที่ 3.8 การประกอบวงจรชุดต้นแบบ ..... 23

# สารบัญรูปภาพ (ต่อ)

หน้า

ภาพที่ 3.9 ผังแสดงการทำงานของโมดูลวัดค่ากระแสไฟฟ้าภายในห้องคอมพิวเตอร์แม่ข่าย ..... 24

ภาพที่ 3.10 ผังแสดงการทำงานของโมดูลวัดค่าระดับน้ำจากถาดรองน้ำในเครื่องปรับอากาศ ..... 24

ภาพที่ 3.11 ผังแสดงการทำงานของโมดูลวัดค่าอุณหภูมิและความชื้นภายในห้องเซิร์ฟเวอร์ ..... 25

ภาพที่ 3.12 ผังแสดงการทำงานของโมดูลวัดค่าอุณหภูมิและความชื้นของเครื่องคอมพิวเตอร์แม่ข่าย ..... 26

ภาพที่ 3.13 แสดงการแจ้งเตือนผ่านแอปพลิเคชันไลน์ ..... 27

ภาพที่ 3.14 แสดงผลบนหน้า Dashboard Netpie ..... 27

ภาพที่ 3.15 แสดงการเขียนโปรแกรมควบคุมการทำงานของระบบ ..... 28

ภาพที่ 4.1 ภาพแสดงการทำงานของอุปกรณ์ต้นแบบระบบตรวจวัดสภาพแวดล้อมที่จัดทำขึ้นด้วยอินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Everything : IoE) 32

ภาพที่ 4.2 ภาพอุปกรณ์ต้นแบบระบบตรวจวัดสภาพแวดล้อมที่จัดทำขึ้นด้วยอินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Everything : IoE) ..... 32

มหาวิทยาลัย
สุโขทัยธรรมาธิราช

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

การดำเนินงานของภาคธุรกิจและโรงงานอุตสาหกรรมในปัจจุบันจะต้องใช้ข้อมูลจำนวนมากและการติดต่อสื่อสารที่รวดเร็วเพื่อเพิ่มศักยภาพในการแข่งขันให้กับองค์กร การจัดเก็บข้อมูลและอุปกรณ์ต่าง ๆ ในการติดต่อสื่อสารจึงจำเป็นจะต้องมีระบบการบริหารการจัดการสำหรับห้องคอมพิวเตอร์แม่ข่าย ที่เป็นศูนย์บริการข้อมูลและให้บริการด้านการสื่อสารที่สำคัญยิ่ง รวมไปถึงสภาพแวดล้อมภายในห้องคอมพิวเตอร์แม่ข่ายหรือห้องเซิร์ฟเวอร์ก็มีความสำคัญเช่นกัน สำหรับห้องเซิร์ฟเวอร์ที่มีต้นทุนแบบประหยัดกรณีเกิดไฟฟ้าดับทำให้อุณหภูมิภายในห้องสูงส่งผลกระทบกับอุปกรณ์ต่าง ๆ ภายในห้องเสียหายการดำเนินงานหยุดชะงัก หรือมีน้ำหยดจากการใช้ระบบเครื่องปรับอากาศแบบแยกส่วน (split type) ที่ใช้ในการทำความเย็นภายในห้อง อาจทำให้โดนอุปกรณ์หรืออุณหภูมิในห้องสูงขึ้นเนื่องจากเครื่องปรับอากาศทำงานไม่เต็มที่ ดังนั้นจึงต้องมีเฝ้าสังเกตสภาพแวดล้อมห้องเซิร์ฟเวอร์ที่มีความเสี่ยงและส่งผลกระทบไปยังอุปกรณ์ที่อยู่ในห้องเซิร์ฟเวอร์ทำให้เกิดเหตุขัดข้องและหยุดการทำงานได้

ในปัจจุบันเทคโนโลยีอินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Everything : IoE) เข้ามามีบทบาทในภาคธุรกิจเป็นอย่างมาก ไม่ได้เป็นเพียงแค่เชื่อมต่ออุปกรณ์ต่าง ๆ เพื่อช่วยในการรับส่งข้อมูลผ่านอินเทอร์เน็ตให้ทำงานได้แบบอัตโนมัติ แต่ยังมีการเชื่อมต่อระบบและข้อมูลต่าง ๆ ในทุก ๆ สิ่ง ประกอบไปด้วย คน อุปกรณ์ ข้อมูล กระบวนการในการรวบรวม ประมวลผลและนำมาวิเคราะห์เพื่อนำไปสู่การปรับปรุงให้ดียิ่งขึ้น โดยเฉพาะการเฝ้าสังเกตสภาพแวดล้อมของห้องเซิร์ฟเวอร์และแจ้งเตือนเพื่อป้องกันเหตุการณ์ไม่พึงประสงค์ที่จะเกิดขึ้น ยิ่งไปกว่านั้นระบบที่ใช้ในการตรวจวัดสำเร็จรูปที่มีขายในปัจจุบันมีราคาค่อนข้างสูงเมื่อเปรียบเทียบกับงบประมาณของธุรกิจที่มีขนาดเล็กจะไม่สามารถเข้าถึงระบบที่มีราคาสูงเหล่านั้นได้ ดังนั้นงานวิจัยในครั้งนี้ได้นำเอาอุปกรณ์ควบคุมขนาดเล็ก ESP8266 พร้อมทั้งอุปกรณ์เซ็นเซอร์ต่าง ๆ เข้ามาใช้เพื่อเฝ้าสังเกตสภาพแวดล้อมของห้องเซิร์ฟเวอร์ในกรณีเกิดไฟฟ้าดับและมีการหยดของน้ำจากเครื่องปรับอากาศได้แก่ การตรวจจับไฟฟ้าดับ, การตรวจวัดระดับน้ำของเครื่องปรับอากาศ, การตรวจวัดอุณหภูมิและความชื้นภายในห้อง

คอมพิวเตอร์แม่ข่าย และการทำงานของอุปกรณ์ตรวจวัดทุกตัวจะทำการแจ้งเตือนเมื่อค่าที่ตรวจวัดได้เกินกว่าที่กำหนดผ่านไลน์แอปพลิเคชั่น และแสดงผลเป็นกราฟผ่าน NETPIE แพลตฟอร์ม

## 1.2 วัตถุประสงค์ในการศึกษา

1. **เพื่อประยุกต์การเฝ้าสังเกต** ด้วยการตรวจวัดค่าสภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์ม
2. **เพื่อทดสอบประสิทธิภาพการทำงาน** ของการพัฒนาระบบต้นแบบที่เสนอ

## 1.3 กรอบความคิดทางทฤษฎี

ระบบการแจ้งเตือนสำหรับห้องห้องเซิร์ฟเวอร์ ด้วยการนำอินเทอร์เน็ตประสานสรรพสิ่งมาประยุกต์ใช้ ผู้วิจัยได้ทำการศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องเพื่อใช้เป็นแนวทางในการพัฒนาระบบ ดังนี้

1. Internet of Things, Internet of Everything, NETPIE
2. ความสามารถของอุปกรณ์ ESP8266 และ เซ็นเซอร์ในการตรวจจับ
3. ระบบการจัดการฐานข้อมูล NoSQL
4. เครื่องมือและภาษาคอมพิวเตอร์ในการพัฒนา
5. ระบบการตรวจจับและแจ้งเตือน
6. งานวิจัยที่เกี่ยวข้อง

## 1.4 ขอบเขตของการศึกษา

1. **พัฒนาระบบตรวจวัดด้วยอุปกรณ์ควบคุมขนาดเล็ก ESP8266** สำหรับควบคุมอุปกรณ์การตรวจจับไฟฟ้าดับ ความชื้น เซนเซอร์การตรวจวัดระดับน้ำของเครื่องปรับอากาศ เซนเซอร์ตรวจวัดอุณหภูมิและความชื้นพร้อมการควบคุมระบบพัดลมอัตโนมัติ และเซนเซอร์ตรวจวัดอุณหภูมิและความชื้นเครื่องแม่ข่าย

***1.1 ส่วนฮาร์แวร์*** การติดตั้งอุปกรณ์ NodeMCU ESP8266, P2H1588A0 Relay Module 5 V, Rain/Water Detection Sensor Module และ DHT11 Temperature and Humidity Sensor Module ต่อสายไฟในวงจรประกอบออกมาเป็นชุดต้นแบบ

***1.2 ส่วนซอฟต์แวร์*** Arduino IDE สำหรับเขียนและอัปโหลดโปรแกรมไปยังบอร์ด NodeMCU ESP8266 เพื่อควบคุมเซนเซอร์ที่เชื่อมต่อกับบอร์ดให้ทำงานตามคำสั่ง

2. **สร้างการแจ้งเตือนอุณหภูมิและความชื้น** ผ่าน Line Notification
3. **แสดงผลการตรวจวัดของเซ็นเซอร์ต่าง ๆ** เพื่อเฝ้าระวังการทำงานของเครื่องคอมพิวเตอร์แม่ข่ายบน NETPIE แพลตฟอร์ม

## 1.5 นิยามศัพท์เฉพาะ

1. **อินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Everything : IoE)** คือ การเชื่อมโยงระหว่างมนุษย์ กระบวนการ ข้อมูล และสิ่งต่าง ๆ ให้เชื่อมต่อถึงกันอย่างชาญฉลาด มีการพัฒนาระบบเชื่อมต่อกับอุปกรณ์ต่าง ๆ เกิดขึ้นเป็นไปอย่างต่อเนื่องและส่งผลให้มีข้อมูลจำนวนมากเพิ่มขึ้น เครือข่ายรวมของอุปกรณ์ที่สามารถเชื่อมต่อถึงกันและเทคโนโลยีที่ให้ความสะดวกในการสื่อสารระหว่างอุปกรณ์กับระบบคลาวด์และระหว่างอุปกรณ์ด้วยกันเอง เกิดขึ้นจากชิปคอมพิวเตอร์และการสื่อสารโทรคมนาคมที่มีแบนด์วิดท์สูง ทำให้อุปกรณ์ต่าง ๆ ที่ใช้เซนเซอร์เพื่อรวบรวมข้อมูลและตอบสนองต่อผู้ใช้งานได้อย่างชาญฉลาด
2. **คอมพิวเตอร์แม่ข่าย** คือ เครื่องคอมพิวเตอร์ที่ให้บริการและการจัดการระบบเครือข่ายให้กับเครื่องคอมพิวเตอร์ที่เชื่อมต่อมาในระบบเครือข่ายสามารถใช้งานและแบ่งปันทรัพยากรต่าง ๆ ได้ และยังทำหน้าที่บริหารจัดการความปลอดภัยของข้อมูล ดังนั้นคอมพิวเตอร์แม่ข่ายจะต้องมีความเสถียรสูง เพื่อรองรับผู้ใช้เป็นจำนวนมาก
3. **สภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายหรือห้องเซิร์ฟเวอร์** คือ สิ่งต่าง ๆ ที่อยู่ภายในห้องคอมพิวเตอร์แม่ข่าย เช่น อุปกรณ์คอมพิวเตอร์ อุปกรณ์เก็บข้อมูล ระบบปฏิบัติการ ระบบอินเทอร์เน็ต ระบบการรักษาความปลอดภัย ระบบไฟฟ้า ระบบระบายความร้อน ระบบการสำรองข้อมูล เป็นต้น
4. **การควบคุมอุณหภูมิ** คือ กระบวนการในการควบคุมปรับค่าอุณหภูมิให้อยู่ในสภาพแวดล้อมที่เหมาะสม ตามเงื่อนไขที่กำหนดให้อยู่ในขอบเขตที่เหมาะสมกับการทำงาน เพื่อให้การทำงานเป็นไปได้ด้วยดีและเกิดประสิทธิภาพสูงสุด

## 1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. **ได้องค์ความรู้** การออกแบบของระบบจากการประยุกต์การเฝ้าสังเกตด้วยการตรวจวัดค่าสภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์ม
2. **ได้ต้นแบบของระบบการแจ้งเตือน** สำหรับห้องคอมพิวเตอร์แม่ข่าย และสามารถนำไปประยุกต์ใช้ในสภาพแวดล้อมอื่น ๆ ได้
3. ได้ปัจจัยที่จะมีผลกระทบต่อระบบการทำงาน ของระบบคอมพิวเตอร์แม่ข่ายเพื่อนำไปหาแนวทางป้องกันและแก้ไขปัญหาก่อนที่จะเกิดการสูญเสีย

มหาวิทยาลัย
สุโขทัยธรรมาธิราช

# บทที่ 2
# วรรณกรรมที่เกี่ยวข้อง

การศึกษาและการพัฒนาระบบต้นแบบจากการประยุกต์การเฝ้าสังเกตด้วยการตรวจวัดค่าสภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์มในครั้งนี้ได้ทำการศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องต่าง ๆ เพื่อเป็นแนวทางในการออกแบบและพัฒนาระบบ ดังต่อไปนี้

1. Internet of Things, Internet of Everything, NETPIE
2. ความสามารถของอุปกรณ์ ESP8266 และ เซ็นเซอร์ในการตรวจจับ
3. ระบบการจัดการฐานข้อมูล NoSQL
4. เครื่องมือและภาษาคอมพิวเตอร์ในการพัฒนา
5. ระบบการตรวจจับและแจ้งเตือน
6. งานวิจัยที่เกี่ยวข้อง

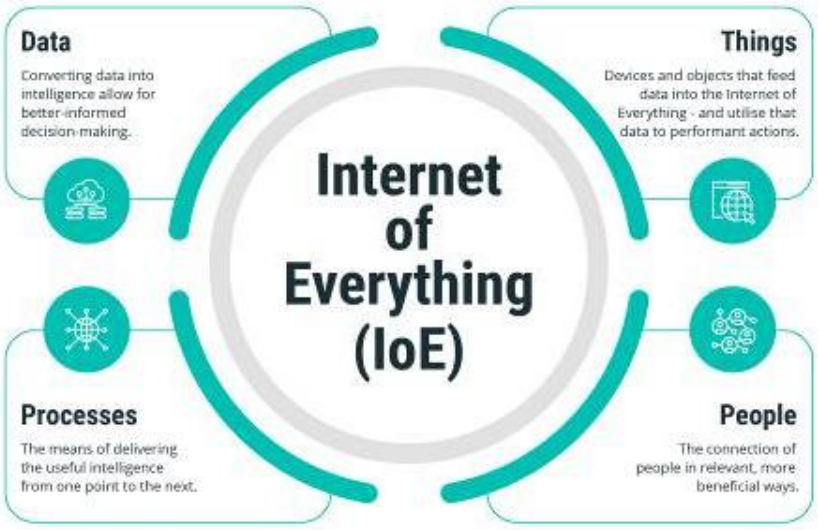
## 2.1 Internet of Things, Internet of Everything, NETPIE

### 2.1.1 Internet of Things (IoT)

คือ อุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ที่สามารถเชื่อมโยงหรือส่งข้อมูลถึงกันได้ด้วยอินเทอร์เน็ต และสื่อสารกันเพื่อให้เกิดการทำงานร่วมกันได้อย่างมีประสิทธิภาพสูงสุดประกอบด้วยอุปกรณ์ IoT โดยมีคอมพิวเตอร์ขนาดเล็กไว้ประมวลผลข้อมูล และมีระบบสั่งการไปยังตัวอุปกรณ์อีกตัวเพื่อสั่งให้ทำงานตามที่ได้รับคำสั่ง อุปกรณ์เกตเวย์ เป็นตัวกลางในการส่งผ่านคำสั่งของอุปกรณ์ IoT ไปยังอุปกรณ์เป้าหมายที่ต้องรับคำสั่งเพื่อทำตามคำสั่งที่ได้รับ ช่วยอำนวยความสะดวกในการสื่อสารกับระบบข้อมูลส่วนกลาง เครื่องบริการ Server/Broker ทำหน้าที่เป็นจุดศูนย์กลางในการประมวลผลสามารถคำนวณคำสั่งที่ซับซ้อนได้ จะรับคำสั่งจากอุปกรณ์ IoT หลังจากนั้นก็จะประมวลผลเพื่อให้ได้คำสั่งอย่างง่ายส่งไปยังอุปกรณ์ IoT ที่เป็นผู้รับ และอุปกรณ์ฝั่งผู้ใช้งาน เป็นส่วนของอุปกรณ์ที่ใช้แสดงผลสถานะอุปกรณ์ IoT ทั้งหมดที่อยู่ในเครือข่าย โดยปกติระบบ IoT จะเป็นการเชื่อมต่อเพื่อสื่อสารกันระหว่างอุปกรณ์อิเล็กทรอนิกส์ ผู้ใช้งานจะต้องมีอุปกรณ์เพื่อใช้ในการเชื่อมต่อกับระบบเพื่อดูสถานะของการทำงาน

### 2.1.2 Internet of Everything (IoE)

คือ การเชื่อมโยงระหว่างมนุษย์ กระบวนการ ข้อมูล และสิ่งต่าง ๆ ให้เชื่อมต่อถึงกันอย่างชาญฉลาด มีการพัฒนาอุปกรณ์ต่าง ๆ มากมาย รวมถึงการเติบโตของข้อมูลก็เพิ่มมากขึ้น ซึ่งข้อมูลเหล่านั้นจะนำไปสู่การวิเคราะห์เพื่อนำไปเพิ่มประสิทธิภาพของกระบวนการทำงานให้กับองค์กร และอุตสาหกรรมต่าง ๆ เป็นอย่างมาก องค์ประกอบหลักของ IoE มีดังนี้ 1) มนุษย์ เป็นการเชื่อมต่อโดยมนุษย์จะได้รับข้อมูลเพื่อนำข้อมูลเหล่านั้นนำไปทำการวิเคราะห์และไปใช้ประโยชน์ในด้านต่าง ๆ ต่อไป 2) กระบวนการ เป็นกระบวนการทำงานเพื่อนำข้อมูลจากอุปกรณ์ที่ทำหน้าที่ส่งข้อมูลไปยังอุปกรณ์ที่รับคำสั่งให้ทำงาน 3) ข้อมูล เป็นการนำข้อมูลมาทำการวิเคราะห์และช่วยให้สามารถตัดสินใจได้ดียิ่งขึ้น 4) สิ่งต่าง ๆ เป็นอุปกรณ์ที่ทำการเชื่อมต่อเพื่อนำข้อมูลเข้าสู่อินเทอร์เน็ตของสิ่งต่าง ๆ และใช้ข้อมูลนั้นในการดำเนินการ




ภาพที่ 2.1 องค์ประกอบของอินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Everything : IoE)

ที่มา : Emerline (2023) The Internet of Everything and the Internet of Things.

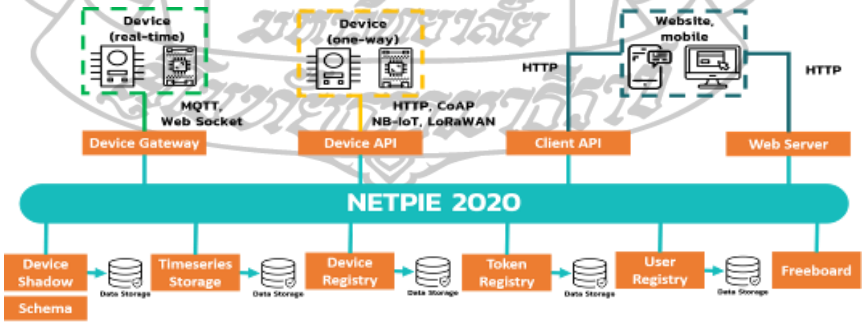
### 2.1.3 NETPIE (Network Platform for Internet of Everything)

คือ cloud platform ที่ถูกออกแบบและพัฒนาขึ้น โดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) สำหรับให้บริการเชื่อมต่อสื่อสารในรูปแบบ IoT เพื่อให้เกิดการสื่อสารระหว่างอุปกรณ์ในเครือข่าย IoT ให้มีประโยชน์ต่อนักพัฒนาและอุตสาหกรรมไทย ช่วยให้อุปกรณ์สามารถสื่อสารกันได้โดยผู้พัฒนาไม่ต้องกังวลว่าอุปกรณ์จะอยู่ที่ไหน ทั้งในส่วนประกอบที่เป็น

อุปกรณ์ หรือส่วนการทำงานของระบบ เพียงแค่นำ NETPIE Library ไปติดตั้งในตัวอุปกรณ์ NETPIE ก็จะทำหน้าที่ในการดูแลการเชื่อมต่อให้ทั้งหมด ไม่ว่าอุปกรณ์จะอยู่ในเครือข่ายชนิดใด ในลักษณะใด หรืออยู่ในสถานที่ไหนก็ตาม ผู้พัฒนาก็สามารถการออกแบบระบบการเข้าถึงอุปกรณ์จากระยะไกลได้ NETPIE ใช้งานง่ายโดยการออกแบบให้อุปกรณ์ถูกค้นพบและเข้าสู่บริการโดยอัตโนมัติ ถูกออกแบบมาให้ผู้พัฒนาสามารถออกแบบได้เองทั้งหมด เช่น ต้องการให้อุปกรณ์ตัวไหนมีสิทธิสื่อสารกับอุปกรณ์ตัวไหน หรือสามารถกำหนดได้ว่าอุปกรณ์ตัวไหนมีสิทธิอ่านหรือเขียนข้อมูลได้ NETPIE มีสถาปัตยกรรมเป็น Cloud ในทุก ๆ ระดับของระบบ ทำให้มีความยืดหยุ่น และความคล่องตัวสูงในการขยายตัว นอกจากนี้โมดูลต่าง ๆ ยังถูกออกแบบให้ทำงานแยกกันเพื่อให้เกิดสภาวะความยืดหยุ่นในการพัฒนาระบบได้ง่าย นอกจากนี้ทางเนคเทคได้เปิด NETPIE Library ในรูปแบบ open-source ให้กับนักพัฒนาเพื่อที่จะสามารถนำไปปรับปรุงให้ตรงกับความต้องการในการใช้งานได้

NETPIE 2020 เป็นแพลตฟอร์มที่ช่วยให้อุปกรณ์ต่าง ๆ สามารถสื่อสารกันได้ มีการรับ-ส่งข้อมูลระหว่างอุปกรณ์แบบ real-time ทำให้ผู้ใช้งานทราบถึงข้อมูลของอุปกรณ์ ณ เวลานั้น ๆ ไม่ว่าผู้ใช้งานจะอยู่ที่ไหนก็ตาม และยังสามารถรองรับการเชื่อมต่ออุปกรณ์ IOT ได้จำนวนมาก

NETPIE 2020 Platform จะมีการแสดงค่าของข้อมูลจากเซนเซอร์ หรืออุปกรณ์แบบ Real-time (Monitoring) ในการควบคุมการทำงานของอุปกรณ์ต่าง ๆ ผ่าน Cloud Platform (Controlling) และทำการเก็บค่าของข้อมูลที่ได้จากเซนเซอร์หรือของอุปกรณ์ (Data Storage) มีระบบการแจ้งเตือนความผิดปกติของเซนเซอร์หรืออุปกรณ์ที่กำหนด ไปยังผู้ดูแลระบบ ทำการแสดงผลและควบคุมการทำงานของอุปกรณ์ผ่าน Dashboard (Dashboard for monitor & control) ดังภาพที่ 2.10




ภาพที่ 2.2 คุณสมบัติหลัก ๆ ของ NETPIE 2020 Platform

ที่มา : Netpie (2023) NETPIE 2020 Overview

## 2.2 ความสามารถของอุปกรณ์ ESP8266 เซนเซอร์ในการตรวจจับ และอุปกรณ์ในการควบคุมสัญญาณไฟฟ้า

### 2.2.1 อุปกรณ์ NodeMCU ESP8266

ESP8266 เป็นชิปของโมดูล ESP8266 สำหรับการติดต่อสื่สารบนมาตรฐาน Wi-Fi ตัวชิปมีราคาถูก ผลิตโดย Espressif Systems ใช้กันอย่างแพร่หลายในอุปกรณ์ IoT โมดูล ESP8266 มีหลายรุ่น และมีการพัฒนาออกมาเรื่อย ๆ เช่น ESP-01 , ESP-03 , ESP-07 ปัจจุบันเป็น ESP-12E ช่วงแรงดันไฟฟ้าของ ESP8266 คือ 3V ถึง 3.6V มาพร้อมกับตัวควบคุมแรงดันไฟฟ้า LDO (Low-Dropout Regulator) เพื่อควบคุมให้แรงดันคงที่ 3.3V ซึ่งสามารถจ่ายกระแสไฟได้สูงสุดถึง 600 mA เนื่องจาก ESP8266 กินกระแสไฟมากที่สุดคือ 80 mA ในระหว่างส่งสัญญาณเอาท์พุทออกไป พิน 3V3 ใช้เพื่อจ่ายพลังงานให้กับอุปกรณ์ภายนอกได้อีกด้วย การจ่ายไฟให้ NodeMCU ESP8266 โดยใช้ช่องเสีย Micro USB หรือมีแหล่งจ่ายไปแรงดัน 5V ก็สามารถป้อนที่พิน VIN เพื่อเป็นแหล่งจ่ายไฟให้ ESP8266 และอุปกรณ์ต่อพ่วงได้โดยตรง สำหรับสวิตช์ออนบอร์ดและไฟแสดงสถานะ LED บนบอร์ดมีสองปุ่ม ปุ่ม RST คือ ปุ่มเพื่อรีเซ็ตชิป ESP8266 ส่วนอีกปุ่มเป็นปุ่ม FLASH อัปโหลดที่ใช้ในขณะทำการอัปโหลดเฟิร์มแวร์ สามารถเขียนโปรแกรมสั่งงานลงไปในชิปได้โดยใช้ Arduino IDE เขียนโปรแกรมด้วยภาษา C/C++ และ ภาษา Lau ทำให้การเขียนโปรแกรมใช้งานเป็นไปได้ง่าย สามารถติดต่อกับอุปกรณ์และเซนเซอร์ต่าง ๆ ได้

ภาพที่ 2.3 NodeMCU ESP8266

บอร์ด NodeMCU V3 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้เฟิร์มแวร์ Lua ได้ถูกออกแบบมาให้ใช้กับโมดูล ESP8266 ESP-12E Wi-Fi ทำให้เหมาะที่จะนำไปพัฒนาและใช้งานในลักษณะของ IoT Applications ข้อดีของตัวบอร์ดนี้คือ มีราคาที่ถูกและเขียนโปรแกรมควบคุมผ่านโปรแกรม Arduino IDE ง่ายต่อการพัฒนาโปรแกรมในการใช้งานร่วมกับเซนเซอร์ต่าง ๆ

คุณสมบัติของ NodeMCU ESP8266 V3

(1) ไมโครคอนโทรลเลอร์ ESP8266 ESP-12E

(2) ตัวประมวลผล Tensilica Xtensa Diamon 32-bit

(3) ใช้กระแสไฟประมาณ 70mA

(4) ขาอินเตอร์เฟสใช้แรงดัน 3.3V

(5) มีสายอากาศสำหรับ Wi-Fi อยู่ภายในบอร์ด

(6) มาตรฐานการติดต่อสื่อสาร 802.11 b/g/n

(7) ความถี่ WiFi ที่ใช้ : 2.4GHz สนับสนุนระบบรักษาความปลอดภัยแบบ WPA / WPA2

(8) ขาติดต่อสื่อสารแบบดิจิตอล จำนวน 9 ขา สามารถกำหนดเป็น input หรือ output ก็ได้ (D0-D8 สามารถใช้เป็นขา GPIO, PWM, IIC เป็นต้น) โดยแต่ละขา สามารถรับและจ่ายกระแสได้สูงสุด 12mA

(9) ขาติดต่อสื่อสารแบบอนาล็อก จำนวน 1 ขา ซึ่งจะสามารถกำหนดเป็น input ได้เท่านั้น มีความละเอียด 10 บิต (0-1023)

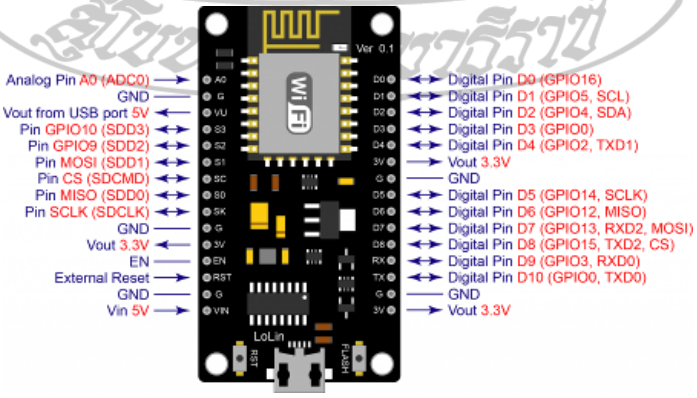
(10) สามารถกำหนด Transfer Rate ได้ตั้งแต่ 110-460,800bps

(11) สนับสนุนการติดต่อสื่อสารแบบ UART / GPIO

(12) ขนาดของ Flash Memory คือ 16 เมกกะไบท์ (โดยถูกจองด้วยโปรแกรม bootloader เป็นจำนวน 0.5 กิโลไบท์)

(13) ขนาดของ SRAM คือ 64 กิโลไบท์ และขนาดของ EEPROM คือ 512 ไบท์

(14) ขนาดบอร์ด ยาว 58 มิลลิเมตร กว้าง 31 มิลลิเมตร

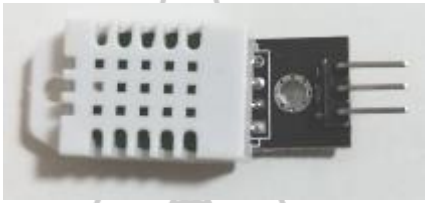



ภาพที่ 2.4 NodeMCU V3 ESP8266 และขาต่าง ๆ

ที่มา : Future Kit (2023) ทำความรู้จักกับบอร์ด NodeMCU

### 2.2.2 DHT22 Temperature and Humidity Sensor Module

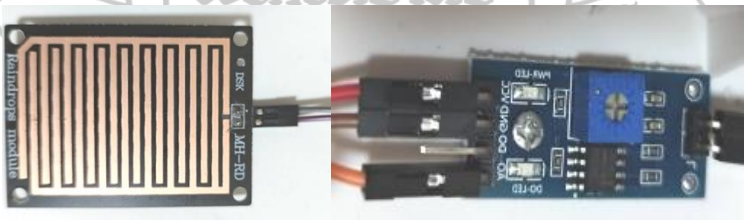
ความชื้นสัมพัทธ์ที่ละเอียดและช่วงการวัดที่สูงกว่า DHT11 โดยใช้ชิป DHT22 ให้ Output ออกมาเป็นแบบ Digital Output ทำงานด้วยไฟ 3.5 ถึง 5.5 V เหมาะสำหรับวัดอุณหภูมิในช่วง -40 ถึง 125 องศาเซลเซียส ความแม่นยำ + 0.5 องศาเซลเซียส และความชื้นสัมพัทธ์ ในช่วง 0 ถึง 100 เปอร์เซ็นต์ ความคลาดเคลื่อน 2-5%



ภาพที่ 2.5 DHT22 Temperature and Humidity Sensor Module

### 2.2.3 Rain/Water Detection Sensor Module

เป็นโมดูลวัดความชื้นในอากาศและระดับน้ำ การตรวจจับน้ำฝนช่วงการวัดอยู่ในช่วงระหว่าง 0.1 – 100 มิลลิเมตรต่อชั่วโมง แต่ถ้าวัดระดับน้ำในบ่อน้ำหรือภาชนะจะวัดตามเส้นผ่านศูนย์กลางของเซนเซอร์โดยวัดตามความสูงของน้ำ ถ้าปริมาณน้ำมาก ค่าความต้านทานที่ได้จะลดลง สามารถปรับค่าความไวในการตรวจวัดได้งาน



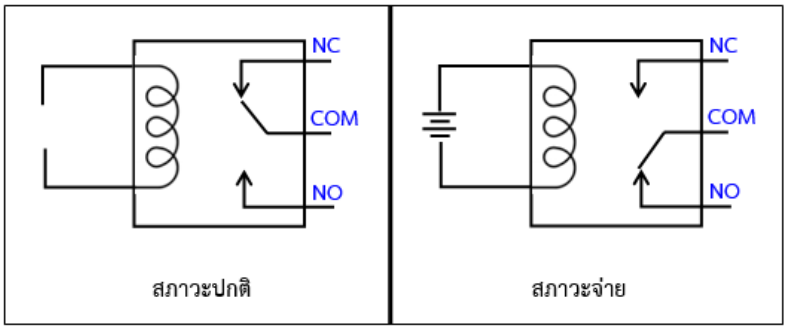

ภาพที่ 2.6 Rain/Water Detection Sensor Module

### 2.2.4 Relay Module 5V

เป็นโมดูลที่ใช้ควบคุมการทำงานของโหลดทางไฟฟ้าทั้งแรงดันไฟฟ้ากระแสตรง (DC) และไฟฟ้ากระแสสลับ (AC) โหลดสูงสุด คือ AC 250V/10A และ DC 30V/10A โดยใช้สัญญาณในการควบคุมการทำงานด้วยสัญญาณโลจิก TTL ทำงานด้วยสัญญาณแบบ Active Low, กระแสขับรีเลย์ (Drive Current) 15-20mA, มี LED แสดงสถานะ Relay

ภาพที่ 2.7 Relay Module 5V 4 Channel

หลักการทำงานของวงจรไฟฟ้าของรีเลย์ เมื่อรีเลย์ได้รับพลังงานไฟฟ้าหน้าสัมผัส NO จะปิดและหน้าสัมผัส NC จะเปิดขึ้น จะปล่อยให้กระแสไฟไหลผ่านวงจร เมื่อรีเลย์ไม่ได้จ่ายไฟหน้าสัมผัส NO จะเปิดขึ้นและหน้าสัมผัส NC จะปิด ทำให้กระแสไฟฟ้าไหลผ่านไม่ได้ ดังภาพที่ 2.6




ภาพที่ 2.8 สัญลักษณ์ในวงจรไฟฟ้าของรีเลย์

## 2.3 ระบบการจัดการฐานข้อมูล NoSQL

ฐานข้อมูล NoSQL ใช้โมเดลข้อมูลที่หลากหลายสำหรับการเข้าถึงและจัดการข้อมูล ฐานข้อมูลประเภทนี้มีการปรับปรุงประสิทธิภาพสำหรับแอพลิเคชันที่ต้องการใช้ข้อมูลปริมาณมาก มีการส่งข้อมูลด้วยความเร็ว และมีโมเดลข้อมูลที่ยืดหยุ่นทำให้ฐานข้อมูล NoSQL เหมาะสำหรับข้อมูลแบบกึ่งมีโครงสร้างและไม่มีโครงสร้าง ทั้งยังมีโมเดลที่หลากหลาย เช่น ค่าหลัก เอกสาร และกราฟ ซึ่งจะปรับให้เหมาะกับประสิทธิภาพและขนาด สามารถขยายระบบในรูปแบบแนวนอนเพื่อรองรับปริมาณข้อมูลที่มากขึ้น

ความแตกต่างของฐานข้อมูล SQL และ NoSQL ฐานข้อมูล SQL เป็นฐานข้อมูลเชิงสัมพันธ์ คือข้อมูลจะถูกจัดระเบียบเป็นตาราง และแต่ละตารางมีโครงสร้างเฉพาะ ตารางเชื่อมต่อกันผ่านความสัมพันธ์ทำให้ฐานข้อมูลมีประสิทธิภาพมากเหมาะสำหรับการจัดเก็บข้อมูลที่เข้าถึงด้วยวิธีใดวิธีหนึ่ง แต่ฐานข้อมูล NoSQL เป็นฐานข้อมูลที่ไม่สัมพันธ์กันข้อมูลจะจัดเก็บไว้ในชุดเอกสาร ไม่มีโครงสร้างเฉพาะและไม่ได้เชื่อมโยงถึงกันผ่านความสัมพันธ์ จึงเหมาะสำหรับการจัดเก็บข้อมูลที่ไม่จำเป็นต้องเข้าถึงในลักษณะใดลักษณะหนึ่ง ฐานข้อมูล SQL ใช้วิธีการปรับขนาดตามแนวตั้งโดยเพิ่มพลังงานให้กับเซิร์ฟเวอร์ ส่วนฐานข้อมูล NoSQL ใช้วิธีปรับขนาดตามแนวนอนปรับขนาดโดยเพิ่มเซิร์ฟเวอร์ให้มากขึ้น ฐานข้อมูล SQL มีราคาแพงกว่าในการบำรุงรักษามากกว่าฐานข้อมูล NoSQL มีค่าใช้จ่ายน้อยกว่าเนื่องจากดูแลระบบที่น้อยกว่า การตัดสินใจเลือกใช้ฐานข้อมูลให้เหมาะสมขึ้นอยู่กับปัจจัยหลายประการ รวมถึงลักษณะของข้อมูล ข้อกำหนดด้านประสิทธิภาพ ข้อกำหนดด้านความสามารถในการปรับขนาด และงบประมาณ

NoSQL มีรูปแบบการเก็บข้อมูลออกเป็น 4 รูปแบบหลัก ๆ ดังนี้

### 2.3.1 Document Based

การเก็บข้อมูลและ Metadata จัดเป็นลำดับชั้นในรูปแบบ Semi-structure data เช่น JSON หรือ XML ในฐานข้อมูล

### 2.3.2 Key-Value

การเก็บ Record จะมีแค่ Key และ Value เท่านั้น สามารถเข้าถึงข้อมูลได้อย่างรวดเร็วโดยการเข้าถึงข้อมูลให้ใช้ Key ก็จะได้ Value ที่ต้องการ

### 2.3.3 Wide-Column SQL

จะใช้ฐานข้อมูลแบบ RDBMS (Relational Database Management System) มีการเก็บข้อมูลในรูปแบบของ Row ที่มี Column ที่แน่นอน แต่ Wide-Column จะมีการเก็บข้อมูลโดยแต่ละ Row จะมีคู่ของ Key Column และ Value Column ที่อาจจะเหมือนหรือต่างกันก็ได้

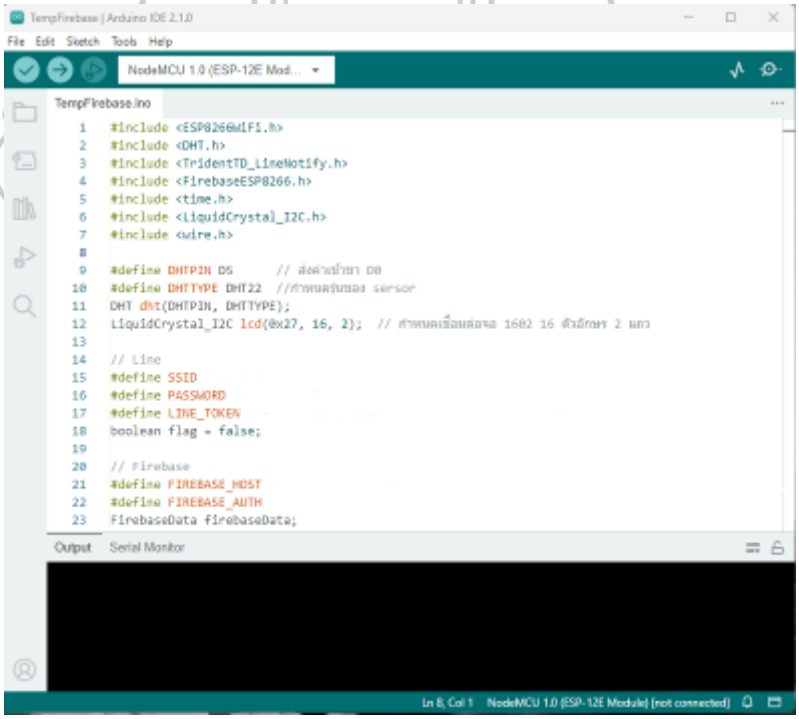
### 2.3.4 Graph

ข้อมูลจะเก็บอยู่ในรูปแบบกราฟแผนภูมิ มี Node และ Edge ที่เชื่อมต่อกัน ไม่ต้องนำข้อมูลมา Join กันเหมือนของ RDBMS

## 2.4 เครื่องมือและภาษาคอมพิวเตอร์ในการพัฒนา

### 2.4.1 Arduino IDE

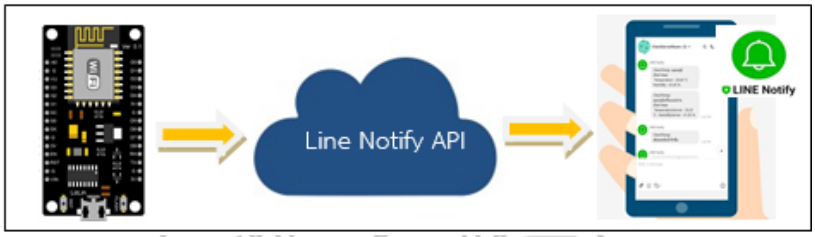
เป็นโปรแกรมที่ใช้งานในลักษณะ Open source ทำหน้าที่ในการติดต่อระหว่างคอมพิวเตอร์ในระบบการ Windows, Mac OS X หรือ Linux กับบอร์ด Arduino หรือ บอร์ด NodeMCU ESP8266 โปรแกรมนี้ออกแบบมาเพื่อให้ง่ายต่อการเขียนและอัปโหลดโปรแกรมที่เขียนอยู่เข้าสู่บอร์ดสำหรับการควบคุมการทำงานของบอร์ด เขียนด้วยภาษา C/C++ และ ภาษา Lau สามารถควบคุมอุปกรณ์ต่าง ๆ เช่น การควบคุมการเปิด-ปิดไฟ, ควบคุมระดับความแรงของพัดลม เป็นต้น และเนื่องจากมีโมดูล Wi-Fi ในตัว จึงสามารถเชื่อมต่อเพื่อส่งข้อมูลหรือสั่งงานผ่านทางอินเทอร์เน็ตได้โดยไม่ต้องหาอุปกรณ์มาต่อเพิ่ม และยังมีราคาที่ถูกช่วยลดต้นทุนเป็นอย่างมากในการนำไปพัฒนาอุปกรณ์ IoT




ภาพที่ 2.9 โปรแกรม Arduino IDE

### 2.4.2 Line Application

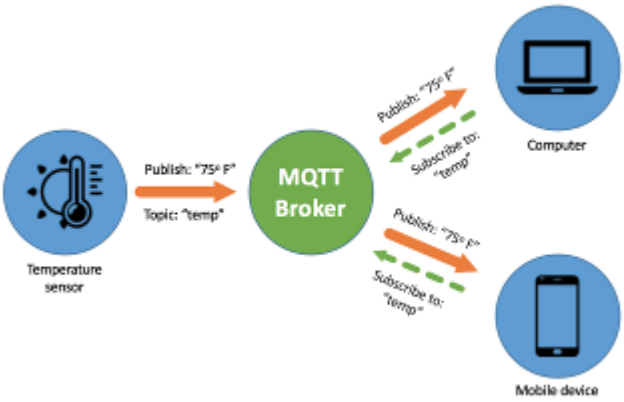
Line เป็นแอปพลิเคชันที่สามารถส่งข้อความ โทร จำเป็นต้องเชื่อมต่ออินเทอร์เน็ต การให้บริการ Line ไม่เสียค่าบริการ ปัจจุบันการใช้งานของ Line ในรูปแบบต่าง ๆ มีมากขึ้น ทั้งการขายสินค้า การชำระเงิน การพูดคุยกันในหน่วยงาน รวมถึงการเป็นส่วนหนึ่งของระบบ IoT ในส่วนของ API ได้มีการเปิดส่วนของ Line Notify ขึ้นมาเพื่อให้นักพัฒนาได้ส่งการแจ้งเตือนข้อความต่าง ๆ ผ่าน Line ได้ง่ายขึ้น Line Notify ออกแบบมาเพื่อใช้สำหรับการแจ้งเตือนโดยเฉพาะการส่งข้อความเพื่อแจ้งเตือนไปยังกลุ่มหรือบุคคล จะมีข้อจำกัดการใช้งานอยู่ที่กลุ่มเดียวหรือบุคคลเดียวเท่านั้น การติดต่อกับ Line ใช้โปรโตคอล HTTPS โดยใช้ Secure Sockets Layer (SSL) หรือ Transport Layer Security) ในการเข้ารหัสข้อมูลที่มีการจราจรอยู่บน HTTP ในการติดต่อสื่อสารเพื่อเพิ่มความปลอดภัยในการส่งข้อมูล และใช้ Token เป็นรหัสผ่านเข้าไปขอส่งข้อความเข้า Line




ภาพที่ 2.10 การแจ้งเตือนในส่วนของ Line Notify

### 2.4.3 Line Application

คือ โปรโตคอลในการส่งข้อมูลที่พัฒนาเพื่อใช้ในระบบ IoT ทำงานแบบ Broker and Clients Network ถูกออกแบบให้สามารถส่งข้อมูลแบบ Real-Time มีปริมาณของข้อมูลน้อย ทำให้ใช้พลังงานต่ำ MQTT พัฒนามาจาก TCP/IP ที่มีการส่งข้อมูลระหว่าง 1 อุปกรณ์กับ 1 อุปกรณ์ โดยที่ต้องทำการเชื่อมโยงกันก่อนถึงจะส่งหรือรับข้อมูลกันได้ และต้องรับประกันว่าข้อมูลนั้นได้ถูกส่งไปครบถ้วนเท่านั้น ถ้าหากมีหลาย ๆ อุปกรณ์อยู่ในระบบจะต้องส่งข้อมูลอุปกรณ์ต่ออุปกรณ์ทีละคู่ทำให้สิ้นเปลืองพลังงานมากไม่เหมาะกับการทำระบบ IoT เนื่องจากระบบ IoT มีการส่งข้อมูลอยู่ตลอดเวลา หลักการทำงานของ MQTT จะมี Broker (Server) และ Clients (Publisher/Subscriber) ทำงานเป็นหลัก โดย Broker จะเป็นตัวกลางในการรับข้อมูลที่เรียกว่า Topic จาก Publisher หลังจากนั้นจะส่ง Topic นั้น ไปยัง Subscriber และ Subscriber ก็จะได้รับข้อมูลทั้งหมดใน Topic นั้นเช่นกัน ตัวอย่างการทำงานของ MQTT ดังภาพที่ 2.11




ภาพที่ 2.11 ตัวอย่างการทำงานของ MQTT

ที่มา : Innorobix Automation (2023) Message Queuing Telemetry Transport (MQTT)

## 2.5 วรรณกรรมที่เกี่ยวข้อง

ผู้ศึกษาได้ทำการทบทวนวรรณกรรมที่เกี่ยวข้องกับงานที่ต้องการศึกษาค้นคว้าในครั้งนี้ โดยแบ่งกลุ่มของการใช้อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ในการควบคุมเซนเซอร์ต่าง ๆ ดังนี้

### 2.5.1 ESP8266

***2.5.1.1 ปกเกศ จันทะกล และ สุจิตรา ผาระนัด (2566) ระบบไฟฟ้าภายในโรงเรือนควบคุมด้วยสมาร์ทโฟน*** ได้ออกแบบและสร้างเครื่องควบคุมระบบไฟฟ้าภายในโรงเรือนด้วยสมาร์ทโฟน ตัวโรงเรือนจำลอง ควบคุม 3 ระบบ คือ ระบบน้ำ พัดลมระบายอากาศ และระบบแสงสว่าง โดยใช้เซนเซอร์ร่วมกับอุปกรณ์ควบคุมไมโครคอนโทรลเลอร์ ผ่านระบบเครือข่ายอินเทอร์เน็ต ติดตั้งเซนเซอร์แสงและอุณหภูมิภายในโรงเรือนและทำการวัดค่าต่าง ๆ ที่ได้ส่งมาที่ไมโครคอนโทรลเลอร์จากนั้นจะทำการประมวลผลค่าที่ได้นำมาเปรียบเทียบกับค่ามาตรฐานที่ตั้งไว้ ถ้าอุณหภูมิต่ำกว่า 30 องศาเซลเซียส จะทำการปิดพัดลมอัตโนมัติ ความเข้มแสงสว่างมากกว่า 300 ลักซ์ ให้ทำการปิดไฟอัตโนมัติ ระบบประมวลผลจะทำการส่งข้อมูลผ่านระบบเครือข่ายอินเทอร์เน็ตมายังโทรศัพท์สมาร์ทโฟน เพื่อให้ผู้ดูแลระบบทราบและทำการควบคุมปัจจัยภายในโรงเรือนโดยสั่งการผ่านสมาร์ทโฟน จากนั้นระบบจะสั่งการให้ไมโครคอนโทรลเลอร์ควบคุมอุปกรณ์ต่าง ๆ ภายในโรงเรือน

***2.5.1.2 นิติคม อริยพิมพ์ และชัยพร อัดโดดดร (2565) การออกบบและสร้างระบบไอโอทีสำหรับบ้านจำลองแบบอัจฉริยะที่ควบคุมด้วยไมโครคอนโทรลเลอร์*** การประยุกต์ใช้ระบบไอโอที ประกอบด้วย ESP8266 ใช้ในการรองรับการเชื่อมต่อเครือข่ายมาตรฐาน IEEE 802.11b/g/n ให้สามารถเชื่อมต่อกับเครือข่าย WiFi ได้เพื่อให้บอร์ดเชื่อมต่อกับอินเทอร์เน็ต และใช้

อุปกรณ์ตรวจจับ MQ-2 ในการตรวจแก๊สและควัน ใช้โมดูล PIR HC-SR501 ตรวจจับความเคลื่อนไหวในระยะ 1-4 เมตร เมื่อตัวอุปกรณ์ตรวจจับสามารถตรวจจับการเคลื่อนไหวได้ จะส่งการแจ้งเตือนไปยังแอพพลิเคชั่น Line ใช้อุปกรณ์ LDR Sensor ในการตรวจสอบสถานะการทำงานของหลอดไฟและควบคุมการเปิด-ปิดหลอดไฟด้วยสมาร์ทโฟนผ่านแอพพลิเคชั่น Blynk

ข้อเสนอแนะควรใช้อุปกรณ์กล้องเข้ามาร่วมในการตรวจจับความเคลื่อนไหวเพื่อให้ได้ประสิทธิภาพที่แม่นยำยิ่งขึ้น เพิ่มระบบเปิด-ปิด ของหลอดไฟแบบตั้งเวลาเปิด-ปิดอัตโนมัติ เพิ่มอุปกรณ์สำรองไฟ กรณีเกิดไฟฟ้าดับเพื่อแผงวงจรควบคุมจะได้ไม่เกิดการเสียหาย

***2.5.1.3 วิจิตตรา คุ้มวงษ์ และสรวง รุ่งประกายพรรณ. (2565). การพัฒนาชุดอุปกรณ์อินเตอร์เน็ตของสรรพสิ่งสำหรับติดตามอุณหภูมิและความชื้นของการเก็บรักษายา*** ได้พัฒนาชุดอุปกรณ์ IoT สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ผ่านคลาวด์เซิฟเวอร์ NETPIE2020 โดยการประยุกต์ใช้ NodeMCU V2 LUA based ESP8266-12E เป็นส่วนในการรับข้อมูลจากเซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ และทำการแจ้งเตือนไปยัง Line Application กรณีที่อุณหภูมิและความชื้นสัมพัทธ์ออกนอกช่วงที่กำหนด

***2.5.1.4 ณัฏฐพล เจริญศิริ, ประชาสันต์ แว่นไธสง และ กฤษณพล เกิดทองคำ. (2563). ระบบแจ้งเตือนอุณหภูมิและความชื้น สำหรับห้องควบคุมไฟฟ้า สำหรับบริษัทสงวนวงษ์อุตสาหกรรม จำกัด*** ได้ทำการศึกษาระบบงานเดิมและทำการออกแบบระบบใหม่เพื่อทำการตรวจวัดค่าอุณหภูมิและความชื้นของห้องควบคุมโดยการประยุกต์ใช้บอร์ด ESP8266 ร่วมด้วยเซนเซอร์ตรวจวัดอุณหภูมิและความชื้นแสดงผลผ่านมือถือด้วย Blynk Application และทำการแจ้งเตือนไปยัง Line Application ผลการทดสอบสามารถทำงานได้จริง ความพึงพอใจของผู้ใช้งานอยู่ในระดับถึงพอใจมาก

***2.5.1.5 บะไบ๋ คาริชมาลย์, จีรัง คำนวนตา และ สัญญา พันธุ์แพง. (2563). พัดลมไอน้ำอัจฉริยะ*** ออกแบบระบบควบคุมการทำงานของพัดลมไอน้ำ โดยใช้ ESP8266 ร่วมด้วยระบบเซนเซอร์วัดค่าอุณหภูมิและปรับความเร็วของพัดลมตามอุณหภูมิได้แบบอัตโนมัติและยังสามารถควบคุมสั่งการผ่านมือถือได้ ผลการทดลองพบว่าสามารถควบคุมพัดลมไอน้ำผ่านระบบมือถือและการทำงานของฟังก์ชันต่าง ๆ ของระบบก็ทำงานได้อย่างถูกต้อง

***2.5.1.6 สิรวิชญ์ โตหมื่นไวย, ศุภกร สุวรรณะ, ศุภกฤษ นาคป้อมฉิน, และ แก้วใจ อาภรณ์พิศาล. (2563). การพัฒนาระบบควบคุมอุณหภูมิและค่าความชื้นผ่านสมาร์ทโฟน กรณีศึกษาฟาร์มเห็ดสุริยนต์ อาเภอนครชัยศรี จังหวัดนครปฐม*** ได้ออกแบบเครื่องมือช่วยวัดค่าอุณหภูมิความชื้นและระบบน้ำแบบเปิด-ปิดอัตโนมัติ เพื่อช่วยให้การเพาะเห็ดมีคุณภาพและได้ผลผลิตมากยิ่งขึ้น โดยการประยุกต์ใช้บอร์ด ESP8266 และเซนเซอร์วัดค่าอุณหภูมิและความชื้นใช้ในการ

ตรวจวัดและส่งค่าไปบน Firebase Database Realtime จากนั้นจะส่งค่าไปที่แอปพลิเคชันบนมือถือเพื่อแสดงค่าที่ได้และทำการแจ้งเตือนหากค่าที่ตรวจพบเกินมาตรฐานที่กำหนด

***2.5.1.7 จันตพงษ์ บุตรลักษณ์. (2560). การพัฒนาระบบเฝ้าระวังสภาพแวดล้อมของเซิร์ฟเวอร์แร็ค*** งานวิจัยนี้ได้เสนอการพัฒนาระบบเฝ้าระวังความปลอดภัยของเซิร์ฟเวอร์แร็ค โดยมีการแจ้งเตือนเมื่อเกิดเหตุการณ์ผิดปกติ จะช่วยป้องกันอุปกรณ์ต่าง ๆ ในศูนย์ข้อมูลไม่ให้เกิดความเสียหาย โดยการใช้ ESP8266 ร่วมด้วยเซนเซอร์ตรวจวัดค่าอุณหภูมิและความชื้นของระบบทำความเย็น การตรวจจับควันและระบบไฟฟ้า ค่าที่วัดได้จะแสดงผลที่หน้าเว็บ Grafana โดยแสดงเป็นกราฟและสถิติต่าง ๆ ของกำลังไฟฟ้า พร้อมทั้งแสดงความชื้นของตู้แร็ค และทำการแจ้งเตือนความผิดปกติผ่าน Line Application

### 2.5.2 Arduino Microcontroller

***2.5.2.1 รวิช ควรประเสริฐ และพรพุฒิ ศุภเอม. (2565). การพัฒนาระบบตรวจวัดอุณหภูมิในงานเทคโนโลยีสถาปัตยกรรมด้วยไมโครคอนโทรลเลอร์แบบอาร์ดุยโน*** โดยการออกแบบระบบตรวจวัดอุณหภูมิจากไมโครคอนโทรลเลอร์แบบอาร์ดุยโนร่วมกับโมดูลนาฬิกาแบบเรียบไทม์ โมดูลเก็บข้อมูลโดยเอสดีการ์ด โมดูลจอภาพแบบโมโนโครม และโมดูลเซนเซอร์อุณหภูมิแบบเทอร์มิสเตอร์ และทำการปรับเทียบค่าระบบ ระหว่างระบบตรวจวัดอุณหภูมิเชิงอุตสาหกรรมโอปุส 200 ของ LUFFT เป็นระบบตรวจวัดเชิงอุตสาหกรรมที่มีมาตรฐานทางไฟฟ้าและการตรวจวัดร่วมกับระบบตรวจวัดอุณหภูมิด้วยไมโครโทรลเลอร์แบบอาร์ดุยโนที่พัฒนาขึ้น โดยเปรียบเทียบค่าข้อมูลที่ได้ใช้รูปแบบสถิติแบบการวิเคราะห์ถดถอย ผลการทดสอบสามารถใช้งานได้จริงในสภาพแวดล้อมทั่วไปได้เป็นอย่างดี

***2.5.2.2 ธานิล ม่วงพูล และวงศกร. (2562). อุดมพัดลมอัจฉริยะที่ควบคุมด้วยอาดูไอโน โดยการออกแบบพัดลมอัจฉริยะที่ควบคุมด้วยอาดูไอโน*** แบ่งเป็น 3 ส่วน 1) ตรวจสอบค่าอุณหภูมิโดยใช้เซนเซอร์วัดอุณหภูมิ และส่งค่าไปยังบอร์ดไมโครคอนโทรลเลอร์ เพื่อคำนวณในการสั่งการควบคุมความแรงลม และจะตัดการทำงานเมื่ออุณหภูมิต่ำกว่า 25 องศาเซลเซียส 2) รับค่าระยะห่างของมนุษย์โดยใช้ เซนเซอร์วัดระยะทาง และส่งค่าไปยังบอร์ดไมโครคอนโทรลเลอร์ เพื่อคำนวณสั่งการบังคับทิศทางของลม และ 3) กาหนดค่าและเขียนคำสั่งโดยใช้โปรแกรม Arduino IDE ซึ่งใช้ภาษา C++ ในการเขียนคำสั่งเพื่อกำหนดค่าควบคุมต่างๆ ผลการทดสอบการปรับความแรงลมสอดคล้องกับอุณหภูมิที่ตรวจวัดได้ และการบังคับทิศทางลมตามที่มนุษย์อยู่สามารถทำงานได้จริง

### 2.5.3 WeMos D1 Mini Node

***2.5.3.1 ชนนิกานต์ รอดมรณ์, มธุรส ผ่านเมือง และ วีรศักดิ์ จงเลขา (2564) การประยุกต์ใช้เครือข่ายเซนเซอร์ไร้สายสาหรับฟาร์มอัจฉริยะ*** ได้ประยุกต์ใช้เทคโนโลยี IoT เข้ามาช่วยในการเพาะปลูกผักกาดหอมพันธุ์กรีนโอ๊ค โดยใช้ DHT11 Sensor Node ในการวัดความชื้นในอากาศและวัดอุณหภูมิ และใช้ Capacitive Soil Moisture Sensor ในการวัดความชื้นในดิน เมื่อได้ค่าจะทำการส่งข้อมูลไปยังบอร์ด WeMos D1 Mini Node และบอร์ด ESP32 (Controller Gateway) ข้อมูลที่รับ-ส่ง จะใช้ MOTT Protocol เพื่อเก็บข้อมูลลงฐานข้อมูลในรูปแบบ JSON ไฟล์ รวมถึงส่งคำสั่งไปควบคุมการเปิด-ปิดปั๊มน้ำ และเมื่อบอร์ด WeMos D1 ESP32 ได้รับข้อมูลจากเซนเซอร์ และพบว่ามีค่าไม่สมดุลตามที่กำหนดไว้ ก็จะสั่งการให้ปั๊มน้ำทำงานรดน้ำในแปลงผักอัตโนมัติและจะหยุดรดน้ำเมื่อพบว่า ความชื้นในดินมีค่าระหว่าง 40-70% และผู้ดูแลระบบยังสามารถควบคุมการทำงานของปั๊มน้ำผ่านแอปพลิเคชันบนโทรศัพท์มือถือหรือเว็บไซต์ได้

# บทที่ 3
# วิธีดำเนินการศึกษา

ขั้นตอนการดำเนินงานในการพัฒนาระบบตรวจวัดค่าและการแจ้งเตือนสภาพแวดล้อมห้องคอมพิวเตอร์แม่ข่าย มีดังต่อไปนี้

ภาพที่ 3.1 แสดงขั้นตอนการดำเนินงานในการพัฒนาระบบการตรวจวัดสภาพแวดล้อมและระบบการแจ้งเตือนของห้องคอมพิวเตอร์แม่ข่าย

## 3.1 การวิเคราะห์ และพัฒนาระบบการตรวจวัดค่าและระบบการแจ้งเตือนสภาพแวดล้อมห้องคอมพิวเตอร์แม่ข่าย

ประกอบด้วย 5 ส่วน ดังนี้

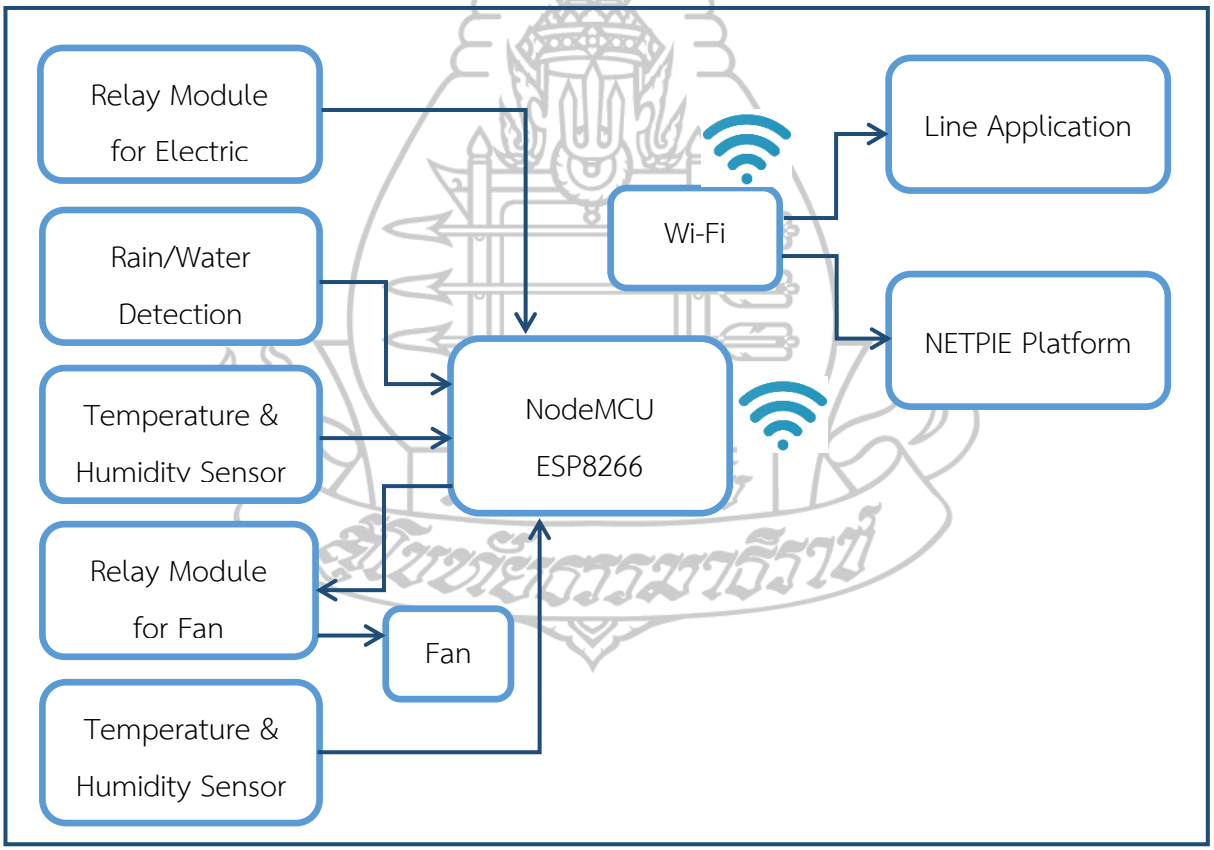
ระบบตรวจจับการทำงานของไฟฟ้า

ระบบการตรวจวัดระดับน้ำจากถาดรองน้ำในเครื่องปรับอากาศ

ระบบการตรวจวัดค่าอุณหภูมิและความชื้นภายในห้องคอมพิวเตอร์แม่ข่าย พร้อมปรับระดับความแรงพัดลมอัตโนมัติ

ระบบการตรวจวัดค่าจากเครื่องคอมพิวเตอร์แม่ข่าย

ระบบแจ้งเตือนผ่านแอปพลิเคชันมือถือ และแสดงผลบนหน้า Dashboard




ภาพที่ 3.2 แสดงขั้นตอนพัฒนาระบบการตรวจวัดสภาพแวดล้อมและระบบการแจ้งเตือนของห้องคอมพิวเตอร์แม่ข่าย

## 3.2 การพัฒนาระบบการตรวจวัดค่าและระบบการแจ้งเตือน

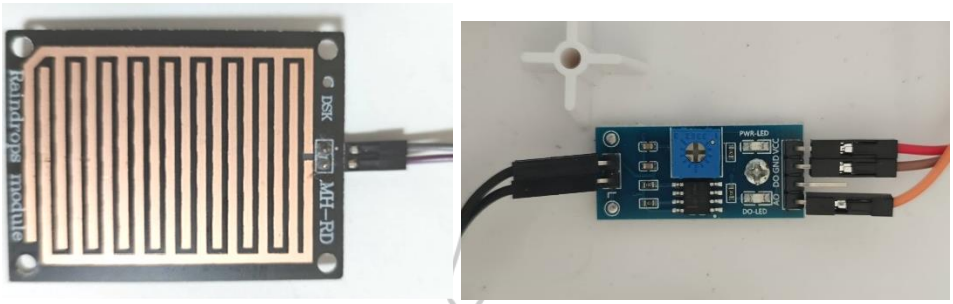
**ประกอบด้วย 2 ส่วน ดังนี้**

### 3.3.1 ฮาร์ดแวร์

เป็นส่วนของการติดตั้งอุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 เป็นส่วนของวงจรควบคุมการทำงานของอุปกรณ์ต่าง ๆ ได้แก่ เซนเซอร์ตรวจวัดอุณหภูมิและความชื้น, เซนเซอร์ตรวจวัดระดับน้ำ, โมดูลรีเลย์ควบคุมการปรับระดับความแรงของพัดลมอัตโนมัติ และตรวจจับกระแสไฟฟ้าดับ และทำการการเชื่อมต่อวงจรที่เป็นส่วนประกอบของชุดต้นแบบ

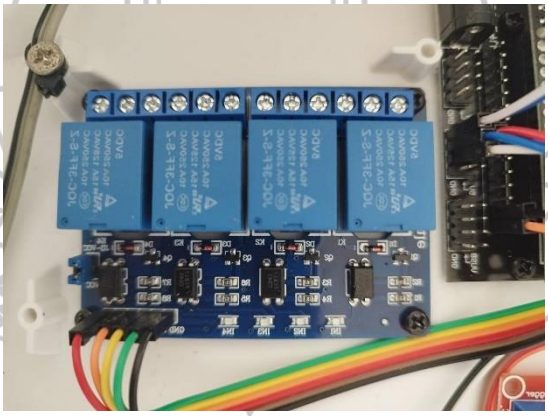
ภาพที่ 3.3 บอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266

ภาพที่ 3.4 Relay Module 5V 1 Channel

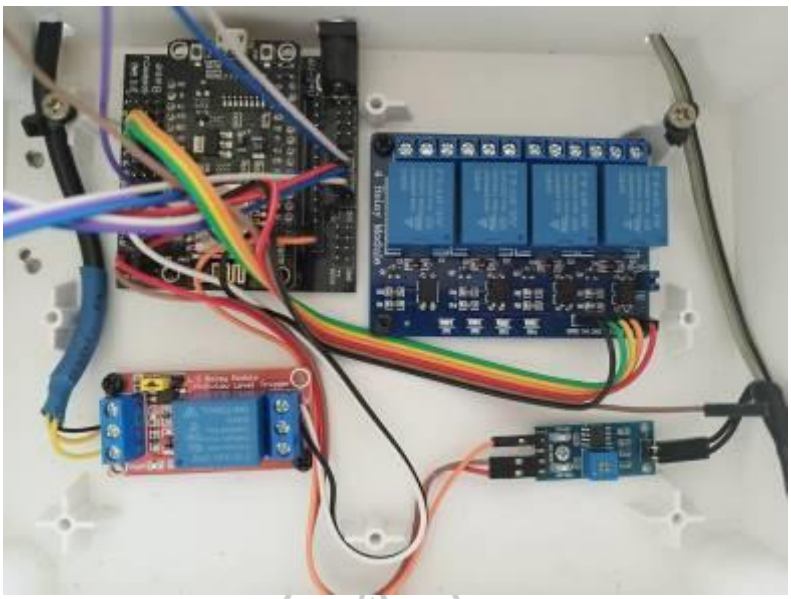



ภาพที่ 3.5 Rain/Water Detection Sensor Module

ภาพที่ 3.6 DHT22 Temperature and Humidity Sensor Module



ภาพที่ 3.7 P2H1588A0 Relay Module 5V 4 Channel

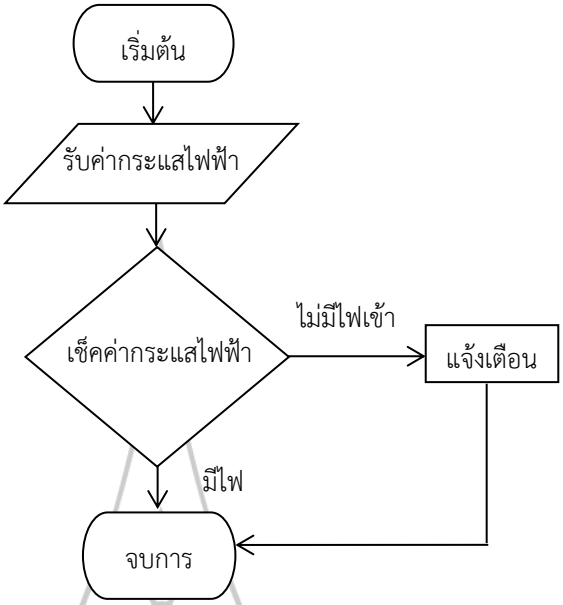


ภาพที่ 3.8 การประกอบวงจรชุดต้นแบบ

### 3.2.2 ซอฟต์แวร์

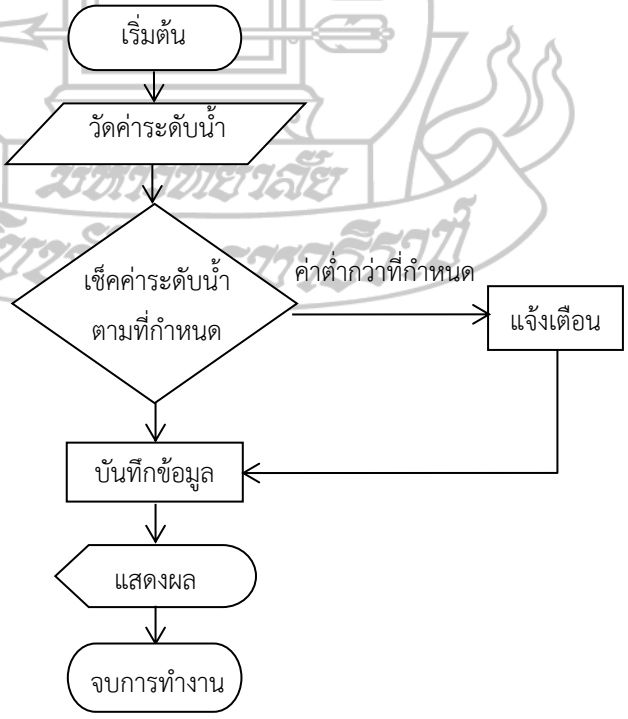
ใช้โปรแกรม Arduino IDE 2.1.0 เป็นเครื่องมือในการเขียนควบคุมการทำงานและรับค่าของอุปกรณ์ในการทำงานของระบบ เป็นส่วนของโปรแกรมประมวลผลการทำงานของบอร์ดไมโครคอนโทรลเลอร์ พัฒนาด้วยภาษา C/C++ โดยสามารถแบ่งการทำงาน ดังต่อไปนี้

***3.2.2.1 โปรแกรมประมวลผลการทำงานของบอร์ดไมโครคอนโทรลเลอร์*** รับค่าจากโมดูลรีเลย์วัดค่ากระแสไฟฟ้า หากไฟฟ้าดับจะทำการส่งค่าโลจิก 0 ให้กับบอร์ดไมโครคอนโทรลเลอร์ นำไปเปรียบเทียบกับค่าที่กำหนดไว้ หากค่าที่ได้รับเป็น 0 คือไฟฟ้าดับ และถ้าค่าเป็น 1 จะมีกระแสไฟฟ้าผ่าน จากนั้นจะทำกาแจ้งเตือนผ่านแอปพลิเคชันไลน์ และแสดงผลบนหน้า Dashboard



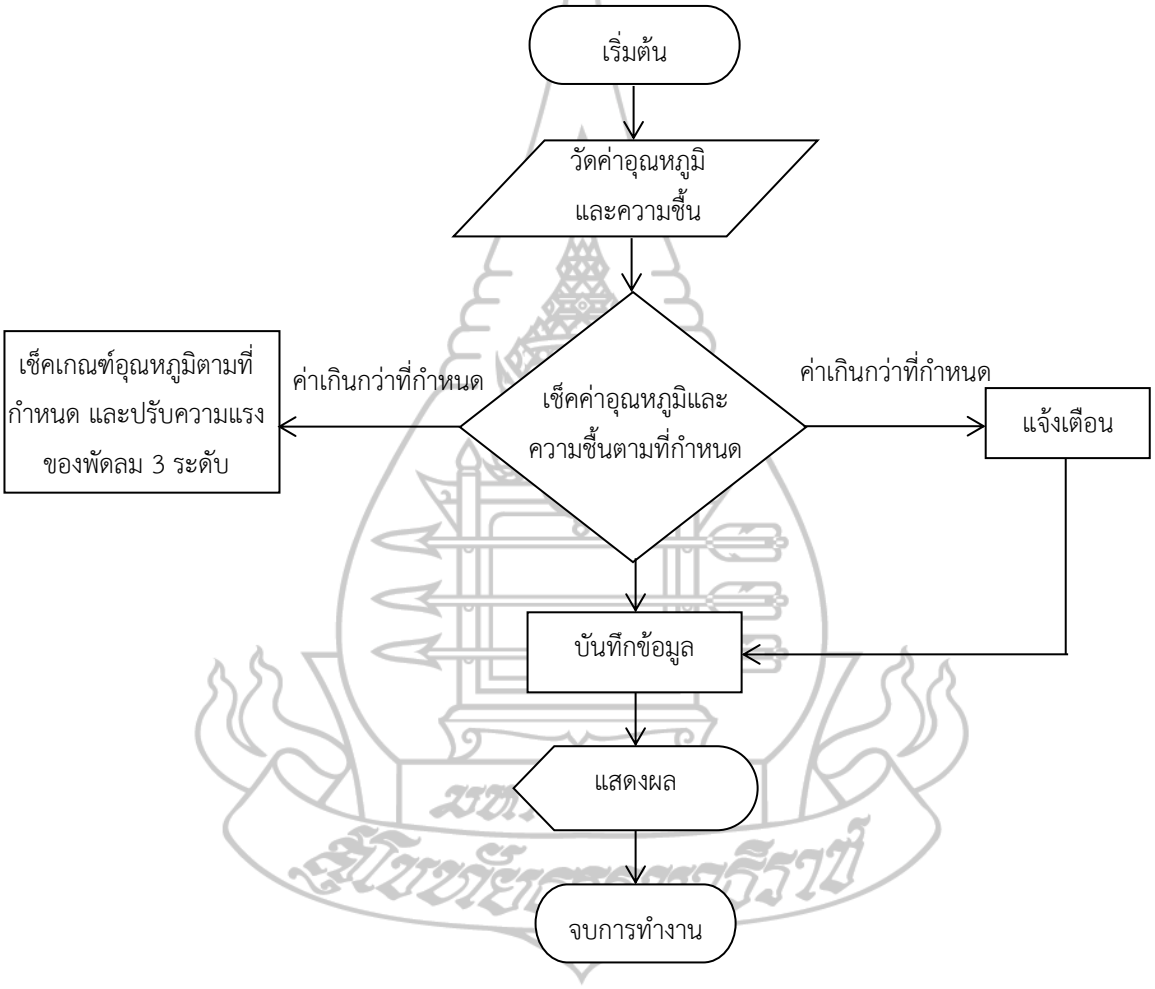

ภาพที่ 3.9 ผังแสดงการทำงานของโมดูลวัดค่ากระแสไฟฟ้าภายในห้องคอมพิวเตอร์แม่ข่าย

***3.2.2.2 โปรแกรมประมวลผลการทำงานของบอร์ดไมโครคอนโทรลเลอร์*** รับค่าการตรวจวัดระดับน้ำจากถาดรองน้ำในเครื่องปรับอากาศ หากมีค่าต่ำกว่าที่กำหนดจะทำการแจ้งเตือนไปยังแอปพลิเคชันไลน์ และแสดงผลบนหน้า Dashboard



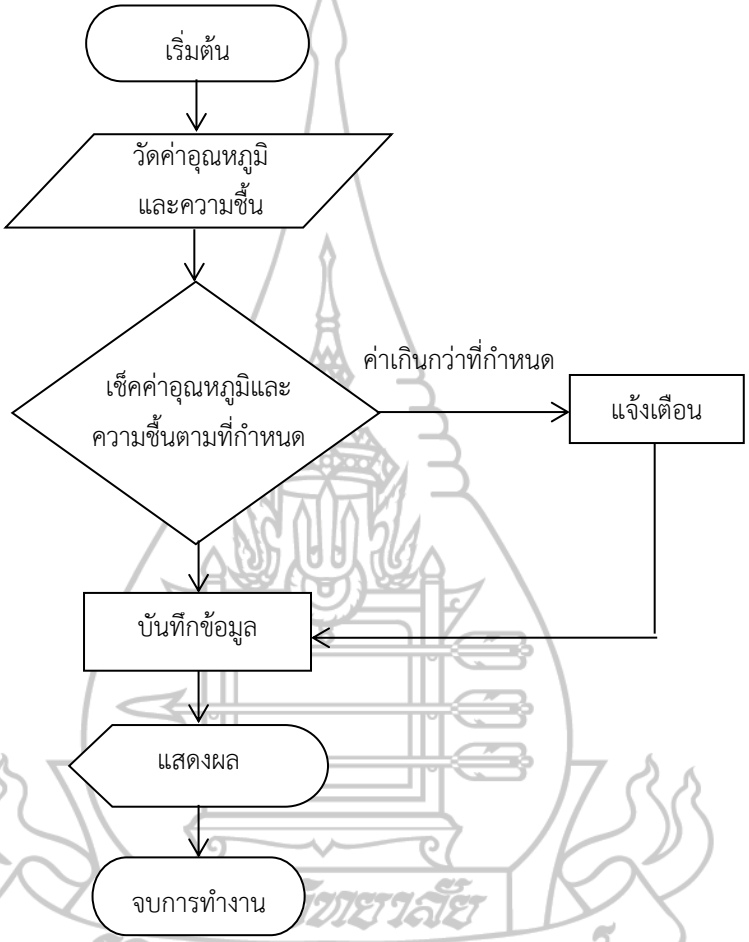

ภาพที่ 3.10 ผังแสดงการทำงานของโมดูลวัดค่าระดับน้ำจากถาดรองน้ำในเครื่องปรับอากาศ

***3.2.2.3 โปรแกรมประมวลผลการทำงานของบอร์ดไมโครคอนโทรลเลอร์*** รับค่าอุณหภูมิและความชื้นภายในห้องคอมพิวเตอร์แม่ข่าย จะทำการเปรียบเทียบค่าที่ตั้งไว้ หากมีค่าเกินกว่าที่กำหนดจะทำการแจ้งเตือนไปยังแอปพลิเคชันไลน์ และทำการสั่งให้โมดูลรีเลย์ของพัดลมเปิดอัตโนมัติพร้อมปรับระดับพัดลมตามอุณหภูมิที่กำหนดไว้มีทั้งหมด 3 ระดับ และแสดงผลบนหน้า Dashboard



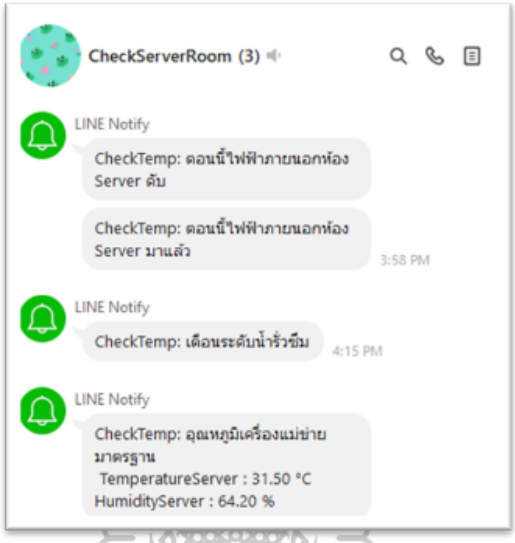

ภาพที่ 3.11 ผังแสดงการทำงานของโมดูลวัดค่าอุณหภูมิและความชื้นภายในห้องเซิร์ฟเวอร์ และทำการปรับระดับความแรงของพัดลมอัตโนมัติตามเงื่อนไขที่กำหนด

***3.2.2.4 โปรแกรมประมวลผลการทำงานของบอร์ดไมโครคอนโทรลเลอร์*** รับค่าอุณหภูมิและความชื้นจากเครื่องคอมพิวเตอร์แม่ข่าย เพื่อทำการเปรียบเทียบค่าที่ตั้งไว้ หากมีค่าเกินกว่าที่กำหนดจะทำการแจ้งเตือนไปยังแอปพลิเคชันไลน์ และแสดงผลบนหน้า Dashboard

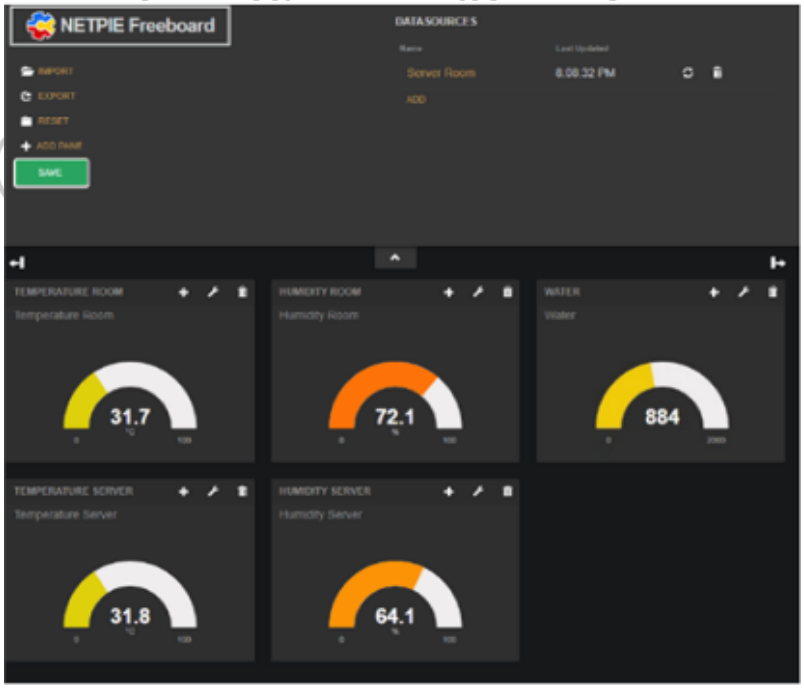



ภาพที่ 3.12 ผังแสดงการทำงานของโมดูลวัดค่าอุณหภูมิและความชื้นของเครื่องคอมพิวเตอร์แม่ข่าย

***2.2.5 โปรแกรมประมวลผลการทำงานของบอร์ดไมโครคอนโทรลเลอร์*** รับค่าการตรวจวัดต่าง ๆ และทำการแจ้งเตือนไปยังแอปพลิเคชันไลน์ และแสดงผลบนหน้า Dashboard

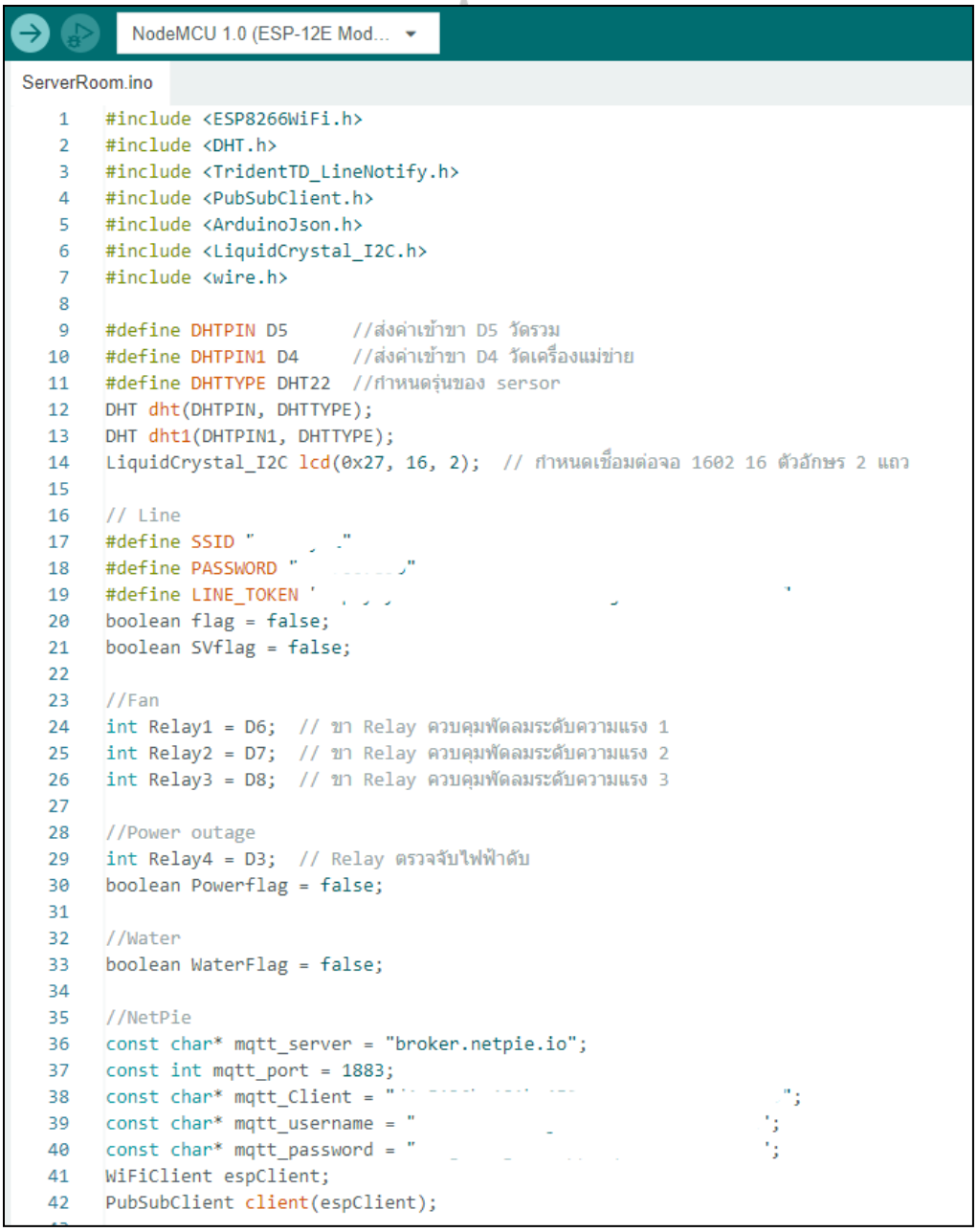



ภาพที่ 3.13 แสดงการแจ้งเตือนผ่านแอปพลิเคชันไลน์




ภาพที่ 3.14 แสดงผลบนหน้า Dashboard Netpie

การเขียนโปรแกรมและการอัพโหลดโปรแกรม ด้วยการใช้ Arduino IDE 2.1.0 ที่ใช้เป็นเครื่องมือในการเขียนควบคุมการทำงานและรับค่าของอุปกรณ์ในการทำงานของระบบที่ได้ทำการออกแบบ มีดังนี้




```
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <TridentTD_LineNotify.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <LiquidCrystal_I2C.h>
#include <wire.h>

#define DHTPIN D5      //ส่งค่าเข้าขา D5 วัดรวม
#define DHTPIN1 D4     //ส่งค่าเข้าขา D4 วัดเครื่องแม่ข่าย
#define DHTTYPE DHT22  //กำหนดรุ่นของ sersor
DHT dht(DHTPIN, DHTTYPE);
DHT dht1(DHTPIN1, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);  // กำหนดเบื้องต่อจอ 1602 16 ตัวอักษร 2 แถว

// Line
#define SSID "
#define PASSWORD "
#define LINE_TOKEN '
boolean flag = false;
boolean SVflag = false;

//Fan
int Relay1 = D6;  // ขา Relay ควบคุมพัดลมระดับความแรง 1
int Relay2 = D7;  // ขา Relay ควบคุมพัดลมระดับความแรง 2
int Relay3 = D8;  // ขา Relay ควบคุมพัดลมระดับความแรง 3

//Power outage
int Relay4 = D3;  // Relay ตรวจจับไฟฟ้าดับ
boolean Powerflag = false;

//Water
boolean WaterFlag = false;

//NetPie
const char* mqtt_server = "broker.netpie.io";
const int mqtt_port = 1883;
const char* mqtt_Client = "
";
const char* mqtt_username = "
';
const char* mqtt_password = "
';
WiFiClient espClient;
PubSubClient client(espClient);
```

ภาพที่ 3.15 แสดงการเขียนโปรแกรมควบคุมการทำงานของระบบ

## 3.3 การทดสอบและการประเมินประสิทธิภาพของการทำงานของการพัฒนาระบบต้นแบบจากการประยุกต์การเฝ้าสังเกตด้วยการตรวจวัดค่าสภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์ม

ทำการทดสอบระบบโดยนำระบบที่ทำการออกแบบและพัฒนาเรียบร้อยแล้วติดตั้งเพื่อวัดค่าของระบบต่าง ๆ โดยแบ่งการทดสอบของระบบออกเป็น 5 ระบบ ดังต่อไปนี้

### 3.3.1 ระบบการตรวจจับการทำงานของไฟฟ้า

ทำการทดสอบโดยการต่อเสียบอุปกรณ์ไว้กับเต้ารับไฟฟ้าห้องคอมพิวเตอร์แม่ข่าย หากไม่มีกระแสไฟฟ้าเข้ามาจะทำการแจ้งเตือนผ่านแอปพลิเคชันไลน์ว่าไฟดับ และเมื่อกระแสไฟฟ้าไหลผ่านปกติจะทำการแจ้งเตือนว่าไฟมาแล้ว

### 3.3.2 ระบบการตรวจวัดระดับน้ำจากถาดรองน้ำในเครื่องปรับอากาศ

ทำการทดสอบโดยการนำแผ่นตรวจวัดติดตั้งในถาดรองน้ำเครื่องปรับอากาศ กำหนดค่าระดับน้ำของเซนเซอร์แล้วทำการตรวจวัดค่าของระดับน้ำ ถ้าระดับน้ำมีปริมาณสูงขึ้นค่าแรงดันจะต่ำลง ทางผู้ศึกษาได้ทำการทดสอบและทำการปรับค่าให้ตรงกับการใช้งานดังนี้ จากแผ่นตรวจวัดระดับน้ำเทียบความสูงของระดับน้ำกับค่าที่ได้จากการวัดของระบบโดยจะได้ค่าของแรงดันน้ำมีค่าต่ำกว่า 650 จะทำการแจ้งเตือนผ่านแอปพลิเคชันไลน์ การตั้งเกณฑ์วัดระดับน้ำต้องขึ้นอยู่กับการบริหารจัดการของระยะเวลาเมื่อเกิดเหตุการณ์ว่าจะสามารถเข้ามาแก้ไขได้ทันเวลาหรือไม่

### 3.3.3 ระบบการตรวจวัดค่าอุณหภูมิและความชื้นภายในห้องเซิร์ฟเวอร์และระบบปรับระดับความแรงของพัดลมอัตโนมัติ

ผู้จัดทำได้กำหนดค่าอุณหภูมิของห้องไม่เกิน 25 องศาเซลเซียส ถ้าหากเกินค่าที่กำหนดให้ทำการแจ้งเตือนไปยังแอปพลิเคชันไลน์ และทำการเปิดพัดลมอัตโนมัติตามความแรงที่กำหนดโดยกำหนดค่าไว้ดังนี้ อุณหภูมิสูงกว่า 25 องศาเซลเซียส แต่ไม่เกิน 28 องศาเซลเซียส ความแรงพัดลมอยู่ที่ระดับ 1 อุณหภูมิสูงกว่า 28 องศาเซลเซียส แต่ไม่เกิน 31 องศาเซลเซียส ความแรงพัดลมอยู่ที่ระดับ 2 อุณหภูมิตั้งแต่ 31 องศาเซลเซียส ความแรงพัดลมอยู่ที่ 3 ทำการทดสอบและประเมินประสิทธิภาพโดยการตรวจวัดค่าอุณหภูมิและความชื้น พร้อมทั้งทำการแจ้งเตือน และผลของการทำงานของพัดลมสามารถปรับระดับความแรงตามเกณฑ์ที่กำหนดไว้หรือไม่

### 3.3.4 ระบบการตรวจวัดค่าจากเครื่องคอมพิวเตอร์แม่ข่าย

ทำการทดสอบโดยการติดตั้งอุปกรณ์ไว้กับตัวเครื่องคอมพิวเตอร์แม่ข่าย กำหนดค่าอุณหภูมิของเครื่องคอมพิวเตอร์แม่ข่ายไม่เกิน 60 องศาเซลเซียส ถ้าหากเกินค่าที่กำหนดให้ทำการแจ้งเตือนไปยังแอปพลิเคชันไลน์ ทำการทดสอบและประเมินประสิทธิภาพของการตรวจวัดค่าอุณหภูมิและความชื้น พร้อมทั้งทำการแจ้งเตือน

### 3.3.5 การแสดงผลบนหน้า Dashboard

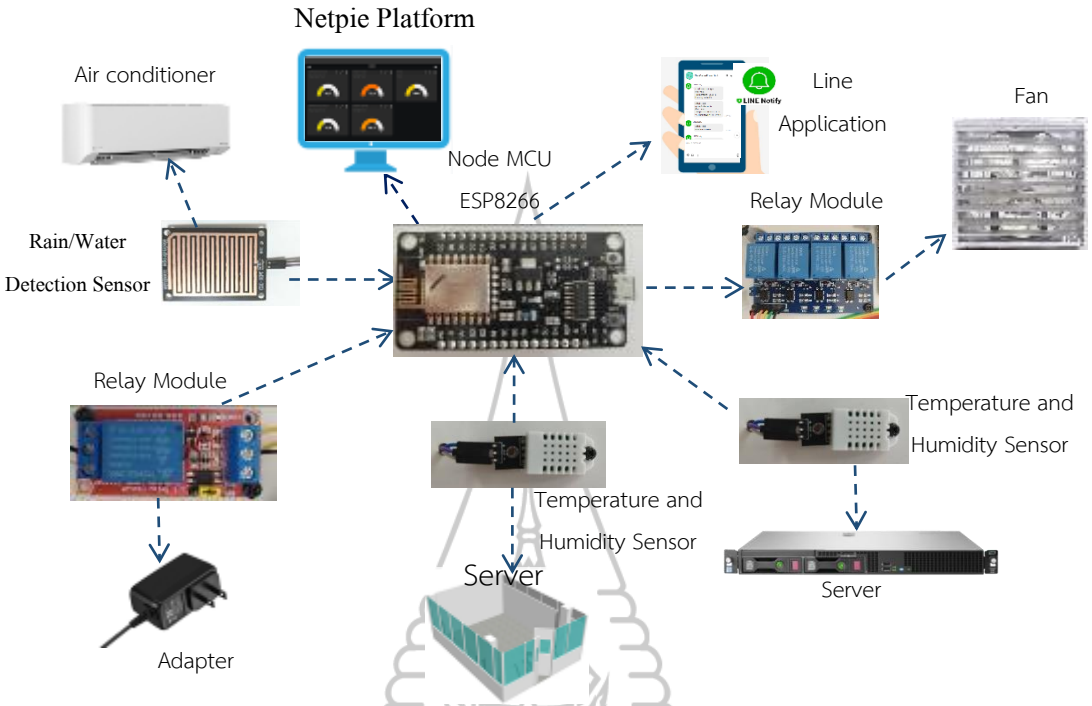
การตรวจวัดค่าจากอุปกรณ์ต่าง ๆ จะต้องแสดงผลที่ได้บนหน้า Dashboard เพื่อให้ผู้ดูแลระบบรู้ข้อมูลของค่าต่าง ๆ ที่ตรวจวัดได้ในหน้าเดียวกัน ว่าอุปกรณ์ตัวไหนมีค่าสูงกว่าปกติ เพื่อใช้ข้อมูลนั้นมาช่วยในการวิเคราะห์แก้ไขปัญหาได้ตรงจุดและทันเวลา

# บทที่ 4
# ผลการวิเคราะห์ข้อมูล

ผลการวิเคราะห์ข้อมูลของการพัฒนาระบบต้นแบบจากการประยุกต์การเฝ้าสังเกตด้วยการตรวจวัดค่าสภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์ม แบ่งออกเป็น 2 ตอน ผลการทดสอบมีความสามารถในการทำงานตรงตามขอบเขตที่ได้กำหนดไว้ ตอนที่ 1 การพัฒนาอุปกรณ์ตรวจจับการเกิดไฟฟ้าดับ ตรวจวัดค่าระดับน้ำในถาดรองน้ำเครื่องปรับอากาศ ระบบการตรวจวัดค่าสภาพแวดล้อมภายในห้องห้องเซิร์ฟเวอร์ที่เป็นอุปกรณ์ต้นแบบตรวจวัดค่าอุณหภูมิและความชื้นภายในห้องพร้อมปรับระดับความแรงของพัดลมอัตโนมัติ ตรวจวัดค่าอุณหภูมิและความชื้นของเครื่องคอมพิวเตอร์แม่ข่าย การแจ้งเตือนผ่านแอพพลิเคชันไลน์และแสดงรายงานผลบน Dashboard และตอนที่ 2 การประเมินประสิทธิภาพระบบการตรวจวัดค่า การแจ้งเตือนผ่านแอพพลิเคชันไลน์ และแสดงรายงานผลบน Dashboard

## 4.1 การพัฒนาระบบต้นแบบจากการประยุกต์การเฝ้าสังเกตด้วยการตรวจวัดค่าสภาพแวดล้อมของห้องคอมพิวเตอร์แม่ข่ายด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 และ NETPIE แพลตฟอร์ม

อุปกรณ์ต้นแบบระบบตรวจวัดสภาพแวดล้อมที่จัดทำขึ้นด้วยอุปกรณ์เซนเซอร์ร่วมกับอุปกรณ์ ESP8266 ในการพัฒนาอุปกรณ์ได้ตามขอบเขตที่กำหนดไว้ ทำการตรวจจับไฟฟ้าดับภายในห้อง ตรวจวัดระดับน้ำในถาดรองเครื่องปรับอากาศ ตรวจวัดอุณหภูมิและความชื้นภายในห้องคอมพิวเตอร์แม่ข่ายสามารถเปิดพัดลมแบบอัตโนมัติโดยปรับระดับความแรงของพัดลมตามอุณหภูมิของห้องที่กำหนด และตรวจวัดอุณหภูมิและความชื้นของเครื่องคอมพิวเตอร์แม่ข่าย




ภาพที่ 4.1 ภาพแสดงการทำงานของอุปกรณ์ต้นแบบระบบตรวจวัดสภาพแวดล้อมที่จัดทำขึ้นด้วยอินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Everything : IoE)

ภาพที่ 4.2 ภาพอุปกรณ์ต้นแบบระบบตรวจวัดสภาพแวดล้อมที่จัดทำขึ้นด้วยอินเทอร์เน็ตประสานสรรพสิ่ง (Internet of Everything : IoE)

## 4.2 การประเมินประสิทธิภาพระบบการตรวจวัดค่าและระบบการแจ้งผ่านแอปพลิเคชันไลน์

ทำการติดตั้งอุปกรณ์ในห้องเซิร์ฟเวอร์จำลอง ขนาดห้องกว้าง 2.40 x ยาว 2.40 เมตร ใช้เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ 2 ชุด ใช้โปรแกรมPower MAX ให้ซีพียู (CPU) และ กราฟิกการ์ด (GPU) ของเครื่องคอมพิวเตอร์ทำงาน 100% นำอุปกรณ์ตรวจวัดทำการติดตั้งได้แก่ เซนเซอร์ตรวจจับไฟฟ้าดับโดยทำการเสียบปลั๊กที่เต้ารับก่อนเข้าเครื่องสำรองไฟฟ้าของห้องเซิร์ฟเวอร์ 1 จุด เซนเซอร์ตรวจวัดอุณหภูมิและความชื้น 2จุด ติดตั้งสำหรับตรวจวัดอุณหภูมิภายในห้อง 1 จุด และสำหรับตรวจวัดอุณหภูมิหลังเครื่องเซิร์ฟเวอร์ 1 จุด เซนเซอร์ตรวจวัดระดับน้ำติดตั้งในถาดจำลองความสูง 3 เซนติเมตร 1 จุด และในการตรวจวัดอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่าย โดยใช้อุปกรณ์ที่พัฒนาขึ้น เทียบกับเครื่องวัดเทียบจำนวน 2 อุปกรณ์ ได้แก่ 1. เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 สามารถตรวจวัดอุณหภูมิและความชื้นในบริเวณภายในห้อง ไม่สามารถตรวจวัดเฉพาะจุดหลังเครื่องความพิวเตอร์แม่ข่ายได้ และ 2. เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 สามารถตรวจวัดอุณหภูมิแต่ไม่สามารถตรวจวัดความชื้นได้ โดยกำหนดเงื่อนไขในการทดสอบระบบดังนี้

ตารางที่ 4.1 เงื่อนไขการทดสอบระบบที่พัฒนา

| ลำดับ | สถานการณ์ | การทำงานของพัดลม | แจ้งเตือนแอปพลิเคชันไลน์ |
|---|---|---|---|
| 1 | - ไฟฟ้าดับ<br>- อุณหภูมิภายในห้องเซิร์ฟเวอร์ต่ำกว่า 25°C | พัดลมไม่ทำงาน | แจ้งเตือนไฟฟ้าดับ |
| 2 | - ไฟฟ้าดับ<br>- อุณหภูมิภายในห้องเซิร์ฟเวอร์ 25°C ถึง 27°C | พัดลมทำงานความแรงระดับ 1 | แจ้งเตือนอุณหภูมิห้องเกินกำหนด |
| 3 | - ไฟฟ้าดับ<br>- อุณหภูมิภายในห้องเซิร์ฟเวอร์ 28°C ถึง 30°C | พัดลมทำงานความแรงระดับ 2 | ไม่แจ้งเตือน |

ตารางที่ 4.1 (ต่อ)

| ลำดับ | สถานการณ์ | การทำงานของพัดลม | แจ้งเตือนแอปพลิเคชันไลน์ |
|---|---|---|---|
| 4 | - ไฟฟ้าดับ<br>- อุณหภูมิภายในห้องเซิร์ฟเวอร์สูง 31°C | พัดลมทำงานความแรงระดับ 3 | ไม่แจ้งเตือน |
| 5 | - ไฟฟ้ากลับมาใช้งานได้ปกติ<br>- อุณหภูมิภายในห้องเซิร์ฟเวอร์สูง 31°C | พัดลมทำงานความแรงระดับ 3 | แจ้งเตือนไฟฟ้าใช้งานได้ปกติ |
| 6 | - ไฟฟ้ากลับมาใช้งานได้ปกติ<br>- อุณหภูมิภายในห้องเซิร์ฟเวอร์ต่ำกว่า 25°C | พัดลมไม่ทำงาน | แจ้งเตือนอุณหภูมิห้องปกติ |
| 7 | อุณหภูมิหลังเครื่องเซิร์ฟเวอร์ต่ำกว่า 60°C | พัดลมไม่ทำงาน | แจ้งเตือนอุณหภูมิเครื่องแม่ข่าย เกินกำหนด |
| 8 | อุณหภูมิหลังเครื่องเซิร์ฟเวอร์สูง 60°C | พัดลมไม่ทำงาน | แจ้งเตือนอุณหภูมิเครื่องแม่ข่าย ปกติ |
| 9 | ระดับน้ำในถาดรองน้ำอยู่ในช่วง 0 - 650 Ohm | พัดลมไม่ทำงาน | แจ้งเตือนระดับน้ำหยด |
| 10 | ระดับน้ำในถาดรองน้ำอยู่ในช่วง 651 - 1024 Ohm | พัดลมไม่ทำงาน | แจ้งเตือนไม่มีน้ำหยด |

หลังจากนั้นทำการทดสอบระบบตรวจวัดในสถานะการณ์ต่าง ๆ และบันทึกข้อมูล พร้อมกับหาค่าความผิดพลาดของระบบที่พัฒนา ดังนี้

สถานการณ์ทดลองที่ 1 ไฟฟ้าและเครื่องปรับอากาศทำงานปกติ

ทำการตรวจสอบอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่ายในสภาวะไฟฟ้าทำงานปกติ ที่อุณหภูมิห้องต่ำกว่า 25°C และทำการบันทึกข้อมูลของอุณหภูมิและความชื้นภายในห้อง และอุณหภูมิและความชื้นด้านหลังเครื่องแม่ข่าย

ตารางที่ 4.2 สถานการณ์ทดลองที่ 1ไฟฟ้าและเครื่องปรับอากาศทำงานปกติ

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและ<br>ความชื้นภายในห้อง | | <br>อุณหภูมิห้อง<br> <br>ความชื้นภายในห้อง |
| อุณหภูมิและ<br>ความชื้นด้านหลัง<br>เครื่องเซิร์ฟเวอร์ | | <br> |

ตารางที่ 4.3 การตรวจวัดอุณหภูมิไฟฟ้าและเครื่องปรับอากาศทำงานปกติ

| **รายการตรวจวัด** | **สถานะการทำงานของระบบที่พัฒนา (A)** | **แสดงค่าตรวจวัดบน Dashboard** | **เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B)** | **ค่าความผิดพลาด ((A-B/B)X100)** | **เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C)** | **ค่าความผิดพลาด ((A-C/C)X100)** |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | - อุณหภูมิ 23.4°C<br>- ความชื้น 52.3% | - อุณหภูมิ 23.4°C<br>- ความชื้น 52.3% | - อุณหภูมิ 23.5°C<br>- ความชื้น 49% | - อุณหภูมิ -0.42 %<br>- ความชื้น 6.73 % | - อุณหภูมิ 23.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ -0.42 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |
| อุณหภูมิและความชื้นหลังเครื่องเซิร์ฟเวอร์ | - อุณหภูมิ 37.5°C<br>- ความชื้น 26.3% | - อุณหภูมิ 37.5°C<br>- ความชื้น 26.3% | ไม่สามารถตรวจวัดได้ | ไม่สามารถหาค่าความผิดพลาดได้ | - อุณหภูมิ 37.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ 0%<br>-ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.4 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าและเครื่องปรับอากาศทำงานปกติ

| รายการแจ้งเตือน | แอปพลิเคชันไลน์ | หมายเหตุ |
|---|---|---|
| ไฟฟ้าทำงานปกติ | ไม่แจ้งเตือน | ไฟฟ้าทำงานปกติ |
| อุณหภูมิภายในห้องเซิร์ฟเวอร์ | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 25°C |
| อุณหภูมิหลังเครื่องเซิร์ฟเวอร์ | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 60°C |
| น้ำล้นถาดรองเครื่องปรับอากาศ | ไม่แจ้งเตือน | 651 - 1024 Ohm (สภาวะปกติ) |

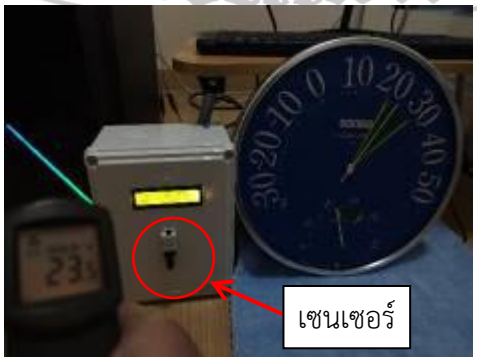
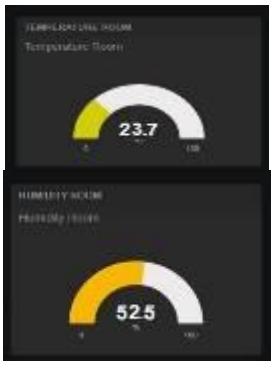
ตารางที่ 4.5 การทำงานของพัดลมขณะไฟฟ้าและเครื่องปรับอากาศทำงานปกติ

| รายการ | ระดับความแรงของพัดลม | หมายเหตุ |
|---|---|---|
| การทำงานของพัดลม | พัดลมไม่ทำงาน | อุณหภูมิภายในห้องเซิร์ฟเวอร์ต่ำกว่า 25°C |

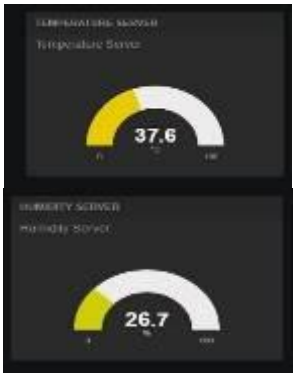
สถานการณ์ทดลองที่ 2 ไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน

ทำการตรวจสอบอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่าย พร้อมทั้งตรวจสอบการแจ้งเตือนผ่านไลน์แอปพลิเคชั่น ขณะไฟฟ้าดับในทันที

ตารางที่ 4.6 สถานการณ์ทดลองที่ 2 ไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน

| อุปกรณ์การตรวจวัด | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง |  |  |

ตารางที่ 4.6 (ต่อ)

| อุปกรณ์การตรวจวัด | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | |  |

ตารางที่ 4.7 การตรวจวัดอุณหภูมิไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | - อุณหภูมิ 23.7°C<br>- ความชื้น 52.5% | - อุณหภูมิ 23.7°C<br>- ความชื้น 52.5 % | - อุณหภูมิ 24°C<br>- ความชื้น 50% | - อุณหภูมิ -1.25 %<br>- ความชื้น 5 % | - อุณหภูมิ 23.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ 0.85%<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.7 (ต่อ)

| **รายการตรวจวัด** | **สถานะการทำงานของระบบที่พัฒนา (A)** | **แสดงค่าตรวจวัดบน Dashboard** | **เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B)** | **ค่าความผิดพลาด ((A-B/B)X100)** | **เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C)** | **ค่าความผิดพลาด ((A-C/C)X100)** |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | - อุณหภูมิ 37.6°C<br>- ความชื้น 26.7% | - อุณหภูมิ 37.6°C<br>- ความชื้น 26.7% | ไม่สามารถตรวจวัดได้ | ไม่สามารถหาค่าความผิดพลาดได้ | - อุณหภูมิ 37°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ 1.62%<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.8 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน

| **รายการแจ้งเตือน** | **แอปพลิเคชันไลน์** | **หมายเหตุ** |
|---|---|---|
| ไฟฟ้าดับ | แจ้งเตือน | ไฟฟ้าดับในทันที |
| อุณหภูมิภายในห้องเซิร์ฟเวอร์ | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 25°C |
| อุณหภูมิหลังเครื่องแม่ข่าย | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 60°C |
| น้ำล้นถาดรองเครื่องปรับอากาศ | ไม่แจ้งเตือน | 651 - 1024 Ohm (สภาวะปกติ) |

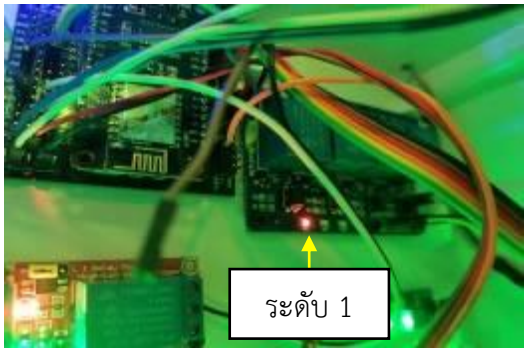
ตารางที่ 4.9 การทำงานของพัดลมขณะไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน

| รายการ | ระดับความแรงของพัดลม | หมายเหตุ |
|---|---|---|
| การทำงานของพัดลม | พัดลมไม่ทำงาน | อุณหภูมิภายในห้องเซิร์ฟเวอร์ต่ำกว่า 25°C |

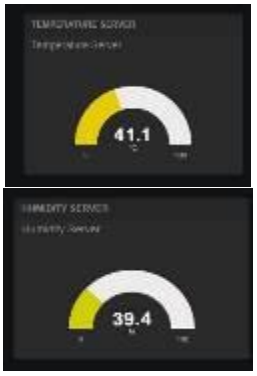
สถานการณ์ทดลองที่ 3 ไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุดการทำงาน

ทำการตรวจสอบอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่าย พร้อมปรับระดับความแรงของพัดลมอัตโนมัติ เมื่อไฟฟ้าดับเป็นเวลา 10 นาที

ตารางที่ 4.10 สถานการณ์ทดลองที่ 3 ไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุดการทำงาน

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | | |
| |  | |

ตารางที่ 4.10 (ต่อ)

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | |  |

ตารางที่ 4.11 การตรวจวัดอุณหภูมิไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุดการทำงาน

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | - อุณหภูมิ 26°C<br>- ความชื้น 53.2% | - อุณหภูมิ 26°C<br>- ความชื้น 53.2% | - อุณหภูมิ 26.5°C<br>- ความชื้น 52% | - อุณหภูมิ -1.88 %<br>- ความชื้น 2.3 % | - อุณหภูมิ 26.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ -1.88 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.11 (ต่อ)

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | - อุณหภูมิ 41.1°C - ความชื้น 39.4% | - อุณหภูมิ 41.1°C - ความชื้น 39.4% | ไม่สามารถตรวจวัดได้ | ไม่สามารถหาค่าความผิดพลาดได้ | - อุณหภูมิ 40.5°C - ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ 1.48 % - ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.12 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุดการทำงาน

| รายการแจ้งเตือน | แอปพลิเคชันไลน์ | หมายเหตุ |
|---|---|---|
| ไฟฟ้าดับ | ไม่แจ้งเตือน | ทำการแจ้งเตือนไฟฟ้าดับในครั้งแรก |
| อุณหภูมิภายในห้องเซิร์ฟเวอร์ | แจ้งเตือน | อุณหภูมิตั้งแต่ 25°C |
| อุณหภูมิหลังเครื่องแม่ข่าย | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 60°C |
| น้ำล้นถาดรองเครื่องปรับอากาศ | ไม่แจ้งเตือน | 651 - 1024 Ohm (สภาวะปกติ) |

ตารางที่ 4.13 การทำงานของพัดลมขณะไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุด

การทำงาน

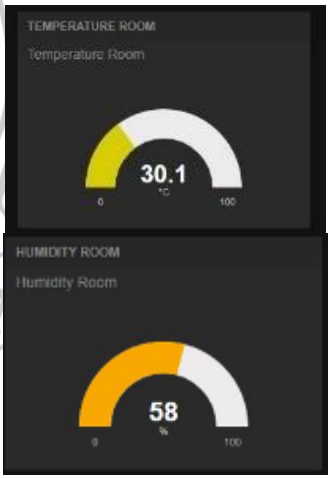
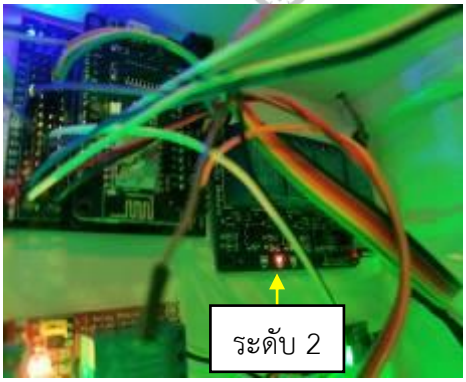
| รายการ | ระดับความแรงของพัดลม | หมายเหตุ |
|---|---|---|
| การทำงานของพัดลม | ทำงาน ความแรงระดับ 1 | อุณหภูมิ 25°C ถึง 27°C |

สถานการณ์ทดลองที่ 4 ไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุดการทำงาน

ทำการตรวจสอบอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่าย พร้อมปรับระดับความแรงของพัดลมอัตโนมัติ เมื่อไฟฟ้าดับเป็นเวลา 30 นาที

ตารางที่ 4.14 สถานการณ์ทดลองที่ 4 ไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุด

การทำงาน

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | |  |
| |  | |

ตารางที่ 4.14 (ต่อ)

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | | |

ตารางที่ 4.15 การตรวจวัดอุณหภูมิไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุดการทำงาน

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | - อุณหภูมิ 30.1°C<br>- ความชื้น 58% | - อุณหภูมิ 30.1°C<br>- ความชื้น 58% | - อุณหภูมิ 30°C<br>- ความชื้น 55% | - อุณหภูมิ 0.33 %<br>- ความชื้น 5.45% | - อุณหภูมิ 30.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ -1.31 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.15 (ต่อ)

| **รายการตรวจวัด** | **สถานะการทำงานของระบบที่พัฒนา (A)** | **แสดงค่าตรวจวัดบน Dashboard** | **เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B)** | **ค่าความผิดพลาด ((A-B/B)X100)** | **เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C)** | **ค่าความผิดพลาด ((A-C/C)X100)** |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | - อุณหภูมิ 41°C<br>- ความชื้น 26.9 % | - อุณหภูมิ 41°C<br>- ความชื้น 26.9 % | ไม่สามารถตรวจวัดได้ | ไม่สามารถหาค่าความผิดพลาดได้ | - อุณหภูมิ 41°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ 0 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.16 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุดการทำงาน

| **รายการแจ้งเตือน** | **แอปพลิเคชันไลน์** | **หมายเหตุ** |
|---|---|---|
| ไฟฟ้าดับ | ไม่แจ้งเตือน | ทำการแจ้งเตือนไฟฟ้าดับในครั้งแรก |
| อุณหภูมิภายในห้องเซิร์ฟเวอร์ | ไม่แจ้งเตือน | ทำการแจ้งเตือนอุณหภูมิตั้งแต่ 25°C ในครั้งแรก |
| อุณหภูมิหลังเครื่องแม่ข่าย | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 60°C |
| น้ำล้นถาดรองเครื่องปรับอากาศ | ไม่แจ้งเตือน | 651 - 1024 Ohm (สภาวะปกติ) |

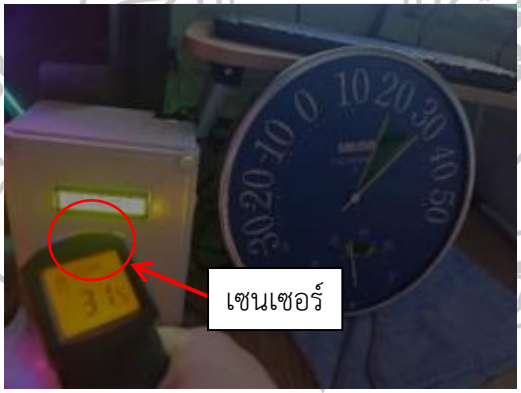
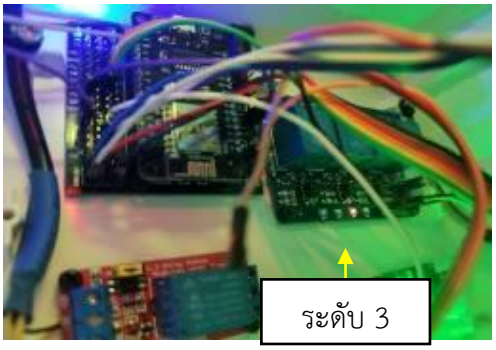
ตารางที่ 4.17 การทำงานของพัดลมขณะไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุดการทำงาน

| รายการ | ระดับความแรงของพัดลม | หมายเหตุ |
|---|---|---|
| การทำงานของพัดลม | ทำงาน ความแรงระดับ 2 | อุณหภูมิ 28°C ถึง 30°C |

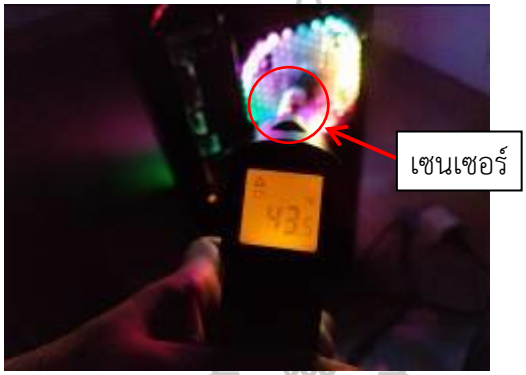
สถานการณ์ทดลองที่ 5 ไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุดการทำงาน

ทำการตรวจสอบอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่าย พร้อมปรับระดับความแรงของพัดลมอัตโนมัติ เมื่อไฟฟ้าดับเป็นเวลา 50 นาที

ตารางที่ 4.18 สถานการณ์ทดลองที่ 5 ไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุดการทำงาน

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง |   | |

ตารางที่ 4.18 (ต่อ)

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย |  | |

ตารางที่ 4.19 การตรวจวัดอุณหภูมิไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุดการทำงาน

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | - อุณหภูมิ 31.6°C<br>- ความชื้น 61% | - อุณหภูมิ 31.6°C<br>- ความชื้น 61% | - อุณหภูมิ 31.5°C<br>- ความชื้น 60% | - อุณหภูมิ 0.32 %<br>- ความชื้น 1.67% | - อุณหภูมิ 31.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ 0.32 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.19 (ต่อ)

| **รายการตรวจวัด** | **สถานะการทำงานของระบบที่พัฒนา (A)** | **แสดงค่าตรวจวัดบน Dashboard** | **เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B)** | **ค่าความผิดพลาด ((A-B/B)X100)** | **เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C)** | **ค่าความผิดพลาด ((A-C/C)X100)** |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | - อุณหภูมิ 43.7°C<br>- ความชื้น 30.2 % | - อุณหภูมิ 43.7°C<br>- ความชื้น 30.2 % | ไม่สามารถตรวจวัดได้ | ไม่สามารถหาค่าความผิดพลาดได้ | - อุณหภูมิ 43.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ 0.46 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.20 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุดการทำงาน

| **รายการแจ้งเตือน** | **แอปพลิเคชันไลน์** | **หมายเหตุ** |
|---|---|---|
| ไฟฟ้าดับ | ไม่แจ้งเตือน | ทำการแจ้งเตือนไฟฟ้าดับในครั้งแรก |
| อุณหภูมิภายในห้องเซิร์ฟเวอร์ | ไม่แจ้งเตือน | ทำการแจ้งเตือนอุณหภูมิตั้งแต่ 25°C ในครั้งแรก |
| อุณหภูมิหลังเครื่องแม่ข่าย | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 60°C |
| น้ำล้นถาดรองเครื่องปรับอากาศ | ไม่แจ้งเตือน | 651 - 1024 Ohm (สภาวะปกติ) |

ตารางที่ 4.21 การทำงานของพัดลมขณะไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุดการทำงาน

| รายการ | ระดับความแรงของพัดลม | หมายเหตุ |
|---|---|---|
| การทำงานของพัดลม | ทำงาน ความแรงระดับ 3 | อุณหภูมิตั้งแต่ 31°C |

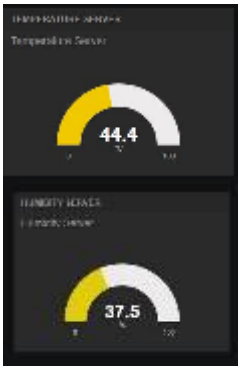
สถานการณ์ทดลองที่ 6 ไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศหยุดการทำงาน

ทำการตรวจสอบอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่าย พร้อมปรับระดับความแรงของพัดลมอัตโนมัติ เมื่อไฟฟ้าดับเป็นเวลา 1 ชั่วโมง 50 นาที

ตารางที่ 4.22 สถานการณ์ทดลองที่ 6 ไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศหยุดการทำงาน

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | | |

ตารางที่ 4.22 (ต่อ)

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | |  |

ตารางที่ 4.23 การตรวจวัดอุณหภูมิไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศหยุดการทำงาน

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | - อุณหภูมิ 33.1°C - ความชื้น 63.5% | - อุณหภูมิ 33.1°C - ความชื้น 63.5% | - อุณหภูมิ 33°C - ความชื้น 62% | - อุณหภูมิ 0.31 % - ความชื้น 2.42% | - อุณหภูมิ 33.5°C - ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ - 1.19 % - ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.23 (ต่อ)

| **รายการตรวจวัด** | **สถานะการทำงานของระบบที่พัฒนา (A)** | **แสดงค่าตรวจวัดบน Dashboard** | **เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B)** | **ค่าความผิดพลาด ((A-B/B)X100)** | **เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C)** | **ค่าความผิดพลาด ((A-C/C)X100)** |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | - อุณหภูมิ 44.4°C<br>- ความชื้น 37.5% | - อุณหภูมิ 44.4°C<br>- ความชื้น 37.5% | ไม่สามารถตรวจวัดได้ | ไม่สามารถหาค่าความผิดพลาดได้ | - อุณหภูมิ 44.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ – 0.22 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.24 การแจ้งเตือนผ่านแอปพลิชันไลน์ขณะไฟฟ้าดับเป็นเวลา 50 - 150 นาที และเครื่องปรับอากาศหยุดการทำงาน

| **รายการแจ้งเตือน** | **แอปพลิเคชันไลน์** | **หมายเหตุ** |
|---|---|---|
| ไฟฟ้าดับ | ไม่แจ้งเตือน | ทำการแจ้งเตือนไฟฟ้าดับในครั้งแรก |
| อุณหภูมิภายในห้องเซิร์ฟเวอร์ | ไม่แจ้งเตือน | ทำการแจ้งเตือนอุณหภูมิตั้งแต่ 25°C ในครั้งแรก |
| อุณหภูมิหลังเครื่องแม่ข่าย | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 60°C |
| น้ำล้นถาดรองเครื่องปรับอากาศ | ไม่แจ้งเตือน | 651 - 1024 Ohm (สภาวะปกติ) |

ตารางที่ 4.25 การทำงานของพัดลมขณะไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศหยุดการทำงาน

| รายการ | ระดับความแรงของพัดลม | หมายเหตุ |
|---|---|---|
| การทำงานของพัดลม | ทำงาน ความแรงระดับ 3 | อุณหภูมิตั้งแต่ 31°C |

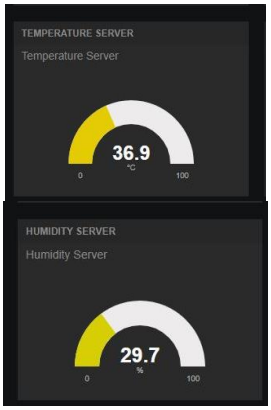
สถานการณ์ทดลองที่ 7 ไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน

ทำการตรวจสอบอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่าย พร้อมทั้งตรวจสอบการแจ้งเตือนผ่านไลน์แอปพลิเคชั่น เมื่อไฟฟ้าเข้าสู่สภาวะปกติในทันที

ตารางที่ 4.26 สถานการณ์ทดลองที่ 7 ไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | | |

ตารางที่ 4.26 (ต่อ)

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | |  |

ตารางที่ 4.27 การตรวจวัดอุณหภูมิเมื่อไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | - อุณหภูมิ 28.3°C<br>- ความชื้น 53.2% | - อุณหภูมิ 28.3°C<br>- ความชื้น 53.2% | - อุณหภูมิ 28°C<br>- ความชื้น 50% | - อุณหภูมิ 1.07 %<br>- ความชื้น 6.4% | - อุณหภูมิ 28°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ 1.07 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.27 (ต่อ)

| **รายการตรวจวัด** | **สถานะการทำงานของระบบที่พัฒนา (A)** | **แสดงค่าตรวจวัดบน Dashboard** | **เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B)** | **ค่าความผิดพลาด ((A-B/B)X100)** | **เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C)** | **ค่าความผิดพลาด ((A-C/C)X100)** |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | - อุณหภูมิ 36.9°C<br>- ความชื้น 29.7% | - อุณหภูมิ 36.9°C<br>- ความชื้น 29.7% | ไม่สามารถตรวจวัดได้ | ไม่สามารถหาค่าความผิดพลาดได้ | - อุณหภูมิ 37°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ - 0.27 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.28 การแจ้งเตือนผ่านแอปพลิชันไลน์เมื่อไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน

| **รายการแจ้งเตือน** | **แอปพลิเคชันไลน์** | **หมายเหตุ** |
|---|---|---|
| ไฟฟ้าภายในห้องเซิร์ฟเวอร์ | แจ้งเตือนไลน์ ไฟฟ้าเข้าสู่สภาวะปกติ | ทำการแจ้งเตือนเมื่อไฟฟ้าทำงาน |
| อุณหภูมิภายในห้องเซิร์ฟเวอร์ | ไม่แจ้งเตือน | ทำการแจ้งเตือนอุณหภูมิตั้งแต่ 25°C ในครั้งแรก |
| อุณหภูมิหลังเครื่องแม่ข่าย | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 60°C |
| น้ำล้นถาดรองเครื่องปรับอากาศ | ไม่แจ้งเตือน | 651 - 1024 Ohm (สภาวะปกติ) |

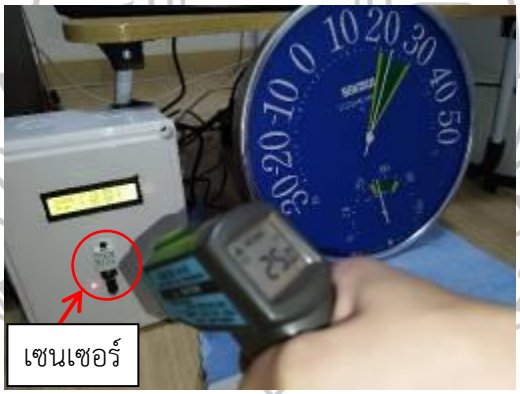
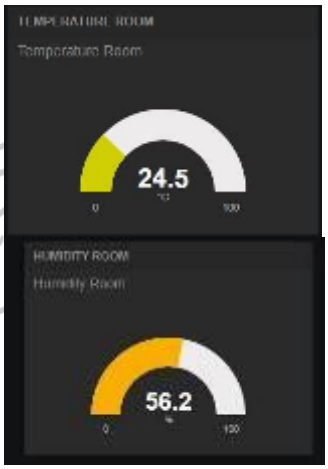
ตารางที่ 4.29 การทำงานของพัดลมเมื่อไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน

| รายการ | ระดับความแรงของพัดลม | หมายเหตุ |
|---|---|---|
| การทำงานของพัดลม | ทำงาน ความแรงระดับ 2 | อุณหภูมิตั้งแต่ 28°C |

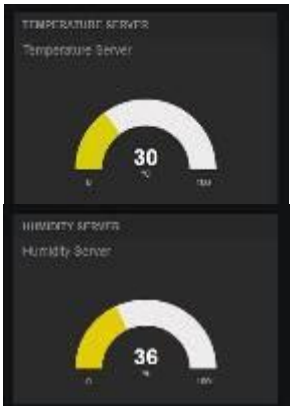
สถานการณ์ทดลองที่ 8 ไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติต่ำกว่า 25°C และเครื่องปรับอากาศทำงาน

ทำการตรวจสอบอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์และเครื่องแม่ข่าย เมื่อไฟฟ้าเข้าสู่สภาวะปกติอุณหภูมิต่ำกว่า 25°C ใช้เวลาประมาณ 30 นาที

ตารางที่ 4.30 สถานการณ์ทดลองที่ 8 ไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติต่ำกว่า 25°C และเครื่องปรับอากาศทำงาน

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง |  |  |

ตารางที่ 4.30 (ต่อ)

| จุดตรวจวัดอุณหภูมิ | การตรวจวัดอุณหภูมิ | Dashboard |
|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | |   |

ตารางที่ 4.31 การตรวจวัดไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติต่ำกว่า 25°C และเครื่องปรับอากาศทำงาน

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นภายในห้อง | - อุณหภูมิ 24.5°C<br>- ความชื้น 56.2% | - อุณหภูมิ 24.5°C<br>- ความชื้น 56.2% | - อุณหภูมิ 24.2°C<br>- ความชื้น 55% | - อุณหภูมิ 1.24 %<br>- ความชื้น 2.18% | - อุณหภูมิ 25°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ -2 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.31 (ต่อ)

| รายการตรวจวัด | สถานะการทำงานของระบบที่พัฒนา (A) | แสดงค่าตรวจวัดบน Dashboard | เครื่องวัดเทียบ 1 เครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 (B) | ค่าความผิดพลาด ((A-B/B)X100) | เครื่องวัดเทียบ 2 เครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 (C) | ค่าความผิดพลาด ((A-C/C)X100) |
|---|---|---|---|---|---|---|
| อุณหภูมิและความชื้นหลังเครื่องแม่ข่าย | - อุณหภูมิ 30°C<br>- ความชื้น 36% | - อุณหภูมิ 30°C<br>- ความชื้น 36% | ไม่สามารถตรวจวัดได้ | ไม่สามารถหาค่าความผิดพลาดได้ | - อุณหภูมิ 30.5°C<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | - อุณหภูมิ -1.64 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น |

ตารางที่ 4.32 การแจ้งเตือนผ่านแอปพลิชันไลน์เมื่อไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติต่ำกว่า 25°C และเครื่องปรับอากาศทำงาน

| รายการแจ้งเตือน | แอปพลิเคชันไลน์ | หมายเหตุ |
|---|---|---|
| ไฟฟ้าภายในห้องเซิร์ฟเวอร์ | ไม่แจ้งเตือน | ทำการแจ้งเตือนเมื่อไฟฟ้ากลับมาทำงานปกติ |
| อุณหภูมิภายในห้องเซิร์ฟเวอร์ | แจ้งเตือนอุณหภูมิเข้าสู่สภาวะปกติ | ทำการแจ้งเตือนอุณหภูมิตั้งแต่ 25°C ในครั้งแรก |
| อุณหภูมิหลังเครื่องแม่ข่าย | ไม่แจ้งเตือน | อุณหภูมิต่ำกว่า 60°C |
| น้ำล้นถาดรองเครื่องปรับอากาศ | ไม่แจ้งเตือน | 651 - 1024 Ohm (สภาวะปกติ) |

ตารางที่ 4.33 ตารางการทำงานของพัดลมเมื่อไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติ ต่ำกว่า 25°C และเครื่องปรับอากาศทำงาน

| รายการ | ระดับความแรงของพัดลม | หมายเหตุ |
|---|---|---|
| การทำงานของพัดลม | พัดลมหยุดทำงาน | อุณหภูมิต่ำกว่า 25°C |

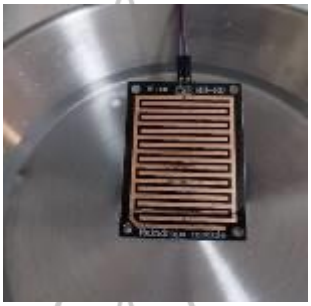
สถานการณ์ทดลองที่ 9 น้ำหยดจากเครื่องปรับอากาศ

ทำการตรวจวัดระดับน้ำ โดยนำถาดรองน้ำความสูง 3 เซนติเมตรมาทดสอบ โดยเกณฑ์การวัดระดับน้ำจะต้องอยู่ในช่วง 651 - 1024 Ohm ถ้าหากต่ำกว่า 651 Ohm จะทำการแจ้งเตือนผ่านไลน์แอปพลิเคชั่น

ตารางที่ 4.34 สถานการณ์ทดลองที่ 9 น้ำหยดจากเครื่องปรับอากาศ

| จุดตรวจวัดระดับน้ำ | การตรวจวัดระดับน้ำ | Dashboard |
|---|---|---|
| สภาวะปกติ ไม่มีการหยดของน้ำ | | |
| | | |

ตารางที่ 4.34 (ต่อ)

| จุดตรวจวัดระดับน้ำ | การตรวจวัดระดับน้ำ | Dashboard |
|---|---|---|
| มีการหยดของน้ำ |  | |
| กลับสู่สภาวะปกติ ไม่มีการหยดของน้ำ | | |

ตารางที่ 4.35 การตรวจวัดระดับน้ำในสภาวะปกติ และเมื่อมีน้ำหยด

| รายการตรวจวัด | การแจ้งเตือนแอปพลิเคชันไลน์ | แสดงค่าตรวจวัดบน Dashboard |
|---|---|---|
| สภาวะปกติ ไม่มีการหยดของน้ำ | ไม่มีการแจ้งเตือน | 651 - 1024 Ohm |
| มีการหยดของน้ำ | แจ้งเตือน | 0 - 650 Ohm |
| กลับสู่สภาวะปกติ ไม่มีการหยดของน้ำ | แจ้งเตือนไม่มีน้ำหยด เข้าสู่สภาวะปกติ | 651 - 1024 Ohm |

**สรุปวิธีการทดสอบระบบที่พัฒนา**

จากการทดสอบนำเครื่องมือมาเปรียบเทียบความเที่ยงตรงของระบบที่พัฒนากับเครื่องวัดอุณหภูมิแบบแอนะล็อก รุ่น Sekisul SM-300 และเครื่องวัดอุณหภูมิแบบอินฟราเรด รุ่น DIGICON DP-88 การทำงานของระบบที่พัฒนา และการแจ้งเตือนผ่านแอปพลิเคชันไลน์สามารถทำได้ตามที่กำหนด ค่าความผิดพลาดของอุณหภูมิที่ยอมรับได้ +/-2% และค่าความชื้นที่ยอมรับได้+/-10% ผลการทดสอบระบบที่พัฒนาสามารถทำงานได้ครบทุกฟังก์ชันตามที่กำหนด ผลสรุปตามตารางดังนี้

ตารางที่ 4.36 สรุปผลการทดลองการตรวจจับไฟฟ้าดับ อุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์ และการทำงานของพัดลม

| สถานการณ์การทดลอง | ค่าความผิดพลาดของระบบที่พัฒนา | | การยอมรับค่าความผิดพลาด (อุณหภูมิ +/-2% / ค่าความชื้น +/-10%) | การแจ้งเตือนแอปพลิเคชันไลน์ | การทำงานของพัดลม | รายงานค่าบน Dashboard |
|---|---|---|---|---|---|---|
| | เครื่องวัดอุณหภูมิแบบแอนะล็อก | เครื่องวัดอุณหภูมิแบบอินฟราเรด | ระบบทำถูกต้องตามเกณฑ์ที่กำหนด (ผ่าน ✓ / ไม่ผ่าน ✗) | | | |
| ไฟฟ้าและเครื่องปรับอากาศทำงานปกติ | - อุณหภูมิ -0.42 %<br>- ความชื้น 6.73 % | - อุณหภูมิ -0.42 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ (ไม่แจ้งเตือน) | ✓ (ไม่ทำงาน) | ✓ |

ตารางที่ 4.36 (ต่อ)

| สถานการณ์การทดลอง | ค่าความผิดพลาดของระบบที่พัฒนา | | การยอมรับค่าความผิดพลาด (อุณหภูมิ +/-2% / ค่าความชื้น +/-10%) | การแจ้งเตือนแอปพลิเคชันไลน์ | การทำงานของพัดลม | รายงานค่าบน Dashboard |
|---|---|---|---|---|---|---|
| | เครื่องวัดอุณหภูมิแบบแอนะล็อก | เครื่องวัดอุณหภูมิแบบอินฟราเรด | ระบบทำถูกต้องตามเกณฑ์ที่กำหนด (ผ่าน ✓ / ไม่ผ่าน ✗) | | | |
| ไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ -1.25 %<br>- ความชื้น 5 % | - อุณหภูมิ 0.85%<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ (แจ้งเตือนไฟฟ้าดับ) | ✓ (ไม่ทำงาน) | ✓ |
| ไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ -1.88 %<br>- ความชื้น 2.3 % | - อุณหภูมิ -1.88 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ (แจ้งเตือนอุณหภูมิสูงตั้งแต่ 25°C) | ✓ (ทำงานระดับ 1) | ✓ |
| ไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ 0.33 %<br>- ความชื้น 5.45% | - อุณหภูมิ -1.31 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ (ไม่แจ้งเตือน) | ✓ (ทำงานระดับ 2) | ✓ |

ตารางที่ 4.36 (ต่อ)

| สถานการณ์การทดลอง | ค่าความผิดพลาดของระบบที่พัฒนา | | การยอมรับค่าความผิดพลาด (อุณหภูมิ +/-2% / ค่าความชื้น +/-10%) | การแจ้งเตือนแอปพลิเคชันไลน์ | การทำงานของพัดลม | รายงานค่าบน Dashboard |
|---|---|---|---|---|---|---|
| | เครื่องวัดอุณหภูมิแบบแอนะล็อก | เครื่องวัดอุณหภูมิแบบอินฟราเรด | ระบบทำถูกต้องตามเกณฑ์ที่กำหนด (ผ่าน ✓ / ไม่ผ่าน ✗) | | | |
| ไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ 0.32 %<br>- ความชื้น 1.67% | - อุณหภูมิ 0.32 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ (ไม่แจ้งเตือน) | ✓ (ทำงานระดับ 3) | ✓ |
| ไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ 0.31 %<br>- ความชื้น 2.42% | - อุณหภูมิ - 1.19 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ (ไม่แจ้งเตือน) | ✓ (ทำงานระดับ 3) | ✓ |
| ไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน | - อุณหภูมิ 1.07 %<br>- ความชื้น 6.4% | - อุณหภูมิ 1.07 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ (แจ้งเตือนไฟฟ้าเข้าสู่สภาวะปกติ) | ✓ (ทำงานระดับ 2) | ✓ |

ตารางที่ 4.36 (ต่อ)

| สถานการณ์การทดลอง | ค่าความผิดพลาดของระบบที่พัฒนา | | การยอมรับค่าความผิดพลาด (อุณหภูมิ +/-2% / ค่าความชื้น +/-10%) | การแจ้งเตือนแอปพลิเคชันไลน์ | การทำงานของพัดลม | รายงานค่าบน Dashboard |
|---|---|---|---|---|---|---|
| | เครื่องวัดอุณหภูมิแบบแอนะล็อก | เครื่องวัดอุณหภูมิแบบอินฟราเรด | ระบบทำถูกต้องตามเกณฑ์ที่กำหนด (ผ่าน ✓ / ไม่ผ่าน ✗) | | | |
| ไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติต่ำกว่า 25°C และเครื่องปรับอากาศทำงาน | - อุณหภูมิ 1.24 %<br>- ความชื้น 2.18% | - อุณหภูมิ -2 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ (แจ้งเตือนอุณหภูมิเข้าสู่สภาวะปกติ) | ✓ (หยุดการทำงาน) | ✓ |

ตารางที่ 4.37 สรุปผลการทดลองการตรวจจับอุณหภูมิและความชื้นของเครื่องแม่ข่าย

| สถานการณ์การทดลอง | ค่าความผิดพลาดของระบบที่พัฒนาเปรียบเทียบกับเครื่องวัดอุณหภูมิแบบอินฟราเรด | การยอมรับค่าความผิดพลาด (อุณหภูมิ +/-2% / ค่าความชื้น +/-10%) | การแจ้งเตือนแอปพลิเคชันไลน์ (ไม่แจ้งเตือนอุณภูมิสูงไม่เกิน 60°C) | รายงานค่าบน Dashboard |
|---|---|---|---|---|
| | | ระบบทำถูกต้องตามเกณฑ์ที่กำหนด (ผ่าน ✓ / ไม่ผ่าน ✗) | | |
| ไฟฟ้าและเครื่องปรับอากาศทำงานปกติ | - อุณหภูมิ 0%<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ | ✓ |
| ไฟฟ้าดับและเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ 1.62%<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ | ✓ |
| ไฟฟ้าดับเป็นเวลา 0 - 10 นาที และเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ 1.48 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ | ✓ |
| ไฟฟ้าดับเป็นเวลา 10 - 30 นาที และเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ 0 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ | ✓ |
| ไฟฟ้าดับเป็นเวลา 30 - 50 นาที และเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ 0.46 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ | ✓ |

ตารางที่ 4.37 (ต่อ)

| สถานการณ์การทดลอง | ค่าความผิดพลาดของระบบที่พัฒนาเปรียบเทียบกับเครื่องวัดอุณหภูมิแบบอินฟราเรด | การยอมรับค่าความผิดพลาด (อุณหภูมิ +/-2% / ค่าความชื้น +/-10%) | การแจ้งเตือนแอปพลิเคชันไลน์ (ไม่แจ้งเตือนอุณภูมิสูงไม่เกิน 60°C) | รายงานค่าบน Dashboard |
|---|---|---|---|---|
| | | ระบบทำถูกต้องตามเกณฑ์ที่กำหนด (ผ่าน ✓ / ไม่ผ่าน ✗) | | |
| ไฟฟ้าดับเป็นเวลา 50 – 150 นาที และเครื่องปรับอากาศหยุดการทำงาน | - อุณหภูมิ – 0.22 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ | ✓ |
| ไฟฟ้าเข้าสู่สภาวะปกติและเครื่องปรับอากาศทำงาน | - อุณหภูมิ -0.27 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ | ✓ |
| ไฟฟ้าเข้าสู่สภาวะปกติ อุณหภูมิเข้าสู่สภาวะปกติต่ำกว่า 25°C และเครื่องปรับอากาศทำงาน | - อุณหภูมิ -1.64 %<br>- ไม่มีฟังก์ชันวัดค่าความชื้น | ✓ | ✓ | ✓ |

ตารางที่ 4.38 สรุปผลการทดลองการตรวจจับน้ำหยดจากเครื่องปรับอากาศ

| สถานการณ์การทดลอง | การแจ้งเตือนแอปพลิเคชันไลน์ | รายงานค่าบน Dashboard |
|---|---|---|
| | ระบบทำถูกต้องตามเกณฑ์ที่กำหนด (ผ่าน ✓ / ไม่ผ่าน ✗) | |
| การหยดของน้ำ | ✓ | ✓ |
| เมื่อน้ำหยุดหยด | ✓ | ✓ |

**วิจารณ์ผลการทดสอบ**

อุปกรณ์ที่พัฒนามีข้อจำกัดของเรื่องระยะเวลาจะต้องไม่เกิน 2 ชั่วโมง ตามระยะเวลาของเครื่องสำรองไฟฟ้าหากเกินกว่าเวลาที่กำหนดควรจะมีการปิดอุปกรณ์เพื่อป้องกันความเสียหาย และถ้าหากความร้อนมีอุณหภูมิที่สูงเกินกว่าจะควบคุมอุณหภูมิได้ให้ทำการดับอุปกรณ์ในห้องเซิร์ฟเวอร์

# บทที่ 5
# สรุปการศึกษาค้นคว้าอิสระ อภิปรายผล และข้อเสนอแนะ

จากการศึกษาค้นคว้าอิสระการเฝ้าสังเกตสภาพแวดล้อมห้องเซิร์ฟเวอร์ด้วยอินเทอร์เน็ตประสานสรรพสิ่งโดยใช้ ESP8266 และ NETPIE สามารถสรุปผลการศึกษาค้นคว้าอิสระ อภิปรายผล และข้อเสนอแนะ ดังต่อไปนี้

## 5.1 สรุปการศึกษาค้นคว้าอิสระ

การศึกษาค้นคว้าอิสระนี้เป็นต้นแบบการออกแบบและพัฒนาระบบการเฝ้าสังเกตสภาพแวดล้อมห้องเซิร์ฟเวอร์ด้วยอินเทอร์เน็ตประสานสรรพสิ่งโดยใช้ ESP8266 และ NETPIE โดยนำอุปกรณ์ NodeMCU ESP8266 มาควบคุมอุปกรณ์เซนเซอร์ต่าง ๆ และควบคุมรีเลย์สั่งการแบบอัตโนมัติ การตรวจจับไฟฟ้าดับเพื่อให้ทราบถึงการดับของไฟฟ้าอาจจะส่งผลกระทบการทำงานของอุปกรณ์ หากไฟฟ้าดับเป็นระยะเวลาที่นานเกินความสามารถของการสำรองไฟของเครื่องสำรองไฟทำให้ส่งผลต่อการหยุดทำงานของอุปกรณ์ต่าง ๆ ในห้องเซิร์ฟเวอร์ ใช้เซนเซอร์ตรวจวัดระดับน้ำในถาดรองน้ำของเครื่องปรับอากาศเพื่อให้รู้ถึงระดับน้ำของการหยดของน้ำจากเครื่องปรับอากาศ ใช้เซนเซอร์นำมาตรวจวัดค่าอุณหภูมิและความชื้นของห้องเซิร์ฟเวอร์ และปรับอุณหภูมิห้องด้วยการใช้รีเลย์ในการควบคุมพัดลมที่สั่งการปรับระดับความแรงแบบอัตโนมัติตามอุณหภูมิห้องที่ตรวจวัดได้ ตรวจวัดค่าอุณหภูมิเครื่องคอมพิวเตอร์แม่ข่ายเพื่อให้รู้ถึงอุณหภูมิการระบายความร้อนของเครื่อง และ ซึ่งสถานะต่าง ๆ ของอุปกรณ์จะทำการแจ้งเตือนไปยังผู้ดูแลระบบผ่านแอปพลิชันไลน์และแสดงรายงานผลของค่าบน NETPIE แพลตฟอร์ม จากการพัฒนาระบบสรุปได้ว่าระบบสามารถทำงานได้ทั้งหมด 9 ฟังก์ชัน ได้แก่ 1) ฟังก์ชันการตรวจจับไฟฟ้าดับ 2) ฟังก์ชันตรวจจับน้ำแอร์หยด 3) ฟังก์ชันตรวจจับอุณหภูมิของห้อง 4) ฟังก์ชันตรวจจับความชื้นของห้อง 5) ฟังก์ชันตรวจจับอุณหภูมิของโฮสต์ 6)ฟังก์ชันตรวจจับความชื้นของโฮสต์ 7) ฟังก์ชันควบคุมสำหรับพัดลม 8) ฟังก์ชันแจ้งเตือน และ 9) ฟังก์ชันรายงานสภาพแวดล้อมของห้อง สำหรับการศึกษาค้นคว้าในครั้งนี้เป็นไปตามวัตถุประสงค์ที่ได้กำหนดไว้ และประโยชน์ที่ได้รับมีดังต่อไปนี้

ประโยชน์ที่ได้รับจากอุปกรณ์ต้นแบบ ด้วยอินเทอร์เน็ตประสานสรรพสิ่ง โดยนำอุปกรณ์ NodeMCU ESP8266 มาควบคุมอุปกรณ์เซนเซอร์ต่าง ๆ ให้ตรวจวัดค่าสภาพแวดล้อมรวมทั้งการ

แจ้งเตือน ผู้ดูแลระบบสามารถรับรู้ถึงสถานการณ์ที่ผิดปกติของห้องเซิร์ฟเวอร์ และแก้ไขสถานการณ์ได้ทันท่วงทีก่อนที่อุปกรณ์ต่าง ๆ จะได้รับความเสียหาย

ประหยัดค่าใช้จ่ายในการจัดหาอุปกรณ์ จากการออกแบบและการเฝ้าสังเกตสภาพแวดล้อมห้องเซิร์ฟเวอร์ด้วยอินเทอร์เน็ตประสานสรรพสิ่งโดยใช้ ESP8266 และ NETPIE พบว่ามีค่าใช้จ่ายของอุปกรณ์ที่ถูกกว่าหากเราไปซื้อระบบที่สำเร็จรูปมาแล้ว ประหยัดค่าใช้จ่ายในส่วนของการลงทุนส่วนนี้อย่างมาก และประสิทธิภาพจากการทดลองระบบมีความสามารถตรงตามวัตถุประสงค์ที่ต้องการ

## 5.2 ข้อเสนอแนะ

การศึกษาค้นคว้าอิสระนี้เป็นต้นแบบของการพัฒนาระบบการเฝ้าสังเกตสภาพแวดล้อมห้องเซิร์ฟเวอร์ด้วยอินเทอร์เน็ตประสานสรรพสิ่งโดยใช้ ESP8266 และ NETPIE หากมีการนำไปพัฒนาเพิ่มเติมควรศึกษาสภาพแวดล้อมอื่น ๆ ที่มีผลกระทบต่ออุปกรณ์ภายในห้อง เพื่อนำมาปรับปรุงแก้ไขให้ดียิ่งขึ้น

การพัฒนาระบบเป็นต้นแบบของการตรวจวัดค่าสถาพแวดล้อม สามารถนำไปประยุกต์ใช้ในพื้นที่ต่าง ๆ ได้ ควรเพิ่มจำนวนจุดในการตรวจวัดค่าอุณหภูมิและความชื้น โดยคำนึงถึงขนาดของสถานที่ที่นำไปติดตั้ง เพื่อจะได้ค่าที่ถูกต้องและแม่นยำยิ่งขึ้น

# บรรณานุกรม


จันตพงษ์ บุตรลักษณ์. (2560). การพัฒนาระบบเฝ้าระวังสภาพแวดล้อมของเซิร์ฟเวอร์แร็ค, *แหล่งทุนสำนักคอมพิวเตอร์และเทคโนโลยีสารสนเทศมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ*.

ชนนิกานต์ รอดมรณ์, มธุรส ผ่านเมือง, วีรศักดิ์ จงเลขา. (2564). การประยุกต์ใช้เครือข่ายเซนเซอร์ไร้สายสำหรับฟาร์มอัจฉริยะ. *วิจัย มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย,* 13(2), 315-329.

ณัฏฐพล เจริญศิริ, ประชาสันต์ แว่นไธสง, กฤษณพล เกิดทองคำ. (2563). ระบบแจ้งเตือนอุณหภูมิและความชื้น สำหรับห้องควบคุมไฟฟ้า สำหรับบริษัทสงวนวงษ์อุตสาหกรรม จำกัด, *วิทยาศาสตร์และเทคโนโลยีแห่งรัตนโกสินทร์,* 2(3), 119-132.

ธานิล ม่วงพูล, วงศกร. (2562). อุดมพัดลมอัจฉริยะที่ควบคุมด้วยอาดูไอโน. *การประชุมวิชาการระดับชาติการจัดการเทคโนโลยีและนวัตกรรม*, 5.

นิติคม อริยพิมพ์, ชัยพร อัดโดดดร. (2565). การออกแบบและสร้างระบบไอโอทีสำหรับบ้านจำลองแบบอัจฉริยะที่ควบคุมด้วยไมโครคอนโทรลเลอร์. *วิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ,* 17(1), 29-39.

บะไบ๋ คาริชมาลย์, จีรัง คำนวณตา, สัญญา พันธุ์แพง. (2563). พัดลมไอน้ำอัจฉริยะ. *การประชุมวิชาการระดับชาติ วิทยาศาสตร์ เทคโนโลยีและนวัตกรรม (มหาวิทยาลัยแม่โจ้),* (1)

ปกเกศ จันทะกล, สุจิตรา ผาระนัด. (2566). ระบบไฟฟ้าภายในโรงเรือนควบคุมด้วยสมาร์ทโฟน. *วิทยาศาสตร์ เทคโนโลยี และนวัตกรรม,* 4(1), 1-10

รวิช ควรประเสริฐ, พรพุฒิ ศุภเอม. (2565). การพัฒนาระบบตรวจวัดอุณหภูมิในงานเทคโนโลยีสถาปัตยกรรมด้วยไมโครคอนโทรลเลอร์แบบอาร์ดุยโน. *สาระศาสตร์,* 1(3), 611-622.

วิจิตตรา คุ้มวงษ์, สรวง รุ่งประกายพรรณ. (2565). การพัฒนาชุดอุปกรณ์อินเตอร์เน็ตของสรรพสิ่งสำหรับติดตามอุณหภูมิและความชื้นของการเก็บรักษายา. *Thai Bull Pharm Sci.,* 17(2), 101-118

สิรวิชญ์ โตหมื่นไวย, ศุภกร สุวรรณ, ศุภกฤษ นาคป้อมฉิน, แก้วใจ อาภรณ์พิศาล. (2563). การพัฒนาระบบควบคุมอุณหภูมิและค่าความชื้นผ่านสมาร์ทโฟน กรณีศึกษาฟาร์มเห็ดสุริยนต์ อำเภอนครชัยศรี จังหวัดนครปฐม. *งานประชุมวิชาการระดับชาติด้านวิทยาศาสตร์ เทคโนโลยี และนวัตกรรม,* (3)

*คู่มือการใช้งาน NETPIE Platform.* (2565). Netpie.
https://docs.netpie.io/index.html#netpie-platform

*เซ็นเซอร์น้ำฝน ความชื้น วัดระดับน้ำ Rain Sensor Module.* (2566). Cybertice.
https://www.cybertice.com/product/30/เซ็นเซอร์น้ำฝน-ความชื้น-วัดระดับน้ำ-rain-sensor-module

*ทำความรู้จักกับบอร์ด NodeMCU.* (2563). Futurekit.
https://www.futurekit.com/th/content/10850/ทำความรู้จักกับบอร์ด-nodemcu

*สอนการใช้งานเซนเซอร์ DHT11 และ DHT22.* (2565). Analogread.
https://www.analogread.com/article/200/สอนการใช้งานบอร์ด-arduino-กับเซนเซอร์-dht-วัดความชื้นและอุณหภูมิ-พร้อมโค้ดตัวอย่าง

*หลักการทำงานรีเลย์.* (2566). Tzarbikeshop. https://tzarbikeshop.co.th/หลักการทำงานรีเลย์

*ESP8266 ตอนที่ 1 รู้จักกับ ESP และรุ่นที่นิยมใช้งาน.* (2564). Artronshop.
https://www.artronshop.co.th/article/11/esp8266

*SQL vs NoSQL*. (2563). Clound HM. https://blog.cloudhm.co.th/sql-vs-nosql/.

# ภาคผนวก

# ภาคผนวก ก

การเขียนโปรแกรม ด้วยการใช้ Arduino IDE 2.1.0 ควบคุมการทำงาน
และรับค่าของอุปกรณ์ในการทำงานของระบบ

การเขียนโปรแกรม ด้วยการใช้ Arduino IDE 2.1.0 ควบคุมการทำงานและรับค่าของอุปกรณ์ในการทำงานของระบบที่ได้ทำการออกแบบ

ServerRoom.ino

```
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <TridentTD_LineNotify.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <LiquidCrystal_I2C.h>
#include <wire.h>

#define DHTPIN D5      //ส่งค่าเข้าขา D5 วัดรวม
#define DHTPIN1 D4     //ส่งค่าเข้าขา D4 วัดเครื่องแม่ข่าย
#define DHTTYPE DHT22  //กำหนดรุ่นของ sersor
DHT dht(DHTPIN, DHTTYPE);
DHT dht1(DHTPIN1, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);  // กำหนดเชื่อมต่อจอ 1602 16 ตัวอักษร 2 แถว

// Line
#define SSID "Thesky01"
#define PASSWORD "0987667890"
#define LINE_TOKEN "d4pmyOyG22en4YFDuSGhJPToLaBjct9T9C86BnKRGoW"
boolean flag = false;
boolean SVflag = false;

//Fan
int Relay1 = D6;  // ขา Relay ควบคุมพัดลมระดับความแรง 1
int Relay2 = D7;  // ขา Relay ควบคุมพัดลมระดับความแรง 2
int Relay3 = D8;  // ขา Relay ควบคุมพัดลมระดับความแรง 3

//Power outage
int Relay4 = D3;  // Relay ตรวจจับไฟฟ้าดับ
boolean Powerflag = false;

//Water
boolean WaterFlag = false;

//NetPie
//Topic การเขียนข้อมูล Shadow
#define UPDATEDATA "@shadow/data/update"
const char* mqtt_server = "mqtt.netpie.io";
const int mqtt_port = 1883;
const char* mqtt_Client = "d4a5136b-181b-4593-b5fd-7f7bca3a1489";
const char* mqtt_username = "abAJN8sjCQ3NjKCYRKNrPuVzhsY2REXt";
const char* mqtt_password = "e1Ngelo-gdP3bQpp*o(9rUHv-lN*PKDJ";
const char* topic_pub = "@shadow/data/update";   // Topic สำหรับส่งคำขอเปลี่ยนแปลง Shadow
const char* topic_sub = "@shadow/data/changed";  // Topic สำหรับรับข้อมูลเมื่อ Shadow เปลี่ยนแปลง
```

ServerRoom.ino

```
const char* topic = "Temperature";
WiFiClient espClient;
PubSubClient client(espClient);

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection…");
    if (client.connect(mqtt_Client, mqtt_username, mqtt_password)) {
      Serial.println("connected");
      //client.subscribe("@msg/home/#");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println("try again in 5 seconds");
      delay(5000);
    }
  }
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  String message;
  for (int i = 0; i < length; i++) {
    message = message + (char)payload[i];
  }
}

void sendShadowRequest(const char* payload) {
  client.publish(topic_pub, payload);
}

void setup() {
  Serial.begin(115200);
  dht.begin();
  dht1.begin();

  Serial.println(LINE.getVersion());

  pinMode(Relay1, OUTPUT);     //กำหนดให้เป็น D6 เป็น output
  pinMode(Relay2, OUTPUT);     //กำหนดให้เป็น D7 เป็น output
  pinMode(Relay3, OUTPUT);     //กำหนดให้เป็น D8 เป็น output
  digitalWrite(Relay1, HIGH);  //กำหนดให้เป็น D6 สถานะ High
```

ServerRoom.ino

```
  digitalWrite(Relay2, HIGH);  //กำหนดให้เป็น D7 สถานะ High
  digitalWrite(Relay3, HIGH);  //กำหนดให้เป็น D8 สถานะ High

  // Connect to WiFi
  WiFi.begin(SSID, PASSWORD);
  Serial.printf("WiFi connecting to %s\n", SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(400);
  }
  Serial.printf("\nWiFi connected\nIP : ");
  Serial.println(WiFi.localIP());
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);

  // กำหนด Line Token
  LINE.setToken(LINE_TOKEN);
  //LINE.notify("CheckTemp");

  // LCD
  lcd.init();
  lcd.backlight();
  //lcd.setCursor(0, 0); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่0 แถวที่ 1 เตรียมพิมพ์ข้อความ
  //lcd.print("LCD1602 I2c Test"); //พิมพ์ข้อความ "LCD1602 I2c Test"
  //lcd.setCursor(2, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรกำแหน่งที่3 แถวที่ 2 เตรียมพิมพ์ข้อความ
  //lcd.print("my name's Khwanthip"); //พิมพ์ข้อความ "myarduino.net"
  //SDA ขา D2
  //SCL ขา D1

  //Power outage
  pinMode(Relay4, INPUT_PULLUP);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  String dataToSend = "Hello, Netpie!";
  client.publish(topic, dataToSend.c_str());  // ส่งค่าขึ้น Netpie ไปยังช่องทางที่ต้องการ
  Serial.println(dataToSend);
  delay(5000);
```

ServerRoom.ino

```
131   //TemRoom
132   float temperature = dht.readTemperature();  // วัดค่าอุณหภูมิห้อง
133   float humidity = dht.readHumidity();        // วัดค่าความชื้นห้อง
134   Serial.print("\t");
135   Serial.print("Temperature :  ");
136   Serial.print(temperature, 1);
137   Serial.print("\t\t");
138   Serial.print("Humidity :  ");
139   Serial.print(humidity, 1);
140   Serial.print("\t\t");
141   Serial.print("\n");
142
143   String CheckTem = "\n Temperature : ";
144   String ResultTem = CheckTem + temperature + " °C";
145   String Tem = String(temperature) + " °C";
146
147   String CheckHum = "\n Humidity : ";
148   String ResultHum = CheckHum + humidity + " %";
149   String Hum = String(humidity) + " %";
150
151   float t = dht.readTemperature();
152   if (t >= 25 && flag == false) {
153     String ResultTem;
154     String CheckTem = "\n Temperature : ";
155     ResultTem = CheckTem + temperature + " °C";
156     String string1 = "อุณหภูมิ เกินกำหนด ";
157     String ResultInclude = string1 + ResultTem + "\t" + " Humidity : " + humidity + " %";
158     LINE.notify(ResultInclude);
159     flag = true;
160   }
161   if (t < 25 && flag == false) {
162     String ResultTem;
163     String CheckTem = "\n Temperature : ";
164     ResultTem = CheckTem + temperature + " °C";
165     String string2 = "อุณหภูมิ มาตรฐาน ";
166     String ResultInclude = string2 + ResultTem + "\t" + " Humidity : " + humidity + " %";
167     LINE.notify(ResultInclude);
168     flag = true;
169   }
```

ServerRoom.ino

```
170
171   //TemSever
172   float temperatureSV = dht1.readTemperature();  // วัดค่าอุณหภูมิแม่ข่าย
173   float humiditySV = dht1.readHumidity();        // วัดค่าความชื้นแม่ข่าย
174   Serial.print("\t");
175   Serial.print("TemperatureServer :  ");
176   Serial.print(temperatureSV, 1);
177   Serial.print("\t\t");
178   Serial.print("HumidityServer :  ");
179   Serial.print(humiditySV, 1);
180   Serial.print("\t\t");
181   Serial.print("\n");
182
183   String CheckTemSV = "\n TemperatureServer : ";
184   String ResultTemSV = CheckTem + temperatureSV + " °C";
185   String TemSV = String(temperatureSV) + " °C";
186
187   String CheckHumSV = "\n HumidityServer : ";
188   String ResultHumSV = CheckHumSV + humiditySV + " %";
189   String HumSV = String(humiditySV) + " %";
190
191   float t1 = dht1.readTemperature();
192   if (t1 >= 60 && SVflag == false) {
193     String ResultTemSV;
194     String CheckTemSV = "\n TemperatureServer : ";
195     ResultTemSV = CheckTemSV + temperatureSV + " °C";
196     String stringSV1 = "อุณหภูมิเครื่องแม่ข่าย เกินกำหนด ";
197     String ResultInclude = stringSV1 + ResultTemSV + "\t" + " HumidityServer : " + humiditySV + " %";
198     LINE.notify(ResultInclude);
199     SVflag = true;
200   }
201   if (t1 < 60 && SVflag == false) {
202     String ResultTem;
203     String CheckTemSV = "\n TemperatureServer : ";
204     ResultTemSV = CheckTemSV + temperatureSV + " °C";
205     String stringSV = "อุณหภูมิเครื่องแม่ข่าย มาตรฐาน ";
206     String ResultInclude = stringSV + ResultTemSV + "\t" + " HumidityServer : " + humiditySV + " %";
207     LINE.notify(ResultInclude);
208     SVflag = true;
209   }
210
```

ServerRoom.ino

```
211   // Fan
212   if (t >= 25 && t < 28) {
213     digitalWrite(Relay1, LOW);   // สั่งเปิดพัดลมเบอร์ 1
214     digitalWrite(Relay2, HIGH);  // สั่งปิดพัดลมเบอร์ 2
215     digitalWrite(Relay3, HIGH);  // สั่งปิดพัดลมเบอร์ 3
216   } else if (t >= 28 && t < 31) {
217     digitalWrite(Relay1, HIGH);  // สั่งปิดพัดลมเบอร์ 1
218     digitalWrite(Relay2, LOW);   // สั่งเปิดพัดลมเบอร์ 2
219     digitalWrite(Relay3, HIGH);  // สั่งปิดพัดลมเบอร์ 3
220   } else if (t >= 31) {
221     digitalWrite(Relay1, HIGH);  // สั่งปิดพัดลมเบอร์ 1
222     digitalWrite(Relay2, HIGH);  // สั่งปิดพัดลมเบอร์ 2
223     digitalWrite(Relay3, LOW);   // สั่งเปิดพัดลมเบอร์ 3
224   } else {
225     digitalWrite(Relay1, HIGH);  // สั่งปิดพัดลมเบอร์ 1
226     digitalWrite(Relay2, HIGH);  // สั่งปิดพัดลมเบอร์ 2
227     digitalWrite(Relay3, HIGH);  // สั่งเปิดพัดลมเบอร์ 3
228   }
229   delay(10000);
230   //LCD print
231   lcd.setCursor(0, 0);
232   lcd.print("Temp : " + String(temperature) + " C");
233   lcd.setCursor(0, 1);
234   lcd.print("Hum  : " + String(humidity) + " %");
235
236   //Water
237   int Water = analogRead(A0);
238   Serial.print(Water);
239   delay(100);
240   if (Water <= 650 && WaterFlag == false) {
241     LINE.notify("เตือนระดับน้ำรั่วซึม");
242     WaterFlag = true;
243   }
244   if (Water >= 750 && WaterFlag == true) {
245     LINE.notify("ไม่มีน้ำรั่วซึม สภาวะปกติ");
246     delay(6000);
247     WaterFlag = false;
248   }
249
```

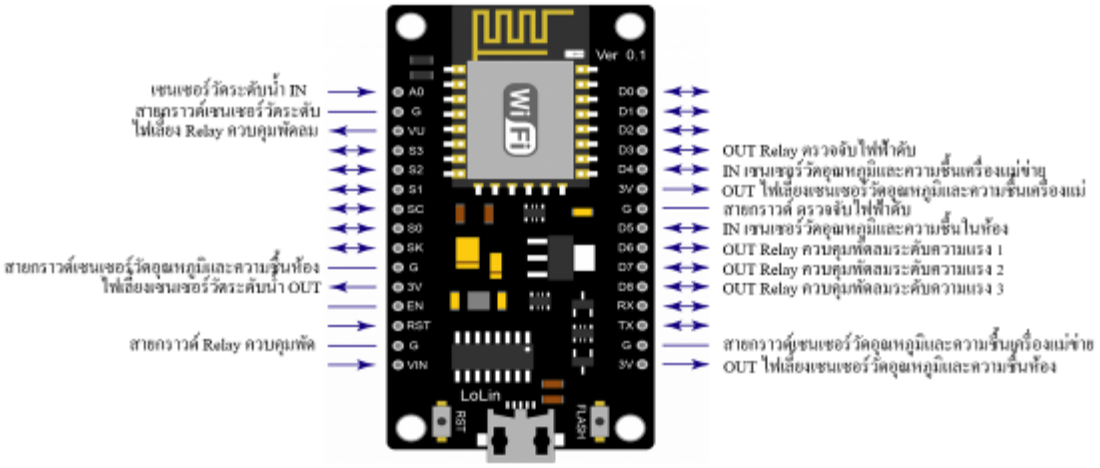
ServerRoom.ino

```
249
250   //Power outage
251   int RelayState = digitalRead(Relay4);
252   if (RelayState == 0) {
253     while (Powerflag == false) {
254       LINE.notify("ตอนนี้ไฟฟ้าภายนอกห้อง Server ดับ");
255       //Serial.print("ทดสอบไฟฟ้าดับ ");
256       delay(3000);
257       Powerflag = true;
258     }
259   }
260   if (RelayState == 1) {
261     while (Powerflag == true)  //while(digitalRead(Relay4)== 0)
262     {
263       LINE.notify("ตอนนี้ไฟฟ้าภายนอกห้อง Server มาแล้ว");
264       delay(3000);
265       Powerflag = false;
266     }
267   }
268
269
270   // กำหนดข้อมูลให้อยู่ในรูปแบบ Json Netpie
271   const size_t capacity = JSON_OBJECT_SIZE(5)+ 48;
272   StaticJsonDocument<capacity> doc;
273
274   JsonObject data = doc.createNestedObject("data");
275   data["Temp"] = String(temperature);
276   data["Hum"] = String(humidity);
277   data["TempSV"] = String(temperatureSV);
278   data["HumSV"] = String(humiditySV);
279   data["Water"] = Water;
280
281
282
283
284
285   Serial.print("Message send: ");
286   Serial.println((doc.as<String>()).c_str());
287
288   //ส่งข้อมูลไปที่ Shadow Device
289   client.publish(UPDATEDATA, (doc.as<String>()).c_str());
290 }
```

**ภาคผนวก ข**

การเชื่อมต่อของอุปกรณ์เซนเซอร์กับบอร์ด ESP8266

การเชื่อมต่อของอุปกรณ์เซนเซอร์กับบอร์ด ESP8266




ที่มา : https://www.futurekit.com/th/content/10850/nodemcu

# ประวัติผู้ศึกษา

**ชื่อสกุล** นางสาวขวัญทิพย์ อิทธิผล

**วัน เดือน ปี เกิด** 27 มิถุนายน 2525

**สถานที่เกิด** กรุงเทพมหานคร

**ที่อยู่ปัจจุบัน** ปริญญาตรี

**ประวัติการศึกษา** บริษัท ท็อปพาร์ท เทรดดิ้ง (ประเทศไทย) จำกัด

**ประวัติการทำงาน** ผู้ควบคุมฝ่ายจัดซื้อ