

**โรงเพาะเห็ดอัจฉริยะโดยใช้เทคโนโลยีการสื่อสารไร้สายระยะไกล**

**Smart Mushroom Farm using Long Range Wireless Communication Technology**

**นายนรินทร สอนชัยภูมิ**

**นายวรรรณณรงค์ สถิตวิทยกูล**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์**

**คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น**

**พ.ศ. 2564**

**ลิขสิทธิ์คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน**

โรงเพาะเห็ดอัจฉริยะโดยใช้เทคโนโลยีการสื่อสารไร้สายระยะไกล

นายนรินทร สอนชัยภูมิ

นายวรรรณณรงค์ สถิตวิทยกูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น

พ.ศ. 2564

ลิขสิทธิ์คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน

Smart Mushroom Farm using Long Range Wireless Communication Technology

Narinthon Sonchaiyaphum

Wannarong Satitwittayakul

A Report Submitted in Partial Fulfillment of the Requirements for

the Degree of Bachelor of Engineering

Department of Computer Engineering

Faculty of Engineering

Rajamangala University of Technology Isan, Khon Kaen Campus

2021

© Faculty of Engineering, Rajamangala University of Technology Isan



**ใบรับรองปริญญานิพนธ์**

**คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน**

หัวข้อปริญญานิพนธ์ : โรงเพาะเห็ดอัจฉริยะโดยใช้เทคโนโลยีการสื่อสารไร้สายระยะไกล

จัดทำโดย : นรินทร สอนชัยภูมิ และวรรรณณรงค์ สถิตวิทยกูล

สาขาวิชา : วิศวกรรมคอมพิวเตอร์

ที่ปรึกษา : อาจารย์ ดร.เศวษ หงษ์ประสิทธิ์

ได้รับอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น

...............................................................คณบดีคณะวิศวกรรมศาสตร์

(อาจารย์ปริญ นาชัยสิทธิ์) วันที่.......เดือน...................พ.ศ. ............

คณะกรรมการสอบปริญญานิพนธ์

............................................ประธานกรรมการสอบ .............................................กรรมการ/ที่ปรึกษา

(อาจารย์นาวา งามวิทยานนท์) (อาจารย์ ดร.เศวษ หงษ์ประสิทธิ์)

.............................................กรรมการ

(อาจารย์อภิวัฒน์ สวัสดิรัตน์)

หัวข้อปริญญานิพนธ์ โรงเพาะเห็ดอัจฉริยะโดยใช้เทคโนโลยีการสื่อสารไร้สายระยะไกล

จัดทำโดย นรินทร สอนชัยภูมิ และวรรรณณรงค์ สถิตวิทยกูล

ปีที่สำเร็จการศึกษา พุทธศักราช 2564

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ที่ปรึกษา อาจารย์ ดร.เศวษ หงษ์ประสิทธิ์

**บทคัดย่อ**

ปริญญานิพนธ์นี้วัตถุประสงค์เพื่อออกแบบและสร้างโรงเพาะเห็ด ที่สามารถควบคุมสภาพแวดล้อมภายในโรงเรือนให้เหมาะสมต่อการออกดอกของเห็ด ผ่านทางเว็บแอปพลิเคชันและวินโดว์แอปพลิเคชัน เพื่อติดตามตรวจสอบและควบคุมระบบ และได้ประยุกต์ใช้งาน LoRa เพื่อเป็นตัวกลางการสื่อสารระหว่างโรงเรือนเพาะเห็ด และพื้นที่ที่มีสัญญาณอินเทอร์เน็ตเพื่อแก้ปัญหาสถานที่ตั้งของโรงเรือนที่อยู่ห่างไกลจากพื้นที่ครอบคลุมของสัญญาณอินเทอร์เน็ต

ระบบควบคุมสภาพแวดล้อมภายในโรงเรือนประกอบไปด้วย ไมโครคอนโทรลเลอร์ (ESP32LoRa) ซึ่งเป็นตัวควบคุมระบบทั้งหมดโดยใช้เซนเซอร์ AM2315 วัดอุณหภูมิและความชื้นBH1750FVI วัดความสว่าง โดยมีอุปกรณ์ที่ใช้ในการควบคุมสภาพแวดล้อม เช่น พัดลม ปั๊ม หลอดไฟ และได้ประยุกต์ใช้งานตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) เพื่อตรวจสอบสถานะการทำงานของอุปกรณ์ ส่วนของเว็บแอปพลิเคชันใช้ไมโครคอนโทรลเลอร์ (ESP32LoRa) เป็น Web Server และใช้แอปพลิเคชัน ngrok ที่ติดไว้บน Raspberry Pi ทำ Port forwarding เพื่อให้สามารถใช้งานเว็บแอปพลิเคชันได้จากทุกที่ที่สามารถเข้าถึงอินเทอร์เน็ต ส่วนของวินโดว์แอปพลิเคชันสามารถติดตามตรวจสอบและควบคุมโรงเรือนผ่านท่าง Serial Port ระหว่าง ไมโครคอนโทรลเลอร์และคอมพิวเตอร์

สรุปผลการดำเนินงานการทดลองการทำงานของระบบสามารถทำงานได้ดี การทดลองเพาะเห็ดใช้เห็ดนางฟ้าในการทำการทดลองทั้งหมด 14 วัน สามารถสรุปผลการทดลองได้ดังนี้ โรงเรือนที่ควบคุมสภาพแวดล้อม มีความกว้างของดอกเฉลี่ยเท่ากับ 7.66 เซนติเมตร มีน้ำหนักทั้งหมดเท่ากับ 1.06 กิโลกรัม โรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อม มีความกว้างของดอกเฉลี่ยเท่ากับ 6.45 เซนติเมตร มีน้ำหนักทั้งหมดเท่ากับ 0.6 กิโลกรัม

Project Title Smart Mushroom Farm using Long Range Wireless

Communication Technology

Proposed by Narinthon Sonchaiyaphum and Wannarong Satitwittayakul

Year 2021

Department Computer Engineering

Project Advisor Dr. Saweth Hongprasit

**Abstract**

The thesis aims to designs and construct a mushroom hatchery, That can control the environment within the hatchery to suit the flowering of mushroom with web applications and window applications that can monitor the environment and control the hatchery system and have applied for LoRa communication to serve as a communication medium between Mushroom Mode and internet area (STA Node) to solve the problem of house location without internet

The environment control system in the mushroom hatchery of the microcontroller (ESP32LoRa) controls the entire system, uses an AM2315 sensor measures temperature and humidity, uses a BH1750FVI sensor measures brightness, environmental control devices such as fans, pumps, lamps, and has applied Current Sense Resistors to check the operating status of the device. The web application uses a microcontroller (ESP32LoRa) as a Web Server and uses the ngrok application installed on the Raspberry Pi to take Port forwarding to enable the web application to be run from anywhere that has access the internet. The window application can monitor and control the mushroom hatchery with the Serial Port between the microcontroller and the computer.

Conclusion of the experimental results The working of the system can work well. The mushroom hatchery experiments used fairy mushrooms for all 14 days of the experiment. The results of the experiment were as follows. Environment controlled greenhouses The average flower width was 7.66 cm and the total weight was 1.06 kg. The average width of the flower was 6.45 cm with the total weight of 0.6 kg.

**กิตติกรรมประกาศ**

ปริญญานิพนธ์ฉบับนี้ได้รับความช่วยเหลืออย่างดียิ่งจาก อาจารย์ ดร.เศวษ หงษ์ประสิทธิ์อาจารย์ที่ปรึกษาปริญญานิพนธ์ ซึ่งกรุณาให้คำแนะนำและถ่ายทอดความรู้ ตลอดจนควบคุมการทำโครงงานจนประสบความสำเร็จ ผู้ทำโครงงานขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณคณาจารย์มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น ที่ได้ประสิทธิประสาทความรู้และคณะตลอดเวลาที่ศึกษาเล่าเรียน

ขอขอบพระคุณบิดามารดาตลอดจนญาติพี่น้องที่คอยเป็นกำลังใจให้ข้าพเจ้าและสนับสนุนส่งเสริมการศึกษามาโดยตลอด

ขอขอบพระคุณเพื่อนๆทุกท่านที่ได้ให้การช่วยเหลือในการทำโครงงาน การเรียน การทำงานและมิตรภาพที่ดีแก่กันเสมอมา

ขอขอบพระคุณทุกท่านที่มิได้เอ่ยนามเป็นอย่างยิ่งในความกรุณาและความช่วยเหลือในด้านต่าง ๆ ที่ทำให้โครงงานนี้สำเร็จได้

ประโยชน์และคุณค่าอันพึงมีจากปริญญานิพนธ์ฉบับนี้ผู้ทำโครงงานขอมอบเป็นกตัญญุตาบูชาแด่ บิดามารดา ตลอดผู้มีพระคุณทุกท่าน

นรินทร สอนชัยภูมิ

วรรรณณรงค์ สถิตวิทยกูล

**สารบัญ**

|  |  |  |
| --- | --- | --- |
|  |  | **หน้า** |
| **บทคัดย่อ** | | ค |
| **กิตติกรรมประกาศ** | | ง |
| **สารบัญตาราง** | | ซ |
| **สารบัญรูป** | | ฌ |
|  | |  |
| **บทที่** | บทนำ | 1 |
| 1. | 1.1 ความเป็นมาและความสำคัญของปัญหา | 1 |
|  | 1.2 วัตถุประสงค์ | 2 |
|  | 1.3 ขอบเขต | 2 |
|  | 1.4 ประโยชน์ที่คาดว่าจะได้รับ | 3 |
| 2. | ทฤษฎีและงานวิจัยที่เกี่ยวข้อง | 4 |
|  | 2.1 ทฤษฎีที่เกี่ยวข้อง | 4 |
|  | 2.1.1 เห็ดนางฟ้าและเห็ดนางฟ้าภูฐาน | 4 |
|  | 2.1.1.1 วงจรชีวิตของเห็ดนางฟ้า | 5 |
|  | 2.1.1.2 สภาพแวดล้อมที่เหมาะสมต่อการเจริญเติบโตของเห็ดนางฟ้า | 6 |
|  | 2.1.1.3 การเพาะเลี้ยงเห็ดนางฟ้านในประเทศไทยและโรงเรือนสำหรับเปิดดอกเห็ด | 7 |
|  | 2.1.1.4 ลักษณะก้อนเชื้อที่นำมาเพาะ | 8 |
|  | 2.1.1.5 การรดน้ำ | 9 |
|  | 2.1.1.6 การเก็บเกี่ยวผลผลิตดอกเห็ด | 10 |
|  | 2.1.2 อุณหภูมิ (Temperature) | 10 |
|  | 2.1.2.1 ความร้อนและอุณหภูมิ | 11 |
|  | 2.1.3 ความชื้น (Humidity) | 11 |
|  | 2.1.3.1 ความชื้นในอากาศ | 11 |
|  | 2.1.3.2 ความชื้นสัมพัทธ์ของประเทศไทยในแต่ละฤดู | 13 |
|  | 2.1.4 แสงสว่าง (Light) | 13 |
|  | 2.1.5 LoRa (Long Range) | 15 |
| **สารบัญ(ต่อ)** | | |
|  |  |  |
|  | 2.1.5.1 คุณสมบัติหลักของเทคโนโลยี LoRa | 18 |
|  | 2.1.5.2 ความแตกต่างระหว่าง LoRa และ LoRaWan | 19 |
|  | 2.1.6 ตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) | 19 |
|  | 2.1.6.1 ตำแหน่งการวางตัวต้านทานตรวจสอบกระแส | 19 |
|  | 2.2 งานวิจัยที่เกี่ยวข้อง | 21 |
|  | 2.2.1 ไอโอทีแพลทฟอร์มโรงเพาะเห็ด | 21 |
|  | 2.2.2 การประยุกต์ใช้เทคโนโลยีไอโอทีควบคุมฟาร์มอัจฉริยะในโรงเรือนเพาะเห็ดนางฟ้า | 22 |
|  | 2.2.3 IOT for Smart Farm: A case study of the Lingzhi Mushroom Farm at Maejo University | 23 |
|  | 2.2.4 IOT BASED DESIGN IMPLEMENTATION OF MUSHROOM FARM MONITORING USING ARDUINO MICROCONTROLLERS & SENSORS | 24 |
| 3. | วิธีการดำเนินงาน | 25 |
|  | 3.1 แผนการดำเนินงาน | 25 |
|  | 3.2 โครงสร้างระบบ | 26 |
|  | 3.3 โรงเรือนเพาะเห็ด (Mushroom Farm) | 28 |
|  | 3.3.1 การออกแบบโรงเรือนเพาะเห็ด | 28 |
|  | 3.3.2 การออกแบบฮาร์ดแวร์ระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ด | 29 |
|  | 3.3.2.1 การอ่านค่าอุณหภูมิและความชื้นจากเซ็นเซอร์ AM2315 | 30 |
|  | 3.3.2.2 การอ่านค่าความสว่างจากเซ็นเซอร์ BH1750FVI | 32 |
|  | 3.3.2.3 การควบคุมพัดลมและปั๊มพ่นหมอก | 34 |
|  | 3.3.2.4 การทำงานของปั๊มพ่นหมอกและพัดลมในโหมดการทำงานแบบอัตโนมัติ (AUTO) | 35 |
|  | 3.3.2.5 การควบคุมความสว่างภายในโรงเรือนเพาะเห็ด | 37 |
|  | 3.3.2.6 การใช้จอแอลซี (LCD) แสดงผล | 39 |
|  | 3.3.2.7 การอ่านค่าแรงดันเอาต์พุตจากวงจรตัวต้านทานตรวจสอบกระแส | 41 |
| **สารบัญ(ต่อ)** | | |
|  |  |  |
|  | 3.4 การออกแบบวงจรตัวต้านทานตรวจสอบกระแส | 42 |
|  | 3.4.1 การคำนวณแรงดันเอาต์พุตของวงจรเมื่อใช้งานกับพัดลมระบายอากาศ | 43 |
|  | 3.4.2 การคำนวณแรงดันเอาต์พุตของวงจรเมื่อใช้งานกับปั๊มพ่นหมอก | 46 |
|  | 3.5 การสื่อสารระหว่างโรงเรือนและพื้นที่ที่มีสัญญาณอินเทอร์เน็ตโดยใช้ลอรา (LoRa Communication | 48 |
|  | 3.5.1 การออกแบบการสื่อสาร Lora ที่ Mushroom Node | 49 |
|  | 3.5.2 การออกแบบการสื่อสาร Lora ที่ STA Node | 60 |
|  | 3.6 เว็บแอปพลิเคชัน (Web Application) | 69 |
|  | 3.6.1 การออกแบบหน้าจอเว็บแอปพลิเคชัน | 69 |
|  | 3.6.1.1 หน้าเข้าสู่ระบบ | 70 |
|  | 3.6.1.2 หน้า Dashboard | 71 |
|  | 3.6.1.3 แถบเมนูนำทาง | 73 |
|  | 3.6.1.4 หน้า Data Logger | 74 |
|  | 3.6.1.5 หน้า Chart | 74 |
|  | 3.6.1.6 หน้า Setting | 75 |
|  | 3.6.2 การทำงานของเว็บแอปพลิเคชัน | 76 |
|  | 3.7 วินโดว์แอปพลิเคชัน (Windows Application) | 81 |
|  | 3.7.1 การออกแบบหน้าจอวินโดว์แอปพลิเคชัน | 82 |
|  | 3.7.1.1 หน้าเข้าสู่ระบบ | 82 |
|  | 3.7.1.2 หน้า Dashboard | 82 |
|  | 3.7.2 การทำงานของวินโดว์แอปพลิเคชัน | 85 |
|  | 3.8 ฐานข้อมูล (Database) | 90 |
|  | 3.8.1 การออกแบบฐานข้อมูล | 90 |
|  | 3.8.2 การบันทึกข้อมูลลงในฐานข้อมูล | 92 |
|  | 3.9 การออกแบบ Use Case Diagram | 96 |
| 4. | ผลการดำเนินงาน | 101 |
|  | 4.1 การทดลองการทำงานของระบบ | 101 |
|  | 4.1.1 การทดลองการทำงานของเว็บแอปพลิเคชัน | 102 |
| **สารบัญ(ต่อ)** | | |
|  |  |  |
|  | 4.1.1.1 ทดลองเปลี่ยนโหมดการทำงาน | 104 |
|  | 4.1.1.2 ทดลองควบคุมพัดลมและปั๊ม | 105 |
|  | 4.1.1.3 ทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด | 107 |
|  | 4.1.1.4 ทดลองกำหนดค่าความชื้นต่ำสุดและความชื้นสูงสุด | 110 |
|  | 4.1.1.5 ทดลองกำหนดค่าความสว่าง | 112 |
|  | 4.1.2 การทดลองการทำงานของวินโดว์แอปพลิเคชัน | 114 |
|  | 4.1.2.1 ทดลองเปลี่ยนโหมดการทำงาน | 117 |
|  | 4.1.2.2 ทดลองควบคุมพัดลมและปั๊ม | 117 |
|  | 4.1.2.3 ทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด | 119 |
|  | 4.1.2.4 ทดลองกำหนดค่าความชื้นต่ำสุดและความชื้นสูงสุด | 120 |
|  | 4.1.2.5 ทดลองกำหนดค่าความสว่าง | 122 |
|  | 4.1.3 การทดลองการทำงานของระบบควบคุมสภาพแวดล้อม | 123 |
|  | 4.2 การทดลองเพาะเห็ด | 134 |
|  | 4.2.1 ผลการทดลองเพาะเห็ดรุ่นที่ 1 | 136 |
|  | 4.2.2 ผลการทดลองเพาะเห็ดรุ่นที่ 2 | 139 |
| 5. | สรุปผลการดำเนินงาน | 143 |
|  | 5.1 สรุปผล | 143 |
|  | 5.2 ปัญหาที่เกิด | 144 |
|  | 5.3 ข้อเสนอแนะและแนวทางการพัฒนา | 144 |
| บรรณานุกรม | | 145 |
| ภาคผนวก | |  |
|  | ภาคผนวก ก คู่มือการใช้งาน |  |
| ประวัติผู้เขียน | |  |

**สารบัญตาราง**

|  |  |  |
| --- | --- | --- |
| **ตารางที่** |  | หน้า |
| 2.1 | แสดงสถิติความชื้นสัมพัทธ์เฉลี่ย (%) ของประเทศไทยในช่วงฤดูกาลต่างๆ | 13 |
| 2.2 | แสดงความถี่ที่ใช้สำหรับ LoRa (หน่วยเป็น MHz) | 15 |
| 3.1 | แสดงแผนการดำเนินงาน | 26 |
| 3.2 | แสดงการทำงานของปั๊มพ่นหมอกและพัดลมในโหมดการทำงานแบบอัตโนมัติ (AUTO) | 36 |
| 3.3 | แสดง Data Dictionary SensorData | 91 |
| 3.4 | แสดง Login Use Case | 97 |
| 3.5 | แสดง Logout Use Case | 97 |
| 3.6 | แสดง View Dashboard Use Case | 98 |
| 3.7 | แสดง Control Use Case | 98 |
| 3.8 | แสดง Setting Use Case | 99 |
| 3.9 | แสดง View Data Use Case | 99 |
| 3.10 | แสดง View Status Use Case | 100 |
| 4.1 | สรุปผลการทดลองการทำงานของเว็บแอปพลิเคชัน | 113 |
| 4.2 | สรุปผลการทดลองการทำงานของวินโดว์แอปพลิเคชัน | 123 |
| 4.3 | สรุปผลการทดลองการทำงานของระบบวันที่ 1 | 124 |
| 4.4 | สรุปผลการทดลองการทำงานของระบบวันที่ 2 | 125 |
| 4.5 | สรุปผลการทดลองการทำงานของระบบวันที่ 3 | 126 |
| 4.6 | สรุปผลการทดลองการทำงานของระบบวันที่ 4 | 127 |
| 4.7 | สรุปผลการทดลองการทำงานของระบบวันที่ 5 | 128 |
| 4.8 | สรุปผลการทดลองการทำงานของระบบวันที่ 6 | 129 |
| 4.9 | สรุปผลการทดลองการทำงานของระบบวันที่ 7 | 130 |
| 4.10 | สรุปผลการทดลองการทำงานของระบบวันที่ 8 | 131 |
| 4.11 | สรุปผลการทดลองการทำงานของระบบวันที่ 9 | 132 |
| 4.12 | สรุปผลการทดลองการทำงานของระบบวันที่ 10 | 133 |
| 4.13 | แสดงการทำงานของปั๊มพ่นหมอกและพัดลมในโหมดการทำงานแบบอัตโนมัติ (AUTO) | 134 |

**สารบัญรูป**

|  |  |  |
| --- | --- | --- |
| **รูปที่** |  | หน้า |
| 2.1 | แสดงภาพเห็ดนางฟ้าภูฐานที่เจริญเติบโตจากถุงเชื้อ | 5 |
| 2.2 | แสดงภาพวงจรชีวิตของเห็ดนางฟ้า | 6 |
| 2.3 | แสดงภาพโรงเรือนเพาะเห็ด | 8 |
| 2.4 | แสดงภาพของสเปกตรัมแสงที่มองเห็นได้ | 14 |
| 2.5 | แสดงภาพตัวอย่างค่า Data Rate (DR) | 16 |
| 2.6 | แสดงภาพความสัมพันธ์ระหว่าง BITRATE และระยะห่างของอุปกรณ์กับ Gateway | 17 |
| 2.7 | แสดงภาพคุณสมบัติหลักของเทคโนโลยี LoRa | 18 |
| 2.8 | แสดงภาพการวางตัวต้านทานตรวจสอบกระแสแบบการตรวจจับด้านต่ำ  (Low-side Sensing) | 20 |
| 2.9 | แสดงภาพการวางตัวต้านทานตรวจสอบกระแสแบบการตรวจจับด้านสูง  (High-side Sensing | 20 |
| 2.10 | แสดงภาพโครงสร้างระบบของงานวิจัยที่เกี่ยวข้อง(1) | 21 |
| 2.11 | แสดงภาพโครงสร้างระบบของงานวิจัยที่เกี่ยวข้อง(2) | 22 |
| 2.12 | แสดงภาพโครงสร้างระบบของงานวิจัยที่เกี่ยวข้อง(3) | 23 |
| 2.13 | แสดงภาพโครงสร้างระบบของงานวิจัยที่เกี่ยวข้อง(4) | 24 |
| 3.1 | แสดงภาพโครงสร้างระบบ | 26 |
| 3.2 | แสดงภาพการออกแบบโรงเพาะเห็ด | 28 |
| 3.3 | แสดงภาพส่วนประกอบของฮาร์ดแวร์ระบบควบคุมสภาพแวดล้อมภายในโรงเรือน | 29 |
| 3.4 | แสดงภาพ Schematic การเชื่อต่อระหว่าง ESP32LoRa และ AM2315 | 30 |
| 3.5 | แสดงภาพโปรแกรมที่ใช้อ่านค่าจากเซ็นเซอร์ AM2315 | 31 |
| 3.6 | แสดงภาพ Schematic การเชื่อต่อระหว่าง ESP32LoRa และ BH1750FVI | 32 |
| 3.7 | แสดงภาพโปรแกรมที่ใช้อ่านค่าจากเซ็นเซอร์ BH1750FVI | 33 |
| 3.8 | แสดงภาพ Schematic ของการควบคุมพัดลมและปั๊มพ่นหมอก | 34 |
| 3.9 | แสดงภาพโปรแกรมส่วนที่ใช้ควบคุมการทำงานของ Relay | 35 |
| 3.10 | แสดงภาพโปรแกรมที่ใช้ควบคุมการทำงานของปั๊มพ่นหมอกและพัดลม | 36 |
| 3.11 | แสดงภาพวงจร LED Driver | 37 |
| **สารบัญรูป(ต่อ)** | | |
|  |  |  |
| 3.12 | แสดงภาพ Schematic ของการควบคุมความสว่างภายในโรงเรือนเพาะเห็ด | 38 |
| 3.13 | แสดงภาพโปรแกรมที่ใช้ในการควบคุมความสว่างของหลอดไฟ | 39 |
| 3.14 | แสดงภาพ Schematic การเชื่อต่อระหว่าง ESP32LoRa และ จอแอลซีดี (LCD) | 39 |
| 3.15 | แสดงภาพโปรแกรมทดสอบการทำงานของจอแอลซีดี (LCD) | 40 |
| 3.16 | แสดงภาพ Schematic การเชื่อต่อระหว่าง ESP32LoRa และวงจรตัวต้านทานตรวจสอบกระแส | 41 |
| 3.17 | แสดงภาพโปรแกรมทดสอบการอ่านค่าแรงดันเอาต์พุตจากวงจรตัวต้านทานตรวจสอบกระแส | 42 |
| 3.18 | แสดงภาพวงจรตัวต้านทานตรวจสอบกระแส | 43 |
| 3.19 | แสดงภาพวงจรตัวต้านทานตรวจสอบกระแสเมื่อใช้งานกับพัดลมระบายอากาศ | 44 |
| 3.20 | แสดงภาพวงจรตัวต้านทานตรวจสอบกระแสเมื่อใช้งานกับปั๊มพ่นหมอก | 46 |
| 3.21 | แสดงภาพการสื่อสารแบบ Node-to-Node | 48 |
| 3.22 | แสดงภาพ Flowchart การทำงานของ Mushroom Node (1) | 49 |
| 3.23 | แสดงภาพรูปแบบของข้อมูล (Data Format) | 50 |
| 3.24 | แสดงภาพฟังก์ชัน sendUpdateData() | 52 |
| 3.25 | แสดงภาพโปรแกรมส่วนที่ตรวจสอบข้อมูลที่ส่งมายัง Mushroom Node | 53 |
| 3.26 | แสดงภาพ Flowchart การทำงานของ Mushroom Node (2) | 54 |
| 3.27 | แสดงภาพ Flowchart การทำงานของ Mushroom Node (3) | 54 |
| 3.28 | แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (1) | 55 |
| 3.29 | แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (2) | 55 |
| 3.30 | แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (3) | 56 |
| 3.31 | แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (4) | 56 |
| 3.32 | แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (5) | 57 |
| 3.33 | แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (6) | 57 |
| 3.34 | แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (7) | 58 |
| 3.35 | แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (8) | 59 |
| 3.36 | แสดงภาพ Flowchart การทำงานของ STA Node (1) | 60 |
| 3.37 | แสดงภาพโปรแกรมส่วนที่รับข้อมูลจาก Serial Port | 61 |
| **สารบัญรูป(ต่อ)** | | |
|  |  |  |
| 3.38 | แสดงภาพโปรแกรมส่วนที่ตรวจสอบข้อมูลที่ส่งมายัง STA Node | 62 |
| 3.39 | แสดงภาพ Flowchart การทำงานของ STA Node (2) | 62 |
| 3.40 | แสดงภาพรูปแบบของข้อมูล (Data Format) ที่ส่งมาจาก Mushroom Node | 63 |
| 3.41 | แสดงภาพการใช้ฟังก์ชัน substring() กับข้อมูลที่ได้รับจาก Mushroom Node | 64 |
| 3.42 | แสดงภาพโปรแกรมที่ใช้ในการส่ง Keyword ทุก 60 วินาที | 65 |
| 3.43 | แสดงภาพ Flowchart การทำงานของ STA Node (3) | 66 |
| 3.44 | แสดงภาพ Flowchart การทำงานของ STA Node (4) | 67 |
| 3.45 | แสดงภาพโครงสร้างการทำงานของ Web Application | 69 |
| 3.46 | แสดงภาพหน้าเข้าสู่ระบบก่อนใช้งานเว็บแอปพลิเคชัน (Web Application) | 70 |
| 3.47 | แสดงภาพหน้า Dashboard ของเว็บแอปพลิเคชัน (Web Application) (1) | 71 |
| 3.48 | แสดงภาพหน้า Dashboard ของเว็บแอปพลิเคชัน (Web Application) (2) | 72 |
| 3.49 | แสดงภาพแถบเมนูนำทาง (1) | 73 |
| 3.50 | แสดงภาพแถบเมนูนำทาง (2) | 73 |
| 3.51 | แสดงภาพหน้า Data Logger ของเว็บแอปพลิเคชัน (Web Application) | 74 |
| 3.52 | แสดงภาพหน้า Chart ของเว็บแอปพลิเคชัน (Web Application) | 74 |
| 3.53 | แสดงภาพหน้า Setting ของเว็บแอปพลิเคชัน (Web Application) | 75 |
| 3.54 | แสดงภาพ Flowchart การสื่อสารระหว่าง Web Server และ Client | 76 |
| 3.55 | แสดงภาพตัวอย่างโปรแกรมส่วนที่ใช้ในการขออัปเดตข้อมูลทุกๆ 5 วินาที | 77 |
| 3.56 | แสดงภาพข้อมูลต่างๆบนหน้าเว็บที่อัปเดตทุกๆ 5 วินาที | 78 |
| 3.57 | แสดงภาพฟังก์ชันที่ถูกเรียกใช้เมื่อกดปุ่มเปลี่ยนโหมดการทำงาน | 78 |
| 3.58 | แสดงภาพการแจ้งเตือนเมื่อกดปุ่มเปลี่ยนโหมดการทำงาน | 79 |
| 3.59 | แสดงภาพฟังก์ชันที่ถูกเรียกใช้เมื่อกดปุ่มขออัปเดตข้อมูล | 79 |
| 3.60 | แสดงภาพฟังก์ชันที่ถูกเรียกใช้เมื่อกดปุ่ม เปิด/ปิด ปั๊ม | 80 |
| 3.61 | แสดงภาพฟังก์ชันที่ถูกเรียกใช้เมื่อกดปุ่ม เปิด/ปิด พัดลม | 80 |
| 3.62 | แสดงภาพโครงสร้างการทำงานของ Windows Application | 81 |
| 3.63 | แสดงภาพหน้าเข้าสู่ระบบของวินโดว์แอปพลิเคชัน (Windows Application) | 82 |
| 3.64 | แสดงภาพหน้า Dashboard ของวินโดว์แอปพลิเคชัน (1) | 82 |
| 3.65 | แสดงภาพหน้า Dashboard ของวินโดว์แอปพลิเคชัน (2) | 83 |
| **สารบัญรูป(ต่อ)** | | |
|  |  |  |
| 3.66 | แสดงภาพหน้า Dashboard ของวินโดว์แอปพลิเคชัน (3) | 84 |
| 3.67 | แสดงภาพ Flowchart การทำงานของวินโดว์แอปพลิเคชัน (1) | 85 |
| 3.68 | แสดงภาพ Flowchart การทำงานของวินโดว์แอปพลิเคชัน (2) | 86 |
| 3.69 | แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกดปุ่มเปลี่ยนโหมด | 87 |
| 3.70 | แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกดปุ่ม เปิด/ปิด ปั๊ม | 87 |
| 3.71 | แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกดปุ่ม เปิด/ปิด พัดลม | 87 |
| 3.72 | แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าอุณหภูมิต่ำสุด | 88 |
| 3.73 | แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าอุณหภูมิสูงสุด | 88 |
| 3.74 | แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าความชื้นต่ำสุด | 89 |
| 3.75 | แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าความชื้นสูงสุด | 89 |
| 3.76 | แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าความสว่าง | 89 |
| 3.77 | แสดงภาพโครงสร้างของฐานข้อมูล | 90 |
| 3.78 | แสดงภาพตัวอย่างข้อมูลเก็บอยู่ภายในฐานข้อมูล | 91 |
| 3.79 | แสดงภาพ Flowchart ขั้นตอนการบันทึกข้อมูลลงในฐานข้อมูล | 92 |
| 3.80 | แสดงภาพการบันทึกข้อมูลลงในฐานข้อมูลทุกชั่วโมง | 93 |
| 3.81 | แสดงภาพฟังก์ชันที่ใช้บันทึกข้อมูลลงในฐานข้อมูล | 94 |
| 3.82 | แสดงภาพไฟล์ post-esp-data.php บางส่วน | 95 |
| 3.83 | แสดงภาพ Use Case Diagram | 96 |
| 4.1 | แสดงภาพหน้าเข้าสู่ระบบก่อนใช้งานเว็บแอปพลิเคชัน (Web Application) | 102 |
| 4.2 | แสดงภาพหน้าแรกของเว็บแอปพลิเคชัน (Web Application) | 103 |
| 4.3 | แสดงภาพการทดลองเปลี่ยนแปลงโหมดการทำงาน (1) | 104 |
| **สารบัญรูป(ต่อ)** | | |
|  |  |  |
| 4.4 | แสดงภาพการทดลองเปลี่ยนแปลงโหมดการทำงาน (2) | 104 |
| 4.5 | แสดงภาพการทดลองควบคุมพัดลมและปั๊ม (1) | 105 |
| 4.6 | แสดงภาพการทดลองควบคุมพัดลมและปั๊ม (2) | 105 |
| 4.7 | แสดงภาพการทดลองควบคุมพัดลมและปั๊ม (3) | 106 |
| 4.8 | แสดงภาพการทดลองควบคุมพัดลมและปั๊ม (4) | 106 |
| 4.9 | แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (1) | 107 |
| 4.10 | แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (2) | 107 |
| 4.11 | แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (3) | 108 |
| 4.12 | แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (4) | 108 |
| 4.13 | แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (5) | 109 |
| 4.14 | แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (6) | 109 |
| 4.15 | แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (7) | 110 |
| 4.16 | แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (8) | 110 |
| 4.17 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุด (1) | 111 |
| 4.18 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุด (2) | 111 |
| 4.19 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด (1) | 111 |
| 4.20 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด (2) | 112 |
| 4.21 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความสว่าง (1) | 112 |
| 4.22 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความสว่าง (2) | 112 |
| 4.23 | แสดงภาพหน้า Login ของวินโดว์แอปพลิเคชัน | 114 |
| 4.24 | แสดงภาพแสดงข้อความแจ้งเตือนเมื่อกรอกข้อมูลไม่ถูกต้อง | 114 |
| 4.25 | แสดงภาพวินโดว์แอปพลิเคชันเมื่อยังไม่ได้เชื่อมต่อกับ Serial Port | 115 |
| 4.26 | แสดงภาพวินโดว์แอปพลิเคชันเมื่อเชื่อมต่อกับ Serial Port | 116 |
| 4.27 | แสดงภาพปุ่มเปลี่ยนโหมดการทำงานของวินโดว์แอปพลิเคชัน | 117 |
| 4.28 | แสดงภาพวินโดว์แอปพลิเคชันเมื่อโหมดการทำงานเป็นโหมด MANUAL | 117 |
| 4.29 | แสดงภาพปุ่มควบคุมพัดลม (Turn off Fan) ขณะที่พัดลมกำลังทำงาน | 117 |
| 4.30 | แสดงภาพวินโดว์แอปพลิเคชันหลังจากสั่งปิดพัดลม | 118 |
| 4.31 | แสดงภาพปุ่มควบคุมปั๊ม (Turn on Fan) ขณะที่ปั๊มไม่ทำงาน | 118 |
| 4.32 | แสดงภาพวินโดว์แอปพลิเคชันหลังจากสั่งเปิดปั๊ม | 118 |
| **สารบัญรูป(ต่อ)** | | |
|  |  |  |
| 4.33 | แสดงภาพการทดลองเปลี่ยนแปลงค่าอุณหภูมิต่ำสุด (1) | 119 |
| 4.34 | แสดงภาพการทดลองเปลี่ยนแปลงค่าอุณหภูมิต่ำสุด (2) | 119 |
| 4.35 | แสดงภาพการทดลองเปลี่ยนแปลงค่าอุณหภูมิสูงสุด (1) | 120 |
| 4.36 | แสดงภาพการทดลองเปลี่ยนแปลงค่าอุณหภูมิสูงสุด (2) | 120 |
| 4.37 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุด (1) | 120 |
| 4.38 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุด (2) | 121 |
| 4.39 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด (1) | 121 |
| 4.40 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด (2) | 121 |
| 4.41 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความสว่าง (1) | 122 |
| 4.42 | แสดงภาพการทดลองเปลี่ยนแปลงค่าความสว่าง (2) | 122 |
| 4.43 | แสดงภาพของโรงเรือนที่สร้างขึ้นเพื่อใช้ในการเปรียบเทียบผลการทดลอง | 135 |
| 4.44 | แสดงภาพผลการทดลองวันที่ 9 ตุลาคม 2563 | 136 |
| 4.45 | แสดงภาพผลการทดลองวันที่ 10 ตุลาคม 2563 | 136 |
| 4.46 | แสดงภาพผลการทดลองวันที่ 11 ตุลาคม 2563 | 137 |
| 4.47 | แสดงภาพผลผลิตที่ได้จากการทดลองเพาะเห็ดนางฟ้า | 138 |
| 4.48 | แสดงภาพผลการทดลองวันที่ 19 ตุลาคม 2563 | 139 |
| 4.49 | แสดงภาพผลการทดลองวันที่ 20 ตุลาคม 2563 | 139 |
| 4.50 | แสดงภาพผลการทดลองวันที่ 21 ตุลาคม 2563 | 140 |
| 4.51 | แสดงภาพผลผลิตที่ได้จากการทดลองเพาะเห็ดนางฟ้า | 141 |

**บทที่ 1**

**บทนำ**

**1.1 ความเป็นมาและความสำคัญของปัญหา**

ในปัจจุบันประเทศไทยมีการใช้งานอินเทอร์เน็ตเพิ่มขึ้นอย่างมาก เมื่อเทียบกับในอดีตที่ผ่านมาโดยจะเห็นได้จาก ผลการสำรวจพฤติกรรมผู้ใช้อินเทอร์เน็ตในประเทศไทย ของสำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (สพธอ.) หรือ ETDA (เอ็ตด้า) กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม เผยผลการสำรวจพฤติกรรมผู้ใช้อินเทอร์เน็ตในประเทศไทย ปี 2562 หรือ Thailand Internet User Behavior 2019 ชี้ ทศวรรษที่ผ่านมา คนไทยใช้อินเทอร์เน็ตเพิ่มขึ้นอย่างก้าวกระโดดกว่า 150% ส่งผลให้ปัจจุบันไทยมีผู้ใช้อินเทอร์เน็ต 47.5 ล้านคน หรือราว 70% ของจำนวนประชาชนทั้งหมด จากการสำรวจข้อมูลของประชาชนเกี่ยวกับพฤติกรรมการใช้อินเทอร์เน็ต ประจำปี 2562 ผ่านทางออนไลน์ ช่วงเดือน ส.ค.- ต.ค. 2562 โดยมีคนไทยเข้ามาตอบแบบสอบถามกว่า 17,242 คน ซึ่งจากการวิเคราะห์ข้อมูล พบว่า ปี 2562 คนไทยใช้อินเทอร์เน็ตเฉลี่ยวันละ 10 ชั่วโมง 22 นาที เพิ่มขึ้น 17 นาทีจากปี 2561 [5] จากข้อมูลดังกล่าวจะเห็นได้ว่าในปัจจุบันอินเทอร์เน็ตมีความสำคัญต่อการดำเนินชีวิตประจำวันของคนไทย ทั้งด้านการศึกษา ด้านธุรกิจและพาณิชย์ ด้านการบันเทิง รวมถึงการประยุกต์ใช้งานอินเทอร์เน็ตในภาคการเกษตร เช่น การควบคุมการให้น้ำ การควบคุมอุณหภูมิในโรงเรือน การติดตามสภาพดิน เป็นต้น

การเพาะเห็ดในปัจจุบันนิยมเพาะในโรงเรือน โรงเรือนที่เหมาะสำหรับการเพาะเห็ด ควรเป็นโรงเรือนที่สามารถเก็บความชื้นได้ดี มีระบบถ่ายเทอากาศดี และสะดวกต่อการทำความสะอาด เห็ดจะออกดอกได้ดีเมื่อมีสภาพแวดล้อมเหมาะสม ดังนั้นการควบคุมสภาพแวดล้อมภายในโรงเรือนให้เหมาะสมต่อการออกดอกของเห็ดโดยมนุษย์อาจทำให้สภาพแวดล้อมภายในโรงเรือนไม่เหมาะสำหรับการออกดอกของเห็ดเท่าที่ควร อาจส่งผลให้ขนาดของดอกเห็ดมีขนาดเล็กและมีผลผลิตต่ำ จึงได้มีการนำไมโครคอนโทรลเลอร์มาควบคุมสภาพแวดล้อมภายในโรงเรือน โดยนำค่าที่อ่านได้จากเซ็นเซอร์วัดสภาพแวดล้อมมาประมวลผล แล้วนำค่าที่ได้มาควบคุมการทำงานของอุปกรณ์ปรับสภาพแวดล้อมต่างๆ เช่น พัดลมระบายอากาศ ปั๊มพ่นหมอก หลอดไฟ เพื่อปรับสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ด และเมื่อสามารถควบคุมสภาพแวดล้อมภายในโรงเรือนได้แล้ว จึงนำเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง (Internet of Things) หรือ ไอโอที (IoT) มาประยุกต์ใช้งานในการเพาะเห็ด ทำให้สามารถมอนิเตอร์ (Monitor) สภาพแวดล้อมภายในโรงเรือน และความคุมโรงเรือน จากที่ไหนก็ได้ที่สามารถใช้อินเทอร์เน็ตได้ [1]-[4] จากการศึกษางานวิจัยที่เกี่ยวข้องพบว่ายังมีข้อจำกัดคือ ที่ตั้งของโรงเรือนต้องมีสัญญาณอินเทอร์เน็ตหากไม่สัญญาณอินเทอร์เน็ตก็จะไม่สามารถมอนิเตอร์และควบคุมโรงเรือนผ่านอินเทอร์เน็ตได้ และในการแสดงค่าสถานะการทำงานของอุปกรณ์ควบคุมสภาพแวดล้อมภายในโรงเรือนเป็นค่าที่ได้จากซอฟต์แวร์ ซึ่งหากอุปกรณ์เกิดขัดข้องหรือมีปัญหาเกิดขึ้นจะไม่สามารถทราบได้ว่าอุปกรณ์ทำงานหรือไม่

ดังนั้นจากปัญหาที่กล่าวมาผู้จัดทำจึงได้ออกแบบและสร้างโรงเพาะเห็ด ที่สามารถควบคุมสภาพแวดล้อมภายในโรงเรือนให้เหมาะสมต่อการออกดอกของเห็ด ออกแบบและสร้างเว็บแอปพลิเคชันและวินโดว์แอปพลิเคชัน ที่สามารถมอนิเตอร์ค่าสภาพแวดล้อมและควบคุมโรงเรือนได้ และประยุกต์ใช้งานเทคโนโลยีการสื่อสารไร้สายระยะไกล (LoRa) เพื่อเป็นตัวกลางการสื่อสารระหว่างโรงเรือนเพาะเห็ด (Mushroom Node) และพื้นที่ที่มีสัญญาณอินเทอร์เน็ต (STA Node) เพื่อแก้ปัญหาที่ตั้งของโรงเรือนไม่มีสัญญาณอินเทอร์เน็ต และผู้จัดทำได้ประยุกต์ใช้งานตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) เพื่อใช้ในการตรวจสอบว่าอุปกรณ์ทำงานจริงตามที่ควบคุมหรือไม่ เพื่อให้การแสดงค่าสถานะการทำงานของอุปกรณ์ควบคุมสภาพแวดล้อมที่ถูกต้อง

**1.2. วัตถุประสงค์**

1.2.1 เพื่อออกแบบและสร้างเว็บแอปพลิเคชันสำหรับมอนิเตอร์และควบคุมโรงเรือนเพาะเห็ด

1.2.2 เพื่อออกแบบและสร้างวินโดว์แอปพลิเคชันสำหรับมอนิเตอร์และควบคุมโรงเรือนเพาะเห็ด

1.2.3 เพื่อออกแบบและสร้างโรงเรือนที่ใช้สำหรับเพาะเห็ด

1.2.4 เพื่อออกแบบระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ด

1.2.5 เพื่อศึกษาและประยุกต์ใช้งานเทคโนโลยีการสื่อสารไร้สายระยะไกล (LoRa) สำหรับการควบคุมโรงเรือนเพาะเห็ด

1.2.6 เพื่อศึกษาการพัฒนาเว็บแอปพลิเคชันบนไมโครคอนโทรลเลอร์ (ESP32LoRa)

**1.3 ขอบเขต**

1.3.1 เว็บแอปพลิเคชัน

1) สามารถมอนิเตอร์และควบคุมโรงเรือนได้จากทุกที่ที่สามารถเข้าถึงอินเทอร์เน็ต

2) ใช้แอปพลิเคชัน ngrok ที่ติดไว้บน Raspberry Pi ทำ Port forwarding Web Application เพื่อให้สามารถใช้งานเว็บแอปพลิเคชันได้จากทุกที่ที่สามารถเข้าถึงอินเทอร์เน็ต

3) ใช้ Raspberry Pi 3 Model B เป็น Database และทำ Port forwarding

4) ใช้ไมโครคอนโทรลเลอร์ (ESP32LoRa) เป็น Web Server

1.3.2 วินโดว์แอปพลิเคชัน

1) สามารถมอนิเตอร์และควบคุมโรงเรือนได้ผ่านทาง Serial Port ระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์ (ESP32LoRa)

2) วินโดว์แอปพลิเคชันสามารถใช้งานได้กับไมโครคอนโทรลเลอร์ที่อยู่โรงเรือนเพาะเห็ด (Mushroom Node) และไมโครคอนโทรลเลอร์ที่ทำหน้าที่เป็น Web Server (STA Node)

3) วินโดว์แอปพลิเคชันสามารถนำไปติดตั้งและใช้งานกับคอมพิวเตอร์เครื่องอื่นที่เป็นระบบปฏิบัติการวินโดว์

1.3.3 โรงเรือนที่ใช้สำหรับเพาะเห็ด

1) ออกแบบโรงเรือนสำหรับเพาะเห็ดขนาด (กว้าง x ยาว x สูง) 200x150x210 เซนติเมตรเป็นโรงเรือนแบบปิดสามารถควบคุมสภาพแวดล้อมได้

2) ใช้ก้อนเห็ดนางฟ้าที่เชื้อเดินเต็มก้อนมาทำการทดลอง

3) ใช้ไมโครคอนโทรลเลอร์ (ESP32LoRa) เป็นไมโครคอนโทรลเลอร์ที่ใช้ในการควบคุมโรงเรือนเพาะเห็ด

**1.4 ประโยชน์ที่คาดว่าจะได้รับ**

1.4.1 เว็บแอปพลิเคชันสำหรับมอนิเตอร์และควบคุมโรงเรือนเพาะเห็ด

1.4.2 วินโดว์แอปพลิเคชันสำหรับมอนิเตอร์และควบคุมโรงเรือนเพาะเห็ด

1.4.3 โรงเรือนสำหรับเพาะเห็ดที่สามรถควบคุมสภาพแวดล้อมภายในโรงเรือนให้เหมาะสมต่อการเจริญเติบโตของเห็ด

1.4.4 ระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ด

1.4.5 ความรู้และการประยุกต์ใช้งานเทคโนโลยีการสื่อสารไร้สายระยะไกล (LoRa) สำหรับการควบคุมโรงเรือนเพาะเห็ด

1.4.6 ความรู้การพัฒนาเว็บแอปพลิเคชันบนไมโครคอนโทรลเลอร์ (ESP32LoRa)

**บทที่ 2**

**ทฤษฎีและงานวิจัยที่เกี่ยวข้อง**

ในการจัดทำโครงงานโรงเพาะเห็ดอัจฉริยะโดยใช้เทคโนโลยีการสื่อสารไร้สายระยะไกลผู้จัดทำโครงงานได้ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องเพื่อใช้ในการดำเนินโครงงานให้เป็นไปตามวัตถุประสงค์และขอบเขตของโครงงาน

**2.1 ทฤษฎีที่เกี่ยวข้อง**

**2.1.1 เห็ดนางฟ้าและเห็ดนางฟ้าภูฐาน [6]**

เห็ดนางฟ้า (Phoenix Oyster Mushroom) มีชื่อสามัญคือ Sajor-caju ชื่อวิทยาศาสตร์ Pleurotus sajor-caju (Fr.) Sing มีวงจรชีวิตแบบผสม (Heterothallic) ลักษณะสัณฐานของเห็ด นางฟ้าจะประกอบไปด้วยส่วนของหมวกดอก (Cap) ก้านดอก (Stalk) ครีบดอก (Gills) และเส้นใย (Mycelium) ขนาดเส้นผ่านศูนย์กลางประมาณ 3-6 cm มีก้านดอกไม่สมดุล (Eccentric) ออกดอกเป็นดอกเดียวหรือเป็นกลุ่ม สปอร์ของเห็ดมีขนาดประมาณ 3.5x8.6 μm2 เป็นเห็ดที่มี 23 คุณค่าทางอาหาร มีโปรตีนและคาร์โบไฮเดรตสูง ไขมันต่ า อุดมไปด้วยธาตุอาหารต่าง ๆ และ กรดอะมิโนที่จ าเป็นต่อร่างกาย มีรสชาติดี มีความกรอบ และเก็บไว้ได้นาน

เห็ดนางฟ้าภูฐาน (Bhutan Oyster Mushroom) เป็นเห็ดตระกูลเดียวกับเห็ดนางฟ้า ซึ่งพบที่ประเทศภูฐาน จัดเป็นเห็ดในกลุ่มกินอยู่กับอินทรียวัตถุ (Saprophytes) คือเป็นเห็ดที่กิน ซากพืช หรือย่อยสลายซากพืช เช่น กินต้นไม้ที่ตายแล้ว ฟาง หญ้าแห้ง ขี้เลื่อย เศษพืชต่าง ๆ ใบไม้ จะถูกเห็ดราย่อยสลายให้ผุกร่อนลดรูปกลายเป็นสารอินทรีย์ปนไปกับดิน เจริญเติบโตได้เร็วมากในบริเวณที่อากาศชื้นและเย็น ดอกเห็ดมี สีขาวจนถึงสีน้ำตาลอ่อน หมวกดอกเนื้อแน่นสีคล้ำ ก้านดอกสีขาว ขนาดยาวไม่มีวงแหวน ล้อมรอบ ครีบดอกสีขาวอยู่ชิดติดกันมากกว่าครีบดอกเห็ดเป๋าฮื้อ เส้นใยค่อนข้างละเอียด และสิ่งที่ เด่นกว่าเห็ดชนิดอื่นคือการออกดอกเห็ดเร็ว ระยะช่วงห่างของการออกดอกสั้น มีความสามารถในการใช้อาหารสูง ถ้าใส่อาหารมากผลผลิตก็มาก มีความต้านทานราสีเขียวและราสีดำสูง สามารถเพาะได้ตลอดปี เห็ดนางฟ้าภูฐานจึงเป็นผลิตผลทางเกษตรกรรมที่มีความสำคัญ ให้ผลผลิตสูงประมาณ 300-350 กรัมต่อถุงก้อนเชื้อ 1 กิโลกรัม เป็นที่นิยมในการผลิต จำหน่าย และบริโภค อีกทั้งอุดมไปด้วยคุณประโยชน์ในด้านโภชนาการ

|  |
| --- |
| **รูปที่ 2.1** แสดงภาพเห็ดนางฟ้าภูฐานที่เจริญเติบโตจากถุงเชื้อ |

**2.1.1.1 วงจรชีวิตของเห็ดนางฟ้า**

เห็ดนางฟ้ามีวงจรชีวิตแบบผสม (Heterothallic) เห็ดที่เจริญเติบโตเต็มที่จะมีการสร้าง สปอร์ที่บริเวณครีบดอก เมื่อสปอร์แก่ก็จะถูกปล่อยออกมา เมื่อไปตกบริเวณที่เหมาะสมจะเจริญ ไปเป็นเส้นใยขั้นแรก (Primary Mycelium) แต่ไม่สามารถพัฒนาตนเองไปเป็นเส้นใยขั้นที่ 2 (Secondary Mycelium) ต้องอาศัยการผสมกับเส้นใยระยะที่ 1 เส้นอื่นซึ่งเจริญมาจากสปอร์อื่น แล้วจึงจะพัฒนาไปเป็นเส้นใยระยะที่ 2 ได้ จากนั้นก็จะเจริญรวมเป็นกลุ่มก้อน ดอกเห็ดอ่อนที่มี ลักษณะเป็นก้อนกลมคล้ายสำลีจะค่อย ๆ ขยายโตขึ้น เนื้อเยื่อภายในเห็ดอ่อนที่มีแต่เส้นใยอัด แน่นเป็นก้อน จะพัฒนาอย่างช้า ๆ เจริญออกเป็นอวัยวะต่าง ๆ เมื่อดอกโตเต็มที่ก็จะสร้างสปอร์ แล้วบานออกเพื่อกระจายสปอร์ต่อไป

|  |
| --- |
| **รูปที่ 2.2** แสดงภาพวงจรชีวิตของเห็ดนางฟ้า |

**2.1.1.2 สภาพแวดล้อมที่เหมาะสมต่อการเจริญเติบโตของเห็ดนางฟ้า**

1. อุณหภูมิ ในช่วงที่เป็นเส้นใยจนกระทั่งเป็นดอก อุณหภูมิที่เหมาะสมคือ 21-30 °C ดอกเห็ดจะออกเร็วมาก หากอุณหภูมิต่ำกว่า 15 °C หรือสูงกว่า 35 °C เห็ดจะไม่ออกดอก หรือถ้า ออกดอกแล้วก็จะชะงักการเจริญเติบโต และถ้าหนาวเกินไปจะมีสีซีดด้วย

2. อากาศ เห็ดต้องการออกซิเจนค่อนข้างมาก โดยเฉพาะอย่างยิ่งเวลาเกิดดอกเห็ด แต่ในโรงเรือนจะมีการสะสมแก๊สคาร์บอนไดออกไซด์เนื่องจากการหายใจของเห็ด ถ้ามีคาร์บอนไดออกไซด์พอเหมาะจะช่วยกระตุ้นเส้นใยในการสร้างตุ่มเห็ด แต่ถ้ามีมากเกินไป เห็ดนางฟ้าจะมีลำต้นยืดยาวออกไปดอกเห็ดบิดเบี้ยว มีขนาดเล็ก หรือไม่ออกดอก

3. ความชื้น การเพิ่มความชื้นในวัตถุเพาะทำได้โดยการรดน้ำ แต่ต้องระวังไม่ให้มาก เกินไปเพราะจะทำให้อากาศในวัสดุเพาะจะลดลงขาดออกซิเจน เส้นใยชะงักการเจริญเติบโต และอาจทำให้จุลินทรีย์อื่นที่อยู่ด้วยเจริญขึ้นแทนเส้นใยเห็ด แต่ถ้าขาดความชื้นก็จะไม่สามารถ ย่อยสลายอาหาร เส้นใยก็เจริญเติบโตไม่ได้ ส่วนความชื้นในอากาศนั้น ถ้ามีน้อยเกินไปนั้น น้ำในดอกเห็ดจะระเหยออกไป ทำให้ดอกเห็ดชะงักการเจริญ แต่ถ้าความชื้นมากเกินไป จะเกิด เส้นใยฟูขึ้นจากดอกเห็ด ทำให้ฉ่ำน้ำและการเกิดดอกเห็ดจะลดลงมาก

4. แสง มีความจำเป็นในการกระตุ้นให้เส้นใยรวมตัวกันเพื่อให้เกิดดอกเห็ดเร็วขึ้นในระยะเห็ดออกดอก หากมีแสงน้อยเกินไปดอกเห็ดจะไม่สมบูรณ์ แต่เห็ดนางฟ้านั้นต้องการแสง ไม่มาก ขนาดที่เหมาะสมคือ สว่างพอที่จะอ่านหนังสือออก แสงสีน้ำเงินจะกระตุ้นการออกดอก ของเห็ดนางฟ้ามากกว่าสีอื่น

5. ความสะอาด โรงเรือนที่ไม่สะอาดจะเป็นแหล่งสะสมเชื้อโรคที่เป็นผลเสียต่อเห็ด การรักษาความสะอาด ทำได้โดยการไม่ให้โรงเรือนสกปรกรกรุงรังไปด้วยสิ่งที่ไม่เกี่ยวข้องกับ การเพาะเห็ด ไม่ปล่อยให้เศษก้อนเชื้อหกหล่นหมักหมมตามมุมต่าง ๆ มีทางระบายน้ำ

**2.1.1.3 การเพาะเลี้ยงเห็ดนางฟ้านในประเทศไทยและโรงเรือนสำหรับเปิดดอกเห็ด**

เห็ดนางฟ้าจะออกดอกและเป็นดอกสมบูรณ์ได้ดี ในอากาศที่มีความชื้นสูงประมาณ 60-80 % ขึ้นไป จึงจำเป็นต้องเพาะเห็ดในโรงเรือนที่เก็บความชื้นได้ในระดับดังกล่าว โดยทั่วไป หากผลิตเพื่อจำหน่าย จะนิยมสร้างโรงเรือนขนาด 4×6 m2 , 6×6 m2 , 8×16 m2 หรือ 8×20 m2 แล้วแต่ขนาดของพื้นที่และความต้องการของผู้ผลิต และปัจจุบันมีการผลิตเพื่อรับประทานใน ครัวเรือน หรือเป็นงานอดิเรก จะเพาะเพียงจำนวนน้อยและ ไม่มีพื้นที่มาก บ้างก็ใช้ไม้มาต่อเป็นตู้ ขนาดตามความสะดวก ใช้ผ้าพลาสติกปิดโดยรอบ ด้านหนึ่งคลุมให้เปิด-ปิด สามารถรดน้ำ และเก็บเห็ดได้สะดวก ตู้นี้จะต้องวางในมุมอับลมและไม่ถูกแดดส่องโดยตรง นอกจากนี้ ปัจจุบัน ยังนิยมใช้โอ่งดินเผาหรือวงบ่อซีเมนต์ ซึ่งมีคุณสมบัติใกล้เคียงกันในการกักเก็บความชื้น ต้นทุน การผลิตต่ำ สามารถใช้โอ่งดินเผาหรือวงบ่อซีเมนต์ที่แตกชำรุดแล้วมาประยุกต์ได้ มีความสะดวก ในการดูแล รดน้ำ ตลอดจนการเก็บเห็ด โดยนำโอ่งดินเผาหรือวงบ่อซีเมนต์มาวางในแนวนอน สำหรับวงบ่อซีเมนต์จะปิดด้านหนึ่งให้สนิทเหมือนกับก้นโอ่งดินเผา และอีกด้านหนึ่งจะใช้วัสดุคลุม ที่ทึบแสงแต่มีรูระบายอากาศบ้าง เช่น กระสอบป่าน เป็นต้น

|  |
| --- |
| **รูปที่ 2.3** แสดงภาพโรงเรือนเพาะเห็ด |

**2.1.1.4 ลักษณะก้อนเชื้อที่นำมาเพาะ**

ก้อนเชื้อเห็ดนางฟ้าที่เหมาะสำหรับนำมาเปิดดอกให้ความชื้น เพาะให้เป็นดอกนั้น ควรจะมีเส้นใยสีขาวทั่วตลอดทั้งก้อนแล้ว จับตัวเป็นก้อนแข็งดีพอสมควรไม่เหลวเละ ก้อนเชื้อเห็ด ที่มีอายุมากอาจมีสีขาวจัดและอาจมีตุ่มเห็ดสีขาวขึ้นมาบ้างแล้ว เมื่อนำไปเปิดดอกจะได้ดอกเห็ด เร็วในประมาณหนึ่งสัปดาห์ ถ้าหากก้อนเชื้อมีอายุมากเกินไป จะมีน้ำสีเหลืองข้างถุงซึ่งเกิดจาก เส้นใยแตกตัวลงเอง (Autolysis) เมื่อนำไปเปิดดอกจะเกิดดอกเห็ดเร็วมากแต่ผลผลิตรวมต่ำ

การวางถุงก้อนเชื้อในโรงเรือน สามารถทำได้หลายวิธี ดังนี้

1. เปิดจุกสำลีให้ออกดอกเห็ดที่ปากถุง จะวางถุงในแนวนอนกับพื้น โดยวางซ้อนกัน เป็นชั้น พ่นละอองน้ำเป็นฝอยละเอียด เห็ดจะเกิดแล้วโผล่ออกมาทางปากถุงได้เอง เป็นวิธีที่นิยมมากที่สุด สามารถให้ผลผลิตได้หลายรุ่น เมื่อเก็บผลผลิตได้ 2-3 รุ่น ก้อนเชื้อจะยุบตัวลงมา ทำให้ ถุงเชื้อแน่นอยู่ตลอดเวลา เส้นใยเห็ดสามารถส่งอาหารเพื่อทำให้เกิดดอกเห็ดใหม่ได้อีกหลายครั้ง แต่ก้อนเชื้อชั้นล่างๆ มักจะถูกทำลายด้วยราเมือก หรือเน่าเปื่่อยเสียก่อน ดังนั้น จึงไม่ควรวาง ซ้อนกันเกิน 12 ถุง

2. พับปากถุง หลังจากที่แกะคอขวดออกแล้วให้เปิดปากถุงพับลงมา ม้วนปากถุงให้อยู่ในระดับเดียวกับวัสดุเพาะหรือก้อนเชื้อ อาจวางก้อนเชื้อเห็ดได้ทั้งแนวนอนหรือแนวตั้งบนชั้น โดยวางชิด ๆ กัน จะเกิดดอกเห็ดครั้งละหลายดอกแต่ดอกเล็กลงเพราะแย่งอาหารกัน การวาง ก้อนเชื้อเห็ดบนชั้นจะวางได้จำนวนถุงเชื้อน้อย จึงเก็บความชื้นได้น้อย แต่อากาศหมุนเวียนได้ดี จึงต้อง คอยรักษาความชื้นในโรงเรือนไม่ให้แห้งเร็วเกินไป

3. ตัดปากถุง เป็นการเปิดปากถุงโดยใช้มีดโกนปาดปากถุงออกตรงส่วนคอขวด เหลือถุงพลาสติกหุ้มก้อนเชื้อส่วนบนอยู่บางส่วน วิธีนี้จะได้ดอกเห็ดน้อยกว่าการเปิดจุกสำลี แต่จะ ได้มวลเห็ดมากกว่า

4. กรีดข้างถุง เมื่อถอดและจุกสำลีออก รวบปากถุงรัดยางให้แน่น ใช้มีดคม ๆ กรีดข้าง ถุง ให้เป็นแนวยาวประมาณ 5-10 แถว หรือกีดแบบเฉียงเล็กน้อย ยาวประมาณ 6-8 cm หรือกรีด เป็นรอยกากบาทเล็ก ๆ ก็ได้ อาจวางถุงบนชั้นในแนวนอนแล้วกรีดด้านก้นถุงอีกด้านหนึ่ง หรือจะ ไม่วางบนชั้นแต่ใช้เชือกรัดปากถุงให้แน่น แขวนไว้ในแนวตั้งสลับสูงบ้างต่ำาบ้าง ระยะห่างของถุง ประมาณ 5-7 cm วิธีนี้จะเสียเวลาแต่โรงเรือนจะสะอาด โอกาสที่ก้อนเชื้อจะถูกรบกวนจากมดและ ไรน้อยกว่าการวางบนชั้น

5. การเปลือยถุง แกะถุงพลาสติกออกหมดทั้งก้อนแล้ววางเชื้อลงในแบบไม้หรือตะกร้า รดน้ำให้เปียกทั่วทั้งก้อน จะสามารถเกิดดอกเห็ดเร็วและเกิดได้ทุกส่วน แต่ต้องรักษาความชื้น ในโรงเรือนให้สูงมาก เพราะก้อนเชื้อจะสูญเสียความชื้นอย่างรวดเร็ว และดอกเห็ดเล็กเพราะแย่ง อาหารกัน

6. เพาะแบบแขวน ใช้เชือกไนล่อน 4 เส้น ผูกติดกันด้านหัวท้าย ส่วนตรงกลางใส่แผ่น พลาสติกแข็งเจาะรูร้อยเชือกทั้ง 4 เส้น ถ่างห่างออกจากกัน วางก้อนเชื้อซ้อนกันได้หลายถุง แขวนห้อยจากคานด้านบน พื้นเรือนเพาะจึงสะอาด ศัตรูเห็ดมีน้อย

**2.1.1.5 การรดน้ำ**

น้ำเป็นปัจจัยที่ทำให้เกิดการสร้างดอกเห็ด จึงต้องใช้น้ำสะอาด ปราศจากสารเคมีและ สิ่งสกปรก ไม่มีกลิ่น และสารที่เป็นอันตรายต่อเห็ด เช่น เกลือแกง คลอรีน เป็นต้น จึงควรใช้น้ำฝน น้ำบ่อ น้ำบาดาล มีความเป็นกลาง เครื่องมือที่ใช้รดน้ำอาจเป็นระบบพ่นฝอย หรือฝักบัวที่ฝอย ละเอียดมาก ๆ ในช่วงที่เห็ดยังไม่เป็นดอก จำนวนครั้งที่รดน้ำขึ้นอยู่กับความชื้นในอากาศรอบ ๆ ต้องให้มีความชื้นไม่ต่ำกว่า 80 % แต่ต้องระมัดระวังอย่าให้น้ำเข้าไปในก้อนเชื้อเห็ด จะทำให้เชื้อ เห็ดเน่า เมื่อดอกเห็ดอายุ 1-2 วัน สามารถรดนำให้ดอกเห็ดเปียกได้ แต่เมื่อดอกเริ่มบานออก ให้รดน้ำเพียงวันละ 2-4 ครั้ง ขึ้นกับลักษณะการขาดน้ำของเห็ด โดยรดไปตามพื้น เพดาน และ ด้านข้างของก้อนเชื้อ

**2.1.1.6 การเก็บเกี่ยวผลผลิตดอกเห็ด**

หลังจากเปิดดอกให้ความชื้นแล้ว ภายใน 2-3 สัปดาห์ จะเกิดดอกเห็ดเล็ก ๆ ถ้าหาก ดูแลรักษาต่อได้ดี ดอกเห็ดจะโตเต็มที่ภายใน 4-5 วัน ส่วนมากจะเก็บได้ในวันที่ 4 ถ้าปล่อยไว้นาน กว่านั้น ดอกเห็ดจะสร้างสปอร์ออกมาเป็นผงสีขาวละเอียด หลุดร่วงหล่นลงมาด้านล่าง ดอกเห็ด จะเสียคุณภาพ จึงควรสังเกตลักษณะที่เหมาะสม คือ ก้านของดอกเห็ดจะหยุดการเจริญทางความยาว หมวกดอกเริ่มคลี่ออกประมาณครึ่งหนึ่งแล้วเริ่มสร้างสปอร์ ขอบดอกจะหนาและรวมตัวเข้า หากัน เมื่อเจริญเต็มที่แล้วขอบดอกจะคลี่ออกและบางลงกว่าเดิม เป็นระยะที่ควรเก็บเกี่ยวได้ โดย ให้ใช้มือดึงที่โคนออกมาเบา ๆ ไม่ควรใช้มีดตัดเพราะเศษเห็ดที่ติดอยู่กับก้อนเชื้อจะเน่า เกิดเป็น แหล่งสะสมเชื้อโรค โดยทั่วไปแล้ว ก้อนเชื้อเห็ดนางฟ้าสูตรธรรมดาขนาด 1 กิโลกรัม จะให้ผลผลิต ครั้งละ 50-60 กรัม แต่ละก้อนจะให้ผลผลิตประมาณ 4-6 รุ่น แต่ละรุ่นมีช่วงห่างระหว่างการเกิด ดอกประมาณ 10-15 วัน

**2.1.2 อุณหภูมิ (Temperature) [7]**

คือการวัดค่าเฉลี่ยของพลังงานจลน์ของอนุภาคในสสารใด ๆ ซึ่งสอดคล้องกับความร้อนหรือเย็นของสสารนั้น ในอดีตมีแนวคิดเกี่ยวกับอุณหภูมิเกิดขึ้นเป็น 2 แนวทาง คือตามแนวทางของหลักอุณหพลศาสตร์ และตามการอธิบายเชิงจุลภาคทางฟิสิกส์เชิงสถิติ แนวคิดทางอุณหพลศาสตร์นั้น ถูกพัฒนาขึ้นโดยลอร์ดเคลวิน โดยเกี่ยวข้องกับการวัดในเชิงมหภาค ดังนั้นคำจำกัดความอุณหภูมิในเชิงอุณหพลศาสตร์ในเบื้องแรก จึงระบุเกี่ยวกับค่าตัวแปรต่าง ๆ ที่สามารถตรวจวัดได้จากการสังเกต ส่วนแนวทางของฟิสิกส์เชิงสถิติจะให้ความเข้าใจในเชิงลึกยิ่งกว่าอุณหพลศาสตร์ โดยอธิบายถึงการสะสมจำนวนอนุภาคขนาดใหญ่ และตีความพารามิเตอร์ต่าง ๆ ในอุณหพลศาสตร์ (เชิงมหภาค) ในฐานะค่าเฉลี่ยทางสถิติของพารามิเตอร์ของอนุภาคในเชิงจุลภาค

ในการศึกษาฟิสิกส์เชิงสถิติ สามารถตีความคำนิยามอุณหภูมิในอุณหพลศาสตร์ว่า เป็นการวัดพลังงานเฉลี่ยของอนุภาคในแต่ละองศาอิสระในระบบอุณหพลศาสตร์ โดยที่อุณหภูมินั้นสามารถมองเป็นคุณสมบัติเชิงสถิติ ดังนั้นระบบจึงต้องประกอบด้วยปริมาณอนุภาคจำนวนมากเพื่อจะสามารถบ่งบอกค่าอุณหภูมิอันมีความหมายที่นำไปใช้ประโยชน์ได้ ในของแข็ง พลังงานนี้พบในการสั่นไหวของอะตอมของสสารในสภาวะสมดุล ในแก๊สอุดมคติ พลังงานนี้พบในการเคลื่อนไหวไปมาของอนุภาคโมเลกุลของแก๊ส

**2.1.2.1 ความร้อนและอุณหภูมิ**

สสารทั้งหลายประกอบด้วย อะตอมรวมตัวกันเป็นโมเลกุล การเคลื่อนที่ของอะตอม หรือการสั่นของโมเลกุล ทำให้เกิดรูปแบบของพลังงานจลน์ ซึ่งเรียกว่า “ความร้อน” (Heat) เราพิจารณาพลังงานความร้อน (Heat energy) จากพลังงานทั้งหมดที่เกิดขึ้นจากการเคลื่อนที่ของอะตอมหรือโมเลกุลทั้งหมดของสสาร

อุณหภูมิ (Temperature) หมายถึง การวัดค่าเฉลี่ยของพลังงานจลน์ซึ่งเกิดขึ้นจากอะตอมแต่ละตัว หรือแต่ละโมเลกุลของสสาร เมื่อเราใส่พลังงานความร้อนให้กับสสาร อะตอมของมันจะเคลื่อนที่เร็วขึ้น ทำให้อุณหภูมิสูงขึ้น แต่เมื่อเราลดพลังงานความร้อน อะตอมของสสารจะเคลื่อนที่ช้าลง ทำให้อุณหภูมิลดต่ำลง

หากเราต้มน้ำด้วยถ้วยและหม้อบนเตาเดียวกัน จะเห็นได้ว่าน้ำในถ้วยจะมีอุณหภูมิสูงกว่า แต่จะมีพลังงานความร้อนน้อยกว่าในหม้อ เนื่องจากปริมาณความร้อนขึ้นอยู่กับมวลทั้งหมดของสสาร แต่อุณหภูมิเป็นเพียงค่าเฉลี่ยของพลังงานในแต่ละอะตอม ดังนั้นบรรยากาศชั้นบนของโลก (ชั้นเทอร์โมสเฟียร์) จึงมีอุณหภูมิสูง แต่มีพลังงานความร้อนน้อย เนื่องจากมีมวลอากาศอยู่อย่างเบาบาง

**2.1.3 ความชื้น (Humidity) [8]**

ความชื้น (Humidity) หมายถึง ปริมาณไอน้ำที่มีอยู่ในอากาศ เมื่อน้ำได้รับความร้อนจะเปลี่ยนสถานะจากของเหลวกลายเป็นไอลอยขึ้นในอากาศ เรียกว่า การระเหย ซึ่งความร้อนที่ใช้ในการทำให้น้ำระเหยกลายเป็นไอ เรียกว่า ความร้อนแฝง (Latent Heat) เมื่ออากาศเย็นลงไอน้ำจะเริ่มกลั่นตัวเป็นละอองและคายความร้อนแฝงออกมาด้วย กาศจะได้รับไอน้ำมากหรือน้อยขึ้นอยู่กับอุณหภูมิ ดังนั้นอุณหภูมิจึงเป็นตัวกำหนดปริมาณไอน้ำในอากาศ อากาศที่มีอุณหภูมิสูงจะรับไอน้ำได้มากกว่าอากาศที่มีอุณหภูมิต่ำ ถ้าอากาศไม่สามารถรับไอน้ำได้เรียกว่า ไอน้ำอิ่มตัว (Saturate)

**2.1.3.1 ความชื้นในอากาศ**

ความชื้นในอากาสสามารถแบ่งออกเป็น 4 ประเภท คือ

1. ความชื้นสัมบูรณ์ (Absolute Humidity)เป็นน้ำหนักของไอน้ำที่มีอยู่จริงในปริมาตรของอากาศจำนวนหนึ่งคำนวณได้จากน้ำหนักของไอน้ำต่อหนึ่งหน่วยปริมาตรของอากาศ หรือกล่าวอีกนัยหนึ่งว่าความชื้นสัมบูรณ์คือความหนาแน่นของไอน้ำในอากาศ หน่วยที่ใช้มักเป็นกรัมต่อลูกบาศก์เมตรความชื้นสัมบูรณ์ไม่นิยมใช้ในทางอุตุนิยมวิทยาเพราะเมื่ออากาศลอยตัวขึ้นหรือจมตัวลงจะทำให้ปริมาตรของอากาศเปลี่ยนแปลงเนื่องจากบริเวณรอบๆ ความกดอากาศจะเปลี่ยนแปลงแม้ว่าไอน้ำที่มีอยู่ในอากาศจะคงที่

2. ความชื้นจำเพาะ (Specific Humidity) คือน้ำหนักหรือความชื้นที่มีอยู่ในอากาศ (Q) เป็นอัตราส่วนระหว่างน้ำหนักไอน้ำ (Mv) ต่อน้ำหนักของอากาศชื้น น้ำหนักของไอน้ำ (Mv) รวมกับน้ำหนักของอากาศ (Ma) มักใช้เป็นกรัมของน้ำต่อ 1 กิโลกรัมของอากาศชื้น ดังสมการที่ 2.1

(2.1)

ความชื้นจำเพาะของอากาศจะมีค่าคงที่ เมื่ออากาศขยายตัวหรือหดตัว โดยที่ความชื้นจะไม่เปลี่ยนแปลงแม้ว่าปริมาตรของอากาศจะขยายตัวหรือหดตัวก็ตาม

3. ความชื้นสัมพัทธ์ (Relative Humidity) คืออัตราส่วนระหว่างน้ำหนักของไอน้ำที่มีอยู่จริงที่อุณหภูมิและความกดดันหนึ่งต่อน้ำหนักของไอน้ำอิ่มตัวที่อุณหภูมิและความกดดันนั้น คิดเป็นค่าร้อยละตัวอย่างเช่น อากาศ 1 ลูกบาศก์เมตร เมื่ออุณหภูมิ 30 องศาเซลเซียสมีไอน้ำอยู่ 9 กรัม และในอุณหภูมินั้นอากาศอิ่มตัวมีไอน้ำอยู่ 30 กรัม ความชื้นสัมพัทธ์เท่ากับ (100\*9)/30 เท่ากับ 30% ความชื้นสัมพัทธ์เป็นวิธีวัดความชื้นในอากาศที่ใช้มากที่สุด การเปลี่ยนแปลงความชื้นสัมพัทธ์จะไม่ทำให้ปริมาณไอน้ำที่มีอยู่ในอากาศเปลี่ยนแปลง แต่อณุหภูมิจะเปลี่ยนแปลง และถ้าอุณหภูมิเปลี่ยนแปลงความชื้นสัมพัทธ์จะเปลี่ยนแปลงด้วย

4. อุณหภูมิจุดน้ำค้าง (Dew Point Temperature) คืออุณหภูมิซึ่งอากาศถูกทำให้เย็นลง (ความกดอากาศคงที่) ถึงอุณหภูมิหนึ่งที่ที่ไอน้ำจุดอิ่มตัวพอดี อุณหภูมิของจุดน้ำค้างจะเป็นเท่าใดก็ได้ขึ้นอยู่กับจำนวนไอน้ำที่มีอยู่จริงในอากาศ ถ้าอากาศมีไอน้ำมากอุณหภูมิของจุดน้ำค้างจะสูง แต่ถ้าไอน้ำมีน้อยอุณหภูมิของจุดน้ำค้างจะต่ำ ถ้าอุณหภูมิของอากาศลดต่ำกว่าจุดน้ำค้างจะมีการกลั่นตัวในรูปของหยดน้ำ เช่น ในฤดูร้อนแก้วน้ำที่ใส่น้ำแข็งตั้งทิ้งไว้ความชื้นของอากาศจะรวมกันเป็นหยดน้ำเกาะอยู่รอบนอกแก้วน้ำ เนื่องจากอุณหภูมิของแก้วน้ำที่ใส่น้ำแข็งจะต่ำกว่าอุณหภูมิจุดน้ำค้างที่อยู่โดยรอบ อุณหภูมิของจุดน้ำค้างจะบอกถึงความไม่สะดวกสบายของมนุษย์ในช่วงที่มีอากาศอุ่นและชื้นได้ดีกว่าความชื้นสัมพัทธ์ ยกเว้นผู้ที่เคยชินกับอากาศร้อนชื้น คนส่วนใหญ่รู้สึกว่าอากาศชื้นไม่สะดวกสบายเมื่ออุณหภูมิจุดน้ำค้างสูงกว่า 17 องศาเซลเซียส ในขนะที่บางคนอาจไม่รู้สึกสบายตัว เมื่ออุณหภูมิจุดน้ำค้างสูงกว่า 20 องศาเซลเซียส

**2.1.3.2 ความชื้นสัมพัทธ์ของประเทศไทยในแต่ละฤดู**

ประเทศไทยตั้งอยู่ในเขตร้อนใกล้เส้นศูนย์สูตร จึงมีอากาศร้อนชื้นปกคลุมเกือบตลอดปี เว้นแต่บริเวณที่ลึกเข้าไปในแผ่นดิน ตั้งแต่ภาคกลางขึ้นไปความชื้นสัมพัทธ์จะลดลงชันเจน ในช่วงฤดูหนาวและฤดูร้อน โดยเฉพาะฤดูร้อนจะเป็นช่วงที่มีความชื้นสัมพัทธ์ลดลงต่ำที่สุดในรอบปี ในบริเวณดังกล่าวมีความชื้นสัมพัทธ์เฉลี่ยตลอดปีที่ 72-74 เปอร์เซ็นต์และจะลดลงเหลือ 62-69 เปอร์เซ็นต์ ในช่วงฤดูร้อน ดังแสดงข้อมูลตามตารางที่ 2.1

**ตารางที่ 2.1** แสดงสถิติความชื้นสัมพัทธ์เฉลี่ย (%) ของประเทศไทยในช่วงฤดูกาลต่างๆ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ภาค** | **ฤดูหนาว** | **ฤดูร้อน** | **ฤดูฝน** | **ตลอดปี** |
| เหนือ | 73 | 62 | 81 | 74 |
| ตะวันออกเฉียงเหนือ | 69 | 65 | 80 | 72 |
| กลาง | 71 | 69 | 79 | 73 |
| ตะวันออก | 71 | 74 | 81 | 76 |
| ใต้ฝังตะวันออก | 81 | 77 | 78 | 79 |
| ใต้ฝังตะวันตก | 77 | 76 | 84 | 80 |

**2.1.4 แสงสว่าง (Light) [9]**

แสงสว่าง เป็นพลังงานรูปหนึ่ง ซึ่งประกอบด้วยคลื่นแม่เหล็กไฟฟ้า ที่มีความยาวคลื่นประมาณ 380 -780 นาโมเมตร ซึ่งเป็นระยะความยาวคลื่นที่มองเห็นได้ (Visible Light) การเปลี่ยนแปลงของความยาวคลื่นของแสงสว่าง จะทำให้ตารู้สึกเห็นเป็นสีต่างๆ ตามความยาวคลื่นนั้น

สเปกตรัมแม่เหล็กไฟฟ้าและแสงที่เห็นได้

แสงคือรังสีแม่เหล็กไฟฟ้าที่อยู่ในช่วงสเปกตรัมของคลื่นแม่เหล็กไฟฟ้าที่สามารถมองเห็นได้ คือ อยู่ในย่านความถี่ 380 THz (3.8×1014 เฮิรตซ์) ถึง 789 THz (7.5×1014 เฮิรตซ์) จากความสัมพันธ์ระหว่าง ความเร็ว () ความถี่ () และ ความยาวคลื่น () ของแสง

(2.2)

และความเร็วของแสงในสุญญากาศมีค่าคงที่ ดังนั้นเราจึงสามารถแยกแยะแสงโดยใช้ตามความยาวคลื่นได้ โดยแสงที่เรามองเห็นได้ข้างต้นนั้นจะมีความยาวคลื่นอยู่ในช่วง 400 นาโนเมตร (ย่อ 'nm') และ 800 nm (ในสุญญากาศ)

การมองเห็นของมนุษย์นั้นเป็นผลมาจากภาวะอนุภาคของแสงโดยเฉพาะ เกิดจากการที่ก้อนพลังงาน (อนุภาคโฟตอน) แสง ไปกระตุ้น เซลล์รูปแท่งในจอตา (rod cell) และ เซลล์รูปกรวยในจอตา (cone cell) ที่จอตา (retina) ให้ทำการสร้างสัญญาณไฟฟ้าบนเส้นประสาท และส่งผ่านเส้นประสาทตาไปยังสมอง ทำให้เกิดการรับรู้มองเห็น

|  |
| --- |
| **รูปที่ 2.4** แสดงภาพของสเปกตรัมแสงที่มองเห็นได้ |

ความเข้มของแสงสว่างหรือปริมาณการส่องสว่าง (Illuminance) หมายถึง ปริมาณแสงสว่างที่ตกกระทบลงบนหนึ่งหน่วยพื้นที่ที่กำหนด

แหล่งกำเนิดของแสงสว่างมี 2 แหล่ง คือ

1. แสงสว่างจากธรรมชาติ (Natural Lighting) แหล่งกำเนิดของแสงสว่างในธรรมชาติที่สำคัญ คือ ดวงอาทิตย์ การใช้ประโยชน์จากแสงอาทิตย์อย่างเหมาะสม จะเป็นการประหยัดค่าใช้จ่ายได้มาก

2. แสงสว่างจากการประดิษฐ์ (Artificial Lighting) เป็นแหล่งกำเนิดแสงสว่างที่มนุษย์ได้ประดิษฐ์คิดค้นโดยอาศัยธรรมชาติและเทคโนโลยี ได้แก่ หลอดไฟฟ้าชนิดต่างๆ เช่น หลอดไฟฟ้าชนิดไส้หลอด, หลอดฟลูออเรสเซนต์,หลอดเมอคิวรี ,หลอดโซเดียม เป็นต้น

**2.1.5 LoRa (Long Range) [10]**

LoRa มากจากคำว่า Long Range เป็นหนึ่งในเทคโนโลยีด้าน LPWAN (Low-Power Wide-Area Network) เครือข่ายสื่อสารแบบกว้างที่เน้นใช้พลังงานต่ำ เป็นเทคโนโลยีที่ถือว่าเป็นหัวใจสำคัญสำหรับงานทางด้าน IoT (Internet of Things) จุดเด่นของ LPWAN เทคโนโลยีได้แก่ [10]

* Secure — Bidirectional Communication
* Simple Star Network Topology ( ไม่จำเป็นต้องมี network ที่ซับซ้อนเช่น Mesh หรือ Repeater )
* Low data rate
* Low cost
* Long battery life

LoRa เป็นเทคโนโลยีการส่งข้อมูลแบบไร้สายในระยะไกล โดยใช้เทคนิค Proprietary Spread Spectrum technology ซึ่งรูปแบบถูกพัฒนาโดย Semtech Corporation ซึ่งมีย่านความถี่ในแต่ละภูมิภาคที่แตกต่างกันดังตารางที่ 2.2

**ตารางที่ 2.2** แสดงความถี่ที่ใช้สำหรับ LoRa (หน่วยเป็น MHz)

|  |  |
| --- | --- |
| ไทย | 920-925 |
| ยุโรป | 867-869 |
| อเมริกาเหนือ | 902-928 |
| จีน | 470-510 |
| เกาหลี | 920-925 |
| ญี่ปุ่น | 920-925 |
| อินเดีย | 865-867 |

โดยประกาศจาก กสทช. ที่เกี่ยวข้องกับเทคโนโลยี LPWAN ว่าถ้ามีกำลังส่งไม่เกิน 500 มิลลิวัตต์ ได้รับยกเว้นไม่ต้องได้รับใบอนุญาตให้ มี ใช้ และนำออก ซึ่งเครื่องวิทยุคมนาคมและใบอนุญาตให้ตั้งสถานีวิทยุคมนาคม แต่ไม่ได้รับยกเว้น ใบอนุญาตให้ทำนำเข้า และค้าซึ่งเครื่องวิทยุ คมนาคม

ถ้ามีกำลังส่งสูงกว่า 500 มิลลิวัตต์จะต้องได้รับใบอนุญาตวิทยุคมนาคมที่ เกี่ยวข้อง และการใช้คลื่นความถี่ดังกล่าว จะต้องได้รับอนุญาตให้ใช้คลื่นความถี่ ตามมาตรา ๔๕ แห่งพระราชบัญญัติองค์กรจัดสรรคลื่นความถี่ และ กำกับการประกอบกิจการวิทยุกระจายเสียง วิทยุโทรทัศน์ และกิจการโทรคมนาคม พ.ศ. ๒๕๕๓ และจะต้องได้รับ ใบอนุญาตประกอบกิจการโทรคมนาคมแบบที่ สาม

|  |
| --- |
| **รูปที่ 2.5** แสดงภาพตัวอย่างค่า Data Rate (DR) |

จากรูปที่ 2.5 ตัวอย่างค่า Data Rate ( DR ) สังเกตจากรูป จะเห็นว่า DR เป็น 0 อุปกรณ์จะสามารถส่งข้อมูลได้ไกลที่สุด โดยสามารถส่งด้วย Bitrate ที่ต่ำที่สุดโดยการกำหนด Data Rate จะถูกกำหนดจาก Spreading Factor ( SF ) ตั้งแต่ 7– 12 โดยที่แบนวิดท์ ช่องสัญญาณ และค่า SF ที่ปรับได้อาจจะเปลี่ยนแปลงตาม Frequency plan ของแต่ละโซน

|  |
| --- |
| **รูปที่ 2.6** แสดงภาพความสัมพันธ์ระหว่าง BITRATE และระยะห่างของอุปกรณ์กับ Gateway |

จากรูปที่ 2.6 จะเห็นว่าเมื่ออุปกรณ์เข้าใกล้ gateway มากก็จะสามารถที่จะส่งข้อมูลด้วย BITRATE ที่สูงขึ้นได้ และการส่งข้อมูลจะเร็วขึ้นอีกด้วย รวมถึงพลังงานที่ใช้ในการส่งถ้าเทียบต่อขนาดของแพกเกตก็จะน้อยกว่าอุปกรณ์ที่อยู่ไกล Gateway ซึ่งในระดับ LoRaWAN จะมีโหมด ADR (Adaptive Data Rate) ที่เซตในแพกเกตการส่งข้อมูลเพื่อให้การเชื่อมต่อระหว่าง Gateway และ Device สามารถปรับ Spreading Factor แบบอัตโนมัติเพื่อประสิทธิภาพในการส่งโดยดูจากระยะการเชื่อมต่อระหว่าง Gateway และ Device โดยสามารถเลือกได้ว่าต้องที่จะ ปรับเพื่อส่งข้อมูลได้เร็วที่สุด หรือ ปรับเพื่ออายุการใช้งานแบตเตอรี่ที่ยาวนานที่สุดเป็นต้น

**2.1.5.1 คุณสมบัติหลักของเทคโนโลยี LoRa**

|  |
| --- |
| **รูปที่ 2.7** แสดงภาพคุณสมบัติหลักของเทคโนโลยี LoRa |

จากรูปที่ 2.7 เทคโนโลยี LoRa มีคุณสมบัติต่างๆ ดังต่อไปนี้

* Long Range สามารถเชื่อมต่ออุปกรณ์ที่ห่างกันไม่เกิน 30 ไมล์ในพื้นที่ชนบทและเจาะเข้าไปในสภาพแวดล้อมในเมืองที่หนาแน่นหรือในที่ลึก
* Geolocation มีแอปพลิเคชันการระบุตำแหน่งที่ใช้พลังงานต่ำ
* Low Power ใช้พลังงานน้อยที่สุดพร้อมอายุการใช้งานแบตเตอรี่ที่ยาวนานถึง 10 ปีลดค่าใช้จ่ายในการเปลี่ยนแบตเตอรี่
* Mobile รักษาการสื่อสารกับอุปกรณ์ที่เคลื่อนไหวโดยไม่ต้องใช้พลังงานมากเกินไป
* Secure มีการเข้ารหัส AES128 แบบ end-to-end การรับรองความถูกต้องซึ่งกันและกัน
* High Capacity รองรับข้อความนับล้านต่อสถานีฐานตอบสนองความต้องการของผู้ให้บริการเครือข่ายสาธารณะที่ให้บริการในตลาดขนาดใหญ่
* Standardized ความสามารถในการทำงานร่วมกันของอุปกรณ์และความพร้อมใช้งานทั่วโลกของเครือข่าย LoRaWAN สำหรับการปรับใช้แอปพลิเคชัน IoT ได้อย่างรวดเร็วทุกที่
* Low Cost ลดการลงทุนโครงสร้างพื้นฐานค่าใช้จ่ายในการเปลี่ยนแบตเตอรี่และค่าใช้จ่ายในการดำเนินงาน

**2.1.5.2 ความแตกต่างระหว่าง LoRa และ LoRaWan**

LoRa ใช้งานได้เฉพาะโปรโตคอลระดับ Link layer เหมาะอย่างยิ่งที่จะใช้ในการสื่อสารแบบ P2P (point to point) ระหว่างโหนด

LoRaWan สามารถใช้งานโปรโตคอลระดับ Network layer ทำให้สามารถสื่อสารกับอุปกรณ์อื่นที่เชื่อมต่อกันผ่าน Cloud platform

**2.1.6 ตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) [11]**

การตรวจสอบกระแสโดยการวัดแรงดันตกคร่อมตัวต้านทานเป็นวิธีทำสามารถทำได้ง่าย แต่ต้องพิจารณาปัจจัยต่างๆก่อนนำมาใช้งาน เช่น การเลือกค่าความต้านทาน ขนาดของตัวต้านทาน ความร้อน การวัดการไหลของกระแสมีพารามิเตอร์ที่สำคัญที่ต้องจัดการเพื่อให้ระบบทำงานอย่างมีประสิทธิภาพ วิธีที่ใช้กันทั่วไปในการตรวจสอบการไหลของกระแสทำได้โดยการนำตัวต้านทานที่มีค่าความต้านทานต่ำ ต่ออนุกรมกับโหลดแล้ววัดแรงดันตกคร่อมตัวต้านทานโดยใช้กฎของโอห์ม

(2.3)

การใช้ตัวต้านทานในการตรวจสอบกระแสเป็นวิธีพื้นฐานที่มีประสิทธิภาพ แต่ยังมีปัญหาด้านการออกแบบรายละเอียดปลีกย่อยอีกมากมายที่ต้องคำนึงถึง เช่น ตำแหน่งของตัวต้านทานตรวจสอบกระแส ค่าความต้านทาน ขนาดของตัวต้านทาน และข้อพิจารณาเชิงกลอีกหลายประการ

**2.1.6.1 ตำแหน่งการวางตัวต้านทานตรวจสอบกระแส**

ตำแหน่งการวางตัวต้านทานตรวจสอบกระแสมีการวางอยู่ 2 แบบคือ การตรวจจับด้านต่ำ (Low-side Sensing) และ การตรวจับด้านสูง (High-side Sensing) ในการตรวจจับด้านต่ำจะวางตัวต้านทานตรวจสอบกระแสระหว่างโหลด (Load) และ กราวด์ (Ground) ดังรูปที่ 2.8 ซึ่งช่วยให้สามารถใช้วงจรขยายสัณญาณวัดแรงดันตกคร่อมตัวต้านทานตรวบสอบกระแส (Current Sense Resistors) และกราวด์ (Ground) ได้ง่าย

|  |
| --- |
| **รูปที่ 2.8** แสดงภาพการวางตัวต้านทานตรวจสอบกระแสแบบการตรวจจับด้านต่ำ  (Low-side Sensing) |

แบบที่ 2 เป็นการตรวจจับด้านสูง (High-side Sensing) โดยการวงตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) ระหว่างแหล่งจ่าย (Power) และ โหลด (Load) ดังรูปที่ 2.9 การตรวจจับแบบนี้จะใช้วงจรขยายสัญญาณแบบ differential amplifier หรือ instrumentation amplifier ในการวัดแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแส (Current Sense Resistors)

|  |
| --- |
| **รูปที่ 2.9** แสดงภาพการวางตัวต้านทานตรวจสอบกระแสแบบการตรวจจับด้านสูง  (High-side Sensing) |

**2.2 งานวิจัยที่เกี่ยวข้อง**

**2.2.1 ไอโอทีแพลทฟอร์มโรงเพาะเห็ด [1]**

ขันติชัย, ชยุต (2560) ได้ศึกษา ออกแบบ และสร้างไอโอทีแพลทฟอร์มสำหรับใช้ในโรงเพาะเห็ด โดยการใช้ Raspberry Pi 3 Model B ซึ่งเป็น Microcontroller มาใช้วัดค่าอุณหภูมิและ ความชื้นจากเซ็นเซอร์ภายในโรงเรือน จากนั้นจึงส่งข้อมูลอุณหภูมิและความชื้นที่วัดได้ไปยัง Platform และเก็บข้อมูลอุณหภูมิและความชื้นลงบนฐานข้อมูลโดยใช้ MySQL เป็นทั้งฐานข้อมูล และตัวจัดการฐานข้อมูล นอกจากนั้นยังมีระบบ Web Application เพื่อใช้แสดงผลค่าอุณหภูมิและ ความชื้นที่ถูกส่งมาจาก Raspberry Pi หรือประวัติของค่าอุณหภูมิและความชื้นที่เก็บไว้ในฐานข้อมูล โดย Protocol ที่เป็นตัวกลางในการรับส่งข้อมูลระหว่าง Raspberry Pi, Platform และ Web Application คือ MQTT Protocol และใช้ร่วมกับ NodeJS ซึ่งเป็น Service หลักที่ใช้ภายใน Platform

|  |
| --- |
| **รูปที่ 2.10** แสดงภาพโครงสร้างระบบของงานวิจัยที่เกี่ยวข้อง(1) |

**2.2.2 การประยุกต์ใช้เทคโนโลยีไอโอทีควบคุมฟาร์มอัจฉริยะในโรงเรือนเพาะเห็ดนางฟ้า [2]**

วีรศักดิ์, สุรพงษ์ม, รัฐสิทธิ์ (2561) ได้ออกแบบระบบควบคุมอุณหภูมิและความชื้นสำหรับโรงเรือนเพาะเห็ดด้วย ไมโครคอนโทรลเลอร์โดยมีการศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องต่อการเพาะเห็ดตลอดจนการออกแบบโครงสร้างโรงเรือนที่เหมาะสมโดยแบ่งการทดสอบออก 2 ส่วนคือการทดสอบในส่วนของระบบควบคุมและการทดสอบผลผลิต ของดอกเห็ดในโรงเรือนที่มีการควบคุมอุณหภูมิและความชื้นโดยนำก้อนเห็ดนางรมและเห็ดนางฟ้ามาทดสอบจำนวน 300 ก้อนและเปรียบเทียบประสิทธิภาพโรงเรือนเพาะเห็ดที่มีการควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นกับโรงเรือน โดยประยุกต์ใช้ระบบไอโอทีที่ใช้เซ็นเซอร์วัดประกอบด้วย วัดอุณหภูมิ ความชื้นสัมพัทธ์ในโรงเรือนเห็ดนางฟ้าและควบคุมการเปิดปิดปั๊มน้ำให้สปริงเกอร์และพ่นหมอกแบบอัตโนมัติ และเซอร์วิสที่ใช้ในการ ส่งข้อมูลขึ้นระบบอินเตอร์เน็ตคือ NETPIE และเซอร์วิสย่อยคือ NETPIE freeboard ในการแสดงสถาณะความชื้น และเวลาแบบเรลไทม์(real time) และ NETPIE FEED ในการบันทึกข้อมูลความชื้นและเวลาและการดึงข้อมูลมาใช้งานคือ Node.JS ผ่านเซอร์วิส NETPIE REST API มาเป็นไฟล์ CSV ในส่วนแสดงสถาณะการทำงานของการให้น้ำ แบบสปริงเกอร์และพ่นหมอกแบบอัตโนมัติผ่านมือถือ

|  |
| --- |
| **รูปที่ 2.11** แสดงภาพโครงสร้างระบบของงานวิจัยที่เกี่ยวข้อง (2) |

**2.2.3 IOT for Smart Farm: A case study of the Lingzhi Mushroom Farm at Maejo University [3]**

Oran, Anukit (2560) ได้นำเทคโนโลยีไอโอทีมาใช้ในการเพาะเห็ดหลินจือ โดยวัดค่าความชื้นในโรงเพาะเห็ดหลินจือ และแสดงผลบนโทรศัพท์และคอมพิวเตอร์ผ่าน NETPIE การควบคุมการทำงานของสปริงเกอร์และปั๊มหมอกเป็นแบบอัตโนมัติและแจ้งเตือนสถานะการทำงานของสปริงเกอร์และปั๊มหมอกผ่าน แอปพลิเคชัน LINE

|  |
| --- |
| **รูปที่ 2.12** แสดงภาพโครงสร้างระบบของงานวิจัยที่เกี่ยวข้อง (3) |

**2.2.4 IOT BASED DESIGN IMPLEMENTATION OF MUSHROOM FARM MONITORING USING ARDUINO MICROCONTROLLERS & SENSORS [4]**

Parvati, Megha (2561) ในงานวิจัยฉบับนี้นำเสนอระบบการตรวจสอบและควบคุมสิ่งแวดล้อมเพื่อตรวจสอบและควบคุมสภาพแวดล้อมในฟาร์มเห็ด ช่วยให้ผู้ใช้สามารถตรวจสอบอุณหภูมิความชื้นความเข้มข้นของก๊าซคาร์บอนไดออกไซด์และความเข้มของแสงในฟาร์มเห็ดบนอุปกรณ์ Android โดยใช้แพลตฟอร์มออนไลน์ thing Speak ส่วนของตัวควบคุมสภาพแวดล้อมในฟาร์มเห็ดจะควบคุมสภาพแวดล้อมให้เป็นไปตามที่ผู้ใช้กำหนด ข้อมูลสถานะของสภาพแวดล้อมในฟาร์มเห็ดจะถูกส่งไปแสดงผลผ่าน ESP8266 WiFi modem

|  |
| --- |
| **รูปที่ 2.13** แสดงภาพโครงสร้างระบบของงานวิจัยที่เกี่ยวข้อง (4) |

**บทที่ 3**

**วิธีการดำเนินงาน**

ในบทนี้จะกล่าวถึงขั้นตอนและวิธีการดำเนินงานจัดทำโครงงานตลอดจนการออกแบบโครงสร้างและส่วนต่างๆ ของโครงงาน ซึ่งได้รับการศึกษาข้อมูลและทฤษฎีที่เกี่ยวข้องในบทที่ 2 จึงได้นำข้อมูลที่ได้มาศึกษาและวิเคราะห์ออกแบบเพื่อเลือกใช้วัสดุอุปกรณ์ที่จะนำมาทำโครงงาน ให้มีความถูกต้องเหมาะสมกับลักษณะงานและดำเนินการสร้างโครงงานให้มีประสิทธิภาพและเป็นไปตามวัตถุประสงค์และขอบเขตของโครงงาน

**3.1 แผนการดำเนินงาน**

ในการดำเนินงานจัดทำโครงงานโรงเพาะเห็ดอัจฉริยะ (Smart Mushroom Farm) ผู้จัดทำได้วางแผนการดำเนินงานในการจัดทำโครงงานดังต่อไปนี้

1. ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง
2. ศึกษา ออกแบบ โครงสร้างของระบบ
3. ศึกษาและออกแบบ เว็บแอปพลิเคชัน วินโดว์แอปพลิเคชัน และ ฐานข้อมูล
4. ศึกษาและออกแบบโรงเรือนสำหรับเพาะเห็ด
5. ทำการทดลองใช้งาน ไมโครคอนโทรลเลอร์ ร่วมกับอุปกรณ์ต่างๆ
6. ทำการสร้าง เว็บแอปพลิเคชัน และ วินโดว์แอปพลิเคชัน
7. ทำการสร้าง โรงเรือนสำหรับเพาะเห็ด และ ติดตั้งอุปกรณ์อุปกรณ์ต่างๆ
8. ทดสอบการทำงาน และ แก้ไขข้องบกพร่อง
9. จัดทำรูปเล่มรายงาน
10. นำเสนอผลงาน

**ตารางที่ 3.1** แสดงแผนการดำเนินงาน

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **กิจกรรม** | **ก.ค. 2563** | | | | **ส.ค. 2563** | | | | **ก.ย. 2563** | | | | **ต.ค. 2563** | | | |
| 1. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**3.2 โครงสร้างระบบ**

|  |
| --- |
| **รูปที่ 3.1** แสดงภาพโครงสร้างระบบ |

จากรูปที่ 3.1 แสดงภาพโครงสร้างระบบของโครงงานโรงเพาะเห็ดอัจฉริยะโดยใช้เทคโนโลยีการสื่อสารไร้สายระยะไกล โดยภาพรวมของระบบประกอบด้วยโรงเรือนเพาะเห็ดที่สามารถควบคุมสภาพแวดล้อมภายในโรงเรือนได้โดยใช้ไมโครคอนโทรลเลอร์ (ESP32LoRa) เป็นตัวควบคุมระบบและทำหน้าที่ในการสื่อสารกับพื้นที่ที่มีสัญญาณอินเทอร์เน็ตผ่านลอรา (LoRa)

พื้นที่ที่มีสัญญาณอินเทอร์เน็ตจะประกอบไปด้วยไมโครคอนโทรลเลอร์ (ESP32LoRa) ทำหน้าที่สื่อสารระหว่างพื้นที่ที่มีสัญญาณอินเทอร์เน็ตและโรงเรือนและเป็น Web Server ส่วน Raspberry Pi ทำหน้าที่เป็นฐานข้อมูลเพื่อให้สามารถเรียกดูค่าสภาพแวดล้อมย้อนหลังได้ และใช้โปรแกรม ngrok ในการทำ Port forwarding เพื่อให้สามารถใช้งานเว็บแอปพลิเคชัน (Web Application) ได้จากทุกที่ที่สามารถใช้งานอินเทอร์เน็ต

เว็บแอปพลิเคชัน (Web Application) สามารถมอนิเตอร์ค่าสภาพแวดล้อมภายในโรงเรือนและควบคุมระบบ โดยผู้จัดทำได้ใช้ภาษา HTML และ JavaScript ในการสร้างเว็บแอปพลิเคชัน (Web Application) จากนั้นนำเว็บแอปพลิเคชัน (Web Application) เก็บไว้ในหน่วยความจำ (Flash Memory) ของไมโครคอนโทรลเลอร์ (ESP32LoRa) และใช้ ESPAsyncWebServer library เพื่อให้ไมโครคอนโทรลเลอร์ (ESP32LoRa) สามารถทำหน้าที่เป็น Web Server ได้

วินโดว์แอปพลิเคชัน (Windows Application) สามารถมอนิเตอร์ค่าสภาพแวดล้อมภายในโรงเรือนและควบคุมระบบ โดยผู้จัดทำได้ใช้ภาษา Visual Basic ของ .NET Core ในการสร้างวินโดว์แอปพลิเคชัน (Windows Application) วินโดว์แอปพลิเคชัน (Windows Application) สามารถใช้งานได้กับไมโครคอนโทรลเลอร์ (ESP32LoRa) ทั้งสองตัวผ่านทาง Serial Port หากใช้งานวินโดว์แอปพลิเคชัน (Windows Application) กับไมโครคอนโทรลเลอร์ (ESP32LoRa) (Mushroom Node) เมื่อควบคุมการทำงาน เช่น กดปุ่มเปลี่ยนโหมดการทำงาน วินโดว์แอปพลิเคชัน (Windows Application) จะส่งข้อมูลผ่าน Serial Port มายังไมโครคอนโทรลเลอร์ (ESP32LoRa) จากนั้น ไมโครคอนโทรลเลอร์ (ESP32LoRa) จะนำข้อมูลที่ได้มาประมวลผลเพื่อควบคุมการทำงานของระบบ หากใช้งานกับวินโดว์แอปพลิเคชัน (Windows Application) กับ ไมโครคอนโทรลเลอร์ (ESP32LoRa) (STA Node) เมื่อควบคุมการทำงาน เช่น กดปุ่มเปลี่ยนโหมดการทำงาน วินโดว์แอปพลิเคชัน (Windows Application) จะส่งข้อมูลผ่าน Serial Port มายังไมโครคอนโทรลเลอร์ (ESP32LoRa) จากนั้น ไมโครคอนโทรลเลอร์ (ESP32LoRa) จะส่งข้อมูลที่ได้จาก Serial Port ผ่าน LoRa ไปยังยังไมโครคอนโทรลเลอร์ (ESP32LoRa) (Mushroom Node) จากนั้นจะนำข้อมูลที่ได้มาประมวลผล

ฐานข้อมูล (Database) ในโครงงานนี้ผู้จัดทำได้ใช้ Raspberry Pi เป็นฐานข้อมูล (Database) เพื่อให้สามารถดูค่าสภาพแวดล้อมย้อนหลังได้ ใช้ภาษา PHP ในการบันทึกข้อมูลลงในฐานข้อมูล (Database) การนำข้อมูลจากฐานข้อมูล (Database) มาแสดงในรูปแบบของตารางและกราฟที่สามารถเลือกดูข้อมูลได้ตามช่วงเวลาที่ต้องการ

การแจ้งเตือนไปยัง Line Application (Line Notify Service) ผู้จัดทำได้ใช้ Line Notify Service ในการแจ้งเตือนสถานะต่างๆ ไปยัง Line Application เช่น สาถานะ การบันทึกข้อมูล การเปลี่ยนสถานะการทำงานของอุปกรณ์ควบคุมสภาพแวดล้อม หรือการเปลี่ยนโหมดการทำงานของระบบ

**3.3 โรงเรือนเพาะเห็ด (Mushroom Farm)**

**3.3.1 การออกแบบโรงเรือนเพาะเห็ด**

|  |
| --- |
| **รูปที่ 3.2** แสดงภาพการออกแบบโรงเพาะเห็ด |

จากรูปที่ 3.2 แสดงภาพการออกแบบโรงเพาะเห็ด ในโครงงานนี้ผู้จัดทำได้ใช้โปรแกรม Google Sketchup ในการออกแบบโรงเรือนสำหรับ เพาะเห็ดขนาด (กว้าง x ยาว x สูง) 200x150x210 เซนติเมตร โดยได้ออกแบบให้เป็นโรงเรือนแบบปิด สามารถควบคุมสภาพแวดล้อมต่างๆภายในโรงเรือนได้เช่น อุณหภูมิ ความชื้น และแสง โครงสร้างของ โรงเรือนเพาะเห็ดเป็นเหล็กกล่องขนาด 1x1 นิ้ว เพื่อโรงเรือนมีความแข็งแรงและ สามารถใช้งานได้ในระยะยาว จากรูปจะเห็นว่ามีตัวเลขกำกับในแต่ละส่วนโดยมีรายละเอียดดังนี้ ส่วนที่ 1 เป็นส่วนของหัวพ่นหมอกใช้ในการควบคุมความชื้นในโรงเรือน ส่วนที่ 2 พัดลมระบายอากาศใช้ในการควบคุมอุณหภูมิและความชื้นภายในโรงเรือน ส่วนที่ 3 หลอดไฟใช้ในการควบคุมความสว่างในโรงเรือน ส่วนที่ 4 ชั้นสำหรับวางก้อนเห็ด ส่วนที่ 5 ประตูทางเข้าโรงเรือน ส่วนที่ 6 หมุดยกระดับโรงเรือนเพื่อไม่ให้โรงเรือนอยู่ติดกับพื้นจนเกินไปเพื่อป้องกันการเกิดสนิม

**3.3.2 การออกแบบฮาร์ดแวร์ระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ด**

ในการออกแบบฮาร์ดแวร์ระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ด ผู้จัดทำได้ใช้ AM2315 เป็นเซ็นเซอร์วัดอุณหภูมิและความชื้นภายในโรงเรือน ใช้ BH1750FVI เป็นเซ็นเซอร์วัดความสว่างของแสงภายในโรงเรือน ใช้พัดลมในการระบายอากาศ และปั๊มพ่นหมอกในการควบคุมอุณหภูมิและความชื้นภายในโรงเรือน ใช้หลอด LED ในการให้แสงสว่างภายในโรงเรือน ใช้ตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) สำหรับตรวจสอบสถานะการทำงานของพัดลมและปั๊มพ่นหมอก ใช้จอแอลซีดี (LCD) ในการแสดงผลค่าต่างๆ ภายในระบบ

|  |
| --- |
| **รูปที่ 3.3** แสดงภาพส่วนประกอบของฮาร์ดแวร์ระบบควบคุมสภาพแวดล้อมภายในโรงเรือน |

**3.3.2.1 การอ่านค่าอุณหภูมิและความชื้นจากเซ็นเซอร์ AM2315**

AM2315 คือเซ็นเซอร์วัดค่าอุณหภูมิและความชื้นคุณภาพสูงที่มีเอาต์พุตเป็นสัญญาณดิจิตอลโดยมีการสอบเทียบสัญญาณดิจิตอลเอาต์พุตแล้ว ใช้โมดูลตรวจจับอุณหภูมิละความชื้นคุณภาพสูงเพื่อให้แน่ใจว่ามีการวัดค่าที่แม่นยำและมีความทนทาน เซ็นเซอร์ประกอบด้วยตัววัดความชื้นประเภทความจุไฟฟ้า อุปกรณ์วัดอุณหภูมิที่มีความแม่นยำสูง และเชื่อมต่อด้วยไมโครคอนโทรลเลอร์ 8 บิตประสิทธิภาพสูง AM2315 สื่อสารด้วย I2C ผู้ใช้สามารถเชื่อต่อ I2C bus ได้โดยตรง

|  |
| --- |
| **รูปที่ 3.4** แสดงภาพ Schematic การเชื่อต่อระหว่าง ESP32LoRa และ AM2315 |

การเชื่อมต่อระหว่าง ESP32LoRa และ AM2315

|  |  |  |
| --- | --- | --- |
| 5V | ต่อกับ | VCC |
| GND | ต่อกับ | GND |
| SDA(GPIO21) | ต่อกับ | SDA |
| SCL(GPIO22) | ต่อกับ | SCL |

|  |
| --- |
| **รูปที่ 3.5** แสดงภาพโปรแกรมที่ใช้อ่านค่าจากเซ็นเซอร์ AM2315 |

จากรูปที่ 3.5 เป็นโปรแกรมภาษา C++ ที่ใช้ในการอ่านค่าจากเซ็นเซอร์ AM2315 โดยมีการเรียกใช้ไลบรารี่ (Library) Wire.h และ Adafruit\_AM2315.h ต่อมาส่วนที่สำคัญที่สุดคือส่วนของการปลุกการทำงานของเซ็นเซอร์ดังรูปที่แสดงด้านบน หากไม่ปลุกการทำงานของเซ็นเซอร์เมื่อไม่มีการใช้งานเซ็นเซอร์เป็นเวลานานจะส่งผลให้ไม่สามารถใช้งานได้

**3.3.2.2 การอ่านค่าความสว่างจากเซ็นเซอร์ BH1750FVI**

เซ็นเซอร์ BH1750FVI เป็นเซ็นเซอร์ที่มีค่าความละเอียดในการอ่านสูงถึง 16 บิตเชื่อมต่อแบบ I2C ทำให้ประหยัดขอ GPIO ของไมโครคอนโทรลเลอร์ที่เชื่อมต่อ ให้ค่าในการวัดเป็นหน่วย lux (ลักซ์) ซึ่งเป็นหน่วยการวัดแบบ SI (The International System of Unit) สำหรับความสว่าง (illuminance) = 1 lumen per square meter

|  |
| --- |
| **รูปที่ 3.6** แสดงภาพ Schematic การเชื่อต่อระหว่าง ESP32LoRa และ BH1750FVI |

การเชื่อมต่อระหว่าง ESP32LoRa และ BH1750FVI

|  |  |  |
| --- | --- | --- |
| 5V | ต่อกับ | VCC |
| GND | ต่อกับ | GND |
| SDA(GPIO21) | ต่อกับ | SDA |
| SCL(GPIO22) | ต่อกับ | SCL |

|  |
| --- |
| **รูปที่ 3.7** แสดงภาพโปรแกรมที่ใช้อ่านค่าจากเซ็นเซอร์ BH1750FVI |

จากรูปที่ 3.7 เป็นโปรแกรมภาษา C++ ที่ใช้ในการอ่านค่าจากเซ็นเซอร์ BH1750FVI โดยมีการเรียกใช้ไลบรารี่ (Library) Wire.h และ BH1750.h ก่อนอ่านค่าจากเซ็นเซอร์ เริ่มต้นการทำงานของเซ็นเซอร์โดยการเรียกใช้ฟังก์ชัน Wire.begin(); และ lightMeter.begin(); ภายในฟังก์ชัน setup() จากนั้นทำการอ่านค่าโดยใช้ฟังก์ชัน lightMeter.readLightLevel(); มาเก็บในตัวแปล lux ที่มีประเภทเป็น float และแสดงค่าที่ได้ทาง Serial Monitor ในหน่วย lux (ลักซ**ฺ**)

**3.3.2.3 การควบคุมพัดลมและปั๊มพ่นหมอก**

การควบคุมพัดลมและปั๊มพ่นหมอกผู้จัดทำได้ใช้ 4-Channel Relay Module ในการควบคุมการทำงานเพราะพัดลมและปั๊มพ่นหมอกใช้แรงดันไฟเลี้ยง 12V 4-Channel Relay Module เป็น Relay แบบ Active LOW จะทำงานเมื่อสั่งขา GPIO ของ ไมโครคอนโทรลเลอร์ ESP32LoRa เป็น LOW

|  |
| --- |
| **รูปที่ 3.8** แสดงภาพ Schematic ของการควบคุมพัดลมและปั๊มพ่นหมอก |

การเชื่อมต่อระหว่าง ESP32LoRa และ 4-Channel Relay Module

|  |  |  |
| --- | --- | --- |
| 5V | ต่อกับ | DC+ |
| GND | ต่อกับ | DC- |
| GPIO33 | ต่อกับ | IN4 |
| GPIO4 | ต่อกับ | IN3 |

|  |
| --- |
| **รูปที่ 3.9** แสดงภาพโปรแกรมส่วนที่ใช้ควบคุมการทำงานของ Relay |

จากรูปที่ 3.9 แสดงภาพโปรแกรมส่วนที่ใช้ควบคุมการทำงานของ Relay มีรายละเอียดคือ ตัวแปล pumpState และ ตัวแปล fanState เป็นตัวแปลประเภท Boolean ค่าสามารถมีค่าที่เป็นไปได้คือ 0 และ 1 หรือ true และ false โดยผู้จัดทำได้ออกแบบระบบให้พัดลมและปั๊มพ่นหมอกทำงานเมื่อตัวแปล pumpState และ ตัวแปล fanState มีค่าเท่ากับ true แต่ Relay ที่นำมาใช้จะทำงานเมื่อตัวแปล pumpState และ ตัวแปล fanState มีค่าเท่ากับ false ผู้จัดทำจึงได้เขียนโปรแกรมเพื่อให้ Relay และระบบสามารถทำงานร่วมกันได้โดยไม่ต้องแก้ไขทั้งระบบ

**3.3.2.4 การทำงานของปั๊มพ่นหมอกและพัดลมในโหมดการทำงานแบบอัตโนมัติ (AUTO)**

การทำงานของปั๊มพ่นหมอกและพัดลมในโหมดการทำงานแบบอัตโนมัติ (AUTO) จะพิจารณาตัวแปล 4 ตัวแปลคือ ค่าอุณหภูมิต่ำสุด ค่าอุณหภูมิสูงสุด ค่าความชื้นต่ำสุด ค่าความชื้นสูงสุด เพื่อรักษาระดับอุณหภูมิและความชื้นให้อยู่ในช่วงที่กำหนด สมมุติกำหนดให้ ค่าอุณหภูมิต่ำสุดเท่ากับ 25°C ค่าอุณหภูมิสูงสุดเท่ากับ 32°C ค่าความชื้นต่ำสุดเท่ากับ 80% ค่าความชื้นสูงสุดเท่ากับ 85%

**ตารางที่ 3.2** แสดงการทำงานของปั๊มพ่นหมอกและพัดลมในโหมดการทำงานแบบอัตโนมัติ (AUTO)

|  |  |  |  |
| --- | --- | --- | --- |
| **อุณหภูมิ(°C)** | **ความชื้น(%)** | **ปั๊มพ่นหมอก** | **พัดลม** |
| < 25 | < 80 | ON | OFF |
| < 25 | 80-85 | OFF | OFF |
| < 25 | > 85 | OFF | OFF |
| 25-32 | < 80 | ON | OFF |
| 25-32 | 80-85 | OFF | OFF |
| 25-32 | > 85 | OFF | OFF |
| > 32 | < 80 | ON | ON |
| > 32 | 80-85 | ON | ON |
| > 32 | > 85 | ON | ON |

|  |
| --- |
| **รูปที่ 3.10** แสดงภาพโปรแกรมที่ใช้ควบคุมการทำงานของปั๊มพ่นหมอกและพัดลม |

จากรุปที่ 3.10 แสดงภาพโปรแกรมภาษา C++ ที่ใช้ควบคุมการทำงานของปั๊มพ่นหมอกและพัดลมในโหมดการทำงานแบบอัตโนมัติ (AUTO) ตัวแปล ctrlMode คือตัวแปลโหมดการทำงานของระบบ ตัวแปล temp คือตัวแปลค่าอุณหภูมิ และตัวแปล humi คือตัวแปลค่าความชื้นที่อ่านได้จากเซ็นเซอร์ AM2315 ตัวแปล set\_temp\_min และ set\_temp\_max คือค่าอุณหภูมิต่ำสุดและสูงสุด ตัวแปล set\_humi\_min และ set\_humi\_max คือค่าความชื้นต่ำสุดและสูงสุด ตัวแปล pumpState และ fanState ค่าตัวแปลที่ใช้ในการควบคุมการทำงานของปั๊มพ่นหมอกและพัดลม

**3.3.2.5 การควบคุมความสว่างภายในโรงเรือนเพาะเห็ด**

ในการควบคุมความสว่างภายในโรงเรือนผู้จัดทำได้ใช้ LED ขนาด 5mm สีน้ำเงินจำนวน 112 หลอดเพื่อให้ความสว่าง จากนั้นใช้ L298N ในการควบคุมความสว่างโดยสัญญาณ PWM จากไมโครคอนโทรลเลอร์

|  |
| --- |
| **รูปที่ 3.11** แสดงภาพวงจร LED Driver |

จากรูปที่ 3.11 จะต้องคำนวณหาค่าของตัวต้านทานที่จะนำมาต่ออนุกรมกับหลอด LED เพื่อจำกัดกระแสตามความต้องการของหลอด LED โดยสามารถคำนวณได้จากสมการที่ 3.1 โดยกำหนดให้ ,

จาก

(3.1)

จะได้

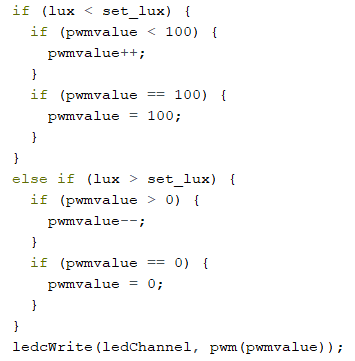
แทนค่า

ดังนั้นค่าของตัวต้านทานที่จะนำมาต่ออนุกรมกับหลอด LED มีค่าเท่ากับ 440 โอห์ม

|  |
| --- |
| **รูปที่ 3.12** แสดงภาพ Schematic ของการควบคุมความสว่างภายในโรงเรือนเพาะเห็ด |

การเชื่อมต่อระหว่าง ESP32LoRa และ L298N

|  |  |  |
| --- | --- | --- |
| GPIO25 | ต่อกับ | IN1 |
| GND | ต่อกับ | GND |
| GND | ต่อกับ | IN2 |



**รูปที่ 3.13** แสดงภาพโปรแกรมที่ใช้ในการควบคุมความสว่างของหลอดไฟ

จากรูปที่ 3.13 โปรแกรมจะทำการตรวจสอบว่าความสว่างต่ำกว่าหรือสูงกว่าค่าที่กำหนดหรือไม่ หากต่ำกว่าค่าที่กำหนดจะทำการเพิ่มค่า pwmvalue เพื่อเพิ่มความสว่างของหลอด LED หากสูงกว่าค่าที่กำหนดจะทำการลดค่า pwmvalue เพื่อลดความสว่างของหลอด LED

**3.3.2.6 การใช้จอแอลซี (LCD) แสดงผล**

ในโครงงานนี้ผู้จัดทำได้ใช้จอแอลซีดี (I2C LCD) ขนาด 16 x 2 ในการแสดงผลข้อมูลต่างๆในระบบ เช่น ค่าอุณหภูมิ ค่าความชื้น ค่าสถานะการทำงานของอุปกรณ์ โหมดการทำงานของระบบ เป็นต้น

|  |
| --- |
| **รูปที่ 3.14** แสดงภาพ Schematic การเชื่อต่อระหว่าง ESP32LoRa และ จอแอลซีดี (LCD) |

การเชื่อมต่อระหว่าง ESP32LoRa และ จอแอลซีดี (LCD)

|  |  |  |
| --- | --- | --- |
| 5V | ต่อกับ | VCC |
| GND | ต่อกับ | GND |
| SDA (GPIO21) | ต่อกับ | SDA |
| SCL (GPIO22) | ต่อกับ | SCL |

|  |
| --- |
| **รูปที่ 3.15** แสดงภาพโปรแกรมทดสอบการทำงานของจอแอลซีดี (LCD) |

จากรูปที่ 3.15 เป็นโปรแกรมภาษา C++ ที่ใช้ในการทดสอบการแสดงผลของจอแอลซีดีขนาด 16 x 2 การเชื่อมต่อแบบ I2C โดยกำหนด I2C Address ของจอแอลซีดี (LCD) เป็น Address 0x27 จากนั้นใช้ฟังก์ชัน lcd.begin() เพื่อเริ่มต้นการทำงานของ LCD ใช้ฟังก์ชัน lcd.serCursor() ในการกำหนดต่ำแหน่งเริ่มต้นของตัวอักษร และใช้ฟังก์ lcd.print() ในการแสดงตัวอักษร รูปที่แสดงด้านล่างแสดงผลลัพธ์ที่ได้จากโปรแกรมทดสอบการทำงานของจอแอลซีดี (LCD) แบบ I2C

**3.3.2.7 การอ่านค่าแรงดันเอาต์พุตจากวงจรตัวต้านทานตรวจสอบกระแส**

ในโครงงานนี้ผู้จัดทำได้นำตัวต้านทานตรวจสอบกระแสมาประยุกต์ใช้ เพื่อใช้ในการตรวจสอบสถานะการทำงานของอุปกรณ์ เพื่อให้แน่ใจอุปกรณ์ทำงานตามที่ควบคุมหรือไม่ จากนั้นผู้จัดทำได้ใช้ ADC (Analog to Digital Converter) ของ ESP32LoRa ในการอ่านค่าแรงดันเอาต์พุตจากวงจรตัวต้านทานตรวจสอบกระแสดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 3.16** แสดงภาพ Schematic การเชื่อต่อระหว่าง ESP32LoRa และวงจรตัวต้านทาน  ตรวจสอบกระแส |

การเชื่อมต่อระหว่าง ESP32LoRa และวงจรตัวต้านทานตรวจสอบกระแส

|  |  |  |
| --- | --- | --- |
| GPIO25 | ต่อกับ | OUT (LM358P) PUMP |
| GPIO35 | ต่อกับ | OUT (LM358P) FAN |

|  |
| --- |
| **รูปที่ 3.17** แสดงภาพโปรแกรมทดสอบการอ่านค่าแรงดันเอาต์พุตจากวงจรตัวต้านทานตรวจสอบ  กระแส |

จากรูปที่ 3.17 แสดงภาพโปรแกรมที่ใช้ในการอ่านค่าแรงดันเอาต์พุตจากวงจรตัวต้านทานตรวจสอบกระแส โดยใช้ GPIO25 และ GPIO35 ของ ESP32LoRa ในการอ่านค่าแรงดันที่ได้จากเอาต์พุตของวงจร ใช้ ADC ความละในการอ่าน 12-bit จากนั้นนำค่าที่ได้มาคำนวลเป็นแรงดันที่สามารถอ่านได้จากเอาต์พุตของวงจร

**3.4 การออกแบบวงจรตัวต้านทานตรวจสอบกระแส**

วงจรตัวต้านทานตรวจสอบกระแส มีวัตถุประสงค์เพื่อใช้การตรวจสอบสถานะการทำงาน ของ พัดลมระบายอากาศ ปั๊มพ่นหมอก ว่าทำงานจริงตามที่ควบคุมหรือไม่ โดยแสดงการออกแบบวงดังรูปด้านล่าง โดยมีหลักการทำงานคือนำตัวต้านทานที่มีค่าความต้านทาน 0.05 โอห์ม มาต่ออนุกรมกับ LOAD จากนั้นนำค่าแรงดันตกคร่อมตัวต้านทานมาขยายแรงดันโดยใช้วงจรขยายวงจรขยายแบบไม่กลับเฟส (Non-inverting Amplifier) เพื่อให้ไมโครคอนโทรลเลอร์สามารถอ่านค่าแรงดันได้

|  |
| --- |
| **รูปที่ 3.18** แสดงภาพวงจรตัวต้านทานตรวจสอบกระแส |

จากรูปที่ 3.18 แสดงภาพของแสดงภาพวงจรตัวต้านทานตรวจสอบกระแส โดยจะประกอบด้วย LOAD คือ พัดลมระบายอากาศ หรือ ปั๊มพ่นหมอก RS คือ ตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) และส่วนของวงจรขยายแบบไม่กลับเฟส (Non-inverting amplifier) ใช้เพื่อขยายแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแส

**3.4.1 การ****คำนวณแรงดันเอาต์พุตของวงจรเมื่อใช้งานกับพัดลมระบายอากาศ**

ในโครงงานนี้ผู้จัดทำใช้พัดลมระบายอากาศขนาด 12 นิ้ว 12V 6.6A ในการควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ด ดังนั้นสามารถคำนวณแรงดันเอาต์พุตของวงจรตัวต้านทานตรวจสอบกระแส ได้จากสมการที่ 3.2

(3.2)

จากสมการด้านบน คือแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) ดังนั้นจึงต้องทราบค่า ก่อน

|  |
| --- |
| **รูปที่ 3.19** แสดงภาพวงจรตัวต้านทานตรวจสอบกระแสเมื่อใช้งานกับพัดลมระบายอากาศ |

จารรูปที่ 3.19 เพื่อให้ไมโครคอนโทรลเลอร์สมารถแรงดันเอาต์พุตของวงจรได้ โดยสามารถหาอัตราการขยายของวงจรได้จากสมการที่ 3.3

จากสูตร

(3.3)

จะได้

อัตราการขยาย

ดังนั้น

อัตราการขยาย เท่า

สามารหาแรงดันตกคร่อมต้านทานตรวจสอบกระแส หรือ ได้จากกฎของโอห์มดังสมการที่ 3.4

(3.4)

*ดังนั้นหาแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแสจากรูปที่ 3.33 กำหนดให้ กระแส ความต้านทานของ*ต้านทานตรวจสอบกระแส

จากสูตร

*จะได้*

*แทนค่า*

*ดังนั้นแรงดันตกคร่องตัวต้านทานตรวจสอบกระแส มีค่าเท่ากับ ดังนั้นสามารถคำนวณแรงดันเอาต์พุตของวงจรเมื่อใช้งานกับพัดลมระบายอากาศได้จากสมการที่ 3.2*

*แทนค่า*

*ดังนั้นแรงดันเอาต์พุตของวงจรเมื่อใช้งานกับพัดลมระบายอากาศ* () *เท่ากับ สามารถคำนวณ**กำลัง*วัตต์ของตัวต้านทานตรวจสอบกระแส

*จากสูตร*

(3.5)

*จะได้*

*แทนค่า*

*ดังนั้นกำลัง*วัตต์ของตัวต้านทานตรวจสอบกระแส เท่ากับ 2.178

**3.4.2 การคำนวณแรงดันเอาต์พุตของวงจรเมื่อใช้งานกับปั๊มพ่นหมอก**

ในโครงงานนี้ผู้จัดทำใช้ปั๊มพ่นหมอกขนาดแรงดัน 4.8 bar / 70 PSI 12V 2A ในการควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ด ดังนั้นสามารถคำนวณแรงดันเอาต์พุตของวงจรตัวต้านทานตรวจสอบกระแส ได้จากสมการที่ 3.2

จากสมการด้านบน คือแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) ดังนั้นจึงต้องทราบค่า ก่อน

|  |
| --- |
| **รูปที่ 3.20** แสดงภาพวงจรตัวต้านทานตรวจสอบกระแสเมื่อใช้งานกับปั๊มพ่นหมอก |

จารรูปที่ 3.20 เพื่อให้ไมโครคอนโทรลเลอร์สมารถแรงดันเอาต์พุตของวงจรได้ โดยสามารถหาอัตราการขยายของวงจรได้จากสมการที่ 3.3

จากสูตร

จะได้

อัตราการขยาย

ดังนั้น

อัตราการขยาย เท่า

สามารหาแรงดันตกคร่อมต้านทานตรวจสอบกระแส หรือ ได้จากกฎของโอห์มดังสมการที่ 3.4

*ดังนั้นหาแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแสจากรูปที่ 3.4 กำหนดให้ กระแส ความต้านทานของ*ต้านทานตรวจสอบกระแส

จากสูตร

*จะได้*

*แทนค่า*

*ดังนั้นแรงดันตกคร่อง**ตัวต้านทานตรวจสอบกระแส มีค่าเท่ากับ ดังนั้นสามารถคำนวณแรงดันเอาต์พุตของวงจรเมื่อใช้งานกับปั๊มพ่นหมอกได้จากสมการที่ 3.2*

*แทนค่า*

*ดังนั้นแรงดันเอาต์พุตของวงจรเมื่อใช้งานกับปั๊มพ่นหมอก เท่ากับ* 3.1 *สามารถคำนวณ**กำลัง*วัตต์ของตัวต้านทานตรวจสอบกระแส ได้จากสมการที่ 3.5

*จากสูตร*

*จะได้*

*แทนค่า*

*ดังนั้นกำลัง*วัตต์ของตัวต้านทานตรวจสอบกระแส เท่ากับ 0.2

**3.5 การสื่อสารระหว่างโรงเรือนและพื้นที่ที่มีสัญญาณอินเทอร์เน็ตโดยใช้ลอรา (LoRa Communication)**

ในโครงงานนี้ผู้จัดทำได้นำเทคโนโลยีการสื่อสาร Lora มาประยุกต์ใช้เพื่อเป็นตัวกลางการสื่อสารระหว่างโรงเรือนเพาะเห็ด (Mushroom Node) และพื้นที่ที่มีสัญญาณอินเทอร์เน็ต (STA Node) เพื่อแก้ปัญหาที่ตั้งของโรงเรือนไม่มีสัญญาณอินเทอร์เน็ต โดยเป็นการสื่อสารแบบ Point to Point หรือ Node-to-Node Communication ดังรูปด้านล่าง

|  |
| --- |
| **รูปที่ 3.21** แสดงภาพการสื่อสารแบบ Node-to-Node |

จากรูปที่ 3.21 Node แต่ละ Node จะมี IP Address เป็นของตัวเองเพื่อใช้ในการระบุตัวตน เช่น Mushroom Node มี IP Address คือ X1 และ STA Node มี IP Address คือ X0

**3.5.1 การออกแบบการสื่อสาร Lora ที่ Mushroom Node**

ในการออกแบบการสื่อสารผ่าน Lora ที่ Mushroom Node ผู้จัดทำจะขออธิบายรายละเอียดต่างๆ โดยใช้ Flowchart ประกอบการอธิบาย

|  |
| --- |
| **รูปที่ 3.22** แสดงภาพ Flowchart การทำงานของ Mushroom Node (1) |

จากรูปที่ 3.22 แสดงการทำงานของ Mushroom Node เริ่มต้นระบบจะตรวจสอบว่าถ้ามีการเปลี่ยนแปลงของข้อมูลก็จะส่งข้อมูลการเปลี่ยนแปลง (C3) ไปยัง STA Node เช่น เมื่อผู้ใช้กดปุ่มเปลี่ยนโหมดการทำงานที่ Mushroom Node จากนั้น Mushroom Node ก็จะส่งข้อมูลการเปลี่ยนแปลงโหมดการทำงานไปยัง STA Node เพื่อให้ STA Node ทราบถึงการเปลี่ยนแปลงของข้อมูลโดยมีรูปแบบของข้อมูล (Data Format) ดังรูปที่ 3.23

|  |
| --- |
| **รูปที่ 3.23** แสดงภาพรูปแบบของข้อมูล (Data Format) |

จากรูปที่ 3.23 แสดงรูปแบบของข้อมูล (Data Format) ที่ส่งไปยัง STA Node เมื่อข้อมูลมีการเปลี่ยนแปลงโดยมีรายละเอียดดังต่อไปนี้

* X0 คือ IP Address ของ STA Node หรือ Node ปลายทางที่ต้องการส่งไปเพื่อให้ปลายทางหรือ STA Node ทราบว่าข้อมูลที่ส่งไปนั้นเป็นข้อมูลที่ต้องการส่งไปยังปลายทางหรือ STA Node
* X1 คือ IP Address ของ Mushroom Node หรือ Node ต้นทางโดยมีวัตถุประสงค์คือเพื่อให้ Node ปลายทางหรือ STA Node ทราบว่าข้อมูลที่ส่งมาเป็นของ Node ไหน ผู้จัดทำได้ออกแบบในลักษณะนี้เพื่อกรณีมี Node ต้นทางมากกว่า 1 Node
* T32.30 คือค่าอุณหภูมิที่วัดได้จากเซ็นเซอร์ AM2315 โดย T คือ Keyword ที่ใช้บอกว่าค่าตัวเลขที่ต่อท้ายมาด้วยจำนวน 4 หลักเป็นค่าอุณหภูมิ
* H71.40 คือค่าความชื้นที่วัดได้จากเซ็นเซอร์ AM2315 โดย H คือ Keyword ที่ใช้บอกว่าค่าตัวเลขที่ต่อท้ายมาด้วยจำนวน 4 หลักเป็นค่าความชื้น
* 30 คือค่าอุณหภูมิสูงสุดที่ใช้ในการควบคุมอุณหภูมิ
* 80 คือค่าอุณหภูมิต่ำสุดที่ใช้ในการควบคุมความชื้น
* M0 คือโหมด (Mode) การทำงานของระบบควบคุมอุณหภูมิและความชื้นภายในโรงเรือนโดย M คือ Keyword ที่ใช้บอกว่าค่าตัวเลขที่ต่อท้ายมาด้วยจำนวน 1 หลักเป็นค่าโหมดการทำงานของระบบ 0 คือเป็นโหมดการทำงานแบบอัตโนมัติ (AUTO Mode) และถ้าเป็น 1 คือเป็นโหมดการทำงานแบบควบคุมเอง (MANUAL Mode)
* P1 คือค่าสถานะการทำงานของปั๊ม (Pump) ที่อ่านได้จากวงจรตัวต้านทานตรวจสอบกระแสโดย P คือ Keyword ที่ใช้บอกว่าค่าตัวเลขที่ต่อท้ายมาด้วยจำนวน 1 หลักเป็นค่าสถานะการทำงานของปั๊ม (Pump) 1 คือค่าที่บอกว่าปั๊ม (Pump) กำลังทำงานถ้าเป็น 0 คือค่าที่บอกว่าปั๊ม (Pump) ไม่มีการทำงาน
* F1 คือค่าสถานะการทำงานของพัดลม (Fan) ที่อ่านได้จากวงจรตัวต้านทานตรวจสอบกระแสโดย F คือ Keyword ที่ใช้บอกว่าค่าตัวเลขที่ต่อท้ายมาด้วยจำนวน 1 หลักเป็นค่าสถานะการทำงานของพัดลม (Fan) 1 คือค่าที่บอกว่าพัดลม (Fan) กำลังทำงานถ้าเป็น 0 คือค่าที่บอกว่าพัดลม (Fan) ไม่มีการทำงาน
* 20 คือค่าอุณหภูมิต่ำสุดที่ใช้ในการควบคุมอุณหภูมิ
* 85 คือค่าอุณหภูมิสูงสุดที่ใช้ในการควบคุมความชื้น
* L0200.33 คือค่าความสว่างที่วัดได้จากเซ็นเซอร์ BH1750FVI โดย L คือ Keyword ที่ใช้บอกว่าค่าตัวเลขที่ต่อท้ายมาด้วยจำนวน 8 หลักเป็นค่าความสว่าง
* 200 คือค่าความสว่างที่กำหนดไว้

การส่งข้อมูลการเปลี่ยนแปลงไปยัง Node ปลายทางหรือ STA Node จะเรียกใช้ฟังก์ชัน sendUpdateData() ในการส่งข้อมูล ฟังก์ชัน sendUpdateData() เป็นฟังก์ชันที่เขียนด้วยภาษา C++ ดังรูปที่ 3.24

|  |  |
| --- | --- |
|  |  |
| **รูปที่ 3.24** แสดงภาพฟังก์ชัน sendUpdateData() | |

จากรูปที่ 3.24 แสดงฟังก์ sendUpdateData() ที่ใช้ส่งข้อมูลเมื่อมีการเปลี่ยนแปลงโดยรายละเอียดมีการทำงานดังต่อไปนี้ ก่อนส่งข้อมูลจะตรวจสอบค่าแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแสก่อนเพื่อให้ได้ค่าสถานะการทำงานของอุปกรณ์ที่ตรงกับค่าจริง จากนั้นจะส่งข้อมูลที่มีรูปแบบ (Data Format) เหมือนรูปที่ 3.23 ตัวแปลต่างๆที่ถูกใช้ในฟังก์ชันมีรายละเอียดดังต่อไปนี้

* Rs\_pump\_voltage แรงดันที่อ่านได้จากวงจรตัวต้านทานตรวจสอบกระแสที่ใช้กับปั๊ม (Pump) ผ่าน ADC Pin
* Rs\_fan\_voltage แรงดันที่อ่านได้จากวงจรตัวต้านทานตรวจสอบกระแสที่ใช้กับพัดลม (Fan) ผ่าน ADC Pin
* pump\_check ค่าที่บอกสถานะการทำงานของปั๊ม (Pump)
* fan\_check ค่าที่บอกสถานการณ์ทำงานของพัดลม (Fan)
* des IP Address ของ STA Node หรือ Node ปลายทาง
* ipAddr IP Address ของ Mushroom Node หรือ Node ต้นทาง
* temp ค่าอุณหภูมิที่วัดได้จากเซ็นเซอร์ AM2315
* tempStr ค่าที่ใช้ส่งแทนค่า temp ในกรณีที่อุณหภูมิต่ำกว่า 10.00 เพื่อป้องกันความยาวของข้อมูลคลาดเคลื่อน
* humi ค่าความชื้นที่วัดได้จากเซ็นเซอร์ AM2315
* humiStr ค่าที่ใช้ส่งแทนค่า humi ในกรณีที่ความชื้นต่ำกว่า 10.00 เพื่อป้องกันความยาวของข้อมูลคลาดเคลื่อน
* set\_temp\_max คือค่าอุณหภูมิสูงสุดที่ใช้ในการควบคุมอุณหภูมิ
* set\_humi\_min คือค่าอุณหภูมิต่ำสุดที่ใช้ในการควบคุมความชื้น
* ctrlMode คือโหมด (Mode) การทำงานของระบบควบคุมอุณหภูมิและความชื้นภายในโรงเรือน
* set\_temp\_min คือค่าอุณหภูมิต่ำสุดที่ใช้ในการควบคุมอุณหภูมิ
* set\_humi\_max คือค่าอุณหภูมิสูงสุดที่ใช้ในการควบคุมความชื้น
* lux คือค่าความสว่างที่วัดได้จากเซ็นเซอร์ BH1750FVI
* set\_lux คือค่าความสว่างที่กำหนด

จากรายละเอียดด้านบนเป็นการส่งข้อมูลการเปลี่ยนแปลงไปยัง STA Node ในกรณีที่ข้อมูลมีการเปลี่ยนแปลง แต่ถ้าหากข้อมูลไม่มีการเปลี่ยนแปลงก็จะทำการตรวจสอบเงื่อนไขข้อต่อไปคือการตรวจสอบว่าข้อมูลส่งมาจาก STA Node หรือไม่ ถ้าไม่มีข้อมูลที่ส่งมาก็จะกลับไปตรวจสอบการเปลี่ยนแปลงของข้อมูล

หากมีข้อมูลส่งมาจาก STA Node ก็จะทำการตรวจสอบข้อมูลโดยหา IP Address -ของตัวเอง (Mushroom Node IP Address) ว่าข้อมูลที่ส่งมาเป็นข้อมูลของตัวเองหรือไม่ ถ้าใช่ก็จะนำข้อมูลที่ได้มาประมวลผลต่อไปดังรูปที่ 3.25 เป็นโปรแกรมส่วนตรวจสอบข้อมูล

|  |
| --- |
| **รูปที่ 3.**25 แสดงภาพโปรแกรมส่วนที่ตรวจสอบข้อมูลที่ส่งมายัง Mushroom Node |
| **รูปที่ 3.26** แสดงภาพ Flowchart การทำงานของ Mushroom Node (2) |
| **รูปที่ 3.27** แสดงภาพ Flowchart การทำงานของ Mushroom Node (3) |

จากรูปที่ 3.26 และ รูปที่ 3.27 เมื่อได้รับข้อมูลที่ส่งมาจาก STA Node แล้วขั้นตอนต่อไปคือการนำข้อมูลมาประมวลผลโดยตัวอย่างของข้อมูลที่ถูกส่งมาจาก STA Node มีดังนี้ X1M, X1Y20, X1T30, X1H80, X1J85, X1P, X1F, หรือ X1R เป็นต้น มีรายละเอียดการทำงานในแต่ละข้อมูลดังนี้

|  |
| --- |
| **รูปที่ 3.28** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (1) |

จากรูปที่ 3.28 ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “M” หมายความว่าต้องการเปลี่ยนโหมดการทำงานของระบบควบคุมอุณหภูมิและความชื้นโดยการทำงานคือจะตรวจสอบว่าถ้าโหมดการทำงานเป็นโหมด AUTO (ctrlMode = false) จะเปลี่ยนเป็นโหมด MANUAL (ctrlMode = true) และถ้าโหมดการทำงานเป็นโหมด MANUAL (ctrlMode = true) ก็จะเปลี่ยนโหมดการทำงานเป็นโหมด AUTO (ctrlMode = false) กล่าวคือเป็นการทำงานแบบสลับ (toggle) โหมดการทำงาน เมื่อเปลี่ยนโหมดการทำงานสำเร็จแล้วก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.29** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (2) |

จากรูปที่ 3.29 ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “Y” หมายความว่าต้องการเปลี่ยนแปลงค่าอุณหภูมิต่ำสุดที่ใช้ในการควบคุมอุณหูมิ เช่น ปัจจุบันค่าอุณหภูมิต่ำสุดคือ 20°C ถ้าได้รับ Keyword “Y” ส่งมาจาก STA Node แสดงว่าต้องการเปลี่ยนค่าอุณหภูมิต่ำสุด โดยข้อมูลที่ส่งมาอาจจะเป็น Y21 ค่าตัวเลข 2 หลักต่อจาก Keyword เป็นค่าอุณหภูมิต่ำสุดที่ต้องการเปลี่ยน เมื่อได้รับค่าที่ต้องการเปลี่ยนแล้วก็จะเปลี่ยนค่าแล้วบันที่ค่าที่เปลี่ยนลงใน EEPROM ตำแหน่งที่ 1 จากนั้นก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.30** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (3) |

จากรูปที่ 3.30 ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “T” หมายความว่าต้องการเปลี่ยนแปลงค่าอุณหภูมิสูงสุดที่ใช้ในการควบคุมอุณหูมิ เช่น ปัจจุบันค่าอุณหภูมิสูงสุดคือ 32°C ถ้าได้รับ Keyword “T” ส่งมาจาก STA Node แสดงว่าต้องการเปลี่ยนค่าอุณหภูมิสูงสุด โดยข้อมูลที่ส่งมาอาจจะเป็น T30 ค่าตัวเลข 2 หลักต่อจาก Keyword เป็นค่าอุณหภูมิสูงสุดที่ต้องการเปลี่ยน เมื่อได้รับค่าที่ต้องการเปลี่ยนแล้วก็จะเปลี่ยนค่าแล้วบันที่ค่าที่เปลี่ยนลงใน EEPROM ตำแหน่งที่ 0 จากนั้นก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.31** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (4) |

จากรูปที่ 3.31 ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “H” หมายความว่าต้องการเปลี่ยนแปลงค่าความชื้นต่ำสุดที่ใช้ในการควบคุมความชื้น เช่น ปัจจุบันค่าความชื้นต่ำสุดคือ 80% ถ้าได้รับ Keyword “H” ส่งมาจาก STA Node แสดงว่าต้องการเปลี่ยนค่าความชื้นต่ำสุด โดยข้อมูลที่ส่งมาอาจจะเป็น H79 ค่าตัวเลข 2 หลักต่อจาก Keyword เป็นค่าความชื้นต่ำสุดที่ต้องการเปลี่ยน เมื่อได้รับค่าที่ต้องการเปลี่ยนแล้วก็จะเปลี่ยนค่าแล้วบันที่ค่าที่เปลี่ยนลงใน EEPROM ตำแหน่งที่ 10 จากนั้นก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.32** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (5) |

จากรูปที่ 3.32 ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “J” หมายความว่าต้องการเปลี่ยนแปลงค่าความชื้นสูงสุดที่ใช้ในการควบคุมความชื้น เช่น ปัจจุบันค่าความชื้นสูงสุดคือ 90% ถ้าได้รับ Keyword “H” ส่งมาจาก STA Node แสดงว่าต้องการเปลี่ยนค่าความชื้นสูงสุด โดยข้อมูลที่ส่งมาอาจจะเป็น H85 ค่าตัวเลข 2 หลักต่อจาก Keyword เป็นค่าความชื้นสูงสุดที่ต้องการเปลี่ยน เมื่อได้รับค่าที่ต้องการเปลี่ยนแล้วก็จะเปลี่ยนค่าแล้วบันที่ค่าที่เปลี่ยนลงใน EEPROM ตำแหน่งที่ 9 จากนั้นก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.33** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (6) |

จากรูปที่ 3.33 ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “P” หมายความว่าต้องการ เปิด/ปิด การทำงานของปั๊ม (Pump) โดยมีรายละเอียดการทำงานดังต่อไปนี้ ขั้นแรกจะตรวจสอบว่าถ้าค่าควบคุมการทำงานของปั๊ม (pumpState) มีค่าเท่ากับ false (pumpState == false) หรือสั่งให้ปั๊มหยุดทำงาน ก็จะเปลี่ยนค่าควบคุมการทำงานของปั๊ม (pumpState) ให้มีค่าเท่ากับ ture (pumpState = true) เพื่อให้ปั๊มกลับมาทำงาน เช่นเดียวกันถ้าปัจจุบันค่าควบคุมการทำงานของปั๊ม (pumpState) มีค่าเท่ากับ ture ก็จะเปลี่ยนค่าควบคุมการทำงานของปั๊ม (pumpState) ให้มีค่าเท่ากับ false กล่าวคือเป็นการทำงานแบบสลับ (toggle) ค่าควบคุมการทำงานของปั๊ม (pumpState) เมื่อเปลี่ยนสำเร็จแล้วก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.34** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (7) |

จากรูปที่ 3.34 ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “F” หมายความว่าต้องการ เปิด/ปิด การทำงานของพัดลม (Fan) โดยมีรายละเอียดการทำงานดังต่อไปนี้ ขั้นแรกจะตรวจสอบว่าถ้าค่าควบคุมการทำงานของพัดลม (fanState) มีค่าเท่ากับ false (fanState == false) หรือสั่งให้พัดลมหยุดทำงาน ก็จะเปลี่ยนค่าควบคุมการทำงานของพัดลม (fanState) ให้มีค่าเท่ากับ ture (fanState = true) เพื่อให้พัดลมกลับมาทำงาน เช่นเดียวกันถ้าปัจจุบันค่าควบคุมการทำงานของพัดลม (fanState) มีค่าเท่ากับ ture ก็จะเปลี่ยนค่าควบคุมการทำงานของพัดลม (fanState) ให้มีค่าเท่ากับ false กล่าวคือเป็นการทำงานแบบสลับ (toggle) ค่าควบคุมการทำงานของพัดลม (fanState) เมื่อเปลี่ยนสำเร็จแล้วก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

ในส่วนต่อมาถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “R” หมายความว่า STA Node ต้องการอัปเดตข้อมูลการเปลี่ยนแปลงโดยให้ Mushroom Node ส่งข้อมูลการเปลี่ยนแปลงไปให้

|  |
| --- |
| **รูปที่ 3.35** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (8) |

จากรูปที่ 3.35 ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “L” หมายความว่าต้องการเปลี่ยนแปลงค่าความสว่างโดยมีรายละเอียดการทำงานดังต่อไปนี้ เมื่อเจอ Keyword “L” จะนำข้อมูลต่อท้ายจำนวน 3 หลักมาเปลี่ยนค่าความสว่างที่กำหนดหรือกำหนดระดับความเข้มของแสง โดยจะทำการตรวจสอบค่าความสว่างที่กำหนดหากมีค่าต่ำกว่า 100 ก็จะนำเลข 0 มีแทรกด้านหน้าของข้อมูลเพื่อไม่ให้ความยาวของข้อมูลเปลี่ยนแปลง จากนั้นทำการบันทึกค่าความสว่างที่ถูกส่งมาจาก STA Node ลงใน EEPROM และส่งข้อมูลการเปลี่ยนแปลงกับไปยัง STA Node

**3.5.2 การออกแบบการสื่อสาร Lora ที่ STA Node**

ในการออกแบบการสื่อสารผ่าน Lora ที่ STA Node ผู้จัดทำจะขออธิบายรายละเอียดต่างๆ โดยใช้ Flowchart ประกอบการอธิบาย

|  |
| --- |
| **รูปที่ 3.36** แสดงภาพ Flowchart การทำงานของ STA Node (1) |

จากรูปที่ 3.36 แสดงการทำงานของ STA Node เริ่มต้นจะส่ง Keyword “R” ไปยัง Mushroom Node เพื่อขออัปเดตข้อมูลก่อนเริ่มระบบ โดยข้อมูลที่ส่งไปมีรูปแบบดังนี้ X1R เมื่อ X1 คือ IP Address ของ Mushroom Node เพื่อให้ทราบว่าต้องการส่งข้อมูลไปยัง Mushroom Node และ Keyword “R” คือ Keyword ที่ส่งไปเพื่อให้ Mushroom Node ทราบว่าต้องการอัปเดตข้อมูล

ขั้นตอนต่อไปจะตรวจสอบว่ามีข้อมูลส่งมาจาก Serial Port หรือ Windows Application หรือไม่ถ้ามีก็จะส่งผ่านข้อมูลไปยัง Mushroom Node เลยโดยไม่มีการประมวลผลใดๆ ตัวอย่างข้อมูลที่ส่งมาจาก Serial Port มีดังต่อไปนี้

* M คือ Keyword ที่ส่งเมื่อกดปุ่มเปลี่ยนโหมด
* P คือ Keyword ที่ส่งเมื่อกดปุ่มเปิดหรือปิดปั๊ม
* F คือ Keyword ที่ส่งเมื่อกดปุ่มเปิดหรือปิดพัดลม
* Y คือ Keyword ที่ส่งเมื่อเปลี่ยนค่าอุณหภูมิต่ำสุดตามด้วยตัวเลขจำนวน 2 หลัก เช่น Y25 เมื่อ 25 คือค่าที่ต้องการเปลี่ยน
* T คือ Keyword ที่ส่งเมื่อเปลี่ยนค่าอุณหภูมิสูงสุดตามด้วยตัวเลขจำนวน 2 หลัก เช่น T32 เมื่อ 32 คือค่าที่ต้องการเปลี่ยน
* H คือ Keyword ที่ส่งเมื่อเปลี่ยนค่าความชื้นต่ำสุดตามด้วยตัวเลขจำนวน 2 หลัก เช่น H80 เมื่อ 80 คือค่าที่ต้องการเปลี่ยน
* J คือ Keyword ที่ส่งเมื่อเปลี่ยนค่าความชื้นสูงสุดตามด้วยตัวเลขจำนวน 2 หลัก เช่น J90 เมื่อ 90 คือค่าที่ต้องการเปลี่ยน
* L คือ Keyword ที่ส่งเมื่อต้องการเปลี่ยนค่าความสว่างหรือระดับความสว่างตามด้วยตัวเลขจำนวน 2 หลักหรือ 3 หลัก เช่น L90, L200

จากรายละเอียดด้านบนเป็นข้อมูลจาก Serial Port เมื่อได้รับข้อมูลจาก Serial Port แล้วก็จะส่งข้อมูลต่อไปยัง Mushroom Node ทันทีดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 3.37** แสดงภาพโปรแกรมส่วนที่รับข้อมูลจาก Serial Port |

จากรูปที่ 3.37 จะรับข้อมูลจาก Serial Port แล้วส่งข้อมูลไปยัง Mushroom Node ด้วยฟังก์ชัน loraSend() โดยข้อมูลที่ส่งจะมีรูปแบบ (Data Format) ดังนี้ X1M เมื่อ X1 คือ IP Address ของ Mushroom Node และ M คือ Keyword ที่ส่งไปเพื่อควบคุมการทำงานของระบบดังรายละเอียดด้านบน จากนั้นจะล้างข้อมูลที่ได้รับจาก Serial Port

เมื่อระบบทำงานผ่านขั้นตอนต่างๆ ด้านบนมาแล้วก็ขั้นตอนต่อไปจะทำการตรวจสอบเงื่อนไขข้อต่อไปคือการตรวจสอบว่าข้อมูลส่งมาจาก Mushroom Node หรือไม่ ดังรูปที่ 3.49 ถ้าไม่มีข้อมูลที่ส่งมาก็จะส่งข้อมูลเพื่อขออัปเดตข้อมูล (C3) ทุกๆ 60 วินาที ถ้ามีข้อมูลส่งมาจาก Mushroom Node ก็จะทำการหา IP Address -ของตัวเอง (STA Node IP Address) ว่าข้อมูลที่ส่งมาเป็นข้อมูลของตัวเองหรือไม่ ถ้าใช่ก็จะนำข้อมูลที่ได้มาประมวลผลต่อไป

|  |
| --- |
| **รูปที่ 3.38** แสดงภาพโปรแกรมส่วนที่ตรวจสอบข้อมูลที่ส่งมายัง STA Node |

|  |
| --- |
| **รูปที่ 3.39** แสดงภาพ Flowchart การทำงานของ STA Node (2) |

เมื่อได้รับข้อมูลที่ส่งมาจาก Mushroom Node แล้วก็จะนำข้อมูลที่ได้มาเก็บไว้ในตัวแปลต่างๆ โดยใช้ฟังก์ชัน substring() เพื่อนำข้อมูลที่ได้ไปใช้งานบน Web Application และ Windows Application ตัวอย่างรูปแบบของข้อมูล (Data Format) ที่ส่งมาจาก Mushroom Node ดังรูปที่ 3.40

|  |
| --- |
| **รูปที่ 3.40** แสดงภาพรูปแบบของข้อมูล (Data Format) ที่ส่งมาจาก Mushroom Node |

จากรูปที่ 3.40 แสดงรูปแบบของข้อมูลที่ส่งมาจาก Mushroom Node เมื่อได้รับข้อมูลก็จะนำข้อมูลที่ได้มา substring() ดังรูปที่ 3.41

|  |  |
| --- | --- |
|  |  |
| **รูปที่ 3.41** แสดงภาพการใช้ฟังก์ชัน substring() กับข้อมูลที่ได้รับจาก Mushroom Node | |

จากรูปที่ 3.41 แสดงการใช้ฟังก์ชัน substring() เพื่อนำข้อมูลที่ได้รับมาเก็บไว้ในตัวแปลที่ถูกต้องโดยมีรายละเอียดดังต่อไปนี้

* ตัวแปล node เป็นตัวแปลที่ใช้เก็บข้อมูลว่าข้อมูลที่ได้รับถูกส่งมาจาก Node ไหน โดยการ substring(0, 2) จากตำแหน่งที่ 0 ถึงตำแหน่งที่ 2
* ตัวแปล temp เป็นค่าอุณหภูมิ โดยการ substring(3, 8) จากตำแหน่งที่ 3 ถึงตำแหน่งที่ 8 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Float
* ตัวแปล humi เป็นค่าความชื้น โดยการ substring(9, 14) จากตำแหน่งที่ 9 ถึงตำแหน่งที่ 14 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Float
* ตัวแปล set\_temp\_max เป็นค่าอุณหภูมิสูงสุด โดยการ substring(14, 16) จากตำแหน่งที่ 14 ถึงตำแหน่งที่ 16 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Int
* ตัวแปล set\_humi\_min เป็นค่าความชื้นต่ำสุด โดยการ substring(16, 18) จากตำแหน่งที่ 16 ถึงตำแหน่งที่ 18 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Int
* ตัวแปล ctrlMode เป็นโหมดการทำงานของระบบควบคุมอุณหภูมิและความชื้นภายในโรงเรือน โดยการ substring(19, 20) จากตำแหน่งที่ 19 ถึงตำแหน่งที่ 20 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Int
* ตัวแปล pumpState เป็นค่าสถานะการทำงานของปั๊มที่ได้จากวงจรตัวต้านทานตรวจสอบกระแสโดยการ substring(21, 22) จากตำแหน่งที่ 21 ถึงตำแหน่งที่ 22 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Int
* ตัวแปล fanState เป็นค่าสถานะการทำงานของพัดลมที่ได้จากวงจรตัวต้านทานตรวจสอบกระแสโดยการ substring(23, 24) จากตำแหน่งที่ 23 ถึงตำแหน่งที่ 24 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Int
* ตัวแปล set\_temp\_min เป็นค่าอุณหภูมิต่ำสุด โดยการ substring(24, 26) จากตำแหน่งที่ 24 ถึงตำแหน่งที่ 26 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Int
* ตัวแปล set\_humi\_max เป็นค่าความชื้นต่ำสุด โดยการ substring(26, 28) จากตำแหน่งที่ 26 ถึงตำแหน่งที่ 28 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Int
* ตัวแปล lux เป็นค่าความสว่างที่ได้จากเซ็นเซอร์ BH1750FVI โดยการ substring(29, 36) จากตำแหน่งที่ 29 ถึงตำแหน่งที่ 36 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Float
* ตัวแปล set\_lux เป็นค่าความสว่างที่กำหนดไว้ โดยการ substring(36, 39) จากตำแหน่งที่ 36 ถึงตำแหน่งที่ 39 แล้วเปลี่ยนประเภทของข้อมูลเป็นประเภท Int

เมื่อจัดการกลับข้อมูลที่ส่งมาเรียบร้อยแล้วขั้นตอนต่อไปคือการส่ง Keyword เพื่อขออัปเดตข้อมูลทุกๆ 60 วินาที ดังรูปที่ 3.42

|  |
| --- |
| **รูปที่ 3.42** แสดงภาพโปรแกรมที่ใช้ในการส่ง Keyword ทุก 60 วินาที |

จากรูปที่ 3.42 มีหลักการทำงานคือ STA Node จะส่ง Keyword “R” ทุกๆ 60 วินาทีโดยใช้ฟังก์ชัน millis() ในการนับเวลา ซึ่งตัวแปล node1 มีค่าเท่ากับ X1 คือ IP Address ของ Mushroom Node รูปแบบข้อมูล (Data Format) คือ X1R

|  |
| --- |
| **รูปที่ 3.43** แสดงภาพ Flowchart การทำงานของ STA Node (3) |

จากรุปที่ 3.43 เมื่อ Keyword เพื่อขออัปเดตข้อมูลทุกๆ 60 วินาทีแล้ว ขั้นตอนต่อไปคือการตรวจสอบว่ามีการควบคุมการทำงานของระบบจากเว็บแอปพลิเคชัน (Web Application) หรือไม่ โดยมีรายละเอียดการทำงานดังต่อไปนี้

* ถ้ามีการกดปุ่มเปลี่ยนโหมดการทำงานจากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “M” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการเปลี่ยนโหมดการทำงาน โดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1M
* ถ้ามีการกดปุ่ม เปิด/ปิด ปั๊มจากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “M” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการ เปิด/ปิด ปั๊ม โดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1P
* ถ้ามีการกดปุ่ม เปิด/ปิด พัดลมจากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “F” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการ เปิด/ปิด พัดลม โดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1F
* ถ้ามีการกดปุ่มขออัปเดตข้อมูลจากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “R” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการอัปเดตข้อมูลโดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1R

|  |
| --- |
| **รูปที่ 3.44** แสดงภาพ Flowchart การทำงานของ STA Node (4) |

จากรูปที่ 3.44 หากมีการกรอกข้อมูลจากเว็บแอปพลิเคชัน (Web Application) STA Node จะส่งข้อมูลต่อไปยัง Mushroom Node เพื่อเปลี่ยนแปลงข้อมูลตามที่ผู้ใช้กรอกในแต่ละ input โดยมีรายละเอียดดังนี้

* ถ้ามีการกรอกข้อมูลที่ input1 จากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “T” + “ข้อมูลที่กรอก” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการเปลี่ยนค่าอุณหภูมิสูงสุด โดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1T32
* ถ้ามีการกรอกข้อมูลที่ input2 จากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “H” + “ข้อมูลที่กรอก” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการเปลี่ยนค่าความชื้นต่ำสุด โดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1H80
* ถ้ามีการกรอกข้อมูลที่ input3 จากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “Y” + “ข้อมูลที่กรอก” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการเปลี่ยนค่าอุณหภูมิต่ำสุด โดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1Y25
* ถ้ามีการกรอกข้อมูลที่ input4 จากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “J” + “ข้อมูลที่กรอก” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการเปลี่ยนค่าความชื้นสูงสุด โดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1J90
* ถ้ามีการกรอกข้อมูลที่ input5 จากเว็บแอปพลิเคชัน (Web Application) จะส่ง Keyword “L” + “ข้อมูลที่กรอก” ไปยัง Mushroom Node เพื่อแจ้งว่าผู้ใช้ต้องการเปลี่ยนค่าความความสว่าง โดยมีรูปแบบของข้อมูลที่ส่งดังนี้ X1L90, X1L200

**3.6 เว็บแอปพลิเคชัน (Web Application)**

ในโครงงานนี้ผู้จัดทำได้ใช้ภาษา HTML CSS Javascript และ PHP ในการสร้าง Web Application และใช้แอปพลิเคชัน ngrok ที่ติดไว้บน Raspberry Pi ทำ Port forwarding ของ Web Application เพื่อให้สามารถใช้งานเว็บแอปพลิเคชันได้จากทุกที่ที่สามารถใช้งานอินเทอร์เน็ตได้ Web Application ส่วนที่แสดงค่าสถานะต่างๆและควบคุมการทำงานของระบบจะถูกเก็บอยู่ที่ ESP32LoRa และ Web Application ส่วนที่เกี่ยวกับฐานข้อมูล เช่น บันทึกข้อมูลลงในฐานข้อมูล นำข้อมูลในฐานข้อมูลมาแสดงบน Web Application จะถูกเก็บอยู่ที่ Raspberry Pi 3 Model B เพราะใช้ Raspberry Pi เป็นฐานข้อมูลดังรูปที่ 3.45

|  |
| --- |
| **รูปที่ 3.45** แสดงภาพโครงสร้างการทำงานของ Web Application |

**3.6.1 การออกแบบหน้าจอเว็บแอปพลิเคชัน**

ในโครงงานนี้ผู้จัดทำออกแบบหน้าเว็บแอปพลิเคชัน ให้มีการปรับเปลี่ยนขนาดของหน้าจอเว็บแอปพลิเคชันให้เหมาะสบกับการแสดงผลบนหน้าจอขนาดต่างๆ และความละเอียดของหน้าจอในอุปกรณ์ที่แตกต่างกัน เช่น คอมพิวเตอร์ โน้ตบุ๊ค โทรศัพท์มือถือ แท็บเล็ต เป็นต้น

**3.6.1.1 หน้าเข้าสู่ระบบ**

|  |
| --- |
| **รูปที่ 3.46** แสดงภาพหน้าเข้าสู่ระบบก่อนใช้งานเว็บแอปพลิเคชัน (Web Application) |

จากรูปที่ 3.46 จะต้องทำการกรอก Username และ Password ให้ถูกต้องก่อนเข้าใช้งาน เว็บแอปพลิเคชัน (Web Application) หากกรอก Username หรือ Password ไม่ถูกต้องระบบจะเด้งกลับมาให้กรอกอีกครั้ง

**3.6.1.2 หน้า Dashboard**

|  |
| --- |
| **รูปที่ 3.47** แสดงภาพหน้า Dashboard ของเว็บแอปพลิเคชัน (Web Application) (1) |

จากรูปที่ 3.47 เมื่อทำการกรอก Username และ Password ถูกต้องแล้วจะเข้าสู่หน้า Dashboard ของเว็บแอปพลิเคชัน (Web Application) ซึ่งแสดงโหมดการทำงานของระบบในโหมดการทำงานแบบอัตโนมัติ (AUTO)

|  |
| --- |
| **รูปที่ 3.48** แสดงภาพหน้า Dashboard ของเว็บแอปพลิเคชัน (Web Application) (2) |

จากรูปที่ 3.48 แสดงหน้า Dashboard ในโหมดการทำงานแบบควบคุมเอง (MANUAL) จะเห็นว่าเว็บแอปพลิเคชัน (Web Application) จะแสดงปุ่มควบคุมพัดลม (Control -FAN) และควบคุมปั๊ม (Control PUMP)ซึ่งปุ่มควบคุมอุปกรณ์จะเปลี่ยนสีตามสถานะการทำงานของอุปกรณ์นั้นๆ

**3.6.1.3 แถบเมนูนำทาง**

|  |
| --- |
| **รูปที่ 3.49** แสดงภาพแถบเมนูนำทาง (1) |

เมื่อทำการคลิกที่แถบเมนูนำทางเว็บแอปพลิเคชันจะแสดงเมนูนำทางขึ้นมาดังรูปที่ 3.50

|  |
| --- |
| **รูปที่ 3.50** แสดงภาพแถบเมนูนำทาง (2) |

จากรูปที่ 3.50 แสดงภาพแถบเมนูนำทางของเว็บแอปพลิเคชัน โดยแถบเมนูนำทางจะประกอบไปด้วยส่วนต่างๆ เช่น ส่วนของการแสดงข้อมูลที่ถูกเก็บไว้ใน Database ส่วนการตั้งค่าระบบ และการออกจากระบบเว็บแอปพลิเคชัน

**3.6.1.4 หน้า Data Logger**

|  |
| --- |
| **รูปที่ 3.51** แสดงภาพหน้า Data Logger ของเว็บแอปพลิเคชัน (Web Application) |

จากรูปที่ 3.51 แสดงภาพหน้า Data Logger ของเว็บแอปพลิเคชัน ในหน้านี้สามารถดูข้อมูลต่างๆ แบบตาราง ภายในโรงเรือนย้อนหลัง ซึ่งค่าเริ่มต้นจะแสดงข้อมูลในวันที่ปัจจุบันและสามารถเลือกดูข้อมูลตามวันที่ที่ระบุหรือสามารถเลือกดูข้อมูลระหว่างวันที่ถึงวันที่ได้

**3.6.1.5 หน้า Chart**

|  |
| --- |
| **รูปที่ 3.52** แสดงภาพหน้า Chart ของเว็บแอปพลิเคชัน (Web Application) |

จากรูปที่ 3.52 แสดงภาพหน้า Chart ของเว็บแอปพลิเคชัน ในหน้านี้สามารถดูข้อมูลต่างๆ แบบกราฟ ภายในโรงเรือนย้อนหลัง ซึ่งค่าเริ่มต้นจะแสดงข้อมูลในวันที่ปัจจุบันและสามารถเลือกดูข้อมูลตามวันที่ที่ระบุได้

**3.6.1.6 หน้า Setting**

|  |
| --- |
| .  **รูปที่ 3.53** แสดงภาพหน้า Setting ของเว็บแอปพลิเคชัน (Web Application) |

จากรูปที่ 3.53 แสดงหน้า Setting ของเว็บแอปพลิเคชันในหน้านี้จะสามารถกำหนดค่าสภาพแวดล้อมภายในโรงเรือน และเปิดปิดงการแจ้งเตือนของ Line Notify ได้

**3.6.2 การทำงานของเว็บแอปพลิเคชัน**

การสื่อสารระหว่าง Web Server และ Client เป็นการสื่อสารผ่านทาง Hypertext Transfer Protocol (HTTP) โดยมีรายละเอียดการทำงานดังรูปที่ 3.54

|  |
| --- |
| **รูปที่ 3.54** แสดงภาพ Flowchart การสื่อสารระหว่าง Web Server และ Client |

จากรูปที่ 3.54 แสดงภาพ Flowchart การสื่อสารระหว่าง Web Server และ Client โดยมีรายละเอียดการทำงานดังต่อไปนี้ เริ่มต้นการทำงาน Client จะส่ง HTTP GET Method เพื่อขออัปเดตข้อมูลทุกๆ 5 วินาที เพื่อให้หน้าเว็บมีการอัปเดตข้อมูลโดยใช้ฟังก์ชัน setInterval() ในภาษา JavaScript จากนั้นใช้ Ajax (Asynchronous JavaScript And XML) ในการส่ง HTTP GET Method ไปยัง Web Server เมื่อ Web Server ตอบกลับมาจะนำข้อมูลที่ได้ไปแสดงบนหน้าเว็บ ดังภาพด้านล่าง

|  |
| --- |
| **รูปที่ 3.55** แสดงภาพตัวอย่างโปรแกรมส่วนที่ใช้ในการขออัปเดตข้อมูลทุกๆ 5 วินาที |

จากรูปที่ 3.55 เป็นตัวอย่างการขออัปเดตค่าอุณหภูมิจาก Web Server ทุกๆ 5 วินาทีเพื่อให้ค่าอุณหภูมิที่แสดงบนหน้าเว็บมีการอัปเดต นอกจากนี้ยังมีค่าต่างๆที่ขออัปเดตทุกๆ 5 วินาทีเหมือนค่าอุณหภูมิ เช่น อุณหภูมิต่ำสุด อุณหภูมิสูงสุด ความชื้น ความชื้นต่ำสุด ความชื้นสูงสุด ความสว่าง ความสว่างที่กำหนด สถานะการทำงานของปั๊มและพัดลม โหมดการทำงาน ดังรูปที่ 3.56

|  |
| --- |
| **รูปที่ 3.56** แสดงภาพข้อมูลต่างๆบนหน้าเว็บที่อัปเดตทุกๆ 5 วินาที |

จากรูปด้านบนแสดงค่าตัวเลขและสถานะต่างๆ Client ขออัปเดตไปยัง Web Server ทุกๆ 5 วินาที ขั้นตอนต่อไปจะทำตรวจสอบว่ามีการความคุมการทำงานของระบบหรือไม่โดยมีรายละเอียดดังต่อไปนี้

|  |
| --- |
| **รูปที่ 3.57** แสดงภาพฟังก์ชันที่ถูกเรียกใช้เมื่อกดปุ่มเปลี่ยนโหมดการทำงาน |

จากรูปที่ 3.57 ถ้า Client กดปุ่มเปลี่ยนโหมดการทำงาน (change mode) จะเรียกใช้ฟังก์ชัน getmode() ที่เขียนด้วยภาษา JavaScript โดยฟังก์ชันมีการทำงานดังนี้ เมื่อมีการเรียกใช้จะส่ง HTTP GET Method “/M” ไปยัง Web Server โดยใช้ Ajax จากนั้น Server จะส่ง Keyword “M” ผ่าน Lora ไปยัง Mushroom Node เพื่อขอเปลี่ยนโหมดการทำงาน เมื่อ Web Server ตอบกลับมายัง Client ว่าส่งข้อมูลสำเร็จจะแจ้งเตือนดังรูปที่ 3.58

|  |
| --- |
| **รูปที่ 3.58** แสดงภาพการแจ้งเตือนเมื่อกดปุ่มเปลี่ยนโหมดการทำงาน |

|  |
| --- |
| **รูปที่ 3.59** แสดงภาพฟังก์ชันที่ถูกเรียกใช้เมื่อกดปุ่มขออัปเดตข้อมูล |

จากรูปที่ 3.59 ถ้า Client กดปุ่มขออัปเดตข้อมูล (request update) จะเรียกใช้ฟังก์ชัน requestupdate() ที่เขียนด้วยภาษา JavaScript โดยฟังก์ชันมีการทำงานดังนี้ เมื่อมีการเรียกใช้จะส่ง HTTP GET Method “/requestupdate” ไปยัง Web Server โดยใช้ Ajax จากนั้น Web Server จะส่ง Keyword “R” ผ่าน Lora ไปยัง Mushroom Node เพื่อขออัปเดตข้อมูล

|  |
| --- |
| **รูปที่ 3.60** แสดงภาพฟังก์ชันที่ถูกเรียกใช้เมื่อกดปุ่ม เปิด/ปิด ปั๊ม |

จากรูปที่ 3.60 ถ้า Client กดปุ่มควบคุมการทำงานของปั๊ม (control PUMP) จะเรียกใช้ฟังก์ชัน toggleCheckbox\_pump(element) ที่เขียนด้วยภาษา JavaScript โดยฟังก์ชันมีการทำงานดังนี้ เมื่อมีการเรียกใช้จะส่ง HTTP GET Method “/P” ไปยัง Web Server โดยใช้ Ajax จากนั้น Web Server จะส่ง Keyword “P” ผ่าน Lora ไปยัง Mushroom Node เพื่อควบคุมการทำงานของปั๊ม

|  |
| --- |
| **รูปที่ 3.61** แสดงภาพฟังก์ที่ถูกเรียกใช้เมื่อกดปุ่ม เปิด/ปิด พัดลม |

จากรูปที่ 3.61 ถ้า Client กดปุ่มควบคุมการทำงานของพัดลม (control FAN) จะเรียกใช้ฟังก์ชัน toggleCheckbox\_fan(element) ที่เขียนด้วยภาษา JavaScript โดยฟังก์ชันมีการทำงานดังนี้ เมื่อมีการเรียกใช้จะส่ง HTTP GET Method “/F” ไปยัง Web Server โดยใช้ Ajax จากนั้น Web Server จะส่ง Keyword “F” ผ่าน Lora ไปยัง Mushroom Node เพื่อควบคุมการทำงานของพัดลม

**3.7 วินโดว์แอปพลิเคชัน (Windows Application)**

ในโครงงานนี้ผู้จัดทำได้มีวัตถุประสงค์ในการนำวินโดว์แอปพลิเคชัน (Windows Application) มาประยุกต์ใช้ภายในโครงงาน ใช้ภาษา Visual Basic ของ .NET Core ในการสร้างวินโดว์แอปพลิเคชัน (Windows Application)

วินโดว์แอปพลิเคชัน (Windows Application) สามารถใช้งานได้กับ ESP32LoRa ได้ทั้งสองตัว ดังรูปที่ 3.62 โดยสื่อสารกันผ่าน Serial Port

|  |
| --- |
| **รูปที่ 3.62** แสดงภาพโครงสร้างการทำงานของ Windows Application |

หากใช้งานกับวินโดว์แอปพลิเคชัน (Windows Application) กับ ESP32LoRa (Mushroom Node) เมื่อควบคุมการทำงาน เช่น กดปุ่มเปลี่ยนโหมดการทำงาน วินโดว์แอปพลิเคชัน (Windows Application) จะส่งข้อมูลผ่าน Serial Port มายัง ESP32LoRa จากนั้น ESP32LoRa จะนำข้อมูลที่ได้มาประมวลผลตามข้อมูลที่ได้

หากใช้งานกับวินโดว์แอปพลิเคชัน (Windows Application) กับ ESP32LoRa (STA Node) เมื่อควบคุมการทำงาน เช่น กดปุ่มเปลี่ยนโหมดการทำงาน วินโดว์แอปพลิเคชัน (Windows Application) จะส่งข้อมูลผ่าน Serial Port มายัง ESP32LoRa จากนั้น ESP32LoRa จะส่งข้อมูลที่ได้จาก Serial Port ผ่าน LoRa ไปยังยัง ESP32LoRa (Mushroom Node) จากนั้นจะนำข้อมูลที่ได้มาประมวลผล

**3.7.1 การออกแบบหน้าจอวินโดว์แอปพลิเคชัน**

**3.7.1.1 หน้าเข้าสู่ระบบ**

|  |
| --- |
| **รูปที่ 3.63** แสดงภาพหน้าเข้าสู่ระบบของวินโดว์แอปพลิเคชัน (Windows Application) |

**3.7.1.2 หน้า Dashboard**

|  |
| --- |
| **รูปที่ 3.64** แสดงภาพหน้า Dashboard ของวินโดว์แอปพลิเคชัน (1) |

จากรูปที่ 3.64 แสดงหน้า Dashboardของวินโดว์แอปพลิเคชันเมื่อยังไม่ได้เชื่อมต่อกับไมโครคอนโทรลเลอร์

|  |
| --- |
| **รูปที่ 3.65** แสดงภาพหน้า Dashboard ของวินโดว์แอปพลิเคชัน (2) |

จากรูปที่ 3.65 แสดงภาพวินโดว์แอปพลิเคชันเมื่อเชื่อมต่อกับ Serial Port แล้ว ESP32LoRa จะส่งข้อมูลผ่าน Serial Port มายังวินโดว์แอปพลิเคชันโดยส่งฟังก์ชัน Serial.print() ตัวอย่างข้อมูลคือ T30.50H78.703075M0P1F02685L200.33200 จากนั้นวินโดว์แอปพลิเคชันจะนำข้อมูลที่ได้ประมวลผล

|  |
| --- |
| **รูปที่ 3.66** แสดงภาพหน้า Dashboard ของวินโดว์แอปพลิเคชัน (3) |

จากรูปที่ 3.66 เมื่อโหมดการทำงานเป็นโหมดควบคุมเอง (MANUAL) วินโดว์แอปพลิเคชัน (Windows Application) จะแสดงปุ่มควบคุมพัดลมและปั๊มขึ้นมา

**3.7.2 การทำงานของวินโดว์แอปพลิเคชัน**

การทำงานของวินโดว์แอปพลิเคชัน (Windows Application) มีรายละเอียดการทำงานดดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 3.67** แสดงภาพ Flowchart การทำงานของวินโดว์แอปพลิเคชัน (1) |
| **รูปที่ 3.68** แสดงภาพ Flowchart การทำงานของวินโดว์แอปพลิเคชัน (2) |

จากรูปที่ 3.67 และ 3.68 แสดงภาพ Flowchart การทำงานของวินโดว์แอปพลิเคชันโดยมีรายละเอียดดังต่อไปนี้ เมื่อแอปพลิเคชันเริ่มต้นทำงานจะแสดงหน้าเข้าสู่ระบบ ให้กรอก Username และ Password ก่อนเพื่อป้องกันความปลอดภัย เมื่อทำการ Login สำเร็จจะแสดงหน้า Dashboard ของวินโดว์แอปพลิเคชัน หากผู้ใช้มีการกำหนดค่าสภาพแวดล้อมหรือควบคุมระบบวินโดว์แอปพลิเคชัน จะส่งข้อมูลผ่านทาง Serial Port ไปยังไมโครคอนโทรลเลอร์เพื่อให้ไมโครคอนโทรลเลอร์ควบการทำงานของระบบต่อไป

เมื่อกดปุ่มเปลี่ยนโหมดการทำงาน (Change mode) วินโดว์แอปพลิเคชันจะส่ง Keyword “M” ผ่าน Serial Port ไปยัง ESP32LoRa โดยใช้ฟังก์ชัน SerialPort1.Write("M") เพื่อให้ ESP32LoRa นำ Keyword ที่ได้ไปประมวลผลต่อไปดังรูปที่ 3.69

|  |
| --- |
| **รูปที่ 3.69** แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกดปุ่มเปลี่ยนโหมด |

เมื่อกดปุ่มเปิดปั๊ม (Turn on Pump) หรือ ปุ่มปิดปั๊ม (Turn off Pump) วินโดว์แอปพลิเคชันจะส่ง Keyword “P” ผ่าน Serial Port ไปยัง ESP32LoRa โดยใช้ฟังก์ชัน SerialPort1.Write("P") เพื่อให้ ESP32LoRa นำ Keyword ที่ได้ไปประมวลผลต่อไปดังรูปที่ 3.70

|  |
| --- |
| **รูปที่ 3.70** แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกดปุ่ม เปิด/ปิด ปั๊ม |

เมื่อกดปุ่มเปิดพัดลม (Turn on Fan) หรือ ปุ่มปิดพัดลม (Turn off Fan) วินโดว์แอปพลิเคชันจะส่ง Keyword “F” ผ่าน Serial Port ไปยัง ESP32LoRa โดยใช้ฟังก์ชัน SerialPort1.Write("F") เพื่อให้ ESP32LoRa นำ Keyword ที่ได้ไปประมวลผลต่อไปดังรูปที่ 3.71

|  |
| --- |
| **รูปที่ 3.71** แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกดปุ่ม เปิด/ปิด พัดลม |

เมื่อกำหนดค่าอุณหภูมิต่ำสุดวินโดว์แอปพลิเคชันจะส่ง Keyword “Y” + ข้อมูลที่กรอก ผ่าน Serial Port ไปยัง ESP32LoRa โดยใช้ฟังก์ชัน SerialPort1.Write("Y" + TextBox1.Text) เพื่อให้ ESP32LoRa นำ Keyword ที่ได้ไปประมวลผลต่อไปดังรูปที่ 3.72

|  |
| --- |
| **รูปที่ 3.72** แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าอุณหภูมิ  ต่ำสุด |

เมื่อกำหนดค่าอุณหภูมิสูงสุดวินโดว์แอปพลิเคชันจะส่ง Keyword “T” + ข้อมูลที่กรอก ผ่าน Serial Port ไปยัง ESP32LoRa โดยใช้ฟังก์ชัน SerialPort1.Write("T" + TextBox4.Text) เพื่อให้ ESP32LoRa นำ Keyword ที่ได้ไปประมวลผลต่อไปดังรูปที่ 3.73

|  |
| --- |
| **รูปที่ 3.73** แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าอุณหภูมิ  สูงสุด |

เมื่อกำหนดค่าความชื้นต่ำสุดวินโดว์แอปพลิเคชันจะส่ง Keyword “H” + ข้อมูลที่กรอก ผ่าน Serial Port ไปยัง ESP32LoRa โดยใช้ฟังก์ชัน SerialPort1.Write("H" + TextBox2.Text) เพื่อให้ ESP32LoRa นำ Keyword ที่ได้ไปประมวลผลต่อไปดังรูปที่ 3.74

|  |
| --- |
| **รูปที่ 3.74** แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าความชื้น  ต่ำสุด |

เมื่อกำหนดค่าความชื้นสูงสุดวินโดว์แอปพลิเคชันจะส่ง Keyword “J” + ข้อมูลที่กรอก ผ่าน Serial Port ไปยัง ESP32LoRa โดยใช้ฟังก์ชัน SerialPort1.Write("J" + TextBox3.Text) เพื่อให้ ESP32LoRa นำ Keyword ที่ได้ไปประมวลผลต่อไปดังรูปที่ 3.75

|  |
| --- |
| **รูปที่ 3.75** แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าความชื้น  สูงสุด |

เมื่อกำหนดค่าความสว่างวินโดว์แอปพลิเคชันจะส่ง Keyword “L” + ข้อมูลที่กรอก ผ่าน Serial Port ไปยัง ESP32LoRa โดยใช้ฟังก์ชัน SerialPort1.Write("L" + TextBoxSetLux.Text) เพื่อให้ ESP32LoRa นำ Keyword ที่ได้ไปประมวลผลต่อไปดังรูปที่ 3.76

|  |
| --- |
| **รูปที่ 3.76** แสดงภาพการส่งข้อมูลผ่าน Serial Port ไปยัง ESP32LoRa เมื่อกำหนดค่าความสว่าง |

**3.8 ฐานข้อมูล (Database)**

**3.8.1 การออกแบบฐานข้อมูล**

ในการออกแบบฐานข้อมูลในโครงงานนี้จะใช้แค่ 1 ตาราง (entity) เท่านั้นเพื่อใช้เก็บข้อมูลต่างๆที่เกี่ยวข้องเช่น ค่าอุณหภูมิ (temp) ค่าความชื้น (humi) ค่าความสว่าง (lux) ค่าอุณหภูมิต่ำสุด (temp\_limit\_min) ค่าอุณหภูมิสูงสุด (temp\_limit\_max) ค่าความชื้นต่ำสุด (humi\_limit\_min) ค่าความชื้นสูงสุด (humi\_imit\_max) ค่าความสว่างที่กำหนด (set\_lux) โหมดการทำงาน (ctrl\_mode) สถานะการทำงานของปั๊ม (pump\_state) สถานการณ์ทำงานของพัดลม (fan\_state) และเวลาที่บันทึกข้อมูล (save\_time) ดังรูปที่ 3.77 โดยตารางมีชื่อว่า SensorData

|  |
| --- |
| **รูปที่ 3.77** แสดงภาพโครงสร้างของฐานข้อมูล |

จากรูปที่ 3.77 โครงสร้างของฐานข้อมูลโดยมี Primary key (PK) คือ id สามารถบันทึกข้อมูลเป็นตัวเลขจำนวนเต็มแบบไม่มีเครื่องหมายไม่เกิน 6 หลัก เป็นลำดับของข้อมูลแบบเพิ่มขึ้นอัตโนมัติ (auto increment) มี save\_time attribute ใช้ในการเก็นค่าวันเวลาที่บันทึกข้อมูล และ #temp #humi #lux #temp\_limit\_min #temp\_limit\_max #humi\_limit\_min #humi\_limit\_max #set\_lux #ctrl\_mode #pump\_state #fan\_state attribute เป็น attribute ที่ใช้ในการเก็บข้อมูลต่างๆภายในระบบ

**ตารางที่ 3.3** แสดง Data Dictionary SensorData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Example | Key |
| id | ลำดับของข้อมูล | int(6) | 1 | PK |
| temp | อุณหภูมิ | varchar(5) | 32.10 |  |
| humi | ความชื้น | varchar(5) | 80.70 |  |
| lux | ความสว่าง | varchar(6) | 213.22 |  |
| temp\_limit\_min | อุณหภูมิต่ำสุด | varchar(2) | 20 |  |
| temp\_limit\_max | อุณหภูมิสูงสุด | varchar(2) | 29 |  |
| humi\_limit\_min | ความชื้นต่ำสุด | varchar(2) | 80 |  |
| humi\_limit\_max | ความชื้นสูงสุด | varchar(2) | 85 |  |
| Set\_lux | ความสว่างที่กำหนด | varchar(5) | 200 |  |
| ctrl\_mode | โหมดการทำงาน | varchar(6) | AUTO |  |
| pump\_state | สถานะการทำงานของปั๊ม | varchar(3) | ON |  |
| fan\_state | สถานะการทำงานของพัดลม | varchar(3) | ON |  |
| save\_time | วันเวลาที่บันทึกข้อมูล | timestamp | 2020-09-08 18:00:00 |  |

|  |
| --- |
| **รูปที่ 3.78** แสดงภาพตัวอย่างข้อมูลเก็บอยู่ภายในฐานข้อมูล |

**3.8.2 การบันทึกข้อมูลลงในฐานข้อมูล**

ในการจัดทำโครงงานนี้ผู้จัดทำได้ใช้ Raspberry pi 3 b เป็นฐานข้อมูลเพื่อเก็บค่าต่างๆและใช้ไมโครคอนโทรลเลอร์ ESP32 LoRa เป็น Web Server โดยจะบันทึกข้อมูลลงในฐานข้อมูลทุกๆชั่วโมง เช่น 00.00, 01.00, 02.00, 12.00, 13.00 เป็นต้น ดัง Flowchart ด้านล่าง

|  |
| --- |
| **รูปที่ 3.79** แสดงภาพ Flowchart ขั้นตอนการบันทึกข้อมูลลงในฐานข้อมูล |
| **รูปที่ 3.80** แสดงภาพการบันทึกข้อมูลลงในฐานข้อมูลทุกชั่วโมง |

จากรูปที่ 3.80 จะเป็นการดึงค่าเวลาจริงมาจากเซิร์ฟเวอร์ NTP (Network Time Protocol) มาใช้ในการตรวจสอบเพื่อบันทึกข้อมูล โดยจะตรวจสอบเวลาในหลักนาทีและวินาที เช่น ถ้าเวลาในหน่วยนาทีมีค่าเท่ากับ 00 และเวลาในหน่วยวินาทีมีค่าเท่ากับ 00 ก็จะทำการบันทึกข้อมูลโดยเรียกใช้ฟังก์ชัน insertDB() ก็จะสามารถบันทึกข้อมูลทุกๆชั่วโมงได้แล้ว

การบันทึกข้อมูลจากเว็บเซิร์ฟเวอร์ (Web Server) ไปยังฐานข้อมูลที่อยู่บน Raspberry pi 3 b ทุกๆชั่วโมงโดยใช้ฟังก์ชัน insertDB() ที่เขียนด้วยภาษา C++ ดังภาพที่ 3.81



**รูปที่ 3.81** แสดงภาพฟังก์ชันที่ใช้บันทึกข้อมูลลงในฐานข้อมูล

จากรูปที่ 3.81 เป็นฟังก์ชันที่ทำหน้าบันทึกข้อมูลลงในฐานข้อมูลที่อยู่บน Raspberry pi 3 b โดยตัวแปลที่เป็นประเภท String ที่มีชื่อว่า serverName เป็นตัวแปลที่เก็บที่อยู่ของไฟล์ PHP ที่มีชื่อว่า post-esp-data.php ทำหน้าที่บันทึกข้อมูลลงในฐานข้อมูลที่อยู่บน Raspberry pi โดยที่อยู่นี้ได้จากการทำ port forwarding โดยใช้โปรแกรม Ngrok ตัวอย่างที่อยู่ http://bc14e80e8393.ap.ngrok.io/post-esp-data.php

ขั้นตอนต่อมาเป็นการจัดรูปแบบเพื่อเตรียมการก่อนการบันทึกข้อมูลผ่าน HTTP POST REQUEST ตัวแปลที่เป็นประเภท String ที่มีชื่อว่า httpRequestData เป็นตัวแปลที่เก็บข้อมูลต่างๆ ที่จะบันทึกลงในฐานข้อมูล ตัวแปลที่มีชื่อว่า httpResponseCode จะทำหน้าที่ POST ข้อมูลไปยังไฟล์ post-esp-data.php แล้วเก็บค่า http response เช่นเซิร์ฟเวอร์ตอบกลับว่า 200 OK แสดงว่าบันทึกข้อมูลสำเร็จ จากนั้นทำการแจ้งเตือนสถานะการบันทึกข้อมูลทาง LINE แอปพลิเคชัน โดยเรียกใช้ฟังก์ชัน NotifyLine()

|  |
| --- |
| **รูปที่ 3.82** แสดงภาพไฟล์ post-esp-data.php บางส่วน |

จากรูปที่ 3.82 แสดง code บางส่วนของไฟล์ post-esp-data.php โดยมีตัวแปลที่มีชื่อว่า $api\_key\_value ที่มีค่าเท่ากับ “tPmAT5Ab3j7F9” เป็นค่าที่ใช้ตรวจสอบว่าข้อมูลที่ส่งมา มาจากแหล่งที่มีที่ถูกต้อง เช่น ถ้า ESP32 LoRa ต้องการที่จะบันทึกข้อมูลลงในฐานข้อมูลจะต้องส่งค่า $api\_key\_value มาด้วยผ่าน HTTP POST

ต่อมาไฟล์ post-esp-data.php จะตรวจสอบว่าถ้ามีการ POST มาจะตรวจสอบ $api\_key\_value ถ้าตรงกันจะทำการเชื่อมต่อฐานข้อมูลแล้วบันทึกข้อมูล ถ้าไม่ตรงกันจะแสดงข้อความ "Wrong API Key provided." เพื่อเป็นการป้องกันฐานข้อมูลเบื่องต้น

**3.9 การออกแบบ Use Case Diagram**

|  |
| --- |
| **รูปที่ 3.83** แสดงภาพ Use Case Diagram |

รายละเอียด Use Case Diagram จากตารางที่ 3.4 ถึงตารางที่ 3.10 ด้านล่างต่อไปนี้แสดงถึงรายละเอียดการใช้งานของแต่ละ Use Case ตามรูปแบบของ UML Use Case Description

**ตารางที่ 3.4** แสดงLogin Use Case

|  |  |
| --- | --- |
| Use Case Name | Login |
| Actors | User |
| Use Case Purpose | เพื่อเข้าสู่ระบบ |
| Pre-conditions | - |
| Post-conditions | เข้าสู่ระบบเพื่อดูค่าสถานะต่างๆ ควบคุมการทำงานของอุปกรณ์ กำหนดค่าสูงสุดต่ำสุดของอุณหภูมิและความชื้น กำหนดค่าความสว่าง |
| Main Course | 1.กรอก Username และ Password  2.กดปุ่ม Login  3.เข้าสู่ระบบเสร็จสิ้น |
| Exceptions | Username หรือ Password ไม่ถูกต้อง |

**ตารางที่ 3.5** แสดงLogout Use Case

|  |  |
| --- | --- |
| Use Case Name | Logout |
| Actors | User |
| Use Case Purpose | เพื่อใช้ในการออกจากระบบ |
| Pre-conditions | เข้าสู่ระบบมาแล้วก่อนหน้า |
| Post-conditions | ออกจากระบบสำเร็จ |
| Main Course | 1.กดปุ่ม Logout  2.ออกจากระบบสำเร็จ |
| Exceptions | - |

**ตารางที่ 3.6** แสดง View Dashboard Use Case

|  |  |
| --- | --- |
| Use Case Name | View Dashboard |
| Actors | User |
| Use Case Purpose | เพื่อใช้ในการแสดงสถานะต่างๆ และจัดการระบบ |
| Pre-conditions | เข้าสู่ระบบมาแล้วก่อนหน้า |
| Post-conditions | - |
| Main Course | แสดงค่าสถานะต่างๆ เช่น อุณหภูมิ ความชื้น แสงสว่าง สถานะการทำงานของอุปกรณ์ |
| Exceptions | แสดงค่าผิดพลาด |

**ตารางที่ 3.7** แสดง Control Use Case

|  |  |
| --- | --- |
| Use Case Name | Control |
| Actors | User |
| Use Case Purpose | เพื่อใช้ในการควบคุมการทำงานของระบบ เช่น เปลี่ยนโหมดการทำงาน ความคุมการเปิดปิดของอุปกรณ์ |
| Pre-conditions | ใช้ร่วมกับ View Dashboard |
| Post-conditions | - |
| Main Course | 1.กดปุ่ม change mode  2.ถ้าโหมดการทำงานแบบอัตโนมัติระบบจะควบคุมการทำงานของอุปกรณ์โดยอัตโนมัติ เมื่อค่าอุณหภูมิและความชื้นต่ำกว่าหรือสูงกว่าที่กำหนด ถ้าโหมดการทำงานแบบแมนนวลผู้ใช้จะสามารถสั่งเปิดปิดอุปกรณ์ได้ตามต้องการ |
| Exceptions | เกิดความผิดพลาดภายในระบบ |

**ตารางที่ 3.8** แสดง Setting Use Case

|  |  |
| --- | --- |
| Use Case Name | Setting |
| Actors | User |
| Use Case Purpose | เพื่อใช้ในการกำหนดค่าสูงสุดต่ำสุดของอุณหภูมิและความชื้น กำหนดค่าความสว่าง ตั้งค่าการแจ้งเตือนผ่าน LINE Application |
| Pre-conditions | ใช้ร่วมกับ View Dashboard |
| Post-conditions | - |
| Main Course | 1.กำหนดค่าสูงสุดต่ำสุด  2.กำหนดค่าสูงสุดต่ำสุดสำเร็จ |
| Exceptions | เกิดความผิดพลาดภายในระบบ |

**ตารางที่ 3.9** แสดง View Data Use Case

|  |  |
| --- | --- |
| Use Case Name | View Data |
| Actors | User |
| Use Case Purpose | เพื่อใช้ในการแสดงผลข้อมูลที่อยู่ใน database ในรูปแบบต่างๆ |
| Pre-conditions | ใช้ร่วมกับ View Dashboard |
| Post-conditions | - |
| Main Course | 1.กดปุ่ม datalogger หรือ chart  2.แสดงข้อมูลในรูปแบบต่างๆ |
| Exceptions | ไม่สามารถติดต่อฐานข้อมูลได้ |

**ตารางที่ 3.10** แสดง View Status Use Case

|  |  |
| --- | --- |
| Use Case Name | View Status |
| Actors | User, Guest |
| Use Case Purpose | เพื่อใช้ในการแสดงสถานะบนจอ LCD Display ที่โรงเพาะเห็ด |
| Pre-conditions | - |
| Post-conditions | - |
| Main Course | 1.ดูค่าสถานะบนจอ LCD Display |
| Exceptions | เกิดความผิดพลาดภายในระบบ |

**บทที่ 4**

**ผลการดำเนินงาน**

หลังจากที่ได้ดำเนินงานออกแบบโรงเรือนเพาะเห็ด เขียนโปรแกรมควบคุมการทำงานระบบภายในโรงเรือน ออกแบบและสร้างเว็บแอปพลิเคชันและวินโดว์แอปพลิเคชันเพื่อมอนิเตอร์และความคุมการทำงานของโรงเรือนเพาะเห็ด ในบทนี้เป็นการทดลองส่วนต่างๆที่ได้ดำเนินงานในบทที่ 3 และนำระบบทั้งหมดมาใช้งานร่วมกัน โดยได้ทำการทดลองดังนี้

* 1. การทดลองการทำงานของระบบ

4.1.1 การทดลองการทำงานของเว็บแอปพลิเคชัน

4.1.2 การทดลองการทำงานของวินโดว์แอปพลิเคชัน

4.1.3 การทดลองการทำงานของระบบควบคุมสภาพแวดล้อม

* 1. การทดลองเพาะเห็ด
  2. **การทดลองการทำงานของระบบ**

การทดลองการทำงานของระบบเป็นการทดลองระบบทุกส่วน เช่น เว็บแอปพลิเคชัน วินโดว์แอปพลิเคชัน และระบบควบคุมสภาพแวดล้อมที่โรงเรือน เพื่อทดลองการทำงานของแต่ละส่วนว่าสามารถใช้งานและทำงานร่วมกันได้หรือไม่

**4.1.1 การทดลองการทำงานของเว็บแอปพลิเคชัน**

ทดลองใช้งานเว็บแอปพลิเคชัน (Web Application) ในการควบคุมระบบภายในโรงเรือนเพาะเห็ด เช่น การเปลี่ยนโหมดการทำงาน เปิด/ปิดปั๊มพ่นหมอกและพัดลมระบายอากาศ กำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด กำหนดค่าความชื้นต่ำสุดและความชื้นสูงสุด และการกำหมดค่าความสว่าง เป็นต้น

ทดลองใช้งานเว็บแอปพลิเคชัน (Web Application) ในการควบคุมระบบภายในโรงเรือนเพาะเห็ด ขั้นตอนแรกเมื่อเข้าใช้งานเว็บแอปพลิเคชัน (Web Application) จะต้องทำการเข้าสู่ระบบก่อนดังรูปที่ 4.1

|  |
| --- |
| **รูปที่ 4.1** แสดงภาพหน้าเข้าสู่ระบบก่อนใช้งานเว็บแอปพลิเคชัน (Web Application) |

จากรูปที่ 4.1 จะต้องทำการกรอก Username และ Password ให้ถูกต้องก่อนเข้าใช้งาน เว็บแอปพลิเคชัน (Web Application) หากกรอก Username หรือ Password ไม่ถูกต้องระบบจะเด้งกลับมาให้กรอกอีกครั้ง

|  |
| --- |
| **รูปที่ 4.2** แสดงภาพหน้าแรกของเว็บแอปพลิเคชัน (Web Application) |

จากรูปที่ 4.2 เมื่อทำการกรอก Username และ Password ถูกต้องแล้วจะเข้าสู่หน้าแรกของเว็บแอปพลิเคชัน (Web Application)

**4.1.1.1 ทดลองเปลี่ยนโหมดการทำงาน**

จากรูปที่ 4.2 จะเห็นว่าโหมดการทำงานเป็นโหมด AUTO จากนั้นเมื่อคลิกที่ปุ่มเปลี่ยนโหมดการทำงาน (change mode) ดังรูปที่ 4.34

|  |
| --- |
| **รูปที่ 4.3** แสดงภาพการทดลองเปลี่ยนแปลงโหมดการทำงาน (1) |

จากรูปที่ 4.3 เมื่อทำการกดที่ปุ่มเปลี่ยนโหมดการทำงาน (Change mode) เว็บแอปพลิเคชัน (Web Application) จะเปลี่ยนโหมดการทำงานจากโหมด AUTO เป็นโหมด MANUAL และแสดงปุ่มควบคุม เปิด/ปิด พัดลมและปั๊มดังรูปที่ 4.4

|  |
| --- |
| **รูปที่ 4.4** แสดงภาพการทดลองเปลี่ยนแปลงโหมดการทำงาน (2) |

**4.1.1.2 ทดลองควบคุมพัดลมและปั๊ม**

จากรูปที่ 4.4 จะเห็นว่าเว็บแอปพลิเคชัน (Web Application) จะแสดงปุ่มควบคุมพัดลม (Control -FAN) และควบคุมปั๊ม (Control PUMP) ปุ่มควบคุมเป็นสีแดงหมายความว่าอุปกรณ์กำลังทำงาน จากนั้นทำงานสั่งปิดพัดลมโดยกดที่ปุ่มควบคุมพัดลม (Control -FAN) ดังรูปที่ 4.5

|  |
| --- |
| **รูปที่ 4.5** แสดงภาพการทดลองควบคุมพัดลมและปั๊ม (1) |
| **รูปที่ 4.6** แสดงภาพการทดลองควบคุมพัดลมและปั๊ม (2) |

จากรูปที่ 4.6 เว็บแอปพลิเคชัน (Web Application) เมื่อสั่งปิดพัดลมแล้วจะทำการแสดงสถานะการทำงานของพัดลมเป็น OFF และแสดงปุ่มควบคุมพัดลม (Control -FAN) เป็นสีเทา ขั้นตอนต่อมาทำการทดลองควบคุมปั๊ม จากรูปที่ 4.ป สาถานะการทำงานของปั๊มเป็น OFF ดังนั้นทำการทดลองสั่งเปิดปั๊มโดยกดที่ปุ่มควบคุมปั๊ม (Control -PUMP) ดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 4.7** แสดงภาพการทดลองควบคุมพัดลมและปั๊ม (3) |

|  |
| --- |
| **รูปที่ 4.8** แสดงภาพการทดลองควบคุมพัดลมและปั๊ม (4) |

จากรูปที่ 4.8 เว็บแอปพลิเคชัน (Web Application) เมื่อสั่งเปิดปั๊มแล้วจะทำการแสดงสถานะการทำงานของปั๊มเป็น ON และแสดงปุ่มควบคุมปั๊ม (Control -PUMP) เป็นสีแดง

**4.1.1.3 ทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด**

ทำการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุดโดยการคลิกที่แถบเมนูนำทางที่อยู่ด้านบนซ้ายของหน้าจอดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 4.9** แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (1) |

เมื่อทำการคลิกที่แถบเมนูนำทางเว็บแอปพลิเคชันจะแสดงเมนูนำทางขึ้นมาดังรูปที่ 4.10

|  |
| --- |
| **รูปที่ 4.10** แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (2) |

จากรูปที่ 4.10 เมื่อเว็บแอปพลิเคชัน (Web Application) แสดงเมนูนำทางขึ้นมาแล้วทำการคลิกที่ Setting ดังรูปที่แสดงด้านล่างเพื่อทำการกำหนดค่าต่างๆของระบบ

|  |
| --- |
| **รูปที่ 4.11** แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (3) |

จากรูปที่ 4.11 เมื่อคลิกที่ Setting เว็บแอปพลิเคชัน (Web Application) จะแสดงหน้า Setting ขึ้นมาดังรูปที่แสดงด้านล่าง โดยหน้า Setting สามารถกำหนดค่าต่างๆภายในระบบและเปิดปิดการแจ้งเตือนของ LINE Notify ได้

|  |
| --- |
| **รูปที่ 4.12** แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (4) |

ทดลองกำหนดค่าอุณหภูมิต่ำสุดโดยค่าอุณหภูมิต่ำสุดมีค่าเท่ากับ 25 องศาเซลเซียส (°C) จากนั้นทำการเปลี่ยนค่าอุณหภูมิต่ำสุดเป็น 26 องศาเซลเซียส (°C) โดยกรอกที่ textbox ดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 4.13** แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (5) |
| **รูปที่ 4.14** แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (6) |

จากรูปที่ 4.14 เมื่อทำการทดลองเปลี่ยนแปลงค่าอุณหภูมิต่ำสุดแล้วเว็บแอปพลิเคชัน (Web Application) จะทำการแสดงค่าอุณหภูมิต่ำสุดตามที่ทำการทดลอง ขั้นตอนต่อมาทำการทดลองเปลี่ยนแปลงค่าอุณหภูมิสูงสุด โดยทำการทดลองกำหนดค่าอุณหภูมิสูงสุดโดยค่าอุณหภูมิสูงสุดมีค่าเท่ากับ 30 องศาเซลเซียส (°C) จากนั้นทำการเปลี่ยนค่าอุณหภูมิสูงสุดเป็น 31 องศาเซลเซียส (°C) โดยกรอกที่ textbox ดังรูปที่ 4.15

|  |
| --- |
| **รูปที่ 4.15** แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (7) |

|  |
| --- |
| **รูปที่ 4.16** แสดงภาพการทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด (8) |

จากรูปที่ 4.16 เมื่อทำการทดลองเปลี่ยนแปลงค่าอุณหภูมิสูงสุดแล้วเว็บแอปพลิเคชัน (Web Application) จะทำการแสดงค่าอุณหภูมิสูงสุดตามที่ทำการทดลอง

**4.1.1.4 ทดลองกำหนดค่าความชื้นต่ำสุดและความชื้นสูงสุด**

ทดลองกำหนดค่าความชื้นต่ำสุดโดยค่าความชื้นต่ำสุดมีค่าเท่ากับ 80 เปอร์เซ็นต์ (%) จากนั้นทำการเปลี่ยนค่าความชื้นต่ำสุดเป็น 82 เปอร์เซ็นต์ (%) โดยกรอกที่ textbox ดังรูปที่ 4.17

|  |
| --- |
| **รูปที่ 4.17** แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุด (1) |

|  |
| --- |
| **รูปที่ 4.18** แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุด (2) |

จากรูปที่ 4.18 เมื่อทำการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุดแล้วเว็บแอปพลิเคชัน (Web Application) จะทำการแสดงค่าความชื้นต่ำสุดตามที่ทำการทดลอง

ขั้นตอนต่อมาทำการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด โดยทำการทดลองกำหนดค่าความชื้นสูงสุดโดยค่าความชื้นสูงสุดมีค่าเท่ากับ 90 เปอร์เซ็นต์ (%) จากนั้นทำการเปลี่ยนค่าความชื้นสูงสุดเป็น 89 เปอร์เซ็นต์ (%) โดยกรอกที่ textbox ดังรูปที่ 4.19

|  |
| --- |
| **รูปที่ 4.19** แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด (1) |
| **รูปที่ 4.20** แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด (2) |

จากรูปที่ 4.20 เมื่อทำการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุดแล้วเว็บแอปพลิเคชัน (Web Application) จะทำการแสดงค่าความชื้นสูงสุดตามที่ทำการทดลอง

**4.1.1.5 ทดลองกำหนดค่าความสว่าง**

ทดลองกำหนดค่าความสว่างโดยค่าความสว่างมีค่าเท่ากับ 200 ลักซ์ (lx) จากนั้นทำการเปลี่ยนค่าความสว่างเป็น 210 ลักซ์ (lx) โดยกรอกที่ textbox ดังรูปที่ 4.21

|  |
| --- |
| **รูปที่ 4.21** แสดงภาพการทดลองเปลี่ยนแปลงค่าความสว่าง (1) |

|  |
| --- |
| **รูปที่ 4.22** แสดงภาพการทดลองเปลี่ยนแปลงค่าความสว่าง (2) |

จากรูปที่ 4.22 เมื่อทำการทดลองเปลี่ยนแปลงค่าความสว่างแล้วเว็บแอปพลิเคชัน (Web Application) จะทำการแสดงค่าความสว่างตามที่ทำการทดลอง

**ตารางที่ 4.1** สรุปผลการทดลองการทำงานของเว็บแอปพลิเคชัน

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงาน** | **ไม่ทำงาน** | **หมายเหตุ** |
| 1 | ทดลองเปลี่ยนโหมดการทำงาน | ✓ |  |  |
| 2 | ทดลองควบคุมพัดลมและปั๊ม | ✓ |  |  |
| 3 | ทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด | ✓ |  |  |
| 4 | ทดลองกำหนดค่าความชื้นต่ำสุดและความชื้นสูงสุด | ✓ |  |  |
| 5 | ทดลองกำหนดค่าความสว่าง | ✓ |  |  |

**4.1.2 การทดลองการทำงานของวินโดว์แอปพลิเคชัน**

ทดลองใช้งานวินโดว์แอปพลิเคชัน (Windows Application) ในการควบคุมระบบภายในโรงเรือนเพาะเห็ด เช่น การเปลี่ยนโหมดการทำงาน เปิด/ปิดปั๊มพ่นหมอกและพัดลมระบายอากาศ กำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด กำหนดค่าความชื้นต่ำสุดและความชื้นสูงสุด และการกำหมดค่าความสว่าง เป็นต้น

ทดลองใช้งาน Windows Application ในการควบคุมระบบภายในโรงเรือนเพาะเห็ด ขั้นตอนแรกเมื่อเข้าใช้งาน Windows Application จะต้องทำการเข้าสู่ระบบก่อนดังรูปที่ 4.23

|  |
| --- |
| **รูปที่ 4.23** แสดงภาพหน้า Login ของวินโดว์แอปพลิเคชัน |

จากรูปที่ 4.23 จะต้องทำการกรอก Username และ Password ให้ถูกต้องก่อนเข้าใช้งาน Windows Application หากกรอก Username หรือ Password ไม่ถูกต้องจะแสดงข้อความแจ้งเตือนดังรูปที่ 4.24

|  |
| --- |
| **รูปที่ 4.24** แสดงภาพแสดงข้อความแจ้งเตือนเมื่อกรอกข้อมูลไม่ถูกต้อง |
| **รูปที่ 4.25** แสดงภาพวินโดว์แอปพลิเคชันเมื่อยังไม่ได้เชื่อมต่อกับ Serial Port |

จากรูปที่ 4.25 เมื่อทำการกรอก Username และ Password ถูกต้องแล้วจะเข้าสู่หน้าแรกของ Windows Application ที่ยังไม่เชื่อมต่อกับ Serial Port เมื่อเชื่อมต่อกับ Serial Port แล้ว ESP32LoRa จะส่งข้อมูลผ่าน Serial Port เพื่อมาแสดงบน Windows Application ดังรูปที่ 4.26

|  |
| --- |
| **รูปที่ 4.26** แสดงภาพวินโดว์แอปพลิเคชันเมื่อเชื่อมต่อกับ Serial Port |

จากรูปที่ 4.26 เมื่อวินโดว์แอปพลิเคชัน (Windows Application) ทำการเชื่อมต่อกับ Serial Port แล้วจะนำข้อมูลที่ได้มาแสดง เช่น อุณหภูมิ ความชื้น ความสว่าง โหมดการทำงาน ค่าต่างๆที่กำหนดไว้ เป็นต้น ขั้นตอนต่อมาทำการทดลองควบคุมการทำงานของระบบผ่านทางวินโดว์แอปพลิเคชัน (Windows Application)

**4.1.2.1 ทดลองเปลี่ยนโหมดการทำงาน**

จากรูปที่ 4.16 จะเห็นว่าโหมดการทำงานเป็นโหมด AUTO จากนั้นเมื่อคลิกที่ปุ่มเปลี่ยนโหมดการทำงาน (Change mode) ดังรูปที่ 4.27

|  |
| --- |
| **รูปที่ 4.27** แสดงภาพปุ่มเปลี่ยนโหมดการทำงานของวินโดว์แอปพลิเคชัน |

จากรูปที่ 4.27 เมื่อทำการกดที่ปุ่มเปลี่ยนโหมดการทำงาน (Change mode) วินโดว์แอปพลิเคชัน (Windows Application) จะเปลี่ยนโหมดการทำงานจากโหมด AUTO เป็นโหมด MANUAL และแสดงปุ่มควบคุม เปิด/ปิด พัดลมและปั๊มดังรูปที่ 4.28

|  |
| --- |
| **รูปที่ 4.28** แสดงภาพวินโดว์แอปพลิเคชันเมื่อโหมดการทำงานเป็นโหมด MANUAL |

**4.1.2.2 ทดลองควบคุมพัดลมและปั๊ม**

จากรูปที่ 4.28 จะเห็นว่าปุ่มควบคุมพัดลม (Turn off Fan) เป็นสีแดงหมายความว่าพัดลมกำลังทำงาน จากนั้นทำงานสั่งปิดพัดลมโดยกดที่ปุ่มควบคุมพัดลม (Turn off Fan) ดังรูปที่ 4.29

|  |
| --- |
| **รูปที่ 4.29** แสดงภาพปุ่มควบคุมพัดลม (Turn off Fan) ขณะที่พัดลมกำลังทำงาน |
| **รูปที่ 4.30** แสดงภาพวินโดว์แอปพลิเคชันหลังจากสั่งปิดพัดลม |

จากรูปที่ 4.30 เมื่อกดที่ปุ่มควบคุมพัดลม (Turn off Fan) วินโดว์แอปพลิเคชัน (Windows Application) จะเปลี่ยนสถานการณ์ทำงานของพัดลมเป็น OFF ตัวสีแดง ต่อมาทำการควบคุมปั๊มโดยกดที่ปุ่มควบคุมปั๊ม (Turn on Pump) ดังรูปที่ 4.31

|  |
| --- |
| **รูปที่ 4.31** แสดงภาพปุ่มควบคุมปั๊ม (Turn on Fan) ขณะที่ปั๊มไม่ทำงาน |
| **รูปที่ 4.32** แสดงภาพวินโดว์แอปพลิเคชันหลังจากสั่งเปิดปั๊ม |

จากรูปที่ 4.32 เมื่อกดที่ปุ่มควบคุมปั๊ม (Turn on Pump) วินโดว์แอปพลิเคชัน (Windows Application) จะเปลี่ยนสถานการณ์ทำงานของปั๊มเป็น ON ตัวสีเขียว

**4.1.2.3 ทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด**

ทดลองกำหนดค่าอุณหภูมิต่ำสุดโดยค่าอุณหภูมิต่ำสุดมีค่าเท่ากับ 25 องศาเซลเซียส (°C) จากนั้นทำการเปลี่ยนค่าอุณหภูมิต่ำสุดเป็น 26 องศาเซลเซียส (°C) โดยกรอกที่ textbox ดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 4.33** แสดงภาพการทดลองเปลี่ยนแปลงค่าอุณหภูมิต่ำสุด (1) |
| **รูปที่ 4.34** แสดงภาพการทดลองเปลี่ยนแปลงค่าอุณหภูมิต่ำสุด (2) |

จากรูปที่ 4.34 เมื่อทำการทดลองเปลี่ยนแปลงค่าอุณหภูมิต่ำสุดแล้ววินโดว์แอปพลิเคชัน (Windows Application) จะทำการแสดงค่าอุณหภูมิต่ำสุดตามที่ทำการทดลอง ขั้นตอนต่อมาทำการทดลองเปลี่ยนแปลงค่าอุณหภูมิสูงสุด โดยทำการทดลองกำหนดค่าอุณหภูมิสูงสุดโดยค่าอุณหภูมิสูงสุดมีค่าเท่ากับ 30 องศาเซลเซียส (°C) จากนั้นทำการเปลี่ยนค่าอุณหภูมิสูงสุดเป็น 31 องศาเซลเซียส (°C) โดยกรอกที่ textbox ดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 4.35** แสดงภาพการทดลองเปลี่ยนแปลงค่าอุณหภูมิสูงสุด (1) |

|  |
| --- |
| **รูปที่ 4.36** แสดงภาพการทดลองเปลี่ยนแปลงค่าอุณหภูมิสูงสุด (2) |

จากรูปที่ 4.36 เมื่อทำการทดลองเปลี่ยนแปลงค่าอุณหภูมิสูงสุดแล้ววินโดว์แอปพลิเคชัน (Windows Application) จะทำการแสดงค่าอุณหภูมิสูงสุดตามที่ทำการทดลอง

**4.1.2.4 ทดลองกำหนดค่าความชื้นต่ำสุดและความชื้นสูงสุด**

ทดลองกำหนดค่าความชื้นต่ำสุดโดยค่าความชื้นต่ำสุดมีค่าเท่ากับ 80 เปอร์เซ็นต์ (%) จากนั้นทำการเปลี่ยนค่าความชื้นต่ำสุดเป็น 81 เปอร์เซ็นต์ (%) โดยกรอกที่ textbox ดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 4.37** แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุด (1) |
| **รูปที่ 4.38** แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุด (2) |

จากรูปที่ 4.38 เมื่อทำการทดลองเปลี่ยนแปลงค่าความชื้นต่ำสุดแล้ววินโดว์แอปพลิเคชัน (Windows Application) จะทำการแสดงค่าความชื้นต่ำสุดตามที่ทำการทดลอง

ขั้นตอนต่อมาทำการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด โดยทำการทดลองกำหนดค่าความชื้นสูงสุดโดยค่าความชื้นสูงสุดมีค่าเท่ากับ 90 เปอร์เซ็นต์ (%) จากนั้นทำการเปลี่ยนค่าความชื้นสูงสุดเป็น 91 เปอร์เซ็นต์ (%) โดยกรอกที่ textbox ดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 4.39** แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด (1) |

|  |
| --- |
| **รูปที่ 4.40** แสดงภาพการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุด (2) |

จากรูปที่ 4.40 เมื่อทำการทดลองเปลี่ยนแปลงค่าความชื้นสูงสุดแล้ววินโดว์แอปพลิเคชัน (Windows Application) จะทำการแสดงค่าความชื้นสูงสุดตามที่ทำการทดลอง

**4.1.2.5 ทดลองกำหนดค่าความสว่าง**

ทดลองกำหนดค่าความสว่างโดยค่าความสว่างมีค่าเท่ากับ 200 ลักซ์ (lx) จากนั้นทำการเปลี่ยนค่าความสว่างเป็น 230 ลักซ์ (lx) โดยกรอกที่ textbox ดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 4.41** แสดงภาพการทดลองเปลี่ยนแปลงค่าความสว่าง (1) |

|  |
| --- |
| **รูปที่ 4.42** แสดงภาพการทดลองเปลี่ยนแปลงค่าความสว่าง (2) |

จากรูปที่ 4.42 เมื่อทำการทดลองเปลี่ยนแปลงค่าความสว่างแล้ววินโดว์แอปพลิเคชัน (Windows Application) จะทำการแสดงค่าความสว่างตามที่ทำการทดลอง

**ตารางที่ 4.2** สรุปผลการทดลองการทำงานของวินโดว์แอปพลิเคชัน

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงาน** | **ไม่ทำงาน** | **หมายเหตุ** |
| 1 | ทดลองเปลี่ยนโหมดการทำงาน | ✓ |  |  |
| 2 | ทดลองควบคุมพัดลมและปั๊ม | ✓ |  |  |
| 3 | ทดลองกำหนดค่าอุณหภูมิต่ำสุดและอุณหภูมิสูงสุด | ✓ |  |  |
| 4 | ทดลองกำหนดค่าความชื้นต่ำสุดและความชื้นสูงสุด | ✓ |  |  |
| 5 | ทดลองกำหนดค่าความสว่าง | ✓ |  |  |

**4.1.3 การทดลองการทำงานของระบบควบคุมสภาพแวดล้อม**

ในการทดลองนี้เป็นการทดลองการทำงานของระบบควบคุมสภาพแวดล้อมภายในโรงเรือน ทำการทดลองโดย ทดลอง เปิด-ปิด อุปกรณ์ควบคุมสภาพแวดล้อม ทดลองเปลี่ยนโหมดการทำงาน ทดลองกำหนดค่าสภาพแวดล้อม จากนั้นทำการสังเกตการทำงานของระบบและบันทึกผลการทดลองทั้งหมดเป็นระยะเวลา 10 วัน

**การทดลอง เปิด-ปิด อุปกรณ์ควบคุมสภาพแวดล้อม** ทำการทดลองโดยควบคุมการทำงานของอุปกรณ์ควบคุมสภาพแวดล้อมจาก เว็บแอปพลิเคชัน วินโดว์แอปพลิเคชัน ปุ่มควบคุมหน้าโรงเรือน และปุ่มควบคุมที่กล่อง STA

**การทดลองเปลี่ยนโหมดการทำงาน** ทำการทดลองโดยเปลี่ยนโหมดการทำงานของระบบจาก เว็บแอปพลิเคชัน วินโดว์แอปพลิเคชัน ปุ่มควบคุมหน้าโรงเรือน และปุ่มควบคุมที่กล่อง STA

**การทดลองกำหนดค่าสภาพแวดล้อม** ทำการทดลองโดยการกำหนดค่าสภาพแวดล้อมต่างๆ เช่น อุณหภูมิ ความชื้น และแสงสว่าง จากนั้นสังเกตการทำงานของอุปกรณ์ควบคุมสภาพแวดล้อม

กำหนดให้

|  |  |  |
| --- | --- | --- |
| T | เท่ากับ | อุณหภูมิ |
| Tmax | เท่ากับ | อุณหภูมิสูงสุด |
| Tmin | เท่ากับ | อุณหภูมิต่ำสุด |
| H | เท่ากับ | ความชื้น |
| Hmax | เท่ากับ | ความชื้นสูงสุด |
| Hmin | เท่ากับ | ความชื้นต่ำสุด |
| L | เท่ากับ | ความสว่าง |
| Ls | เท่ากับ | ความสว่างที่กำหนด |

**ตารางที่ 4.3** สรุปผลการทดลองการทำงานของระบบวันที่ 1

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.4** สรุปผลการทดลองการทำงานของระบบวันที่ 2

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.5** สรุปผลการทดลองการทำงานของระบบวันที่ 3

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.6** สรุปผลการทดลองการทำงานของระบบวันที่ 4

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.7** สรุปผลการทดลองการทำงานของระบบวันที่ 5

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.8** สรุปผลการทดลองการทำงานของระบบวันที่ 6

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.9** สรุปผลการทดลองการทำงานของระบบวันที่ 7

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.10** สรุปผลการทดลองการทำงานของระบบวันที่ 8

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.11** สรุปผลการทดลองการทำงานของระบบวันที่ 9

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**ตารางที่ 4.12** สรุปผลการทดลองการทำงานของระบบวันที่ 10

|  |  |  |  |
| --- | --- | --- | --- |
| **ลำดับ** | **การทดลอง** | **ทำงานถูกต้อง** | **ทำงานผิดพลาด** |
| 1 | เปิด-ปิด พัดลมระบายอากาศ | ✓ |  |
| 2 | เปิด-ปิด ปั๊มพ่นหมอก | ✓ |  |
| 3 | เปลี่ยนโหมดการทำงานของระบบ | ✓ |  |
| 4 | กำหนดให้ Tmax < T | ✓ |  |
| 5 | กำหนดให้ Tmax > T | ✓ |  |
| 6 | กำหนดให้ Tmin < T | ✓ |  |
| 7 | กำหนดให้ Tmin > T | ✓ |  |
| 8 | กำหนดให้ Hmax < H | ✓ |  |
| 9 | กำหนดให้ Hmax > H | ✓ |  |
| 10 | กำหนดให้ Hmin < H | ✓ |  |
| 11 | กำหนดให้ Hmin > H | ✓ |  |
| 12 | กำหนดให้ Ls < L | ✓ |  |
| 13 | กำหนดให้ Ls > L | ✓ |  |

**4.2** **การทดลองเพาะเห็ด**

การทดลองเพาะเห็ดเพื่อทดสอบประสิทธิภาพของโรงเรือนเพาะเห็ดอัจฉริยะในการทดลองเพาะเห็ดจริง โดยใช้เห็ดนางฟ้าในการทำการทดลอง โดยทำการทดลองเพาะเห็ดนางฟ้าระหว่างวันที่ 8 ตุลาคม พ.ศ. 2563 ถึงวันที่ 21 ตุลาคม พ.ศ. 2563 รวมทั้งหมด 14 วัน

การรักษาสภาพแวดล้อมภายในโรงเรือนเพาะเห็ดให้เหมาะสมต่อการออกดอกของเห็ด จะต้องควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอก เช่น อุณหภูมิ ความชื้น และแสงสว่าง โดยผู้จัดทำได้นำเห็ดนางฟ้ามาใช้ในการทำการทดลองเพาะเห็ด จากการศึกษาสภาพแวดล้อมที่เหมาะสมกับการออกดอกของเห็ดนางฟ้าสามารถสรุปได้ดังนี้ อุณหภูมิควรอยู่ระหว่าง 25-30 องศาเซลเซียส (°C) ความชื้นสัมพัทธ์ 80-90 เปอร์เซ็นต์ (%) [12] แสงสว่างที่มีความเข้มของแสงประมาณ 200 lux เป็นแสงสีน้ำเงินจะมีทำให้เห็ดออกดอกได้ดีที่สุด [13]

โครงสร้างของระบบที่ใช้ในการควบคุมสภาพแวดล้อมภายในโรงเรือนจะประกอบด้วยเซนเซอร์วัดอุณหภูมิและความชื้น เพื่อวัดอุณหภูมิและความชื้นภายในโรงเรือนระบบปิด จากนั้นนำข้อมูลอุณหภูมิและความชื้นที่ได้ไปเปรียบเทียบกับค่าที่ตั้งไว้คือ อุณหภูมิควรอยู่ระหว่าง 25-30 องศาเซลเซียส (°C) ความชื้นสัมพัทธ์ 80-90 เปอร์เซ็นต์ (%) หากอุณหภูมิและความชื้นไม่อยู่ภายในช่วงที่กำหนด จะควบคุมการทำงานของปี๊มพ่นหมอกและพัดลมระบายอากาศ เพื่อใช้อุณหภูมิและความชื้นอยู่ในช่วงที่กำหนด โดยการควบคุมการทำงานของปั๊มพ่นหมอกและพัดลมระบายอากาศจะมีการทำงานตามตารางที่ 4.13

**ตารางที่ 4.13** แสดงการทำงานของปั๊มพ่นหมอกและพัดลมในโหมดการทำงานแบบอัตโนมัติ (AUTO)

|  |  |  |  |
| --- | --- | --- | --- |
| **อุณหภูมิ(°C)** | **ความชื้น(%)** | **ปั๊มพ่นหมอก** | **พัดลม** |
| < 25 | < 80 | ON | OFF |
| < 25 | 80-90 | OFF | OFF |
| < 25 | > 90 | OFF | OFF |
| 25-30 | < 80 | ON | OFF |
| 25-30 | 80-90 | OFF | OFF |
| 25-30 | > 90 | OFF | OFF |
| > 30 | < 80 | ON | ON |
| > 30 | 80-90 | ON | ON |
| > 30 | > 90 | ON | ON |

ในการควบคุมแสงสว่างภายในโรงเรือนผู้จัดทำได้นำ LED ขนาด 5 mm สีน้ำเงินมาใช้ในการให้แสงสว่างภายในโรงเรือน โดยกำหนดค่าความเข้มของแสงที่ 200 lux หากค่าความเข้มของแสงน้อยกว่า 200 lux ก็จะเพิ่มความสว่างของหลอด LED หากความสว่างมากกว่า 200 lux ก็จะลดความสว่างของหลอด LED ทำการควบคุมแสงสว่าง 12 ชั่วโมงต่อวันตั้งแต่เวลา 06.00-18.00 น.

การทดลองเพาะเห็ดผู้จัดทำได้ทำการเปรียบเทียบการเจริญเติบโตของเห็ดโดยทำการเพาะเห็ดนางฟ้าจำนวน 20 ก้อน โดยที่ 10 ก้อนจะเพาะภายในโรงเรือนที่สามารถควบคุมอุณหภูมิความชื้นและแสงสว่างได้ อีก 10 จะเพาะภายในโรงเรือนที่ไม่มีการควบคุมอุณหภูมิความชื้นและแสงสว่างดังรูปที่ 4.43 จากนั้นทำการเปรียบเทียบความกว้างของดอกและน้ำหนักของเห็ดเมื่อทำการเก็บเกี่ยวผลผลิต

|  |
| --- |
| **รูปที่ 4.43** แสดงภาพของโรงเรือนที่สร้างขึ้นเพื่อใช้ในการเปรียบเทียบผลการทดลอง |

ผลการทดลองเพาะเห็ดนางฟ้าจะแบ่งออกเป็น 2 ช่วงเพราะทำการเพาะเห็ดทั้งหมด 2 รุ่นเริ่มเปิดดอกเห็ดวันที่ 8 ตุลาคม 2563 โดยเห็ดรุ่นที่ 1 ทำการเก็บเกี่ยวผลผลิตวันที่ 11 ตุลาคม 2563 รุ่นที่ 2 ทำการเก็บเกี่ยวผลผลิตวันที่ 21 ตุลาคม 2563

**4.2.1 ผลการทดลองเพาะเห็ดรุ่นที่ 1**

|  |  |
| --- | --- |
| (ก.) | (ข.) |
| **รูปที่ 4.44** แสดงภาพผลการทดลองวันที่ 9 ตุลาคม 2563  (ก.) ผลการทดลองภายในโรงเรือนที่มีการควบคุมสภาพแวดล้อม  (ข.) ผลการทดลองภายในโรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อม | |

|  |  |
| --- | --- |
| (ก.) | (ข.) |
| **รูปที่ 4.45** แสดงภาพผลการทดลองวันที่ 10 ตุลาคม 2563  (ก.) ผลการทดลองภายในโรงเรือนที่มีการควบคุมสภาพแวดล้อม  (ข.) ผลการทดลองภายในโรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อม | |
| (ก.) | (ข.) |
| **รูปที่ 4.46** แสดงภาพผลการทดลองวันที่ 11 ตุลาคม 2563  (ก.) ผลการทดลองภายในโรงเรือนที่มีการควบคุมสภาพแวดล้อม  (ข.) ผลการทดลองภายในโรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อม | |

|  |  |
| --- | --- |
| (ก.) | (ข.) |
| **รูปที่ 4.47** แสดงภาพผลผลิตที่ได้จากการทดลองเพาะเห็ดนางฟ้า  (ก.) ผลผลิตที่ได้จากโรงเรือนที่มีการควบคุมสภาพแวดล้อมน้ำหนัก 0.7 กิโลกรัม  (ข.) ผลผลิตที่ได้จากโรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อมน้ำหนัก 0.4 กิโลกรัม | |

จากผลการทดลองด้านบนสามารถสรุปผลการทดลองได้ดังนี้ การเพาะเห็ดในโรงเรือนที่มีการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยก้อนละ 6.18 เซนติเมตร มีน้ำหนักทั้งหมด 0.7 กิโลกรัม การเพาะเห็ดในโรงเรือนที่ไม่มีกาการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยก้อนละ 5.67 เซนติเมตร มีน้ำหนักทั้งหมด 0.4 กิโลกรัม

**4.2.2 ผลการทดลองเพาะเห็ดรุ่นที่ 2**

|  |  |
| --- | --- |
| (ก.) | (ข.) |
| **รูปที่ 4.48** แสดงภาพผลการทดลองวันที่ 19 ตุลาคม 2563  (ก.) ผลการทดลองภายในโรงเรือนที่มีการควบคุมสภาพแวดล้อม  (ข.) ผลการทดลองภายในโรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อม | |

|  |  |
| --- | --- |
| (ก.) | (ข.) |
| **รูปที่ 4.49** แสดงภาพผลการทดลองวันที่ 20 ตุลาคม 2563  (ก.) ผลการทดลองภายในโรงเรือนที่มีการควบคุมสภาพแวดล้อม  (ข.) ผลการทดลองภายในโรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อม | |
| (ก.) | (ข.) |
| **รูปที่ 4.50** แสดงภาพผลการทดลองวันที่ 21 ตุลาคม 2563  (ก.) ผลการทดลองภายในโรงเรือนที่มีการควบคุมสภาพแวดล้อม  (ข.) ผลการทดลองภายในโรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อม | |

|  |  |
| --- | --- |
| (ก.) | (ข.) |
| **รูปที่ 4.51** แสดงภาพผลผลิตที่ได้จากการทดลองเพาะเห็ดนางฟ้า  (ก.) ผลผลิตที่ได้จากโรงเรือนที่มีการควบคุมสภาพแวดล้อมน้ำหนัก 0.36 กิโลกรัม  (ข.) ผลผลิตที่ได้จากโรงเรือนที่ไม่มีการควบคุมสภาพแวดล้อมน้ำหนัก 0.2 กิโลกรัม | |

จากผลการทดลองด้านบนสามารถสรุปผลการทดลองได้ดังนี้ การเพาะเห็ดในโรงเรือนที่มีการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยก้อนละ 9.14 เซนติเมตร มีน้ำหนักทั้งหมด 0.36 กิโลกรัม การเพาะเห็ดในโรงเรือนที่ไม่มีกาการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยก้อนละ 7.22 เซนติเมตร มีน้ำหนักทั้งหมด 0.2 กิโลกรัม

จากการทดลองเพาะเห็ดนางฟ้าระหว่างวันที่ 8 ตุลาคม พ.ศ. 2563 ถึงวันที่ 21 ตุลาคม พ.ศ. 2563 รวมทั้งหมด 14 วัน สามารถสรุปผลการทดลองได้ดังนี้

รุ่นที่ 1 ทำการเก็บเกี่ยวผลผลิตวันที่ 11 ตุลาคม 2563 การเพาะเห็ดในโรงเรือนที่มีการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ด มีความกว้างของดอกเฉลี่ยก้อนละ 6.18 เซนติเมตร มีน้ำหนักทั้งหมด 0.7 กิโลกรัม การเพาะเห็ดในโรงเรือนที่ไม่มีกาการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยก้อนละ 5.67 เซนติเมตร มีน้ำหนักทั้งหมด 0.4 กิโลกรัม

รุ่นที่ 2 ทำการเก็บเกี่ยวผลผลิตวันที่ 21 ตุลาคม 2563 การเพาะเห็ดในโรงเรือนที่มีการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ด มีความกว้างของดอกเฉลี่ยก้อนละ 9.14 เซนติเมตร มีน้ำหนักทั้งหมด 0.36 กิโลกรัม การเพาะเห็ดในโรงเรือนที่ไม่มีกาการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยก้อนละ 7.22 เซนติเมตร มีน้ำหนักทั้งหมด 0.2 กิโลกรัม

ดังนั้นการเพาะเห็ดในโรงเรือนที่มีการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ด มีความกว้างของดอกเฉลี่ยเท่ากับ 7.66 เซนติเมตร มีน้ำหนักทั้งหมดเท่ากับ 1.06 กิโลกรัม การเพาะเห็ดในโรงเรือนที่ไม่มีกาการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยเท่ากับ 6.45 เซนติเมตร มีน้ำหนักทั้งหมดเท่ากับ 0.6 กิโลกรัม

**บทที่ 5**

**สรุปผลการดำเนินงาน**

**5.1 สรุปผล**

ปริญญานิพนธ์นี้วัตถุประสงค์เพื่อออกแบบและสร้างเว็บแอปพลิเคชันและวินโดว์แอปพลิเคชันสำหรับมอนิเตอร์และควบคุมระบบโรงเรือนเพาะเห็ด ออกแบบและสร้างโรงเรือนที่ใช้สำหรับเพาะเห็ด ออกแบบระบบควบคุมอุณหภูมิความชื้นและแสงสว่างภายในโรงเรือนเพาะเห็ด ศึกษาและประยุกต์ใช้งาน Lora Communication สำหรับการควบคุมโรงเรือนเพาะเห็ด และเพื่อศึกษาการพัฒนาเว็บแอปพลิเคชันบนไมโครคอนโทรลเลอร์ ESP32LoRa

การทดลองการทำงานของระบบประกอบไปด้วย การทดลองการทำงานของเว็บแอปพลิเคชัน การทดลองการทำงานของวินโดว์แอปพลิเคชัน การทดลองการทำงานของระบบควบคุมสภาพแวดล้อม โดยทั้ง 3 ส่วนสามารถทำงานได้เป็นอย่างดี

การทดลองเพาะเห็ดเป็นการทดสอบประสิทธิภาพของโรงเรือนเพาะเห็ดอัจฉริยะในการทดลองเพาะเห็ดจริง โดยใช้เห็ดนางฟ้าในการทำการทดลอง จากการทดลองเพาะเห็ดนางฟ้าระหว่างวันที่ 8 ตุลาคม พ.ศ. 2563 ถึงวันที่ 21 ตุลาคม พ.ศ. 2563 รวมทั้งหมด 14 วัน สามารถสรุปผลการทดลองได้ดังนี้

รุ่นที่ 1 ทำการเก็บเกี่ยวผลผลิตวันที่ 11 ตุลาคม 2563 การเพาะเห็ดในโรงเรือนที่มีการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ด มีความกว้างของดอกเฉลี่ยก้อนละ 6.18 เซนติเมตร มีน้ำหนักทั้งหมด 0.7 กิโลกรัม การเพาะเห็ดในโรงเรือนที่ไม่มีกาการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยก้อนละ 5.67 เซนติเมตร มีน้ำหนักทั้งหมด 0.4 กิโลกรัม

รุ่นที่ 2 ทำการเก็บเกี่ยวผลผลิตวันที่ 21 ตุลาคม 2563 การเพาะเห็ดในโรงเรือนที่มีการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ด มีความกว้างของดอกเฉลี่ยก้อนละ 9.14 เซนติเมตร มีน้ำหนักทั้งหมด 0.36 กิโลกรัม การเพาะเห็ดในโรงเรือนที่ไม่มีกาการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยก้อนละ 7.22 เซนติเมตร มีน้ำหนักทั้งหมด 0.2 กิโลกรัม

ดังนั้นการเพาะเห็ดในโรงเรือนที่มีการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ด มีความกว้างของดอกเฉลี่ยเท่ากับ 7.66 เซนติเมตร มีน้ำหนักทั้งหมดเท่ากับ 1.06 กิโลกรัม การเพาะเห็ดในโรงเรือนที่ไม่มีกาการควบคุมสภาพแวดล้อมให้เหมาะสมต่อการออกดอกของเห็ดโดยมีความกว้างของดอกเฉลี่ยเท่ากับ 6.45 เซนติเมตร มีน้ำหนักทั้งหมดเท่ากับ 0.6 กิโลกรัม

**5.2 ปัญหาที่เกิด**

5.2.1 ในการนำอุปกรณ์ประเภทความเหนี่ยวนำหรือ Inductive load เช่น ปั๊มพ่นหมอกและพัดลมระบายอากาศ มาใช้งานจะต้องคำนึงแรงดันที่พุ่งขึ้นอย่างเฉียบพลันเมื่อกระแสที่ไหลใน Inductive load ลดลงอย่างฉับพลัน เพื่อไม่ให้แรงดันนั้นไหลย้อนกลับเข้าสู่แหล่งจ่ายกระแสไฟแล้วสร้างความเสียหายให้แก่อุปกรณ์

5.2.2 การเลือกใช้งานตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) เพื่อใช้ในการตรวจสอบสถานะการทำงานของอุปกรณ์ ผู้จัดทำได้นำตัวต้านทานที่มีอัตราทนกำลังวัตต์ต่ำเกินไปทำให้เมื่อ พัดลมระบายอากาศทำงาน จะทำให้ต้านต้านทานเกิดความร้อนสูง

**5.3 ข้อเสนอแนะและแนวทางการพัฒนา**

5.3.1 ปริญญานิพนธ์นี้สามารถพัฒนาต่อได้โดยการนำจอสัมผัส (Touch Screen) เช่น จอ Touch Screen (HMI TFT LCD Touch) มาใช้งานในการแสดงผลและควบระบบหน้าโรงเรือนหากนำจอสัมผัส (Touch Screen) มาใช้งานจะช่วยให้สามารถแสดงค่าสาถานะต่างๆ ภายในโรงเรือนได้อย่างครบถ้าน และสามารถสร้างปุ่มเพื่อใช้ในการควบคุมระบบเพื่อลดการใช้งานขา GPIO ของไมโครคอนโทรลเลอร์จากการใช้ปุ่มควบคุมแบบฮาร์ดแวร์

5.3.2 ปริญญานิพนธ์นี้สามารถพัฒนาต่อได้โดยการพัฒนาระบบให้สามารถรองรับโรงเรือนที่มากกว่า 1 โรงเรือน

**บรรณานุกรม**

[1] ขันติชัย รุจิตระการโชติกุล และชยุต สุรกุล. 2560. **“ไอโอทีแพลทฟอร์มโรงเพาะเห็ด”**.ปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

[2] วีรศักดิ์ ฟองเงิน สุรพงษ์ เพ็ชร์หาญ และรัฐสิทธิ์ ยะจ่อ. 2560. **“****การประยุกต์ใช้เทคโนโลยีไอโอทีควบคุมฟาร์มอัจฉริยะในโรงเรือนเพาะเห็ดนางฟ้า”**. ปริญญาตรี คณะวิทยาศาสตร์ มหาวิทยาลัยราชภัฏลำปาง

[3] Oran Chieochan and Anukit Saokaew. 2017. **“****IOT for Smart Farm: A case study of the Lingzhi Mushroom Farm at Maejo University”**. Bachelor’s Thesis, Chiang Mai University, Thailand.

[4] Parvati Bhandari and Megha Kimothi. 2018. **“****IOT BASED DESIGN IMPLEMENTATION OF MUSHROOM FARM MONITORING USING ARDUINO MICROCONTROLLERS & SENSORS”**. Bachelor’s Thesis, Dept. of Wireless Mobile Communication Engg, GRD IMT, Dehradun, India

[5] สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (สพธอ.) (2563) “**ETDA เผย ปี 62 คนไทยใช้อินเทอร์เน็ตเพิ่มขึ้นเฉลี่ย 10 ชั่วโมง 22 นาที Gen Y ครองแชมป์ 5 ปีซ้อน”** [ออนไลน์] เข้าถึงได้จากURL: https://www.etda.or.th/content/thailand-internet-user-behavior-2019-press-release.html (8 สิงหาคม 2563)

[6] นานทรี หุ้นเหี้ยง. (2563) **“การศึกษาปัจจัยทางกายภาพที่มีผลต่อความชื้นสัมพัทธ์และอุณหภูมิภายในห้องเพาะเลี้ยงเห็ดนางฟ้าภูฐานซึ่งสร้างขึ้นด้วยวงบ่อซีเมนต์ตามแบบจำลองหม้อดินเก็บความเย็น** ปริญญาโทสาขาวิชาฟิสิกส์ศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา

[7] วิกิพีเดียสารานุกรมเสรี. (2563) “**อุณหภูมิ”** [ออนไลน์] เข้าถึงได้จากURL: https://th.wikipedia.org/wiki/อุณหภูมิ (8 สิงหาคม 2563)

[8] วิกิพีเดียสารานุกรมเสรี. (2563) “**ความชื้น”** [ออนไลน์] เข้าถึงได้จากURL: https://th.wikipedia.org/wiki/ความชื้น (8 สิงหาคม 2563)

[9] วิกิพีเดียสารานุกรมเสรี. (2563) “**แสง”** [ออนไลน์] เข้าถึงได้จากURL: https://th.wikipedia.org/wiki/แสง (10 สิงหาคม 2563)

[10] kritsada arjchariyaphat. (2561) “**LoRA, LoRaWAN คืออะไร มารู้จักกันดีกว่า”** [ออนไลน์] เข้าถึงได้จากURL: https://medium.com/deaware/lora-lorawan-คืออะไร-มารู้จักกันดีกว่า-98d20055a4ca (10 สิงหาคม 2563)

[11] Bill Schweber. (2561) “**Using Resistors for Current Sensing: It’s More Than Just I = V/R”** [ออนไลน์] เข้าถึงได้จากURL: https://www.powerelectronics.com/technologies/power-management/article/21864130/using-resistors-for-current-sensing-its-more-than-just-i-vr (10 สิงหาคม 2563)

[12] Peter.Oei (2005) “**Small-scale mushroom cultivation”** [ออนไลน์] เข้าถึงได้จากURL: http://www.cogumelohobby.com/IMAGENS/Agromisa-AD-40-E.pdf (10 สิงหาคม 2563)

[13] EXTENDED ESSAY BIOLOGY (2015) “**The effect of the colour of light on the growth of fruiting bodies in Pleurotus ostreatus”** [ออนไลน์] เข้าถึงได้จากURL: http://slowacki.kielce.eu/IB/PSlusarczyk.pdf (10 สิงหาคม 2563)