การพัฒนาระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ดอัจฉริยะด้วยเทคโนโลยี IoT

Development of a Smart Mushroom Cultivation House Environmental Control System Using IoT Technology

นาย สรวิศ ปัญญา

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

วิทยานิพนธ์ระดับปริญญาตรี เสนอภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม
คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม
เพื่อเป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาภูมิศาสตร์
ตุลาคม 2568

อาจารย์ที่ปรึกษา ประธานบริหารหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาภูมิศาสตร์และหัวหน้าภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม คณะเกษตรศาสตร์ทรัพยากรธรรมชาติและสิ่งแวดล้อมได้พิจารณาวิทยานิพนธ์ระดับปริญญาตรีเรื่อง **"การพัฒนาระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ดอัจฉริยะด้วยเทคโนโลยี IoT"** (Development of a Smart Mushroom Cultivation House Environmental Control System Using IoT Technology) ของ สรวิศ ปัญญา เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาภูมิศาสตร์มหาวิทยาลัยนเรศวร

.....

(รองศาสตราจารย์ ดร.สิทธิชัย ชูสำโรง)

อาจารย์ที่ปรึกษาวิทยานิพนธ์

.....

(อาจารย์ ธัญญาลัคน์ จันทร์สมบัติ)

ประธานบริหารหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาภูมิศาสตร์

.....

(ผู้ช่วยศาสตราจารย์ ว่าที่ ร.ต.ดร. รังสรรค์ เกตุอ๊อด)

หัวหน้าภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม

กิตติกรรมประกาศ

วิทยานิพนธ์ระดับปริญญาตรี เรื่อง การพัฒนาระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ดอัจฉริยะด้วยเทคโนโลยี IoT (Development of a Smart Mushroom Cultivation House Environmental Control System Using IoT Technology) ฉบับนี้ได้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความช่วยเหลือและการสนับสนุนจากหลายฝ่าย

ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูงต่อ รองศาสตราจารย์ ดร. สิทธิชัย ชูสำโรง อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ซึ่งได้ให้ความเมตตา อุทิศเวลา และแรงกายแรงใจในการให้คำแนะนำแก่ข้าพเจ้าตลอดระยะเวลาการดำเนินการวิจัย อาจารย์ได้ให้คำปรึกษาอย่างใกล้ชิดในทุกขั้นตอน ตั้งแต่การวางแนวคิด กำหนดขอบเขตของการวิจัย การออกแบบระบบ การเก็บรวบรวมข้อมูล การวิเคราะห์ผล ไปจนถึงการจัดเรียงเนื้อหาอย่างเป็นระบบ ทำให้ข้าพเจ้าได้เรียนรู้ถึงกระบวนการทำงานเชิงวิชาการอย่างลึกซึ้งและมีระเบียบแบบแผน

อาจารย์ที่ปรึกษาได้ให้คำชี้แนะด้วยความอดทนและเอาใจใส่ ทั้งยังให้กำลังใจเมื่อข้าพเจ้าประสบปัญหา หรือรู้สึกท้อแท้ระหว่างดำเนินการวิจัย คำแนะนำของอาจารย์ไม่ได้มีเพียงด้านวิชาการเท่านั้น แต่ยังเป็นแรงบันดาลใจสำคัญที่ช่วยเสริมสร้างทัศนคติที่ดีต่อการเรียนรู้ การคิดวิเคราะห์อย่างมีเหตุผล และการแก้ไขปัญหาด้วยความรอบคอบ ซึ่งถือเป็นบทเรียนล้ำค่าที่จะติดตัวข้าพเจ้าไปตลอดชีวิตการทำงานและการศึกษาในอนาคต

ท้ายที่สุด ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่เป็นแรงใจสำคัญในทุกย่างก้าวของการศึกษา คอยให้การสนับสนุนทั้งด้านจิตใจและกำลังทรัพย์ รวมถึงความรักและความเข้าใจที่มีให้มาโดยตลอด หากปราศจากแรงสนับสนุนจากครอบครัว ข้าพเจ้าอาจไม่สามารถก้าวมาถึงจุดนี้ได้

นาย สรวิศ ปัญญา

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

ชื่อเรื่อง	การพัฒนาระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ดอัจฉริยะด้วยเทคโนโลยี IoT
ผู้ศึกษาวิจัย	สรวิศ ปัญญา
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร สิทธิชัย ชูสำโรง
ประเภทสารนิพนธ์	วิทยานิพนธ์ปริญญาตรี วท.บ. ภูมิศาสตร์, มหาวิทยาลัยนเรศวร, 2565
คำสำคัญ	IoT , Smartfarming , Web Application

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาระบบฟาร์มเห็ดอัจฉริยะที่ใช้เทคโนโลยี Internet of Things (IoT) สำหรับโรงเรือนเพาะเห็ดนางฟ้าภูฐาน โดยระบบสามารถควบคุมปัจจัยสิ่งแวดล้อมที่สำคัญ (อุณหภูมิ, ความชื้น, ความเข้มแสง, และระดับก๊าซคาร์บอนไดออกไซด์) ได้อย่างแม่นยำ จัดเก็บข้อมูลเข้าสู่ฐานข้อมูล และแสดงผลข้อมูลแบบเรียลไทม์ผ่าน Web Application

ระบบฮาร์ดแวร์ประกอบด้วย ESP32 เป็นไมโครคอนโทรลเลอร์หลัก [4]ที่ทำหน้าที่รับค่าจากเซ็นเซอร์ ได้แก่ DHT22 สำหรับวัดอุณหภูมิและความชื้น , MH-Z14A สำหรับวัดระดับ CO2, และ LDR สำหรับตรวจวัดความเข้มของแสง และใช้ Relay Module ในการสั่งงานอุปกรณ์ควบคุม (เช่น พัดลมระบายอากาศ, ระบบพ่นหมอก) เพื่อรักษาสภาพแวดล้อมให้อยู่ในช่วงที่เหมาะสมต่อการเจริญของเห็ดนอกจากนี้ ยังมี Arduino Mega 2560 ทำหน้าที่แสดงผลข้อมูลบนหน้าจอ TFT Monitor สำหรับผู้ดูแลหน้างาน

ผลการดำเนินงานยืนยันว่าระบบสามารถทำงานได้อย่างมีประสิทธิภาพในการตรวจวัดและควบคุมปัจจัยสิ่งแวดล้อมให้อยู่ในช่วงที่กำหนด สามารถจัดเก็บข้อมูลอย่างเป็นระบบ และแจ้งเตือนความผิดปกติแบบเรียลไทม์ได้ ซึ่งช่วยให้ผู้ใช้งานสามารถตัดสินใจและควบคุมการทำงานของโรงเรือนได้อย่างแม่นยำ ลดภาระแรงงานคน และเพิ่มโอกาสในการได้ผลผลิตที่มีคุณภาพและปริมาณที่ดีขึ้น

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

Title	Development of a Smart Mushroom Cultivation House Environmental Control System Using IoT Technology
Author	Sorawit Panya
Advisor	Associate Professor Dr. Sittichai Choosumrong
Academic Paper	Thesis B.S. in Geography,Naresuan University, 2025
Keyword	IoT , Smartfarming , Web Application

ABSTRACT

This research aims to design and develop a Smart Mushroom Farm System utilizing Internet of Things (IoT) technology for the cultivation of Bhutan Oyster Mushrooms. The system is designed to accurately control critical environmental factors, including temperature, humidity, light intensity, and carbon dioxide levels. It also stores data in a database and displays real-time information through a Web Application.

The hardware system is centered around the ESP32 microcontroller, which functions as the main processing unit. It receives data from multiple sensors, including the DHT22 for measuring temperature and humidity, the MH-Z14A for monitoring CO_2 levels, and an LDR sensor for detecting light intensity. A Relay Module is used to control associated devices such as ventilation fans, fogging systems, and solenoid valves to maintain optimal environmental conditions for mushroom growth. Additionally, an Arduino Mega 2560 is integrated to display sensor readings and device statuses directly on a TFT monitor for on-site supervision.

The operational results confirm that the system efficiently monitors and controls environmental factors within the defined thresholds. It successfully stores data systematically and provides real-time alerts for abnormal conditions, enabling users to make precise decisions and manage the mushroom house effectively. This capability significantly reduces human labor and enhances the probability of achieving high-quality and high-yield mushroom production.

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright© by Naresuan University
All rights reserved

สารบัญ

บทที่ **หน้า**

1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบเขตงานวิจัย 3

1.4 นิยามศัพท์เฉพาะ 3

1.5 สมมติฐานงานวิจัย 4

1.6 กรอบแนวคิดงานวิจัย 5

2 เอกสารและงานวิจัยที่เกี่ยวข้อง 6

2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง 7

2.1.1เทคโนโลยี Internet of Things (IoT) ในภาคเกษตร 7

2.1.2 ความรู้พื้นฐานเกี่ยวกับการเพาะเห็ด 8

2.1.3 ปัจจัยที่มีผลต่อการเจริญเติบโตของเห็ด 8

2.1.4 โรงเรือนระบบอีแวป (EVAP) สำเร็จรูป ขนาด 1.6 x 4 เมตร 10

2.1.5 ซอฟต์แวร์ที่เกี่ยวข้องกับการพัฒนาระบบ 11

2.2 เอกสารงานวิจัยที่เกี่ยวข้อง 16

3 วิธีดำเนินการวิจัย 19

3.1 การเก็บข้อมูล 19

3.2 วัสดุอุปกร์ที่ใช้ในงานวิจัย 20

3.2.1 ส่วนของฮาร์ดแวร์ 20

3.2.2 ส่วนของซอฟต์แวร์ 20

3.3 การออกแบบระบบ 22

3.3.1 การทำงานของส่วนฮาร์ดแวร์ 22

3.3.2 การทำงานของส่วนซอฟต์แวร์ 23

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

สารบัญ(ต่อ)

3.3.3 การต่อวงจรเซนเซอร์ 24

3.4 การพัฒนาระบบ 25

3.4.1 การออกแบบและการสร้างฐานข้อมูล 25

3.4.2 การส่งข้อมูลเข้าสู่ฐานข้อมูล 25

3.4.3 การพัฒนา Web Application 26

3.4.4 การพัฒนาโปรแกรมไมโครคอนโทรลเลอร์ 27

3.4.5 การสร้างระบบแจ้งเตือน 34

4 ผลการดำเนินงาน 36

4.1 ผลการออกแบบอุปกรณ์เซนเซอร์ 36

4.2 ผลการพัฒนา Web Application 38

5 สรุปผลการวิจัย 41

การอภิปรายผล 41

บรรณานุกรม 43

ภาคผนวก 44

ประวัติผู้วิจัย 46

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

สารบัญภาพ

ภาพ **หน้า**

1	กรอบแนวคิดวิจัย	5
2	เห็ดนางฟ้าภูฐาน	8
3	โรงเรือนระบบ EVAP	10
4	หลักการทำงานระบบทำความเย็นด้วยวิธีการระเหยน้ำ	10
5	หลักการทำงานของ MQTT	15
6	หน้าต่างโปรแกรม Arduino IDE	16
7	การลงพื้นที่สำรวจข้อมูล	19
8	ขาของบอร์ด ESP32	20
9	ขาของบอร์ด Arduino MEGA 2560	21
10	การส่งข้อมูลไปยัง Node-RED เพื่อจัดเก็บฐานข้อมูล	23
11	การรับส่งข้อมูลโดยใช้ MQTT	23
12	การต่อวงจรเซนเซอร์	25
13	การสร้างฐานข้อมูลบน PgAdmin4	26
14	การสร้างฐานข้อมูลบน PgAdmin4	26
15	การสร้างฐานข้อมูลบน PgAdmin4	27
16	URL	27
17	Flow Node-RED	27
18	function node “Parse JSON & Prepare SQL”	28
19	SQL Query	26
20	เชื่อมต่อกับ MQTT broker	29
21	รับข้อความจาก ESP32 แล้วอัปเดตค่าเซ็นเซอร์บน Dashboard	29
22	function ปุ่ม เปิด/ปิด	30
23	การดึงข้อมูลหลังจาก Postgres	31
24	การกำหนดขาเซนเซอร์	32
25	การกำหนดค่าปัจจัยสภาพแวดล้อม	33
26	MQTT และ HTTP	33
27	การใช้คำสั่ง /newbot	34

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

สารบัญภาพ(ต่อ)

28 ตั้งชื่อผู้ใช้ ที่ลงท้ายด้วยคำว่า bot..... 34
29 Token และ ChatID..... 35
30 การกำหนดการแจ้งเตือนและส่งแจ้งเตือน..... 35
31 หน้าตู้ควบคุมอุปกรณ์เซนเซอร์..... 36
32 ภายในตู้ควบคุมอุปกรณ์เซนเซอร์..... 37
33 ผลการเชื่อมต่อ ESP32 ไปยัง Arduino MEGA..... 37
34 แสดงผลเซนเซอร์แบบเรียลไทม์..... 38
35 กราฟข้อมูลย้อนหลัง..... 38
36 ภาพรวมหน้าเว็บ..... 39
37 ผลการจัดเก็บข้อมูลใน PostgresSQL..... 39
38 ผลการแจ้งเตือนผ่าน Telegram..... 40
39 โค้ดที่ใช้ในการพัฒนาระบบ..... 45

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

สารบัญตาราง

ตาราง **หน้า**

1	แสดงเครื่องมือที่ใช้ในการวิจัยของส่วนฮาร์ดแวร์.....	**21**
2	กำหนดปัจจัยสิ่งแวดล้อมเพื่อควบคุมอุปกรณ์	**24**
3	แสดงรายละเอียดที่จัดเก็บข้อมูลเซนเซอร์	**25**
4	ชุดคำสั่ง.....	**32**

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบัน ภาคเกษตรของประเทศไทยกำลังเผชิญกับความท้าทายที่ซับซ้อนและหลากหลาย ซึ่งส่งผลกระทบโดยตรงต่อความมั่นคงทางอาหารและความเป็นอยู่ของเกษตรกร ทั้งนี้ ปัจจัยที่เป็นความท้าทายหลักประกอบด้วย ภัยแล้งที่เกิดบ่อยครั้ง การเปลี่ยนแปลงของสภาพภูมิอากาศ ซึ่งทำให้ฤดูกาลเพาะปลูกไม่แน่นอน และเพิ่มความเสี่ยงต่อผลผลิต ต้นทุนการผลิตที่สูงขึ้นจากราคาปุ๋ย เมล็ดพันธุ์ และพลังงาน รวมถึงปัญหาการขาดแคลนแรงงานในภาคเกษตร สิ่งเหล่านี้ล้วนสร้างแรงกดดันต่อเกษตรกรไทยให้ต้องปรับตัวอย่างต่อเนื่องเพื่อรักษาผลผลิตและรายได้ให้อยู่ในระดับที่ยั่งยืน

ในบรรดาพืชเศรษฐกิจ เห็ดยังคงเป็นพืชที่ได้รับความนิยมอย่างต่อเนื่อง เนื่องจากปลูกง่าย ใช้พื้นที่และต้นทุนน้อย แต่ให้ผลตอบแทนสูง ทั้งยังมีตลาดรองรับทั้งในและต่างประเทศ เห็ดเป็นพืชที่เหมาะสำหรับเกษตรกรรายย่อยและผู้ประกอบการขนาดกลาง เนื่องจากสามารถสร้างรายได้อย่างสม่ำเสมอและเร็วเมื่อเทียบกับพืชชนิดอื่น อย่างไรก็ตาม การเพาะเห็ดให้ได้ผลผลิตที่มีคุณภาพและปลอดภัยนั้นจำเป็นต้องควบคุมปัจจัยสิ่งแวดล้อมหลายด้านอย่างแม่นยำ เช่น อุณหภูมิ ความชื้น ความเข้มของแสง และระดับก๊าซคาร์บอนไดออกไซด์ ซึ่งปัจจัยเหล่านี้มีความสำคัญต่อการเจริญเติบโตและคุณภาพของผลผลิต หากการควบคุมทั้งหมดอาศัยแรงงานคนเพียงอย่างเดียว จะเกิดความเสี่ยงต่อความผิดพลาด การไม่สม่ำเสมอ และความเหนื่อยล้าของผู้ดูแล ทำให้ผลผลิตอาจไม่เป็นไปตามมาตรฐาน เพิ่มโอกาสเกิดโรคหรือเชื้อรา และลดประสิทธิภาพในการผลิต

เพื่อแก้ไขปัญหาเหล่านี้ งานวิจัยนี้มุ่งเน้นการพัฒนาระบบฟาร์มเห็ดอัจฉริยะที่นำเทคโนโลยี Internet of Things (IoT) มาประยุกต์ใช้ร่วมกับบอร์ดควบคุม Arduino Mega2560 และ ESP32 ระบบนี้สามารถตรวจวัดและแสดงผลข้อมูลจากเซ็นเซอร์ต่าง ๆ แบบเรียลไทม์ เพื่อให้เกษตรกรสามารถติดตามสถานะของโรงเรือนได้อย่างต่อเนื่องและมีข้อมูลประกอบการตัดสินใจที่แม่นยำ เซ็นเซอร์ที่ใช้ประกอบด้วย DHT22 สำหรับวัดอุณหภูมิและความชื้น, MH-Z14A สำหรับวัดค่า CO_2 และ LDR สำหรับวัดความเข้มของแสง ข้อมูลจากเซ็นเซอร์ทั้งหมดจะถูกส่งไปยัง Web Dashboard ผ่านโปรโตคอล MQTT ซึ่งช่วยให้ผู้ใช้งานสามารถตรวจสอบค่าปัจจุบันของโรงเรือนได้แบบเรียลไทม์

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

นอกจากการแสดงผลแบบเรียลไทม์ ข้อมูลยังสามารถส่งผ่าน HTTP ไปยัง Node-RED เพื่อต่อไปจัดเก็บใน PostgreSQL Database ทำให้เกษตรกรสามารถเรียกดูข้อมูลย้อนหลัง วิเคราะห์แนวโน้มของสภาพแวดล้อม และวางแผนการเพาะปลูกในอนาคตได้อย่างเป็นระบบ การจัดเก็บข้อมูลอย่างเป็นระบบนี้

ยังช่วยให้สามารถเปรียบเทียบค่าของแต่ละวันหรือแต่ละช่วงเวลา วิเคราะห์ความสัมพันธ์ระหว่างปัจจัยสิ่งแวดล้อม และตรวจสอบประสิทธิภาพของอุปกรณ์ที่ใช้ในโรงเรือน

ความสำคัญของระบบยังอยู่ที่ **การควบคุมอุปกรณ์โดยผู้ใช้งานผ่าน Web Interface** ผู้ใช้สามารถเปิดหรือปิดรีเลย์เพื่อควบคุมอุปกรณ์ต่าง ๆ เช่น พัดลมระบายความร้อน หัวพ่นหมอก ไฟ LED หรือปั๊มน้ำตามค่าที่ตรวจวัดได้จากเซ็นเซอร์ แม้ว่าระบบจะไม่ทำงานแบบอัตโนมัติ แต่ยังช่วยให้ผู้ใช้งานสามารถควบคุมโรงเรือนได้อย่างสะดวกและแม่นยำ นอกจากนี้ ระบบยังสามารถตั้งค่าขีดจำกัดของปัจจัยสิ่งแวดล้อม เมื่อค่าที่วัดได้ผิดปกติ เช่น อุณหภูมิสูง/ต่ำ ความชื้นสูง/ต่ำ หรือ CO_2 สูง ระบบจะส่งแจ้งเตือนผ่าน Telegram หรือช่องทางออนไลน์อื่น ๆ ทำให้ผู้ใช้งานสามารถปรับการทำงานของอุปกรณ์ได้ทันเวลา ลดความเสี่ยงต่อความเสียหายของผลผลิต และเพิ่มความมั่นใจในการจัดการโรงเรือน

Web Application ที่พัฒนาขึ้นโดยใช้ PHP, HTML, JavaScript ร่วมกับ PostgreSQL Database ช่วยให้เกษตรกรสามารถเข้าถึงข้อมูลทั้งแบบเรียลไทม์และย้อนหลังจากทุกที่ โดยไม่ต้องพึ่งพาแอปพลิเคชันเฉพาะ ข้อมูลยังสามารถเชื่อมโยงกับระบบ GIS เพื่อบันทึกตำแหน่งของโรงเรือนและวิเคราะห์เชิงพื้นที่ ทำให้สามารถติดตามและบริหารจัดการโรงเรือนในระดับชุมชน จังหวัด หรือประเทศ ซึ่งเป็นพื้นฐานสำคัญสำหรับการวางแผนเชิงภูมิศาสตร์และการจัดการเกษตรกรรมอย่างยั่งยืน

ระบบนี้ยังออกแบบให้สามารถใช้งานได้ในพื้นที่ห่างไกลหรือไม่มีไฟฟ้าจากการไฟฟ้า โดยใช้พลังงานแสงอาทิตย์เป็นแหล่งจ่ายไฟฟ้าหลักหรือสำรอง เพิ่มความยืดหยุ่นและต่อเนื่องในการทำงานของระบบ นอกจากนี้ ระบบยังแสดงผลข้อมูลบนจอ TFT ผ่าน Arduino Mega เพื่อให้ผู้ดูแลหน้างานสามารถตรวจสอบค่าของเซ็นเซอร์ได้โดยตรง โดยไม่ต้องพึ่งพา Web Dashboard เพียงอย่างเดียว

ด้วยเหตุนี้ งานวิจัยนี้จึงมีความสำคัญอย่างยิ่งต่อการพัฒนาฟาร์มเห็ดอัจฉริยะที่ **สามารถตรวจวัดและแสดงผลข้อมูลได้แบบเรียลไทม์** ช่วยให้ผู้ใช้งานสามารถตัดสินใจควบคุมอุปกรณ์ได้อย่างแม่นยำ ลดภาระแรงงานคน เพิ่มความสม่ำเสมอและคุณภาพของผลผลิต สร้างความมั่นคงทางรายได้และอาหารให้กับเกษตรกร และเป็นต้นแบบสำหรับการบริหารจัดการฟาร์มแบบอัจฉริยะในระดับชุมชน จังหวัด หรือประเทศในอนาคต งานวิจัยนี้ยังเปิดโอกาสให้สามารถประยุกต์ใช้เทคโนโลยี IoT ระบบจัดเก็บและวิเคราะห์ข้อมูล และระบบ GIS กับเกษตรกรรมรูปแบบอื่น ๆ ที่ต้องการความแม่นยำและประสิทธิภาพสูง ซึ่งจะช่วยยกระดับความสามารถการแข่งขันของภาคเกษตรไทยให้เทียบเท่ามาตรฐานสากลอย่างยั่งยืน

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

1.2 วัตถุประสงค์

เพื่อออกแบบและพัฒนา **ระบบฟาร์มเห็ดอัจฉริยะ** ที่สามารถควบคุมปัจจัยสำคัญของโรงเรือนได้แบบอัตโนมัติ เก็บข้อมูลเข้าสู่ฐานข้อมูล และแสดงผลข้อมูลเรียลไทม์ผ่าน Web Application

1.3 ขอบเขตของงานวิจัย

1.3.1 ขอบเขตด้านพื้นที่

คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม มหาวิทยาลัยนเรศวร จังหวัด พิษณุโลก

1.3.2 ขอบเขตด้านการศึกษา

ระบบฟาร์มเห็ดอัจฉริยะ ที่สามารถควบคุมปัจจัยสำคัญของโรงเรือนได้ เก็บข้อมูลเข้าสู่ฐานข้อมูล และนำเสนอผลผ่าน Dashboard

1.4 นิยามศัพท์เฉพาะ

เห็ดนางฟ้า (Oyster Mushroom) คือเห็ดชนิดหนึ่งในสกุล *Pleurotus* ที่นิยมเพาะเลี้ยงในประเทศไทย มีลักษณะหมวกกลม แบน ก้านสั้น เจริญเติบโตได้ดีในสภาพแวดล้อมที่มีความชื้นสูงและอุณหภูมิอุ่น เหมาะสำหรับการเพาะในโรงเรือนแบบควบคุม

ESP32 ไมโครคอนโทรลเลอร์ที่มี WiFi และ Bluetooth ในตัว ใช้สำหรับประมวลผลและส่งข้อมูลจากเซ็นเซอร์ไปยังระบบเว็บเซิร์ฟเวอร์หรือฐานข้อมูลแบบเรียลไทม์ เหมาะสำหรับงานด้าน IoT

Arduino MEGA 2560 บอร์ดไมโครคอนโทรลเลอร์ที่มีขาพอร์ตอินพุต–เอาต์พุตจำนวนมาก เหมาะสำหรับงานที่ต้องใช้เซ็นเซอร์และอุปกรณ์หลายชนิด ใช้ประมวลผลและควบคุมอุปกรณ์ภายในระบบ

TFT Monitor หน้าจอแสดงผลแบบสี (Thin Film Transistor) ใช้แสดงค่าข้อมูลจากเซ็นเซอร์และสถานะของอุปกรณ์ภายในระบบแบบเรียลไทม์

เซ็นเซอร์ (Sensor) อุปกรณ์ที่ใช้ตรวจวัดค่าทางกายภาพ เช่น อุณหภูมิ ความชื้น แสง และก๊าซ แล้วส่งข้อมูลไปยังระบบควบคุม ตัวอย่างเซ็นเซอร์ที่ใช้ในโครงงาน ได้แก่ DHT22, MH-Z14A, LDR

อินเทอร์เน็ตของสรรพสิ่ง (IoT – Internet of Things) แนวคิดที่อุปกรณ์ต่าง ๆ สามารถเชื่อมต่ออินเทอร์เน็ตและสื่อสารกันได้ เพื่อทำให้ระบบต่าง ๆ ทำงานร่วมกันอย่างอัตโนมัติ

PHP (Hypertext Preprocessor) ภาษาสคริปต์ที่ทำงานฝั่งเซิร์ฟเวอร์ ใช้สำหรับสร้างระบบจัดการข้อมูล เช่น รับข้อมูลจากเซ็นเซอร์ เก็บเข้าสู่ฐานข้อมูล และสร้าง API สำหรับเชื่อมต่อกับเว็บแอปพลิเคชัน

อรสิรี มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

HTML (Hypertext Markup Language) ภาษามาร์กอัปที่ใช้สำหรับกำหนดโครงสร้างของหน้าเว็บ เช่น การจัดวางข้อความ ปุ่ม และแบบฟอร์ม เพื่อให้ผู้ใช้งานสามารถโต้ตอบกับระบบได้ผ่านเบราว์เซอร์

JavaScript ภาษาสคริปต์ที่ทำงานฝั่งผู้ใช้ (Client-Side) ใช้ควบคุมการแสดงผลข้อมูลแบบโต้ตอบ เช่น การโหลดข้อมูลจาก API โดยไม่ต้องรีเฟรชหน้าเว็บ และการอัปเดตกราฟหรือค่าจากเซ็นเซอร์แบบเรียลไทม์

PostgreSQL ระบบจัดการฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database) แบบ Open Source ใช้เก็บข้อมูลจากเซ็นเซอร์เพื่อใช้ในการวิเคราะห์ย้อนหลังและแสดงผลใน Dashboard

รีเลย์ (Relay) อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็น “สวิตช์ไฟฟ้า” ที่ควบคุมโดยไมโครคอนโทรลเลอร์ ใช้สำหรับเปิด–ปิดพัดลม ปั๊มน้ำ หรือหลอดไฟในระบบอัตโนมัติ

Web Application แอปพลิเคชันที่ใช้งานผ่านเว็บเบราว์เซอร์ โดยไม่ต้องติดตั้งเพิ่มเติม โดยระบบของงานวิจัยนี้ใช้เพื่อควบคุม ติดตาม และแสดงผลข้อมูลของโรงเรือนเพาะเห็ด

Dashboard หน้าจอแสดงผลภาพรวมของข้อมูลที่สำคัญ เช่น กราฟอุณหภูมิ ความชื้น ค่าก๊าซ หรือสถานะของอุปกรณ์ต่าง ๆ เพื่อให้ผู้ใช้สามารถเข้าใจและติดตามระบบได้ง่ายขึ้น

พลังงานแสงอาทิตย์ (Solar Power) การนำพลังงานจากแสงอาทิตย์มาแปลงเป็นพลังงานไฟฟ้า โดยใช้แผง Solar Cell และแบตเตอรี่ เพื่อจ่ายไฟให้กับระบบควบคุมและเซ็นเซอร์ภายในโรงเรือน

ระบบสารสนเทศภูมิศาสตร์ (GIS – Geographic Information System) ระบบที่ใช้ในการจัดการวิเคราะห์ และแสดงข้อมูลเชิงพื้นที่ เช่น การแสดงตำแหน่งของโรงเรือนเห็ดบนแผนที่ การวิเคราะห์ความเหมาะสมของพื้นที่เพาะเห็ด และการติดตามฟาร์มในหลายตำแหน่ง

Node-RED เครื่องมือพัฒนาแบบลากวาง (Flow-Based Programming) ที่ใช้เชื่อมต่ออุปกรณ์ IoT เซ็นเซอร์ และฐานข้อมูลเข้าด้วยกัน โดยทำงานบน Node.js สามารถสร้างและควบคุมระบบอัตโนมัติ รวมถึงแสดงผลข้อมูลแบบเรียลไทม์ได้อย่างสะดวกในรูปแบบ Dashboard

HTTP(Hypertext Transfer Protocol) โปรโตคอลมาตรฐานสำหรับการสื่อสารข้อมูลระหว่างเว็บเบราว์เซอร์และเว็บเซิร์ฟเวอร์ ใช้ในการส่งข้อมูลจาก ESP32 ไปยัง Web Application และฐานข้อมูล

MQTT(Message Queuing Telemetry Transport) โปรโตคอลสื่อสารแบบ Publish–Subscribe ที่ใช้ส่งข้อมูลระหว่างอุปกรณ์ IoT ผ่านเครือข่ายอินเทอร์เน็ต มีน้ำหนักเบา ใช้แบนด์วิธต่ำ เหมาะสำหรับระบบที่ต้องส่งข้อมูลแบบเรียลไทม์จากเซ็นเซอร์ไปยังเซิร์ฟเวอร์หรือ Dashboard

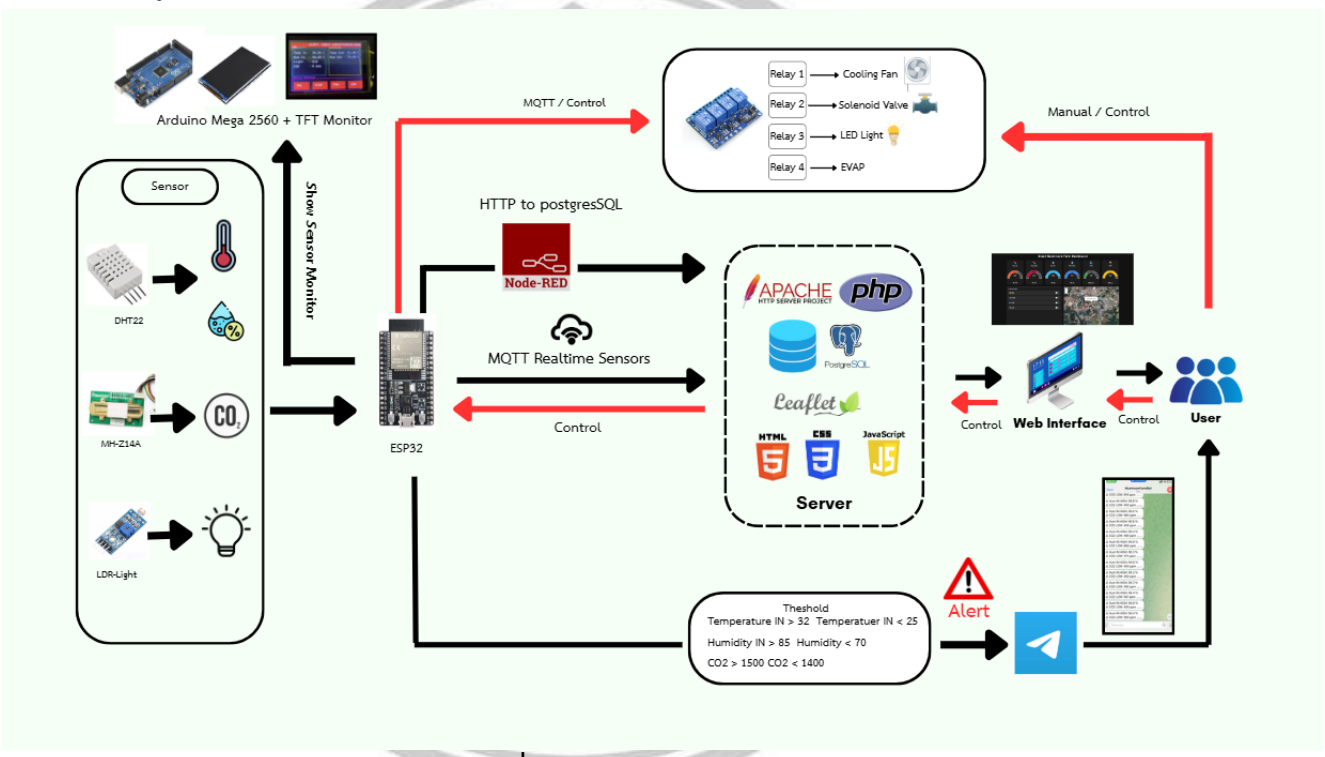
ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

1.5 สมมติฐานงานวิจัย

ระบบควบคุมสภาพแวดล้อมที่พัฒนาขึ้นจะสามารถควบคุมอุณหภูมิ ความชื้น แสง และระดับก๊าซคาร์บอนไดออกไซด์ให้อยู่ในช่วงที่เหมาะสมต่อการเพาะเห็ดได้อย่างมีประสิทธิภาพ และการแสดงผลข้อมูลผ่าน Web Application จะช่วยให้สามารถติดตามและวิเคราะห์แนวโน้มของสภาพแวดล้อมได้แม่นยำมากขึ้น ส่งผลให้ผลผลิตมีคุณภาพและปริมาณที่ดีขึ้นในรอบถัดไป

1.6 กรอบแนวคิดงานวิจัย

ระบบนี้ออกแบบให้ ESP32 เป็นสมองกลางในการจัดการข้อมูล รับค่าจากเซ็นเซอร์ ส่งข้อมูลไปยัง Web Dashboard พร้อมแจ้งเตือนและควบคุมอุปกรณ์ ส่วน Node-RED และ PostgreSQL ทำหน้าที่เก็บข้อมูลย้อนหลังเพื่อวิเคราะห์แนวโน้ม และ Web Interface เป็นช่องทางให้ผู้ใช้งานสามารถโต้ตอบ ดูข้อมูลเรียลไทม์ ควบคุมอุปกรณ์ และรับแจ้งเตือน ในขณะที่ Arduino Mega + TFT ทำหน้าที่แสดงข้อมูลบนหน้าจอในฟาร์มสำหรับผู้ดูแลหน้างานโดยตรง ทำให้ภาพรวมของระบบทำงานเชื่อมโยงกันอย่างราบรื่นและมีประสิทธิภาพสูงสุด.



ภาพที่ 1 กรอบแนวคิดวิจัย

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

จากการทบทวนเอกสาร แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง พบว่ามีข้อมูลและเทคนิคต่าง ๆ ที่สามารถนำมาเป็นแนวคิดในการประยุกต์ใช้เทคโนโลยีสำหรับการควบคุมสภาพแวดล้อมในโรงเรือนเพาะเห็ดได้อย่างมีประสิทธิภาพ ระบบเหล่านี้มักเน้นการตรวจวัดค่าปัจจัยสิ่งแวดล้อมที่สำคัญ เช่น อุณหภูมิ ความชื้น ความเข้มแสง และระดับก๊าซคาร์บอนไดออกไซด์ โดยใช้เซ็นเซอร์ที่สามารถเชื่อมต่อผ่านระบบเครือข่ายไร้สาย (Wireless Network) เพื่อส่งข้อมูลแบบเรียลไทม์ไปยังศูนย์ควบคุมหรือแพลตฟอร์ม Web Dashboard นอกจากนี้ ยังมีการศึกษาวิธีการจัดเก็บและวิเคราะห์ข้อมูลย้อนหลัง การแจ้งเตือนผู้ใช้งานเมื่อค่าที่วัดได้ผิดปกติ และการพัฒนาระบบ Web Application สำหรับการแสดงผลข้อมูลและควบคุมอุปกรณ์

นอกจากเทคโนโลยีและอุปกรณ์แล้วงานวิจัยก่อนหน้านี้ยังเน้นการศึกษาปัจจัยสำคัญที่มีผลต่อการเจริญเติบโตของเห็ด เช่น อุณหภูมิ ความชื้น แสง และระดับ CO_2 ซึ่งข้อมูลเหล่านี้ช่วยให้การออกแบบระบบสามารถตอบสนองต่อความต้องการของโรงเรือนและการเพาะปลูกได้อย่างเหมาะสม บทที่ 2 ของงานวิจัยนี้จึงนำเสนอแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง เพื่อเป็นพื้นฐานสำหรับการออกแบบและพัฒนาระบบฟาร์มเห็ดอัจฉริยะ โดยจัดเป็นหัวข้อสำคัญดังต่อไปนี้

2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1.1เทคโนโลยี Internet of Things (IoT) ในภาคเกษตร

2.1.2 ความรู้พื้นฐานเกี่ยวกับการเพาะเห็ด

2.1.3 ปัจจัยที่มีผลต่อการเจริญเติบโตของเห็ด

2.1.4 โรงเรือนระบบอีแวป (EVAP) สำเร็จรูป ขนาด 1.6 x 4 เมตร

2.1.5 ซอฟต์แวร์ที่เกี่ยวข้องกับการพัฒนาระบบ

2.2 เอกสารงานวิจัยที่เกี่ยวข้อง

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1.1เทคโนโลยี Internet of Things (IoT) ในภาคเกษตร

แนวคิดของ Internet of Things (IoT)

แนวคิดของ Internet of Things (IoT) หรือ “อินเทอร์เน็ตของสรรพสิ่ง” หมายถึงระบบที่เชื่อมโยงอุปกรณ์ต่าง ๆ เข้ากับเครือข่ายอินเทอร์เน็ตเพื่อให้สามารถรับส่งข้อมูลระหว่างกันได้โดยอัตโนมัติ อุปกรณ์เหล่านี้อาจเป็นเซ็นเซอร์ เครื่องจักร หรืออุปกรณ์อิเล็กทรอนิกส์ขนาดเล็กที่สามารถตรวจวัดค่าต่าง ๆ และส่งข้อมูลไปยังระบบประมวลผลกลางได้ การทำงานของ IoT เป็นการเชื่อมโยงระหว่างโลกกายภาพกับโลกดิจิทัล เพื่อให้เกิดการเก็บข้อมูล วิเคราะห์ และควบคุมการทำงานได้อย่างชาญฉลาด โดยมีองค์ประกอบสำคัญ 4 ส่วน ได้แก่ Sensor หรือ Device ทำหน้าที่ตรวจวัดข้อมูลจากสิ่งแวดล้อม เช่น อุณหภูมิ ความชื้น แสง หรือก๊าซต่าง ๆ Connectivity ระบบการเชื่อมต่อข้อมูลผ่านเครือข่าย เช่น Wi-Fi, Bluetooth, LoRa หรือ MQTT Cloud หรือ Server ทำหน้าที่จัดเก็บและประมวลผลข้อมูลจากอุปกรณ์ต่าง ๆ เพื่อให้สามารถเรียกใช้งานได้ทุกที่ทุกเวลา และ Application หรือระบบแสดงผลข้อมูลที่ช่วยให้ผู้ใช้งานสามารถตรวจสอบ ควบคุม และวิเคราะห์ข้อมูลได้แบบเรียลไทม์ หลักการทำงานของระบบ IoT เริ่มจากเซ็นเซอร์ตรวจวัดค่าจากสิ่งแวดล้อมจริงแล้วส่งข้อมูลผ่านเครือข่ายไปยังระบบจัดเก็บหรือประมวลผล ซึ่งจะนำข้อมูลนั้นมาแสดงผลในรูปแบบที่เข้าใจง่าย เช่น Dashboard หรือ Web Application เพื่อให้ผู้ใช้สามารถสั่งควบคุมอุปกรณ์ต่าง ๆ กลับไปยังหน่วยควบคุมได้โดยตรง จึงช่วยให้กระบวนการทำงานเป็นอัตโนมัติ ลดความผิดพลาด และเพิ่มประสิทธิภาพของระบบโดยรวม ระบบ IoT จึงเป็นเทคโนโลยีสำคัญที่ถูกนำมาใช้ในหลายสาขา เช่น บ้านอัจฉริยะ (Smart Home) อุตสาหกรรม (Industry 4.0) การติดตามสุขภาพ (Health Monitoring) และการเกษตรอัจฉริยะ (Smart Farming)

แนวคิดของ Smart Farming

การเกษตรอัจฉริยะ หมายถึงการนำเทคโนโลยีสมัยใหม่ เช่น ระบบเซ็นเซอร์ (Sensor) Internet of Things (IoT) ปัญญาประดิษฐ์ (AI) ระบบสารสนเทศภูมิศาสตร์ (GIS) และการวิเคราะห์ข้อมูล (Data Analytics) มาช่วยในการจัดการกระบวนการทางการเกษตร เพื่อเพิ่มประสิทธิภาพในการผลิต ลดต้นทุน และเพิ่มความแม่นยำในการจัดการทรัพยากร แนวคิดนี้มุ่งเน้นให้ระบบเกษตรสามารถตรวจสอบและตอบสนองต่อการเปลี่ยนแปลงของสิ่งแวดล้อมได้โดยอัตโนมัติ เช่น ระบบควบคุมอุณหภูมิ ความชื้น แสง หรือการรดน้ำพืชโดยใช้ข้อมูลจากเซ็นเซอร์แบบเรียลไทม์ ทำให้ผู้ใช้งานสามารถจัดการฟาร์มได้จากระยะไกลและลดความผิดพลาดจากมนุษย์ Smart Farming ยังช่วยให้สามารถเก็บข้อมูลและวิเคราะห์แนวโน้มย้อนหลัง เช่น อุณหภูมิ ความชื้น หรือผลผลิต เพื่อใช้วางแผนการเพาะปลูกในรอบต่อไปได้อย่างแม่นยำ การประยุกต์ใช้ Smart Farming กับการเพาะเห็ดนางฟ้าจะช่วยให้สามารถควบคุมปัจจัยสำคัญ เช่น อุณหภูมิ ความชื้น แสง และระดับก๊าซคาร์บอนไดออกไซด์ให้อยู่ในระดับที่เหมาะสมต่อการเจริญเติบโตของเห็ด ลดภาระการเฝ้าดูแลของผู้เพาะ และเพิ่มประสิทธิภาพในการผลิตให้ได้ผลผลิตที่มีคุณภาพสูงและสม่ำเสมอ ดังนั้น Smart Farming

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

จึงถือเป็นแนวทางสำคัญในการยกระดับภาคการเกษตรไทยให้เข้าสู่ยุคดิจิทัลที่มีความยั่งยืนและใช้ทรัพยากรอย่างคุ้มค่า.

2.1.2 ความรู้พื้นฐานเกี่ยวกับการเพาะเห็ด

เห็ด (Mushroom) คือสิ่งมีชีวิตชนิดหนึ่งที่อยู่ในอาณาจักรฟังไจ (Fungi Kingdom) ซึ่งไม่มีคลอโรฟิลล์ (Chlorophyll) เหมือนพืชทั่วไป จึงไม่สามารถสังเคราะห์แสงได้ แต่ดำรงชีวิตโดยการดูดซึมสารอาหารจากอินทรียวัตถุที่เน่าเปื่อยในธรรมชาติ เช่น ฟางข้าว ขี้เลื่อย หรือเศษพืชอื่น ๆ เห็ดจัดเป็นสิ่งมีชีวิตที่มีโครงสร้างเรียบง่าย ประกอบด้วยเส้นใย (Mycelium) ที่เจริญอยู่ในวัสดุเพาะ และเมื่อมีสภาพแวดล้อมที่เหมาะสม เส้นใยเหล่านี้จะรวมตัวกันเป็นดอกเห็ด (Fruiting Body) ซึ่งเป็นส่วนที่ใช้บริโภค

ชนิดของเห็ดที่จะเพาะปลูก

ภาพที่ 2 เห็ดนางฟ้าภูฐาน

เห็ดนางฟ้าภูฐาน (Bhutan Oyster Mushroom, *Pleurotus sajor-caju*) เป็นเห็ดเศรษฐกิจที่นิยมเพาะเลี้ยงอย่างแพร่หลายในประเทศไทย เนื่องจากเพาะง่าย เจริญเติบโตได้ดีในสภาพภูมิอากาศเขตร้อน และให้ผลผลิตสม่ำเสมอตลอดปี เห็ดชนิดนี้มีลักษณะดอกค่อนข้างใหญ่ เนื้อแน่น มีกลิ่นหอมเฉพาะตัว และมีรสชาติดี จึงเป็นที่นิยมบริโภคทั้งในครัวเรือนและในเชิงพาณิชย์

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

ลักษณะทั่วไปของเห็ดนางฟ้าภูฐาน คือ ดอกเห็ดมีรูปร่างคล้ายพัด สีขาวนวลถึงน้ำตาลอ่อน เส้นใยของเห็ด (Mycelium) มีการเจริญเติบโตเร็ว สามารถยึดเกาะวัสดุเพาะได้ดี วัสดุเพาะที่นิยมใช้คือขี้เลื่อยไม้เนื้ออ่อน ผสมรำละเอียด และแคลเซียมคาร์บอเนต เพื่อเพิ่มสารอาหารให้เส้นใยเจริญเติบโตได้สมบูรณ์

2.1.3 ปัจจัยที่มีผลต่อการเจริญเติบโตของเห็ด

การเจริญเติบโตของเห็ดขึ้นอยู่กับปัจจัยแวดล้อมหลายประการ ซึ่งมีผลโดยตรงต่อการเจริญของเส้นใย การสร้างดอก และคุณภาพของผลผลิต ปัจจัยหลักที่ต้องควบคุมอย่างเหมาะสมได้แก่ อุณหภูมิ ความชื้น ความเข้มของแสง และปริมาณก๊าซคาร์บอนไดออกไซด์ รวมถึงวัสดุเพาะที่ใช้ในการเพาะเลี้ยง

อุณหภูมิ (Temperature)

อุณหภูมิเป็นปัจจัยที่สำคัญที่สุดในการเจริญของเห็ด เพราะมีผลโดยตรงต่อการแบ่งเซลล์ของเส้นใยและการออกดอก โดยทั่วไปเห็ดแต่ละชนิดจะมีช่วงอุณหภูมิที่เหมาะสมแตกต่างกัน สำหรับเห็ดนางฟ้าภูฐานอุณหภูมิที่เหมาะสมต่อการเจริญของเส้นใยอยู่ที่ประมาณ 25–30°C และอุณหภูมิที่เหมาะสมสำหรับการออกดอกอยู่ในช่วง 28–32°C หากอุณหภูมิต่ำเกินไป เส้นใยจะเจริญช้า ส่วนถ้าอุณหภูมิสูงเกินไปจะทำให้เส้นใยตายหรือดอกเห็ดแห้ง เหี่ยว และมีลักษณะผิดปกติ ดังนั้น การควบคุมอุณหภูมิให้คงที่ภายในโรงเรือนจึงเป็นสิ่งจำเป็น โดยในระบบฟาร์มเห็ดอัจฉริยะจะใช้ พัดลมระบายอากาศ (Cooling Fan) หรือ ระบบพ่นหมอก (Fogging System) เพื่อปรับอุณหภูมิให้อยู่ในระดับที่เหมาะสม

ความชื้นสัมพัทธ์ในอากาศ (Relative Humidity)

ความชื้นเป็นอีกปัจจัยหนึ่งที่มีผลต่อทั้งการเจริญของเส้นใยและการออกดอกของเห็ด เส้นใยของเห็ดต้องการความชื้นในระดับสูงเพื่อให้สามารถดูดซับน้ำและสารอาหารจากวัสดุเพาะได้ดี โดยทั่วไปความชื้นสัมพัทธ์ที่เหมาะสมสำหรับเห็ดนางฟ้าภูฐานอยู่ในช่วง 70–90% RH โดยเฉพาะในระยะออกดอกควรมีความชื้นสูงอย่างต่อเนื่อง หากความชื้นในอากาศต่ำเกินไป ดอกเห็ดจะแห้ง แตก หรือไม่เจริญ ส่วนถ้ามีความชื้นสูงเกินไปอาจทำให้เกิดเชื้อราปนเปื้อน ดังนั้นระบบควบคุมความชื้นภายในโรงเรือนจึงมีความสำคัญอย่างยิ่ง โดยในระบบอัจฉริยะจะใช้ เซ็นเซอร์ DHT22 ตรวจวัดค่าความชื้นและสั่งการ โซลินอยด์วาล์ว (Solenoid Valve) ให้เปิดน้ำเมื่อค่าความชื้นลดลงต่ำกว่าค่ากำหนด

ความเข้มของแสง (Light Intensity)

แม้เห็ดจะไม่ต้องการแสงในการสังเคราะห์อาหาร แต่แสงมีผลสำคัญในกระบวนการสร้างดอก (Primordia Formation) และการพัฒนาโครงสร้างของดอกเห็ด แสงที่เหมาะสมช่วยให้หมวกเห็ดมีขนาดสมบูรณ์และสีสวยสม่ำเสมอ ความเข้มของแสงที่เหมาะสมสำหรับเห็ดนางฟ้าอยู่ที่ประมาณ 50–100 Lux หากแสงแรงเกินไปอาจทำให้ดอกเห็ดไหม้หรือแห้ง แต่ถ้ามืดเกินไปจะทำให้ดอกเห็ดมีลำต้นยาวและหมวกเล็กผิดรูป ระบบฟาร์มเห็ดอัจฉริยะจะใช้ LDR Sensor (Light Dependent Resistor) ตรวจวัดความเข้มแสง และควบคุม หลอดไฟ LED ให้เปิดหรือปิด

ปริมาณก๊าซคาร์บอนไดออกไซด์ (Carbon Dioxide: CO_2)

เห็ดเป็นสิ่งมีชีวิตที่หายใจและปล่อยก๊าซคาร์บอนไดออกไซด์ออกมาในระหว่างการเจริญของเส้นใยและการออกดอก ระดับ CO_2 ที่สูงเกินไปจะส่งผลให้ดอกเห็ดมีลักษณะก้านยาว หมวกเล็ก หรือผิดรูป ส่วนระดับที่ต่ำเกินไปจะทำให้ดอกเห็ดมีการเจริญไม่สมบูรณ์ ปริมาณก๊าซคาร์บอนไดออกไซด์ที่เหมาะสมต่อการออกดอกของเห็ดนางฟ้าภูฐานอยู่ในช่วง 1,500–2,000 ppm ดังนั้น ระบบควบคุมภายในโรงเรือนจึงต้องมีการระบายอากาศที่เหมาะสม โดยใช้ พัดลมระบายอากาศ (Exhaust Fan) ที่สั่งงานอัตโนมัติจากค่าที่วัดได้ด้วยเซ็นเซอร์ MH-Z14A เพื่อรักษาระดับ CO_2 ให้อยู่ในช่วงที่กำหนด

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright© by Naresuan University
All rights reserved

การถ่ายเทอากาศ (Ventilation)

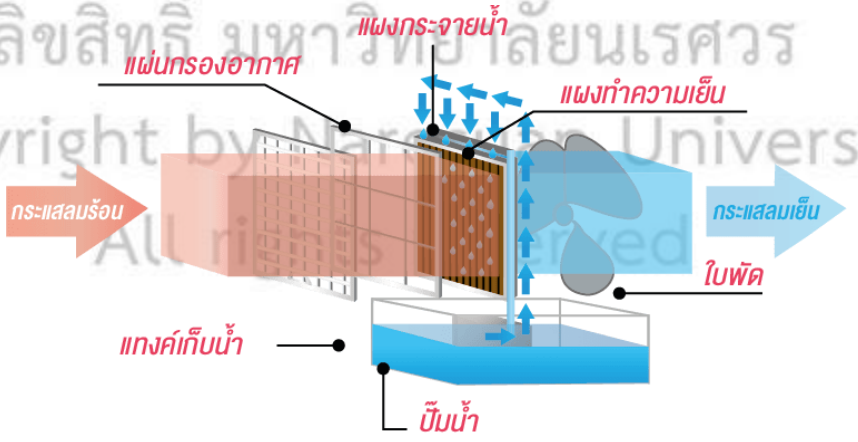
การระบายอากาศมีความสำคัญต่อการควบคุมอุณหภูมิ ความชื้น และระดับก๊าซ CO_2 ภายในโรงเรือน การถ่ายเทอากาศที่ดีช่วยลดความร้อนและป้องกันการสะสมของก๊าซที่ไม่พึงประสงค์ ในระบบฟาร์มเห็ดอัจฉริยะ การระบายอากาศจะถูกควบคุมผ่าน รีเลย์ (Relay Module) ที่สั่งงานพัดลมเมื่อค่าก๊าซ CO_2 หรืออุณหภูมิเกินค่าที่ตั้งไว้

2.1.4 โรงเรือนระบบอีแวป (EVAP) สำเร็จรูป ขนาด 1.6 x 4 เมตร

ภาพที่ 3 โรงเรือนระบบ EVAP

EVAP ย่อมาจาก Evaporative cooling system หมายถึง ระบบทำความเย็นแบบที่ใช้การระเหยของน้ำช่วยในการทำความเย็นเป็นมิตรกับสิ่งแวดล้อม ระบบนี้นำมาทดแทน ระบบแอร์ ช่วยประหยัดค่าไฟไม่น้อยกว่า 80% สามารถ ควบคุม เรื่องความชื้น ให้เหมาะกับ ความต้องการ ในรูปแบบ การใช้งาน ที่ต่างๆ ได้

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved



ภาพที่ 4 หลักการทำงานระบบทำความเย็นด้วยวิธีการระเหยน้ำ

https://www.mechanika.co.th/evaporative-cooling-system/

หลักการทำงานของ EVAP ก็คือ ปล่อยกระแสลม ไหลผ่านตัวกลางที่มีน้ำไหลผ่าน (cooling pad) และการที่อากาศซึ่งมีอุณหภูมิสูงกว่าไหลผ่านน้ำจะทำให้น้ำระเหยกลายเป็นไอ ซึ่งจะเป็นการดึงเอาความร้อนของอากาศออก ทำให้อุณหภูมิของอากาศลดลง

2.1.5 ซอฟต์แวร์ที่เกี่ยวข้องกับการพัฒนาระบบ

การพัฒนาระบบควบคุมสภาพแวดล้อมโรงเรือนเพาะเห็ดอัจฉริยะในครั้งนี้ จำเป็นต้องอาศัยซอฟต์แวร์หลายประเภทที่ทำงานประสานกันในแต่ละชั้นของระบบ ตั้งแต่การเก็บข้อมูล (Data Acquisition) การประมวลผลและควบคุม (Processing & Control) การจัดเก็บข้อมูล (Data Storage) จนถึงการแสดงผล (Visualization) เพื่อให้ระบบสามารถทำงานแบบอัตโนมัติและสื่อสารกันได้อย่างมีประสิทธิภาพ ซอฟต์แวร์ที่ใช้ในการพัฒนามีพื้นฐานทางทฤษฎีดังนี้

Apache

คือซอฟต์แวร์ที่ใช้ทำหน้าที่เป็น Web Server ซึ่งเป็นองค์ประกอบสำคัญในระบบ Web Application โดยมีหน้าที่จัดการคำร้องขอ (Request) จากผู้ใช้ หรือจากอุปกรณ์ IoT แล้วส่งข้อมูลตอบกลับ (Response) ผ่านโปรโตคอล HTTP หรือ HTTPS Apache เป็นซอฟต์แวร์โอเพนซอร์สที่อยู่ภายใต้โครงการของ Apache Software Foundation ได้รับความนิยมสูงเนื่องจากมีเสถียรภาพสูง ปลอดภัย และรองรับการทำงานร่วมกับภาษาสคริปต์ต่าง ๆ เช่น PHP, Python และ Perl รวมถึงสามารถปรับแต่งค่าการทำงาน (Configuration) ได้ยืดหยุ่นผ่านไฟล์ .htaccess ในการวิจัยนี้ Apache ทำหน้าที่เป็น ศูนย์กลางการรับ–ส่งข้อมูลระหว่าง ESP32 และระบบฐานข้อมูล โดยข้อมูลจากเซ็นเซอร์จะถูกส่งมายัง Apache ผ่านโปรโตคอล HTTP จากนั้น Apache จะส่งต่อข้อมูลให้ PHP ประมวลผล และบันทึกลงฐานข้อมูล PostgreSQL รวมถึงทำหน้าที่ให้บริการหน้าเว็บแก่ผู้ใช้ เพื่อให้ผู้ใช้สามารถเรียกดูและควบคุมอุปกรณ์ได้ผ่าน Dashboard แบบเรียลไทม์
Apache ทำงานในรูปแบบของ Client–Server Architecture โดยที่

Client คือ Web Browser หรืออุปกรณ์ IoT ที่ร้องขอข้อมูล

Server คือเครื่องคอมพิวเตอร์ที่ติดตั้ง Apache และรอให้บริการข้อมูล

PHP (Hypertext Preprocessor)

เป็นภาษาสคริปต์ที่ทำงานบนฝั่งเซิร์ฟเวอร์ ซึ่งมีหน้าที่ในการประมวลผลข้อมูลก่อนจะถูกส่งต่อไปยังฝั่งผู้ใช้ โดย PHP ถูกออกแบบมาเพื่อสร้างเว็บไซต์แบบโต้ตอบได้ หรือที่เรียกว่า Dynamic Web Page อีกทั้งยังสามารถเชื่อมต่อฐานข้อมูลได้อย่างมีประสิทธิภาพ จึงได้รับความนิยมอย่างมากในด้านการพัฒนาเว็บไซต์ เนื่องจากเป็นภาษาที่ใช้งานง่าย มีโครงสร้างที่เข้าใจได้ไม่ยาก และสามารถทำงานร่วมกับภาษา HTML ได้โดยตรง นอกจากนี้ PHP ยังรองรับการเชื่อมต่อฐานข้อมูลหลายประเภท เช่น MySQL, PostgreSQL และ Oracle รวมถึงมีชุมชนผู้ใช้ขนาดใหญ่ที่ช่วยกันพัฒนาและจัดทำเอกสารอธิบายไว้อย่างครอบคลุม ในงานวิจัยนี้ PHP ทำหน้าที่เป็นตัวกลางระหว่าง Apache Server และฐานข้อมูล PostgreSQL โดยทำหน้าที่รับข้อมูลที่ส่งมาจากอุปกรณ์ ESP32 ผ่านโปรโตคอล HTTP ในรูปแบบของข้อมูล JSON จากนั้น PHP จะทำการประมวลผลและแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมก่อนบันทึกลงในฐานข้อมูล พร้อมทั้งจัดเตรียม API สำหรับใช้ใน

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright© by Naresuan University
All rights reserved

การดึงข้อมูลกลับมาแสดงผลบนหน้า Dashboard เพื่อให้ผู้ใช้สามารถดูสถานะการทำงานของโรงเรือนได้แบบเรียลไทม์ นอกจากนี้ PHP ยังถูกใช้ในการเขียนโปรแกรมฝั่งเซิร์ฟเวอร์เพื่อควบคุมอุปกรณ์ต่าง ๆ ภายในระบบ เช่น การสั่งเปิดหรือปิดรีเลย์ผ่านการคลิกปุ่มบนหน้า Web Interface ซึ่งช่วยให้ผู้ใช้งานสามารถควบคุมการทำงานของโรงเรือนเพาะเห็ดได้จากระยะไกลผ่านเครือข่ายอินเทอร์เน็ตอย่างสะดวกและมีประสิทธิภาพมากยิ่งขึ้น

HTML (Hypertext Markup Language)

เป็นภาษามาร์กอัปที่ใช้สำหรับกำหนดโครงสร้างและองค์ประกอบของหน้าเว็บ โดย HTML ประกอบด้วยแท็ก (Tags) ซึ่งใช้ในการระบุส่วนต่าง ๆ ของเนื้อหา เช่น ข้อความ รูปภาพ ตาราง หรือปุ่มควบคุม เพื่อจัดรูปแบบและแสดงผลบนเว็บเบราว์เซอร์ได้อย่างถูกต้อง HTML จึงถือเป็นภาษาพื้นฐานสำคัญที่ใช้ในการสร้างหน้าเว็บและ Web Application ทุกประเภท ในระบบนี้ HTML ถูกใช้เพื่อสร้างโครงร่างของหน้า Dashboard โดยกำหนดตำแหน่งและส่วนประกอบต่าง ๆ ที่จะแสดงผลบนหน้าเว็บ เช่น กล่องแสดงค่าอุณหภูมิ ความชื้น แสง และก๊าซคาร์บอนไดออกไซด์แบบเรียลไทม์ รวมถึงปุ่มควบคุมอุปกรณ์ต่าง ๆ อย่าง Cooling Fan, Solenoid Valve และ LED Light นอกจากนี้ยังมีพื้นที่สำหรับแสดงกราฟแนวโน้มข้อมูลที่ดึงมาจากฐานข้อมูล เพื่อให้ผู้ใช้สามารถติดตามสภาพแวดล้อมของโรงเรือนเพาะเห็ดได้อย่างต่อเนื่องและเข้าใจง่าย HTML ยังทำงานร่วมกับภาษาอื่น ๆ ได้แก่ PHP ที่ใช้ประมวลผลข้อมูลจากฝั่งเซิร์ฟเวอร์, CSS ที่ใช้ตกแต่งหน้าเว็บให้สวยงาม และ JavaScript ที่ช่วยเพิ่มความสามารถในการโต้ตอบกับผู้ใช้ ทำให้ Web Application มีความสมบูรณ์ทั้งในด้านการแสดงผลและการทำงานแบบโต้ตอบได้อย่างมีประสิทธิภาพ

CSS (Cascading Style Sheets)

เป็นภาษาที่ใช้ในการตกแต่งและจัดรูปแบบการแสดงผลของเว็บไซต์ โดยมีหน้าที่แยกส่วนของการออกแบบออกจากโครงสร้างเนื้อหาที่กำหนดด้วย HTML เพื่อให้สามารถปรับปรุงหรือแก้ไขรูปแบบการแสดงผลได้อย่างสะดวกและยืดหยุ่น CSS สามารถกำหนดคุณลักษณะต่าง ๆ ขององค์ประกอบบนหน้าเว็บ เช่น สี ขนาดตัวอักษร ระยะห่าง การจัดตำแหน่ง รวมถึงการออกแบบให้หน้าเว็บไซต์สามารถปรับขนาดและแสดงผลได้อย่างเหมาะสมบนทุกอุปกรณ์ ไม่ว่าจะเป็นหน้าจอคอมพิวเตอร์ แท็บเล็ต หรือสมาร์ตโฟน (Responsive Design) ในงานวิจัยนี้ CSS ถูกนำมาใช้เพื่อเพิ่มความสวยงาม ความเป็นระเบียบ และความเข้าใจง่ายให้กับหน้า Web Dashboard โดยเน้นหลักการของ Data Visualization เพื่อให้ข้อมูลที่แสดงออกมามีความหมายและเข้าใจได้อย่างรวดเร็ว ตัวอย่างเช่น การใช้สีที่แตกต่างกันเพื่อบ่งบอกสถานะของค่าที่วัดได้ เช่น สีแดงเมื่อค่ามีการเกิน Threshold การออกแบบกราฟและตัวเลขให้มีขนาดและความชัดเจนเพื่อให้อ่านค่าได้ง่าย รวมถึงการจัดองค์ประกอบของหน้าเว็บให้เหมาะสมกับขนาดหน้าจอทั้งบนอุปกรณ์มือถือและคอมพิวเตอร์ ซึ่งช่วยให้ผู้ใช้สามารถเข้าถึงและเข้าใจข้อมูลได้อย่างมีประสิทธิภาพมากขึ้น

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

JavaScript

เป็นภาษาสคริปต์ที่ทำงานฝั่งผู้ใช้ (Client-side) โดยหลักการทำงานคือช่วยให้เว็บไซต์หรือแอปพลิเคชันสามารถโต้ตอบกับผู้ใช้ได้แบบเรียลไทม์ โดยไม่ต้องรีโหลดหน้าเว็บทุกครั้ง เมื่อผู้ใช้ทำกิจกรรม เช่น คลิกปุ่ม เลื่อนหน้าจอ หรือกรอกข้อมูล JavaScript จะทำงานเพื่อประมวลผลเหตุการณ์นั้น ๆ และปรับเปลี่ยนส่วนต่าง ๆ ของหน้าเว็บตามที่กำหนด JavaScript สามารถทำงานร่วมกับ HTML และ CSS โดย HTML จะเป็นโครงสร้างของหน้าเว็บ ส่วน CSS จะเป็นการตกแต่ง ส่วน JavaScript จะเข้ามาเพิ่มความสามารถโต้ตอบ เช่น การอัปเดตข้อมูลแบบเรียลไทม์ แสดงกราฟ เคลื่อนไหว หรือแจ้งเตือนผู้ใช้ อีกทั้งยังสามารถเชื่อมต่อกับเซิร์ฟเวอร์ผ่านเทคนิคอย่าง AJAX เพื่อดึงหรือส่งข้อมูลโดยไม่ต้องโหลดหน้าใหม่ ทำให้เว็บตอบสนองเร็วขึ้น นอกจากนี้ JavaScript ยังสามารถเชื่อมต่อกับโปรโตคอลต่าง ๆ เช่น WebSocket หรือ MQTT เพื่อรับส่งข้อมูลแบบเรียลไทม์ ทำให้สามารถสั่งงานอุปกรณ์หรืออัปเดตข้อมูลบนหน้าเว็บทันที หลักการทำงานของ JavaScript คือการตรวจจับเหตุการณ์ ประมวลผลข้อมูล และปรับเปลี่ยนหน้าตาของเว็บหรือส่งข้อมูลไปยังเซิร์ฟเวอร์ตามเงื่อนไขที่กำหนด

PostgreSQL

เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) ที่มีความเสถียรสูงและรองรับการจัดเก็บข้อมูลปริมาณมาก หลักการทำงานของ PostgreSQL คือการเก็บข้อมูลในรูปแบบ ตาราง (Tables) ซึ่งประกอบด้วย แถว (Rows) และ คอลัมน์ (Columns) ข้อมูลจะถูกจัดการโดยใช้คำสั่ง SQL ทำให้สามารถสร้าง แก้ไข และดึงข้อมูลออกมาได้อย่างเป็นระบบ PostgreSQL รองรับฟีเจอร์ขั้นสูงหลายอย่าง เช่น Transactions เพื่อให้การทำงานของฐานข้อมูลเป็นไปอย่างถูกต้องครบถ้วน, Triggers สำหรับสั่งงานอัตโนมัติเมื่อเกิดเหตุการณ์บางอย่าง, Views สำหรับสร้างตารางเสมือนที่ช่วยให้เข้าถึงข้อมูลง่ายขึ้น, และ JSON Data Type สำหรับจัดเก็บข้อมูลที่ไม่เป็นแบบตารางโดยตรง

หลักการทำงานของ PostgreSQL คือเมื่อมีคำสั่ง SQL ถูกส่งเข้าไป ระบบจะทำการประมวลผล ตรวจสอบความถูกต้องของข้อมูล และจัดเก็บหรือดึงข้อมูลตามคำสั่ง จากนั้นสามารถส่งผลลัพธ์กลับไปยังผู้ใช้หรือโปรแกรมที่เชื่อมต่อ โดยสามารถทำงานร่วมกับโปรแกรมต่าง ๆ ผ่าน API หรือไลบรารีที่รองรับการเชื่อมต่อ เช่น PHP, Python, Node.js ทำให้ PostgreSQL เป็นฐานข้อมูลที่เหมาะสำหรับการจัดเก็บข้อมูลขนาดใหญ่ รองรับการประมวลผลซับซ้อน และทำงานร่วมกับระบบอื่น ๆ ได้อย่างมีประสิทธิภาพ

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

Node-RED

เป็นเครื่องมือโอเพนซอร์สสำหรับการพัฒนาและจัดการระบบ IoT โดยใช้แนวคิด Flow-based Programming (FBP) ซึ่งช่วยให้ผู้พัฒนาสามารถออกแบบกระบวนการทำงานของระบบได้โดยไม่ต้องเขียนโค้ดมากเกินไป หลักการทำงานของ Node-RED คือการเชื่อมต่อ Node แต่ละตัวเป็นลำดับขั้นตอนของการทำงาน เช่น การรับข้อมูล การประมวลผล และการส่งออกผลลัพธ์ Node แต่ละตัวมีหน้าที่เฉพาะ เช่น Node สำหรับรับข้อมูลจากเซ็นเซอร์, Node สำหรับประมวลผลข้อมูล, Node สำหรับเก็บข้อมูลในฐานข้อมูล หรือ Node สำหรับส่งข้อมูลออกไปยัง Web API ระบบทำงานโดยการลากและวาง Node เข้าด้วยกัน จากนั้น

กำหนดการเชื่อมต่อระหว่าง Node ให้ข้อมูลไหลผ่านตามลำดับที่ตั้งไว้ การทำงานแบบนี้เรียกว่า Flow หลักการสื่อสารของ Node-RED มักใช้โปรโตคอลมาตรฐาน เช่น HTTP, MQTT, WebSocket ทำให้สามารถเชื่อมต่อกับอุปกรณ์ IoT, เซ็นเซอร์, ระบบฐานข้อมูล หรือแอปพลิเคชันต่าง ๆ ได้อย่างง่ายดาย นอกจากนี้ Node-RED มีอินเทอร์เฟซแบบเว็บที่ช่วยให้ผู้ใช้สามารถติดตามการทำงานของ Flow แบบเรียลไทม์ ตรวจสอบข้อมูล และปรับปรุงระบบได้โดยไม่ต้องหยุดการทำงาน

HTTP (Hypertext Transfer Protocol)

คือ โปรโตคอลหลักที่ใช้ในการสื่อสารข้อมูลระหว่างเว็บเบราว์เซอร์กับเว็บเซิร์ฟเวอร์ ซึ่งเป็นหัวใจสำคัญของการทำงานของอินเทอร์เน็ตในปัจจุบัน เมื่อผู้ใช้พิมพ์ที่อยู่เว็บไซต์ (URL) ลงในเบราว์เซอร์ เบราว์เซอร์จะส่งคำขอ (Request) ไปยังเซิร์ฟเวอร์ปลายทาง โดยคำขอนั้นอาจเป็นการร้องขอไฟล์ HTML, CSS, JavaScript, รูปภาพ หรือข้อมูลอื่น ๆ ที่จำเป็นสำหรับการแสดงผลหน้าเว็บ จากนั้นเว็บเซิร์ฟเวอร์จะประมวลผลและส่งข้อมูลกลับมาในรูปแบบของคำตอบ (Response) เพื่อให้เบราว์เซอร์แสดงผลเป็นหน้าเว็บให้ผู้ใช้เห็น HTTP ทำงานบนพอร์ตมาตรฐานคือ พอร์ต 80 และมีลักษณะการทำงานแบบ Stateless Protocol หมายความว่าการเชื่อมต่อแต่ละครั้งจะไม่จดจำสถานะของการสื่อสารก่อนหน้า เช่น เมื่อผู้ใช้ส่งคำขอหนึ่งเสร็จ ระบบจะถือว่าเป็นการสื่อสารที่จบลง ไม่ได้จำว่าผู้ใช้เคยติดต่อมาก่อน เพื่อให้สามารถ “จำสถานะ” หรือ “ติดตามการใช้งาน” ได้ เช่น การเข้าสู่ระบบ (Login) หรือการเก็บสินค้าลงตะกร้า จึงมีการนำเทคโนโลยีเสริมเข้ามา เช่น Cookies, Sessions, และ Tokens เพื่อช่วยให้เว็บไซต์สามารถทำงานแบบมีบริบทต่อเนื่อง HTTP มีโครงสร้างของคำขอและคำตอบที่ชัดเจน โดยคำขอ (HTTP Request) จะประกอบด้วย Request Line, Headers, และ Body ซึ่งระบุข้อมูลว่าเบราว์เซอร์ต้องการทำอะไร เช่น GET สำหรับการดึงข้อมูล, POST สำหรับการส่งข้อมูล, PUT สำหรับการอัปเดตข้อมูล และ DELETE สำหรับการลบข้อมูล ส่วนคำตอบ (HTTP Response) จากเซิร์ฟเวอร์จะประกอบด้วยรหัสสถานะ (Status Code) เช่น 200 (OK), 404 (Not Found), หรือ 500 (Internal Server Error) เพื่อบอกผลลัพธ์ของการประมวลผล

MQTT (Message Queuing Telemetry Transport)

เป็น โปรโตคอลการสื่อสารแบบเบา (Lightweight Protocol) ที่ออกแบบมาเพื่อใช้ในการสื่อสารข้อมูลระหว่างอุปกรณ์ IoT หรือระบบที่มีทรัพยากรจำกัด โดยหลักการทำงานของ MQTT จะอิงกับแนวคิด Publish–Subscribe ซึ่งแตกต่างจากการสื่อสารแบบ Client–Server ปกติ ใน MQTT มีส่วนประกอบหลัก 3 ส่วน คือ **Publisher, Subscriber, และ Broker**

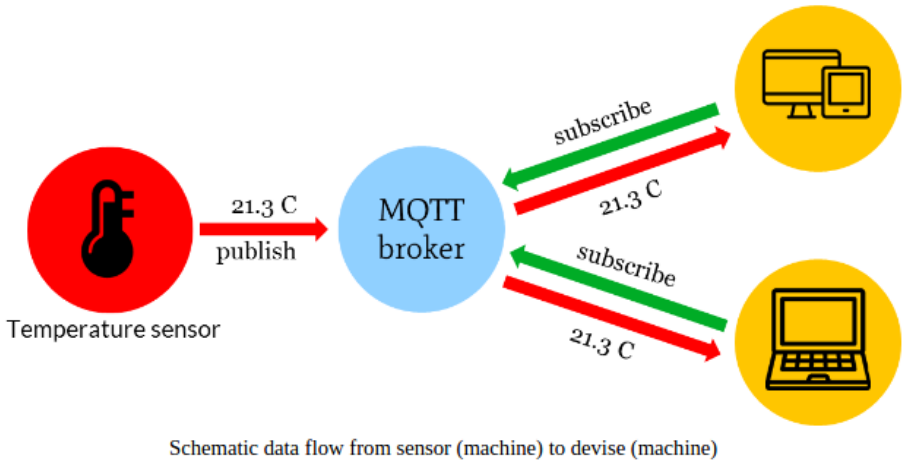
Publisher คืออุปกรณ์หรือโปรแกรมที่ส่งข้อมูลออกไป เช่น เซ็นเซอร์หรือไมโครคอนโทรลเลอร์

Subscriber คืออุปกรณ์หรือโปรแกรมที่รับข้อมูล เช่น แอปพลิเคชันหรือระบบ Web Dashboard

Broker คือเซิร์ฟเวอร์กลางที่ทำหน้าที่รับข้อมูลจาก Publisher และส่งต่อไปยัง Subscriber ตามหัวข้อ (Topic) ที่สมัครรับข้อมูล

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

หลักการทำงานของ MQTT คือ Publisher ส่งข้อความไปยัง Broker พร้อมระบุหัวข้อ ข้อมูลนี้จะถูก Broker จัดการและส่งต่อไปยัง Subscriber ที่สมัครรับหัวข้อเดียวกัน Subscriber จะได้รับข้อมูลแบบเรียลไทม์ทันที โดยไม่ต้องคอยถามข้อมูล (Polling) ทำให้ระบบตอบสนองได้รวดเร็วและใช้ทรัพยากรเครือข่ายน้อย ข้อดีของ MQTT คือน้ำหนักข้อมูลเบา เหมาะสำหรับอุปกรณ์ IoT ที่มีหน่วยความจำและแบนด์วิดท์จำกัด รองรับการส่งข้อมูลแบบเรียลไทม์ ทำงานได้แม้เครือข่ายไม่เสถียร และมีระบบ QoS (Quality of Service) สำหรับควบคุมความน่าเชื่อถือของการส่งข้อความ



ภาพที่ 5 หลักการทำงานของ MQTT

https://grassrootengineer.medium.com

Telegram Bot API

เป็นระบบที่ช่วยให้ผู้พัฒนาสามารถสร้างบอท (Bot) บนแอปพลิเคชัน Telegram เพื่อส่งและรับข้อความอัตโนมัติได้ โดยหลักการทำงานคือบอทจะเชื่อมต่อกับเซิร์ฟเวอร์ผ่าน API ของ Telegram ซึ่งเป็นชุดคำสั่งที่ให้โปรแกรมภายนอกสามารถสื่อสารกับ Telegram ได้ บอทสามารถส่งข้อความ รูปภาพ สติ๊กเกอร์ หรือไฟล์ต่าง ๆ ไปยังผู้ใช้กลุ่มใดก็ได้ตามที่กำหนด รวมถึงสามารถรับคำสั่งหรือข้อความจากผู้ใช้เพื่อนำไปประมวลผลต่อ การทำงานของ Telegram Bot API มีลักษณะเป็นแบบ Client–Server ผู้ใช้ส่งข้อความไปยังบอทผ่าน Telegram Server จากนั้นบอทจะประมวลผลคำสั่งหรือข้อมูล และส่งข้อความตอบกลับหรือแจ้งเตือนตามเงื่อนไขที่กำหนด เช่น การแจ้งเตือนเมื่อเกิดเหตุการณ์สำคัญ ข้อความเตือนความผิดปกติ หรือข้อมูลสถานะต่าง ๆ บอทสามารถทำงานร่วมกับระบบอื่น ๆ ผ่านโปรโตคอล HTTP หรือ Webhook ทำให้สามารถเชื่อมต่อกับเซ็นเซอร์ อุปกรณ์ IoT หรือซอฟต์แวร์ต่าง ๆ เพื่อส่งข้อมูลแจ้งเตือนไปยังผู้ใช้แบบเรียลไทม์

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

Arduino IDE (Integrated Development Environment)

เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้กับบอร์ด Arduino หรือไมโครคอนโทรลเลอร์อื่น ๆ ผู้ใช้เขียนโค้ดด้วยภาษา C/C++ ที่ปรับแต่งให้เหมาะกับฮาร์ดแวร์ จากนั้น IDE จะทำการคอมไพล์โค้ดให้เป็นไฟล์ไบนารีที่บอร์ดสามารถอ่านและรันได้ หลังจากนั้น IDE จะอัปโหลดไฟล์ไบนารีไปยังบอร์ดผ่านพอร์ต USB หรือการเชื่อมต่ออื่น ๆ เมื่อโค้ดถูกอัปโหลดแล้ว บอร์ดจะเริ่มทำงานตามลำดับที่โปรแกรมกำหนด โดยเริ่มจาก

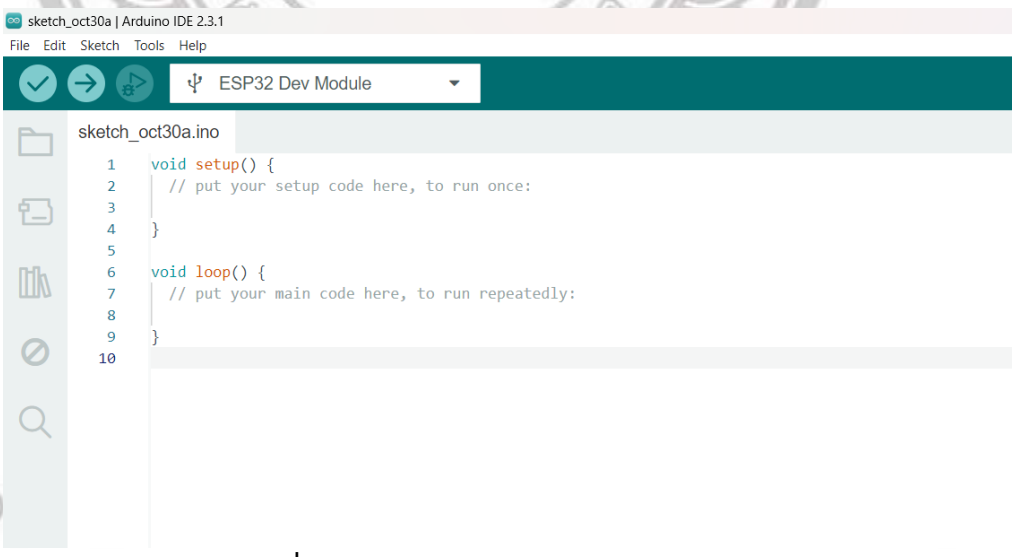
การเตรียมค่าและอุปกรณ์ใน setup() จากนั้นเข้าสู่การทำงานวนซ้ำใน loop() เพื่ออ่านข้อมูล ประมวลผล และตอบสนองต่อสัญญาณหรือคำสั่งต่าง ๆ อย่างต่อเนื่อง

Arduino IDE ยังมีฟีเจอร์ Serial Monitor สำหรับรับและส่งข้อมูลระหว่างคอมพิวเตอร์กับบอร์ด ทำให้สามารถตรวจสอบค่าตัวแปรหรือข้อมูลจากเซ็นเซอร์แบบเรียลไทม์ รวมถึงรองรับการติดตั้งไลบรารีเสริม ซึ่งช่วยให้ใช้งานเซ็นเซอร์ โมดูลสื่อสาร หรืออุปกรณ์เสริมอื่น ๆ ได้ง่ายและสะดวก ทำให้การพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์เป็นไปอย่างรวดเร็วและมีประสิทธิภาพ

1.**Header (ส่วนหัวของโปรแกรม)** ทำหน้าที่เตรียมทุกอย่างก่อนเริ่มโปรแกรม โดยรวมไลบรารีที่จำเป็น ประกาศค่าคงที่ ตัวแปร และกำหนดพินของบอร์ด เพื่อให้โปรแกรมรู้ว่าต้องใช้ฟังก์ชันและอุปกรณ์อะไรบ้าง และทำให้ส่วนอื่นของโค้ดสามารถเรียกใช้งานตัวแปรและพินเหล่านี้ได้อย่างถูกต้อง

2.**void setup()** เป็นฟังก์ชันที่ทำงานเพียงครั้งเดียวเมื่อเริ่มต้นบอร์ด ใช้สำหรับตั้งค่าพื้นฐานของระบบ เช่น กำหนดพินเป็น Input/Output เริ่มการสื่อสาร Serial หรือเชื่อมต่อกับอุปกรณ์ต่าง ๆ ฟังก์ชันนี้ทำให้ระบบพร้อมทำงานและสามารถประมวลผลคำสั่งในขั้นตอนต่อไปได้

3.**void loop()** เป็นฟังก์ชันหลักที่ทำงานซ้ำอย่างต่อเนื่องหลังจาก setup() ทำงานเสร็จ ทำหน้าที่อ่านค่าจากเซ็นเซอร์หรืออินพุตต่าง ๆ ประมวลผลเงื่อนไข ส่งข้อมูล หรือสั่งงานอุปกรณ์ตามเงื่อนไขที่กำหนด การทำงานวนซ้ำนี้ทำให้ระบบสามารถตอบสนองต่อเหตุการณ์หรือข้อมูลที่เปลี่ยนแปลงได้แบบเรียลไทม์ และทำงานต่อเนื่องโดยไม่หยุด



ภาพที่ 6 หน้าต่างโปรแกรม Arduino IDE

Cop All rights reserved

2.2 เอกสารงานวิจัยที่เกี่ยวข้อง

นัฐพันธ์ พูนวิวัฒน์ และคณะ(2022)) “**การพัฒนาโรงเพาะเห็ดควบคุมด้วยระบบอัตโนมัติโดยใช้พลังงานจากเซลล์แสงอาทิตย์**” มีวัตถุประสงค์เพื่อออกแบบและสร้างโรงเพาะเห็ดควบคุมด้วยระบบอัตโนมัติโดยเปลี่ยนมาใช้พลังงานไฟฟ้าจาก เซลล์แสงอาทิตย์ แทนการใช้ไฟฟ้าภายในบ้านเพื่อประหยัดค่าใช้จ่าย โรง

เพาะเห็ดนี้ถูกสร้างขึ้นเพื่อเพาะ เห็ดนางฟ้าภูฐาน เพียงอย่างเดียว ระบบสามารถผลิตกระแสไฟฟ้าจากแผงโซล่าเซลล์ขนาดไม่ต่ำกว่า 330 วัตต์ ผ่านคอนโทรลชาร์จเจอร์ชนิด MPPT เพื่อชาร์จแบตเตอรี่ขนาด 12 โวลต์ 75 แอมแปร์-ชั่วโมง ซึ่งเพียงพอสำหรับอุปกรณ์ในระบบทั้งกลางวันและกลางคืน การควบคุมอุณหภูมิและความชื้นใช้เซ็นเซอร์ควบคุมแบบดิจิตอลรุ่น STC 1000 สำหรับอุณหภูมิ และรุ่น XH-W3005 สำหรับความชื้น โดยมีเป้าหมายควบคุมอุณหภูมิให้อยู่ที่ 21-30 องศาเซลเซียส และความชื้นอยู่ที่ 80 % ระบบจะทำงานแบบอัตโนมัติ: เมื่ออุณหภูมิมากกว่าค่าที่กำหนด พัดลมระบายอากาศ จะทำงาน และถ้าความชื้นต่ำกว่าค่าที่กำหนด หัวพ่นหมอก จะทำงาน นอกจากนี้ ยังมีการเสริมอุปกรณ์เพิ่มเติม เช่น กล้องวงจรปิด Wi-Fi เพื่อให้สามารถดูการเจริญเติบโตของเห็ดผ่านสมาร์ทโฟนได้ และเสริมระบบให้ปุ๋ยกับเห็ดโดยใช้ปั๊มไดอะแฟรม 12 V. ซึ่งทำงานเพียงสัปดาห์ละ 1 ครั้ง ผลการทดลองยืนยันว่าระบบควบคุมอุณหภูมิและความชื้นสามารถทำงานตามค่าที่ตั้งไว้ได้จริง โดยสามารถลดอุณหภูมิจาก 32 องศาเซลเซียส ลงมาที่ 29 องศาเซลเซียส โดยใช้เวลาเฉลี่ย 23.36 นาที และพบว่าเห็ดที่เกิดจากการเพาะในระบบอัตโนมัติมีสีสันที่สวยงามตามลักษณะของเห็ดนางฟ้าภูฐาน

วีรศักดิ์ ฟองเงิน และคณะ(2018) “**การประยุกต์ใช้เทคโนโลยีไอโอทีควบคุมฟาร์มอัจฉริยะในโรงเรือนเพาะเห็ดนางฟ้า**” โดยมีวัตถุประสงค์เพื่อออกแบบและพัฒนาระบบควบคุมอุณหภูมิและความชื้นสำหรับโรงเรือนเพาะเห็ดนางฟ้าและเห็ดนางรม ระบบควบคุมที่พัฒนาขึ้นใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมหลัก โดยมีการใช้เซ็นเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ (เช่น DHT22 ในการทดลอง) และมีการส่งข้อมูลผ่านคลาวด์ไปยังเซอร์วิส NETPIE ซึ่งสามารถแสดงผลข้อมูลบนมือถือหรืออุปกรณ์พกพาแบบเรียลไทม์ผ่าน NETPIE Freeboard ได้ ระบบถูกตั้งโปรแกรมให้ทำงานแบบอัตโนมัติเพื่อรักษาอุณหภูมิและความชื้นให้อยู่ในช่วงที่เหมาะสม โดยอุณหภูมิที่เหมาะสมสำหรับการเจริญของเห็ดนางฟ้าอยู่ที่ประมาณ 25 องศาเซลเซียส และความชื้นของอากาศควรมีความชื้นไม่ต่ำกว่า 80-85 เปอร์เซ็นต์ โดยมีเงื่อนไขการทำงานคือ หากอุณหภูมิต่ำกว่า 15 องศาเซลเซียส จะสั่งการให้ ฮีตเตอร์ ทำงานเพื่อเพิ่มอุณหภูมิ และหากอุณหภูมิมากกว่า 35 องศาเซลเซียส จะสั่งการให้ ปั๊มน้ำ (พ่นละอองน้ำ) และ พัดลมระบายอากาศ ทำงานเพื่อลดอุณหภูมิลง ผลการทดสอบยืนยันว่าระบบสามารถทำงานตามเงื่อนไขที่ออกแบบไว้ และเห็ดที่เก็บจากโรงเรือนควบคุมมีปริมาณมากกว่าโรงเรือนแบบทั่วไป โดยมีน้ำหนักเฉลี่ยต่อก้อนสูงถึง 1.506 กิโลกรัม ซึ่งสูงกว่าเห็ดจากโรงเรือนแบบทั่วไปที่มีน้ำหนักเฉลี่ย 1.206 กิโลกรัม

ธีรพงศ์ เพ็ญเนตร และคณะ(2023) “**การพัฒนาระบบควบคุมสภาพแวดล้อมโรงเรือนเพาะเห็ดนางรมภูฐานโดยใช้ระบบสมองกลฝังตัวไอโอที**” ได้ทำการพัฒนาและประเมินระบบที่คล้ายคลึงกัน โดยมีวัตถุประสงค์เพื่อพัฒนาระบบควบคุมสภาพแวดล้อมโรงเรือนเพาะปลูกเห็ดนางรมภูฐาน และศึกษาต้นทุนการผลิตและรายได้จากการเพาะปลูก ระบบนี้ใช้ NodeMCU ESP8266 เป็นสมองกลฝังตัว และใช้ เซ็นเซอร์ SHT20 ในการตรวจสอบอุณหภูมิและความชื้น โดยมีเป้าหมายควบคุมอุณหภูมิไว้ไม่เกิน 33 องศาเซลเซียส หากอุณหภูมิเกินกว่าที่กำหนด พัดลมระบายอากาศและระบบทำความเย็นก็จะทำงานเพื่อรักษาอุณหภูมิให้เหมาะสม การควบคุมและการแสดงผลสามารถทำได้ผ่าน Blynk application บนโทรศัพท์มือถือ ทั้งในโหมดอัตโนมัติ และโหมดควบคุมโดยผู้ใช้งาน ผลการประเมินประสิทธิภาพของระบบโดยผู้เชี่ยวชาญพบว่าคุณภาพ

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

ของระบบอยู่ในระดับ มากที่สุด โดยมีค่าเฉลี่ยรวม เท่ากับ 4.50 นอกจากนี้ ผลการศึกษาด้านเศรษฐศาสตร์ยังพบว่า โรงเรือนเพาะปลูกเห็ดนางรมภูฐานที่ใช้ระบบควบคุมนี้ จะเริ่มมี กำไรจากการเพาะปลูกเมื่อเข้าเดือนที่ 3 และเดือนที่ 4

Kukkong Kaewkorakot et al.(2024) **“Smart Mushroom Cultivation House: Engineering Development and Data Analysis”** นำเสนอการพัฒนาทางวิศวกรรมของโรงเรือนเพาะเห็ดอัจฉริยะและการวิเคราะห์ข้อมูลอนุกรมเวลา (Time-Series Data Analysis) โรงเรือนต้นแบบตั้งอยู่ที่จังหวัดน่าน โดยระบบที่พัฒนาขึ้นคือระบบควบคุมความชื้นอัตโนมัติ ซึ่งประกอบด้วยปั๊มน้ำ หัวพ่นหมอก เซ็นเซอร์วัดอัตราการไหลของน้ำ (YF-S201) และเซ็นเซอร์อุณหภูมิ/ความชื้น (AM2301) ระบบนี้ควบคุมโดยโปรแกรม C/C++ ที่ติดตั้งในบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 แพลตฟอร์ม NETPIE 2020 ถูกใช้ในการเก็บข้อมูล การเฝ้าติดตามแบบเรียลไทม์ และการปรับค่าพารามิเตอร์ของระบบผ่านเว็บแอปพลิเคชัน โดยค่าที่ถูกควบคุมอัตโนมัติคือความชื้นสัมพัทธ์ ซึ่งสามารถกำหนดค่าต่ำสุด และสูงสุด สำหรับการเปิด/ปิดปั๊มน้ำได้ การวิเคราะห์ข้อมูลอนุกรมเวลาแสดงให้เห็นว่า ระบบสามารถควบคุมความชื้นได้อย่างเสถียรตามช่วงที่กำหนด (เช่น 60% และ 80 % ในช่วงกลางวัน) และยืนยันว่าการควบคุมความชื้นไม่จำเป็นต้องทำในเวลากลางคืนเนื่องจากความชื้นภายนอกสูงและอุณหภูมิต่ำอยู่แล้ว งานวิจัยสรุปว่า ระบบที่พัฒนาขึ้นมีข้อได้เปรียบในด้าน การประหยัดแรงงาน การใช้น้ำ และการใช้พลังงานไฟฟ้า เมื่อเทียบกับวิธีการเพาะเห็ดแบบดั้งเดิมโดยมีการประเมินต้นทุนค่าไฟฟ้าโดยประมาณอยู่ที่ 8 บาท/วัน หรือ 240 บาท/เดือน

Kuntapon Mahamad et al.(2021) “**Application of Smart Farm System to Enhance Phoenix Oyster Mushroom Production**” มีวัตถุประสงค์เพื่อพัฒนาและประยุกต์ใช้ระบบสมาร์ทฟาร์ม (SFS) ในการควบคุมอุณหภูมิและความชื้นสำหรับโรงเรือนเพาะเห็ดนางรมฮ่องเต้ ระบบที่พัฒนาขึ้นใช้บอร์ด ET-ESP8266-RS485 ในการควบคุมและแสดงผลข้อมูลผ่านแอปพลิเคชันบนสมาร์ทโฟน โดยตั้งค่าการควบคุมอุณหภูมิไม่เกิน 30 องศาเซลเซียส และความชื้นไม่น้อยกว่า 80% ระบบสามารถแสดงผลอุณหภูมิและความชื้นได้อย่างแม่นยำ โดยมีค่าเปอร์เซ็นต์ความผิดพลาดเฉลี่ยต่ำเพียง 0.61 % สำหรับอุณหภูมิ และ 1.19 % สำหรับความชื้น ผลการทดลองโดยเปรียบเทียบโรงเรือนที่ติดตั้ง SFS (ชุดทดลอง) กับโรงเรือนที่ไม่ได้ติดตั้ง (ชุดควบคุม) พบว่า ชุดทดลองสามารถรักษาอุณหภูมิและความชื้นให้คงที่ได้ที่เฉลี่ย 27.67องศาเซลเซียส และ 83.36 % ตามลำดับ ในขณะที่ชุดควบคุมมีสภาพแวดล้อมที่ไม่เหมาะสม โดยมีอุณหภูมิเฉลี่ยสูงถึง 32.28 องศาเซลเซียส และความชื้นเฉลี่ยต่ำเพียง 69.32 % ความแม่นยำในการควบคุมสภาพแวดล้อมส่งผลให้ผลผลิตเห็ดของชุดทดลองมีปริมาณเฉลี่ย 8.1 กิโลกรัม ซึ่งสูงกว่าชุดควบคุมที่มีผลผลิตเฉลี่ย 5.7 กิโลกรัม คิดเป็น 1.42 เท่า การวิเคราะห์ความพึงพอใจจากผู้ใช้งาน 30 คน พบว่ามีความพึงพอใจในการใช้ระบบในระดับสูงสุด (4.54 และ S.D.=0.52) และวิเคราะห์ทางเศรษฐศาสตร์พบว่าจุดคุ้มทุนของการผลิตเห็ด (BEQ) อยู่ที่ 28.10 กิโลกรัม

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright© by Naresuan University
All rights reserved

บทที่ 3
วิธีการดำเนินงานวิจัย

ในการดำเนินงานวิจัยครั้งนี้ ผู้วิจัยได้ออกแบบและพัฒนาระบบควบคุมสภาพแวดล้อมภายในโรงเรือนเพาะเห็ดโดยอาศัยเทคโนโลยี Internet of Things (IoT) เพื่อให้สามารถตรวจวัดและควบคุมปัจจัยสำคัญ ได้แก่ อุณหภูมิ ความชื้น ความเข้มแสง และระดับก๊าซคาร์บอนไดออกไซด์ ได้อย่างแม่นยำ และสามารถแสดงผลข้อมูลแบบเรียลไทม์ผ่าน Web Application ที่พัฒนาไว้ ทั้งนี้กระบวนการดำเนินงานวิจัยได้ถูกออกแบบอย่างเป็นระบบ โดยเริ่มตั้งแต่การศึกษาข้อมูลพื้นฐานและความต้องการของผู้ใช้งาน การออกแบบระบบทั้งในส่วนฮาร์ดแวร์และซอฟต์แวร์ การพัฒนาระบบต้นแบบ การเชื่อมโยงข้อมูลเข้าสู่ฐานข้อมูล

3.1 การเก็บข้อมูล

การเก็บข้อมูลเริ่มจากการศึกษาข้อมูลสภาพแวดล้อมที่เหมาะสมในการเพาะเห็ดในโรงเรือน และการลงพื้นที่เพื่อสำรวจการเพาะเห็ดในโรงเรือน

จากการลงพื้นที่สำรวจการเพาะเห็ดในโรงเรือน พบว่าการควบคุมสภาพแวดล้อมภายในมีความสำคัญอย่างยิ่งต่อการเจริญเติบโตของเห็ด โดยเฉพาะเห็ดนางฟ้าที่ต้องการอุณหภูมิและความชื้นในระดับที่เหมาะสม เพื่อให้สามารถออกดอกได้เต็มที่และมีคุณภาพดี เห็ดนางฟ้าจะเติบโตได้ดีในอุณหภูมิไม่เกิน 28 องศาเซลเซียส และมีความชื้นสัมพัทธ์ประมาณ 80% หากสภาพแวดล้อมไม่เหมาะสม เช่น อุณหภูมิสูงเกินไปหรือความชื้นต่ำ เห็ดจะหยุดการเจริญเติบโต ดอกเล็ก หรือแห้งเหี่ยวก่อนเวลาเก็บเกี่ยว

ภาพที่ 7 การลงพื้นที่สำรวจข้อมูล

3.2 วัสดุอุปกร์ที่ใช้ในงานวิจัย

3.2.1 ส่วนของฮาร์ดแวร์

NodeMCU ESP32

Arduino MEGA 2560

TFT Monitor 3.5 inch

Module Relay

DHT22

MH-Z114A

LDR Light

3.2.2 ส่วนของซอฟต์แวร์

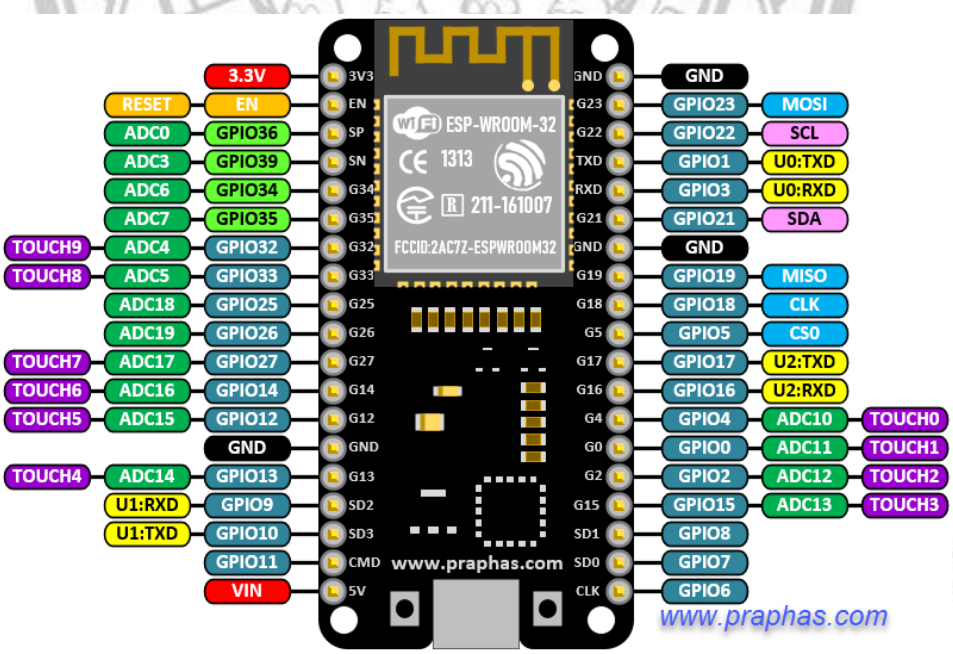
โปรแกรม Arduino IDE

Node-RED

PostgreSQL

Telegram Notify

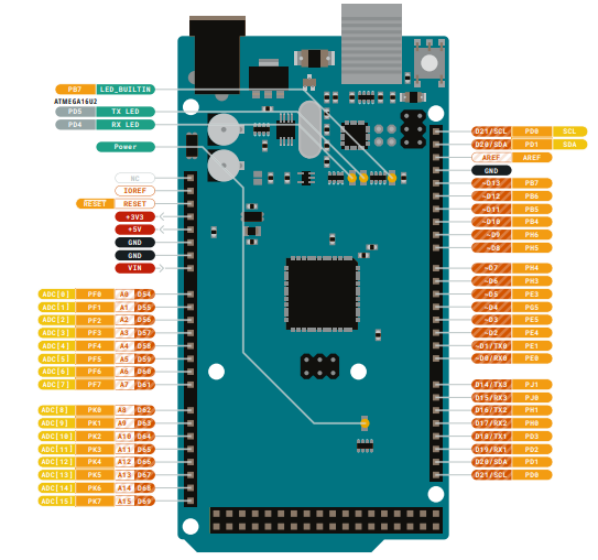
ขาของบอร์ด ESP32



ภาพที่ 8 ขาของบอร์ด ESP32

All rights reserved

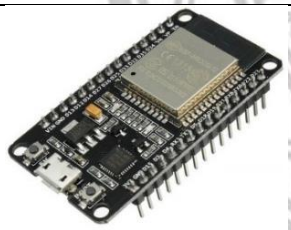
ขาของบอร์ด Arduino MEGA 2560



ภาพที่ 9 ขาของบอร์ด Arduino MEGA 2560

ที่มา https://www.ai-corporation.net/2021/11/25/arduino-mega2560/

ตารางที่ 1 เครื่องมือที่ใช้ในการวิจัยของส่วนฮาร์ดแวร์

ภาพอุปกรณ์	ชื่ออุปกรณ์	การทำงานอุปกรณ์
	NodeMCU ESP32	เป็นไมโครคอนโทรลเลอร์รุ่นใหม่ที่มีความสามารถสูง รองรับการเชื่อมต่อ Wi-Fi และ Bluetooth ในตัว ทำให้สามารถส่งข้อมูลแบบไร้สายหรือควบคุมอุปกรณ์อื่น ๆ ได้
	Arduino MEGA 2560	เป็นไมโครคอนโทรลเลอร์ที่มี พอร์ต I/O มากกว่า Arduino UNO เหมาะสำหรับโปรเจคที่ต้องเชื่อมต่อเซ็นเซอร์และอุปกรณ์หลายตัวพร้อมกัน
	TFT Monitor 3.5	เป็นจอแสดงผลสี สามารถใช้แสดงข้อมูลแบบกราฟิก เช่น กราฟอุณหภูมิ ความชื้น หรือสถานะอุปกรณ์
	Module Relay	รีเลย์เป็นอุปกรณ์สวิตช์ไฟฟ้าที่ควบคุมด้วยสัญญาณไฟฟ้าต่ำ เช่น 3.3V หรือ 5V จากไมโครคอนโทรลเลอร์

Copyright by Naresuan University All rights reserved

	DHT22	เป็นเซ็นเซอร์วัด อุณหภูมิและความชื้น มีความแม่นยำ
	MH-Z14A	เป็นเซ็นเซอร์วัด ก๊าซ CO_2 (Carbon Dioxide) แบบ NDIR
	LDR Light Sensor	เป็นตัวต้านทานที่ค่าความต้านทานเปลี่ยนแปลงตามแสงที่ตกกระทบ

3.3 การออกแบบระบบ

3.3.1 การทำงานของส่วนฮาร์ดแวร์

ส่วนฮาร์ดแวร์ของระบบเป็นองค์ประกอบหลักที่ใช้ในการตรวจวัดและควบคุมอุปกรณ์ภายในโรงเรือน โดยมี **บอร์ด ESP32** เป็นไมโครคอนโทรลเลอร์หลัก ทำหน้าที่ประมวลผลและส่งข้อมูลผ่านสัญญาณ **Wi-Fi** เพื่อจัดเก็บและแสดงผลข้อมูลในระบบออนไลน์

บอร์ด **ESP32** จะรับข้อมูลจากเซนเซอร์หลายชนิด ได้แก่

- **DHT22** สำหรับวัดอุณหภูมิและความชื้นภายในโรงเรือน
- **MH-Z14A** สำหรับตรวจวัดระดับก๊าซคาร์บอนไดออกไซด์ (CO_2)
- **LDR (Light Dependent Resistor)** สำหรับตรวจวัดความเข้มของแสง

ข้อมูลจากเซนเซอร์ทั้งหมดจะถูกส่งเข้ามายัง ESP32 เพื่อทำการประมวลผลเบื้องต้น จากนั้นจึงส่งข้อมูลออกไป 2 รูปแบบ ได้แก่

1. **การส่งข้อมูลแบบเรียลไทม์ผ่านโปรโตคอล MQTT** ไปยังหน้า **Web Dashboard** เพื่อแสดงผลทันที และใช้ในการสั่งควบคุมอุปกรณ์ภายในโรงเรือนแบบเรียลไทม์ เช่น การเปิด–ปิดพัดลม วาล์วน้ำ หรือไฟส่องสว่างผ่านระบบออนไลน์
2. **การส่งข้อมูลแบบ HTTP** ไปยัง **Node-RED** เพื่อจัดเก็บในฐานข้อมูล **PostgreSQL** สำหรับใช้ในการวิเคราะห์ภายหลัง

นอกจากนี้ ESP32 ยังเชื่อมต่อกับ **Relay Module** ซึ่งใช้สำหรับควบคุมการทำงานของอุปกรณ์ต่าง ๆ เช่น พัดลมระบายอากาศ วาล์วน้ำ (Solenoid Valve) และไฟ LED เพื่อรักษาสภาพแวดล้อมให้อยู่ในช่วงที่เหมาะสมต่อการเจริญเติบโตของเห็ด

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

ส่วนของ **Arduino Mega 2560** ทำหน้าที่เป็นไมโครคอนโทรลเลอร์รอง ใช้ในการแสดงผลข้อมูลจากเซนเซอร์บนหน้าจอ **TFT Monitor** เพื่อให้ผู้ดูแลสามารถตรวจสอบสถานะของโรงเรือนได้โดยตรง โดยไม่จำเป็นต้องเข้าผ่านระบบออนไลน์ ซึ่งช่วยเพิ่มความสะดวกในกรณีที่เครือข่าย Wi-Fi มีปัญหา

3.3.2 การทำงานของส่วนซอฟต์แวร์

ระบบซอฟต์แวร์ของฟาร์มเห็ดอัจฉริยะถูกพัฒนาในรูปแบบ **Web Application** เพื่อเป็นศูนย์กลางในการแสดงผลข้อมูลและควบคุมอุปกรณ์ภายในโรงเรือน ระบบทำงานบนสถาปัตยกรรม **Web-Based IoT System** ซึ่งสามารถเข้าถึงได้จากอุปกรณ์หลายประเภท ผ่านเครือข่ายอินเทอร์เน็ตหรือ Wi-Fi

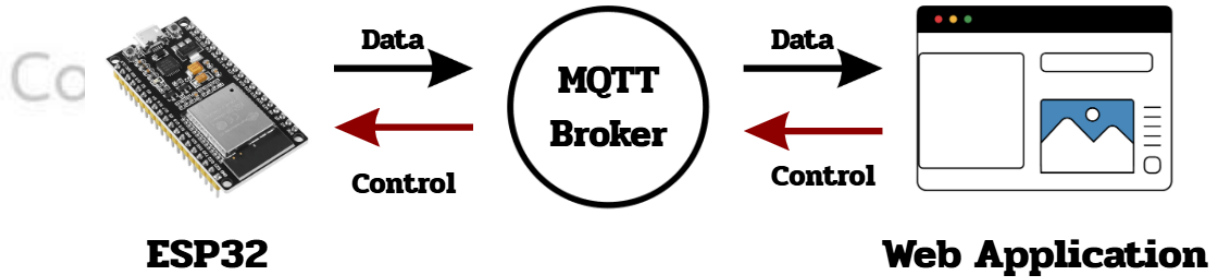
การรับส่งข้อมูลผ่าน HTTP

บอร์ด **ESP32** ส่งข้อมูลจากเซ็นเซอร์ไปยัง **Node-RED** ผ่านโปรโตคอล **HTTP** โดย Node-RED ทำหน้าที่เป็นตัวกลางในการรับข้อมูลและจัดเก็บลงในฐานข้อมูล **PostgreSQL** ในรูปแบบตาราง เพื่อให้สามารถวิเคราะห์และติดตามแนวโน้มข้อมูลย้อนหลังได้

ภาพที่ 10 การส่งข้อมูลไปยัง Node-RED เพื่อจัดเก็บฐานข้อมูล

การรับส่งข้อมูลผ่าน MQTT

ข้อมูลเรียลไทม์สำหรับการแสดงผลบน Web Application และการควบคุมอุปกรณ์จะถูกส่งโดยตรงจาก ESP32 ผ่าน MQTT ทำให้ผู้ใช้สามารถดูสถานะปัจจุบันของโรงเรือนและสั่งควบคุมอุปกรณ์ เช่น พัดลม วาล์วน้ำ หรือไฟส่องสว่าง ผ่านหน้าเว็บได้ทันที



ภาพที่ 11 การรับส่งข้อมูลโดยใช้ MQTT

Web Application

ถูกพัฒนาด้วย HTML, CSS และ JavaScript เพื่อสร้างส่วนติดต่อผู้ใช้ (User Interface) ที่ใช้งานง่าย แสดงข้อมูลจากเซ็นเซอร์ทั้งแบบตัวเลขและกราฟ พร้อมปุ่มควบคุมอุปกรณ์ เช่น พัดลม วาล์วน้ำ และไฟส่องสว่าง เพื่ออำนวยความสะดวกในการจัดการสภาพแวดล้อมภายในโรงเรือน นอกจากนี้ยังมีการประยุกต์ใช้ไลบรารี Leaflet สำหรับแสดงตำแหน่งโรงเรือนบนแผนที่

ซอฟต์แวร์ทั้งหมดทำงานร่วมกันอย่างบูรณาการ ระหว่าง Web Application, Node-RED, Database Management และ IoT Device Control ผ่าน MQTT เพื่อให้ระบบฟาร์มเห็ดอัจฉริยะสามารถตรวจสอบและควบคุมสภาพแวดล้อมได้อย่างแม่นยำ มีประสิทธิภาพ และเข้าถึงข้อมูลจากทุกที่ทุกเวลา

ตารางที่ 2 กำหนดปัจจัยสิ่งแวดล้อมเพื่อควบคุมอุปกรณ์

ปัจจัยสิ่งแวดล้อม	ช่วงค่าที่เหมาะสม	อุปกรณ์ที่เกี่ยวข้อง
อุณหภูมิ	28–32°C	พัดลม / ระบบระเหยน้ำ
ความชื้นสัมพัทธ์	70–90%	ระบบพ่นหมอก / วาล์วน้ำ
ก๊าซคาร์บอนไดออกไซด์	1500–2000 ppm	พัดลมระบายอากาศ

ค่ามาตรฐานเหล่านี้ถูกตั้งไว้ในโปรแกรมของ **ESP32** เพื่อเปรียบเทียบกับค่าที่วัดได้จากเซ็นเซอร์แบบเรียลไทม์ หากพบว่าค่าที่วัดได้อยู่นอกขอบเขตที่กำหนด ระบบจะส่ง **การแจ้งเตือนผ่าน Telegram** ไปยังผู้ดูแล เพื่อให้สามารถตรวจสอบและสั่งเปิด–ปิดอุปกรณ์ที่เกี่ยวข้องด้วยตนเอง

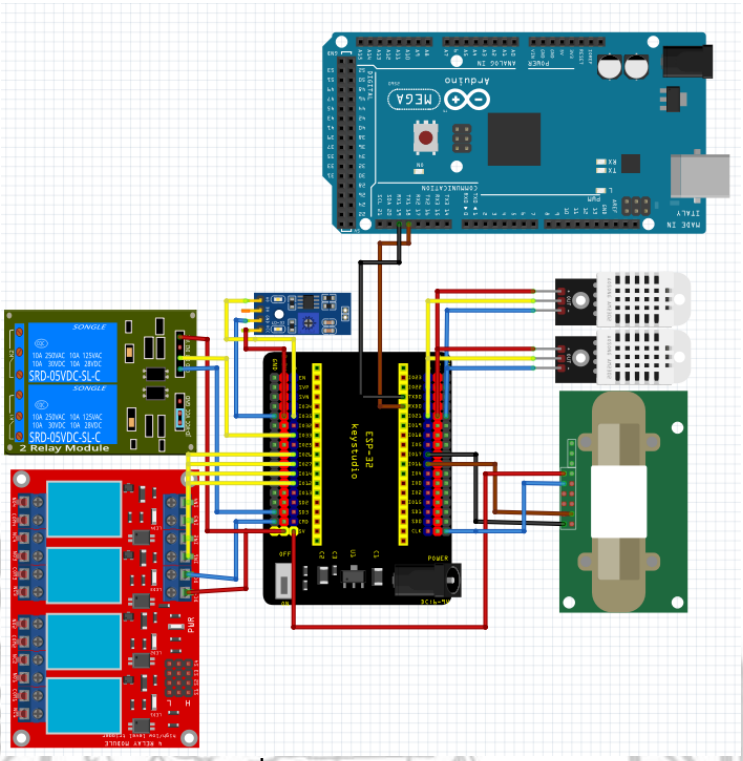
3.3.3 การต่อวงจรเซนเซอร์

GND (Ground) เป็นสายกราวด์หรือจุดอ้างอิงของวงจรไฟฟ้า ทำหน้าที่เป็นศูนย์โวลต์ของระบบ ทุกอุปกรณ์ที่เชื่อมต่อกันต้องต่อ GND ร่วมกันเพื่อให้แรงดันไฟฟ้าที่ส่งสัญญาณมีค่าถูกต้อง

VCC / 5V / 3.3V เป็นสายไฟเลี้ยงให้กับโมดูลหรืออุปกรณ์ โดย VCC หมายถึงไฟบวกของวงจร (Voltage Common Collector) ส่วน 5V และ 3.3V คือแรงดันที่อุปกรณ์ต้องการใช้

TX / RX ใช้สำหรับการสื่อสารแบบอนุกรม (UART) โดย TX (Transmit) เป็นสายส่งข้อมูลจากอุปกรณ์นี้ไปยังอุปกรณ์อื่น ส่วน RX (Receive) เป็นสายรับข้อมูลจากอุปกรณ์อื่น

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved



ภาพที่ 12 การต่อวงจรเซนเซอร์

3.4 การพัฒนาระบบ

3.4.1 การออกแบบและการสร้างฐานข้อมูล

การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลเป็นขั้นตอนสำคัญของการพัฒนาระบบซอฟต์แวร์ เพื่อให้สามารถจัดเก็บและบริหารจัดการข้อมูลได้อย่างเป็นระบบ มีความถูกต้อง และเรียกใช้งานได้อย่างมีประสิทธิภาพ โดยในระบบฟาร์มเห็ดอัจฉริยะนี้ได้เลือกใช้ PostgreSQL เป็นฐานข้อมูลหลัก เนื่องจากเป็นฐานข้อมูลแบบ Open Source ที่มีความเสถียรสูง รองรับการจัดการข้อมูลจำนวนมาก และสามารถทำงานร่วมกับภาษาโปรแกรมต่าง ๆ ได้ดี

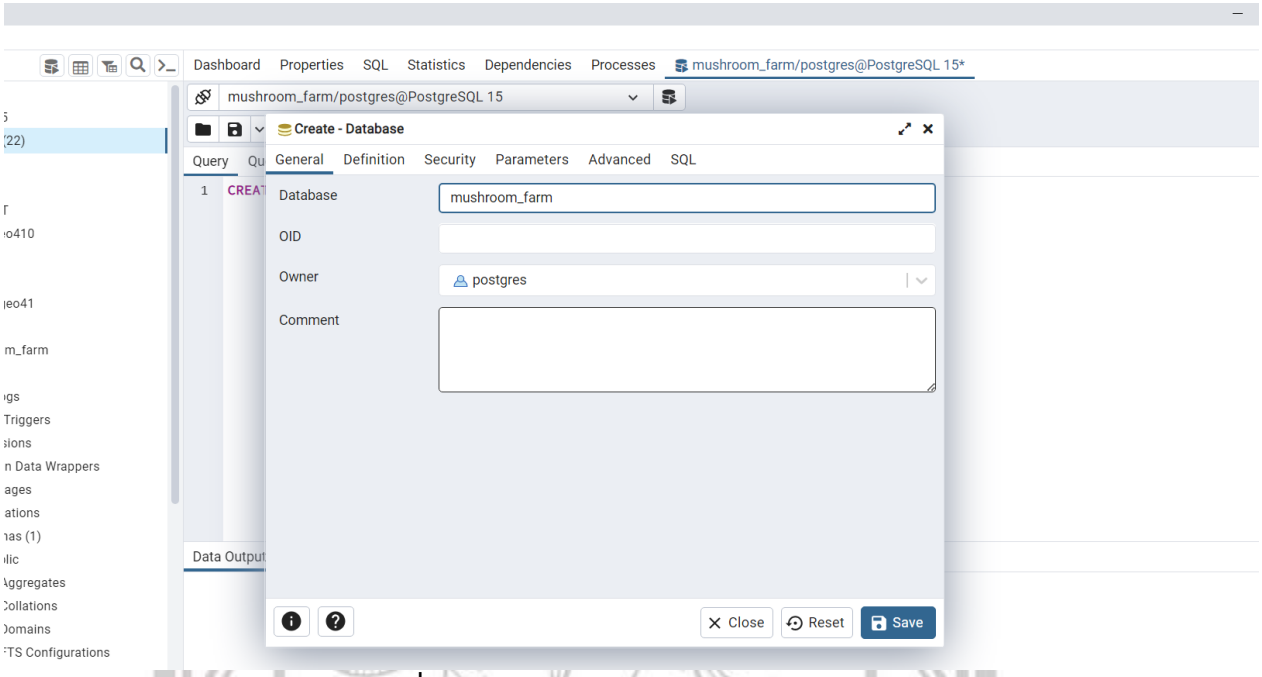
ตารางที่ 3 แสดงรายละเอียดที่จัดเก็บข้อมูลเซนเซอร์

ชื่อตัวแปร	ความหมาย	รูปแบบข้อมูล
id	ลำดับข้อมูล	integer
temp_in	อุณหภูมิภายใน	numeric
temp_out	อุณหภูมิภายนอก	numeric
hum_in	ความชื้นภายใน	numeric
hum_out	ความชื้นภายนอก	numeric
co2	ก๊าซคาร์บอนไดออกไซด์	numeric
idr	แสง	numeric
timestamp	วันที่และเวลา	timestamp without time zone

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

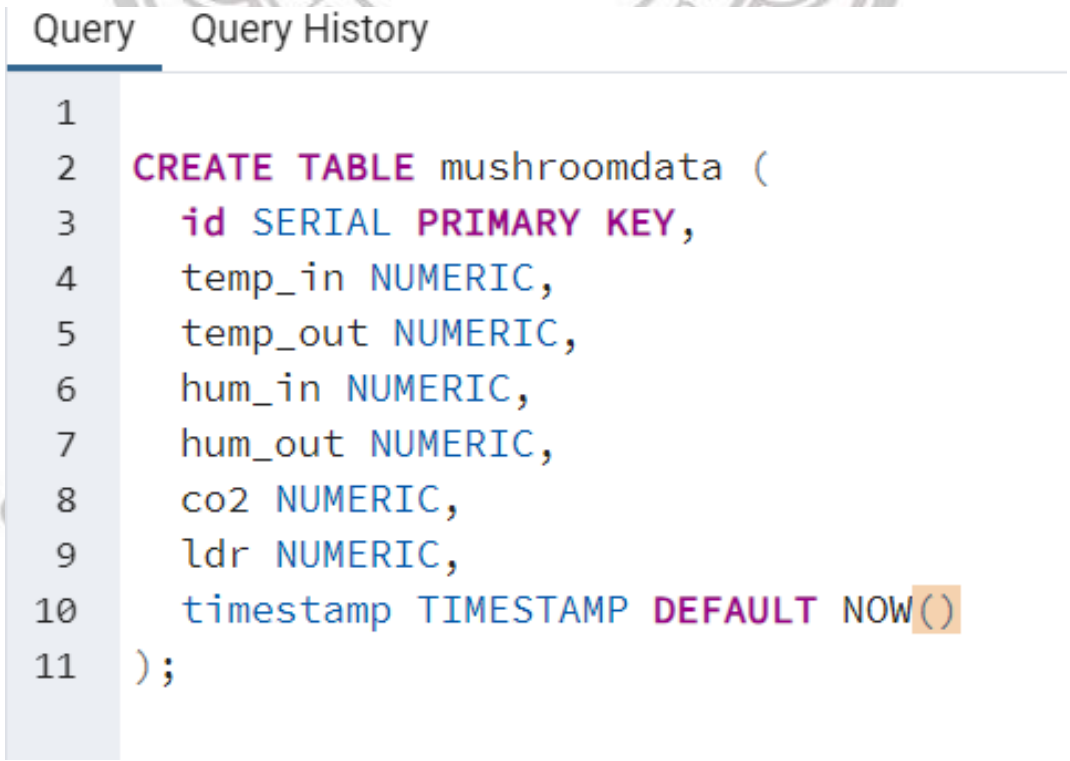
การสร้างฐานข้อมูล

ทำการเข้า PgAdmin4 และทำการ CREATE DATABASE mushroom_farm



ภาพที่ 13 การสร้างฐานข้อมูลบน PgAdmin4

สร้างตารางขึ้นมีชื่อว่า mushroomdata ใช้สำหรับเก็บค่าข้อมูลสิ่งแวดล้อมจากเซ็นเซอร์ เช่น อุณหภูมิ ความชื้น ความเข้มแสง และก๊าซคาร์บอนไดออกไซด์



ภาพที่ 14 การสร้างฐานข้อมูลบน PgAdmin4

ผลที่ได้จากการสร้างตารางฐานข้อมูล

	id [PK] integer	temp_in numeric	temp_out numeric	hum_in numeric	hum_out numeric	co2 numeric	ldr numeric	timestamp timestamp without time zone
1	1	23.9	26.1	52.3	43.2	2813	246	2025-09-29 18:57:22.654443
2	2	23.9	26.1	52.6	43.4	2811	215	2025-09-29 18:57:33.106017
3	3	23.9	26.1	52.5	43.7	2814	214	2025-09-29 18:57:42.724063
4	4	23.9	26.1	52.9	43.9	2810	214	2025-09-29 18:57:52.65928
5	5	23.9	26.1	52.8	43.7	2802	231	2025-09-29 18:58:02.634509
6	6	23.9	26.1	52.4	43.8	2796	215	2025-09-29 18:58:12.728941
7	7	23.9	26.1	52.8	44.3	2798	214	2025-09-29 18:58:22.640895
8	8	23.9	26.1	52.8	44.3	2798	216	2025-09-29 18:58:32.632929
9	9	23.9	26.1	52.9	44.2	2798	215	2025-09-29 18:58:42.731495
10	10	24	26.2	52.5	43.7	2802	217	2025-09-29 18:58:52.664401
11	11	24	26.1	52.1	43.3	2802	216	2025-09-29 18:59:02.699392
12	12	24	26.2	52	43	2810	216	2025-09-29 18:59:12.733554

ภาพที่ 15 การสร้างฐานข้อมูลบน PgAdmin4

3.4.2 การส่งข้อมูลเข้าสู่ฐานข้อมูล

การส่งข้อมูลเข้าสู่ฐานข้อมูลคือการนำค่าที่วัดได้จากเซ็นเซอร์ของ ESP32 ส่งผ่าน HTTP ไปยัง Node-RED ซึ่งทำหน้าที่รับและจัดรูปแบบข้อมูลก่อนส่งต่อไปเก็บใน PostgreSQL เพื่อให้สามารถบันทึก วิเคราะห์ และติดตามแนวโน้มข้อมูลย้อนหลังได้

Arduino IDE

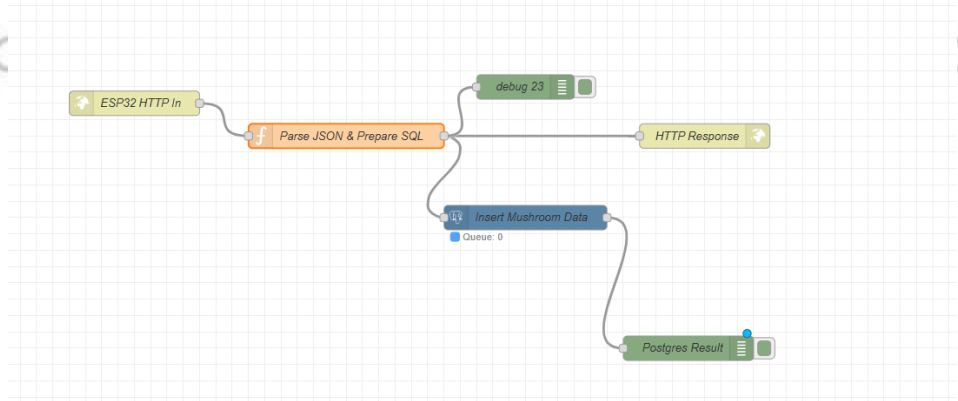
ESP32 อ่านค่าจากเซนเซอร์ จัดข้อมูลให้อยู่ในรูปแบบ JSON ส่งข้อมูลออกไป Node-RED ผ่าน HTPP POST ตาม URL ที่กำหนด

```
const char* NODE_RED_URL = "http://172.20.10.9:1880/mushroom";
```

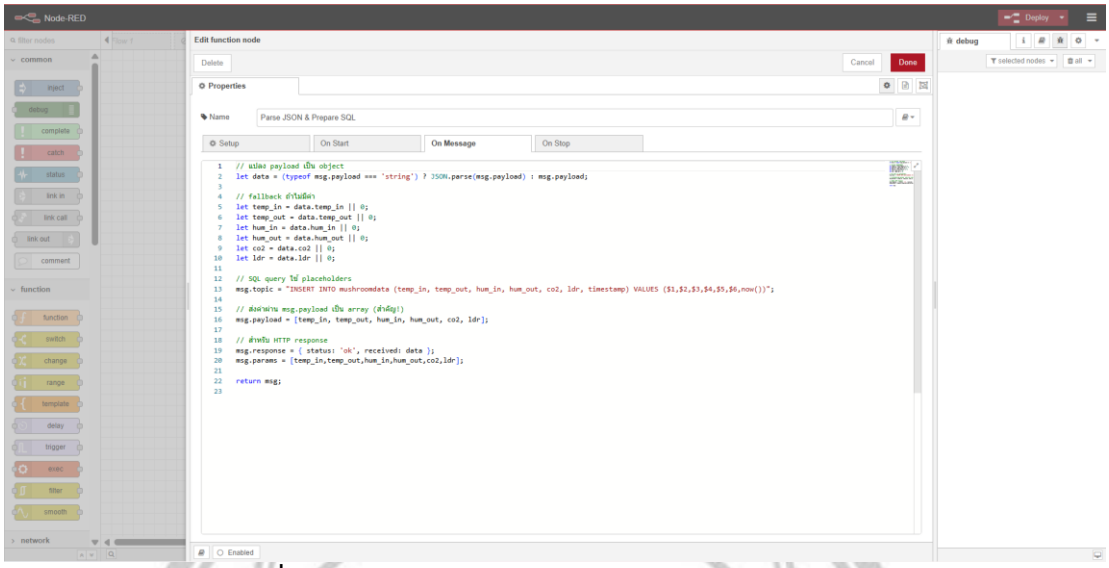
ภาพที่ 16 URL

Node-RED

ทำหน้าที่รับข้อมูลจาก ESP32 แปลง JSON ให้เป็นตัวแปรภายในระบบ และจัดเตรียมคำสั่ง SQL สำหรับการบันทึกลงฐานข้อมูล PostgreSQL รวมถึงตรวจสอบความสมบูรณ์ของข้อมูล เช่น กำหนดค่า fallback หากเซ็นเซอร์ส่งค่าผิดพลาด



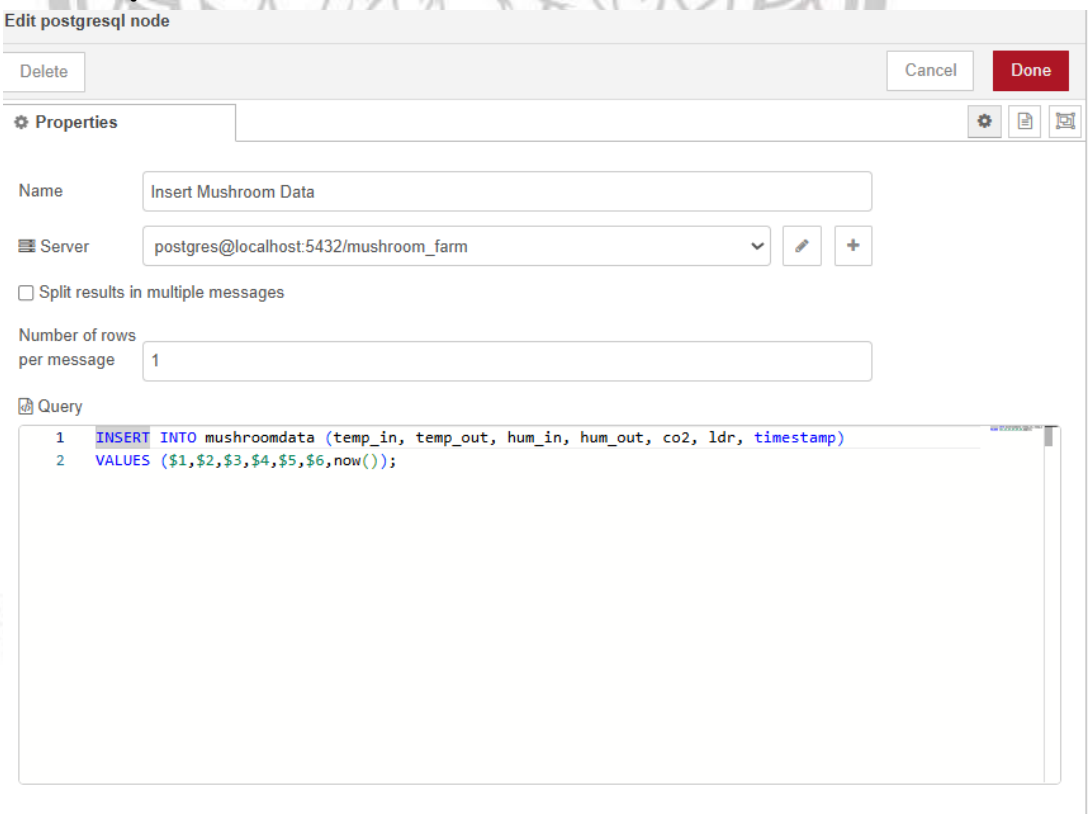
ภาพที่ 17 Flow Node-RED



ภาพที่ 18 function node “Parse JSON & Prepare SQL”

PostgreSQL

Node-RED ส่งข้อมูลไปยัง PostgreSQL โดย insert ลงตาราง mushroomdata พร้อม timestamp เพื่อให้สามารถนำข้อมูลไปวิเคราะห์แนวโน้มย้อนหลังได้



ภาพที่ 19 SQL Query

3.4.3 การพัฒนา Web Application

ระบบ Web Application ถูกพัฒนาขึ้นเพื่อเป็นส่วนกลางในการแสดงผลข้อมูลจากเซนเซอร์และควบคุมอุปกรณ์ภายในโรงเรือนเพาะเห็ด โดยใช้ภาษา PHP, HTML, CSS, และ JavaScript ในการพัฒนา และใช้ Apache เป็น Web Server สำหรับรับ–ส่งข้อมูลระหว่าง ESP32 กับผู้ใช้งานหน้า Dashboard

ส่วนประกอบของ Web Application

Web Application ประกอบด้วยหน้าหลักดังนี้

- หน้า Dashboard

  แสดงค่าจากเซนเซอร์แบบเรียลไทม์ ได้แก่ อุณหภูมิ ความชื้น CO_2 และความเข้มแสง พร้อมกราฟแนวโน้มย้อนหลัง โดยข้อมูลเรียลไทม์มาจาก MQTT ของ ESP32 ส่วนข้อมูลย้อนหลังมาจากฐานข้อมูล PostgreSQL

- หน้า Control

  ใช้ควบคุมอุปกรณ์ในโรงเรือน เช่น การเปิด–ปิดพัดลม ระบบพ่นหมอก และไฟส่องสว่าง โดยส่งคำสั่งผ่าน MQTT ไปยัง ESP32 ทำให้สามารถตอบสนองได้ทันที

- หน้า Map (Leaflet)

  แสดงตำแหน่งโรงเรือนเพาะเห็ดบนแผนที่ และสามารถเชื่อมโยงกับระบบภูมิสารสนเทศ (GIS) เพื่อใช้วิเคราะห์เชิงพื้นที่

การรับข้อมูลเรียลไทม์และอัปเดต Dashboard (JavaScript)

สร้าง MQTT client เชื่อมต่อ broker และ subscribe ข้อมูลสถานะจาก ESP32

```
const client = mqtt.connect(`ws://${broker}:${port}/mqtt`, { clientId });

client.on("connect", () => {
  console.log("MQTT connected");
  client.subscribe(topicStatus);
});
```

ภาพที่ 20 เชื่อมต่อกับ MQTT broker

Copyright by Naresuan University

```
client.on("message", (topic, message) => {
  if (topic === topicStatus) {
    const data = message.toString().split(",");
    data.forEach((p) => {
      const [key, val] = p.split(":");
      const value = parseFloat(val);
```

ภาพที่ 21 รับข้อความจาก ESP32 แล้วอัปเดตค่าเซ็นเซอร์บน Dashboard

อธิบายการทำงาน:

เชื่อมต่อกับ MQTT broker (broker.emqx.io) ผ่าน WebSocket

subscribe topic "farmingmushroom/status" เพื่อรับค่าจาก ESP32

เมื่อมีข้อความใหม่เข้ามา จะ แยกค่า ของแต่ละเซ็นเซอร์

นำค่าที่ได้ไป อัปเดต HTML element ของ Dashboard

การควบคุมอุปกรณ์ผ่าน Control Panel

ผู้ใช้สามารถ คลิกปุ่ม toggle เพื่อสั่งเปิด/ปิดอุปกรณ์ เช่น พัดลม, วาล์วน้ำ, LED โดยโค้ดจะส่งคำสั่งผ่าน MQTT กลับไปยัง ESP32

```
// --- Toggle buttons for all devices ---
["SOL", "EVAP", "FAN", "LED", "SOL2"].forEach((r) => {
  const el = document.getElementById(r);
  if (!el) {
    console.error(`Element ${r} not found`);
    return;
  }

  el.addEventListener("click", () => {
    el.classList.toggle("on"); // Update local state
    const action = el.classList.contains("on") ? r + "_ON" : r + "_OFF";

    if (client.connected()) {
      client.publish(topicControl, action);
      console.log("MQTT published:", action);
    } else {
      console.error("MQTT not connected!");
    }
  });
});
```

ภาพที่ 22 function ปุ่ม เปิด/ปิด

อธิบายการทำงาน:

เพิ่ม Event Listener ให้กับปุ่ม toggle ของแต่ละอุปกรณ์

เมื่อผู้ใช้คลิก จะ **สลับสถานะปุ่ม** และสร้างข้อความคำสั่ง (เช่น "FAN_ON")

ส่งคำสั่งไปยัง ESP32 ผ่าน MQTT topic "farmingmushroom/control"

ESP32 จะรับคำสั่งแล้ว **เปิด/ปิดอุปกรณ์จริง**

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

การดึงข้อมูลย้อนหลังจากฐานข้อมูล (PHP)

ฟังก์ชันนี้ใช้ดึงข้อมูลเซ็นเซอร์ย้อนหลังจาก **PostgreSQL** เพื่อแสดงเป็นกราฟบน Dashboard

history_sensor.php

```php
<?php
header('Content-Type: application/json');
error_reporting(0);

$host='localhost'; $port='5432'; $dbname='mushroom_farm'; $user='postgres'; $password='postgres';
$conn = pg_connect("host=$host port=$port dbname=$dbname user=$user password=$password");
if(!$conn){ echo json_encode([]); exit; }

$sensor=$_GET['sensor']??'temp_in';
$period=$_GET['period']??'3d';
$days=$period==='3d'?3:7;

$allowed_sensors=['temp_in','temp_out','hum_in','hum_out','co2','ldr'];
if(!in_array($sensor,$allowed_sensors)){ echo json_encode([]); exit; }

$query="
SELECT date_trunc('hour', timestamp) AS ts_hour, AVG($sensor) AS value
FROM mushroomdata
WHERE timestamp >= NOW() - INTERVAL '$days days'
GROUP BY ts_hour
ORDER BY ts_hour ASC
";
$result=pg_query($conn,$query);
if(!$result){ echo json_encode([]); exit; }

$data=[];
while($row=pg_fetch_assoc($result)){
    $data[]= [
        'ts_hour'=>date('Y-m-d H:i',strtotime($row['ts_hour'])),
        'value'=>floatval($row['value'])
    ];
}
echo json_encode($data);
pg_close($conn);
```

ภาพที่ 23 การดึงข้อมูลหลังจาก Postgres

อธิบายการทำงาน:

รับค่าพารามิเตอร์ sensor และ period จาก URL

กำหนดช่วงเวลา (3 วันหรือ 7 วัน) สำหรับดึงข้อมูล

ดึงค่าเฉลี่ยของเซ็นเซอร์ ต่อชั่วโมง จากตาราง mushroomdata

ส่งกลับข้อมูลในรูปแบบ JSON เพื่อให้ JavaScript นำไปสร้างกราฟ

3.4.4 การพัฒนาโปรแกรมไมโครคอนโทรลเลอร์

ในส่วนของการพัฒนาโปรแกรมควบคุมไมโครคอนโทรลเลอร์ ESP32 สำหรับระบบฟาร์มเห็ดอัจฉริยะ ได้มีการใช้ ภาษา C++ ร่วมกับ Arduino Framework เพื่อให้สามารถอ่านค่าเซ็นเซอร์, ควบคุมอุปกรณ์, ส่งข้อมูลแบบเรียลไทม์ และแจ้งเตือนผู้ดูแลผ่าน Telegram

โค้ดหลักถูกแบ่งออกเป็นหลายส่วน ได้แก่ การเชื่อมต่อ Wi-Fi, การอ่านค่าเซ็นเซอร์, การควบคุมรีเลย์, การส่งข้อมูลผ่าน MQTT และ HTTP, และการแจ้งเตือนผ่าน Telegram เพื่อให้ระบบสามารถทำงานอัตโนมัติและสามารถมอนิเตอร์แบบเรียลไทม์ได้

ชุดคำสั่งข้อมูล

ชุดคำสั่งในโค้ดของ ESP32 มีหน้าที่หลักในการอ่านค่าเซ็นเซอร์ เช่น DHT22 สำหรับวัดอุณหภูมิและความชื้น, LDR สำหรับวัดความเข้มของแสง และ MH-Z14A สำหรับวัดความเข้มข้นของ CO_2 จากนั้นโปรแกรมจะทำการประมวลผลค่าที่อ่านได้ เช่น การแปลงสัญญาณจาก LDR เป็นค่าความสว่าง (lux) หรือกรองค่าที่ผิดพลาด (NaN) เพื่อใช้ค่าเดิมแทน การตรวจสอบเงื่อนไข Threshold ทำให้ ESP32 สามารถ

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

ตัดสินใจควบคุมรีเลย์ต่าง ๆ เช่น เปิด–ปิดพัดลม, ระบบพ่นหมอก, ไฟส่องสว่าง หรือวาล์วน้ำ โดยอัตโนมัติเพื่อรักษาสภาพแวดล้อมที่เหมาะสมสำหรับการเพาะเห็ด

ข้อมูลที่อ่านและประมวลผลแล้วจะถูกส่งออกไปสองช่องทาง คือแบบเรียลไทม์ผ่านโปรโตคอล MQTT ไปยัง Web Dashboard เพื่อให้ผู้ดูแลสามารถติดตามสถานะและควบคุมอุปกรณ์ได้ทันที และแบบ HTTP POST ไปยัง Node-RED เพื่อจัดเก็บในฐานข้อมูล PostgreSQL สำหรับวิเคราะห์และสร้างกราฟย้อนหลัง

ตารางที่ 4 ชุดคำสั่ง

#include <WiFi.h>	ใช้สำหรับเชื่อมต่อ ESP32 เข้ากับเครือข่าย Wi-Fi
#include <WiFiClientSecure.h>	ใช้สำหรับสร้าง client ที่เชื่อมต่อแบบปลอดภัย
#include <PubSubClient.h>	ใช้สำหรับ MQTT protocol เพื่อส่งและรับข้อมูล
#include <DHT.h>	อ่านค่าจากเซ็นเซอร์ DHT22
#include <HardwareSerial.h>	ใช้สำหรับสื่อสารผ่านพอร์ตอนุกรม (UART)
#include <UniversalTelegramBot.h>	ใช้ส่งข้อความแจ้งเตือนผ่าน Telegram Bot API
#include <HTTPClient.h>	ใช้สำหรับส่ง HTTP POST/GET ไปยัง Node-RED

การทำงานของ ESP32

บอร์ด ESP32 ทำหน้าที่เป็นหน่วยประมวลผลหลักของระบบ (Main Controller) ซึ่งเชื่อมต่อกับเซ็นเซอร์ต่าง ๆ ภายในโรงเรือนเพาะเห็ด ได้แก่

- DHT22 สำหรับตรวจวัดอุณหภูมิและความชื้นภายในและภายนอกโรงเรือน
- MH-Z14A สำหรับตรวจวัดระดับก๊าซคาร์บอนไดออกไซด์ (CO_2)
- LDR สำหรับตรวจวัดระดับแสงภายในโรงเรือน

ลิขสิ
Copyrig

```
// ---------------- Sensors / Pins -
#define DHTPIN_IN   21
#define DHTPIN_OUT  22
#define DHTTYPE     DHT22
DHT dht_in(DHTPIN_IN, DHTTYPE);
DHT dht_out(DHTPIN_OUT, DHTTYPE);

#define LDR_PIN 35 // ADC pin
```

ภาพที่ 24 การกำหนดขาเซนเซอร์

เมื่อ ESP32 ได้รับค่าจากเซ็นเซอร์แล้ว จะนำข้อมูลที่ได้มาประมวลผลเบื้องต้น โดยเปรียบเทียบกับค่า Threshold ที่กำหนดไว้ในโปรแกรม เช่น หากอุณหภูมิสูงเกินค่าที่ตั้ง ระบบจะสั่งเปิดพัดลมเพื่อระบายอากาศ หรือหากความชื้นต่ำกว่ากำหนด จะสั่งเปิดระบบพ่นหมอก เป็นต้น

```
// ---------------- Thresh
#define TEMP_HIGH 32
#define TEMP_LOW  25
#define HUM_LOW   70
#define HUM_HIGH  85
#define CO2_HIGH  1500
#define CO2_LOW   1400
```

ภาพที่ 25 การกำหนดค่าปัจจัยสภาพแวดล้อม

ESP32 ยังทำหน้าที่สื่อสารข้อมูลออกไปสองช่องทางหลัก ได้แก่

โปรโตคอล MQTT – สำหรับส่งข้อมูลแบบเรียลไทม์ไปยัง Web Application (Dashboard) เพื่อให้ผู้ใช้งานสามารถมอนิเตอร์ค่าต่าง ๆ ได้จากระยะไกล รวมถึงควบคุมอุปกรณ์ผ่านเว็บได้ทันที เช่น เปิด–ปิดพัดลมหรือวาล์วน้ำ

โปรโตคอล HTTP – สำหรับส่งข้อมูลไปยัง Node-RED เพื่อบันทึกลงฐานข้อมูล PostgreSQL เพื่อเก็บเป็นข้อมูลย้อนหลังสำหรับการวิเคราะห์แนวโน้ม

```
const char* WIFI_SSID = "Kop";
const char* WIFI_PASS = "66666666";

const char* NODE_RED_URL = "http://172.20.10.9:1880/mushroom";

// MQTT (public broker example)
const char* MQTT_SERVER = "broker.emqx.io";
const uint16_t MQTT_PORT = 1883;
const char* MQTT_TOPIC_STATUS  = "farmingmushroom/status";
const char* MQTT_TOPIC_CONTROL = "farmingmushroom/control";
```

All rights reserved

ภาพที่ 26 MQTT และ HTTP

นอกจากนี้ ESP32 ยังสามารถส่งการแจ้งเตือนไปยัง Telegram เมื่อค่าที่ตรวจวัดได้เกินขอบเขตที่กำหนดไว้ เพื่อให้ผู้ดูแลสามารถตอบสนองต่อเหตุการณ์ได้อย่างรวดเร็ว ระบบนี้จึงทำให้การควบคุมโรงเรือนมีความแม่นยำอย่างแท้จริง

3.4.5 การสร้างระบบแจ้งเตือน

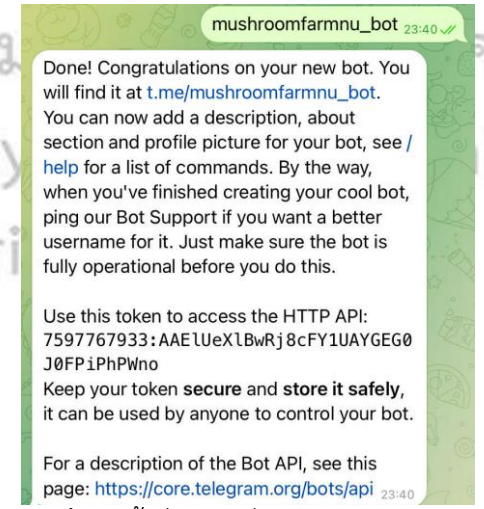
ระบบแจ้งเตือนถูกออกแบบมาเพื่อให้ผู้ดูแลสามารถรับทราบสถานะความผิดปกติภายในโรงเรือนเพาะเห็ดได้แบบเรียลไทม์ โดยใช้ Telegram Bot API เป็นช่องทางสื่อสารหลักระหว่างระบบและผู้ใช้งาน

ในส่วนของ ESP32 จะเป็นหน่วยตรวจสอบค่าจากเซ็นเซอร์ เช่น อุณหภูมิ ความชื้น และระดับก๊าซคาร์บอนไดออกไซด์ (CO_2) อยู่ตลอดเวลา จากนั้นโปรแกรมจะทำการเปรียบเทียบค่าที่อ่านได้กับค่า Threshold ที่กำหนดไว้ในระบบ หากตรวจพบว่าค่าที่ได้เกินหรือต่ำกว่าค่ามาตรฐาน ระบบจะทำการส่งข้อความแจ้งเตือนอัตโนมัติไปยังบัญชี Telegram ของผู้ดูแลผ่าน Telegram Bot ที่สร้างขึ้น โดยมีข้อความระบุสาเหตุของความผิดปกติและค่าปัจจุบันของเซ็นเซอร์นั้น ๆ

การสร้าง Telegram Bot และรับ Token

เปิดแอปพลิเคชัน Telegram แล้วค้นหาชื่อผู้ใช้ @BotFather ซึ่งเป็นบอทหลักของ Telegram สำหรับสร้างบอทอื่นๆ พิมพ์คำสั่ง /newbot เพื่อเริ่มการสร้างบอทใหม่ จากนั้นตั้งชื่อบอท เช่น “MushroomFarmBot” และตั้งชื่อผู้ใช้ ที่ลงท้ายด้วยคำว่า bot เช่น mushroomfarmnu_bot

ภาพที่ 27 การใช้คำสั่ง /newbot



ภาพที่ 28 ตั้งชื่อผู้ใช้ ที่ลงท้ายด้วยคำว่า bot

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

เมื่อสร้างสำเร็จ BotFather จะส่งข้อความตอบกลับมาพร้อมกับ **Token** ซึ่งเป็นรหัสลับที่ใช้เชื่อมต่อระหว่างโปรแกรม ESP32 และ Telegram Bot ให้บันทึก Token ดังกล่าวไว้เพื่อใช้ในโปรแกรม ESP32 ภายหลัง เปิดห้องแชตกับบอทที่สร้าง แล้วพิมพ์คำสั่ง /start เพื่อเปิดการทำงานของบอท

การเชื่อมต่อ Telegram Bot เข้ากับ ESP32

ในฝั่งของ **ESP32** จะเขียนโปรแกรมโดยใช้ภาษา C++ ผ่าน Arduino IDE เพื่อเชื่อมต่อกับ Wi-Fi และส่งข้อความไปยัง Telegram ผ่าน **Telegram Bot API**

ในฝั่งของ **ESP32** จะเขียนโปรแกรมโดยใช้ภาษา C++ ผ่าน Arduino IDE เพื่อเชื่อมต่อกับ Wi-Fi และส่งข้อความไปยัง Telegram ผ่าน Telegram Bot API

การเชื่อมต่อใช้ไลบรารีสำคัญคือ

- WiFi.h สำหรับเชื่อมต่อ Wi-Fi
- WiFiClientSecure.h สำหรับการเชื่อมต่อแบบ HTTPS
- UniversalTelegramBot.h สำหรับสื่อสารกับ Telegram Bot

```
// Telegram (ใส่ค่าแทนที่นี้)
#define BOT_TOKEN "7597767933:AAElUeXlBwRj8cFY1UAYGEG0J0FPiPhPWno"   /,
#define CHAT_ID   "7700755118"      // <-- แทนด้วย chat id ของคุณ
```

ภาพที่ 29 Token และ ChatID

```
// Build alert message based on thresholds
String alertMsg = "";
if (temp_in > TEMP_HIGH) alertMsg += "⚠ Temp IN HIGH: " + String(temp_in,1) + " C\n";
if (temp_in < TEMP_LOW)  alertMsg += "⚠ Temp IN LOW: "  + String(temp_in,1) + " C\n";
if (hum_in > HUM_HIGH)   alertMsg += "⚠ Hum IN HIGH: "   + String(hum_in,1)  + " %\n";
if (hum_in < HUM_LOW)    alertMsg += "⚠ Hum IN LOW: "    + String(hum_in,1)  + " %\n";
if (co2 > CO2_HIGH)      alertMsg += "⚠ CO2 HIGH: "      + String(co2)    + " ppm\n";
if (co2 < CO2_LOW && co2 >= 0) alertMsg += "⚠ CO2 LOW: " + String(co2) + " ppm\n";

// Send Telegram alert if conditions and not sent recently
if (alertMsg.length() > 0 && (now - lastAlert > ALERT_INTERVAL)) {
  Serial.println("Sending Telegram alert...");
  bool sent = bot.sendMessage(CHAT_ID, alertMsg, "");
  if (sent) {
    Serial.println("Telegram alert sent.");
    lastAlert = now;
  } else {
    Serial.println("Telegram send failed.");
  }
}
```

ภาพที่ 30 การกำหนดการแจ้งเตือนและส่งแจ้งเตือน

All rights reserved

บทที่ 4
ผลการดำเนินงาน

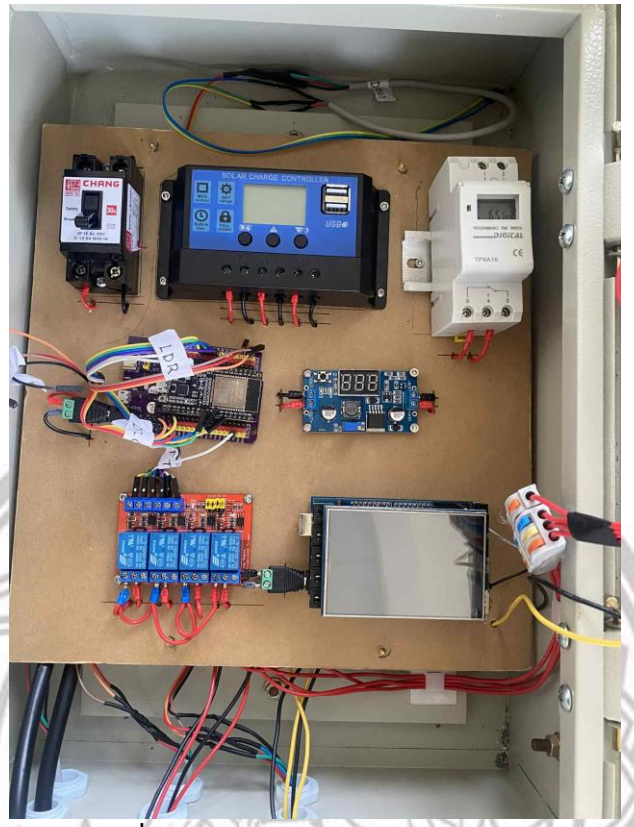
บทนี้นำเสนอผลการดำเนินงานของระบบฟาร์มเห็ดอัจฉริยะที่พัฒนาขึ้น โดยมีวัตถุประสงค์เพื่อสร้างระบบที่สามารถตรวจวัดและแสดงผลข้อมูลสภาพแวดล้อมภายในโรงเรือนเพาะเห็ดได้แบบเรียลไทม์ รวมถึงสามารถควบคุมอุปกรณ์ภายในโรงเรือนผ่าน Web Application และแจ้งเตือนผู้ใช้งานเมื่อค่าที่วัดได้เกินกว่าช่วงที่กำหนด ระบบนี้ถูกออกแบบให้ใช้งานง่าย มีความยืดหยุ่น และสามารถประยุกต์ใช้งานได้จริงในภาคการเกษตร

4.1 ผลการออกแบบอุปกรณ์เซนเซอร์

ภายในตู้ควบคุมจะมีการมีการใช้ Solar Charger เป็นตัวรองรับพลังงานโซล่าเซลล์และจ่ายไฟให้แบตเตอรี่ขนาด 12 โวล แบตเตอรี่มีหน้าที่จ่ายไฟให้ระบบโดยมี Timer เป็นตัวรีเซตระบบป้องกันระบบไม่เสถียร ต่อมาจะมีตัวแปลงไฟจาก 12 โวล แปลงเป็น 5 โวล เพื่อจ่ายไฟเข้าบอร์ด ESP32 เนื่องจาก ESP32 ไม่รองรับไฟเกิน 5โวล ส่วนบอร์ด Arduino MEGA สามารถรองรับการจ่ายไฟ 12 โวลได้จึงไม่ต้องแปลงไฟ และยังมีรีเลย์โมดูลเอาไว้ควบคุมอุปกรณ์ต่างๆ ในโรงเรือน ในส่วนของหน้าตู้ควบคุมจะมีไฟแสดงสถานะเมื่ออุปกรณ์ในโรงเรือนทำงาน ในส่วนของสวิตช์จะเป็น 3 ทาง เพื่อควบคุมอุปกรณ์ในโรงเรือน

ภาพที่ 31 หน้าตู้ควบคุมอุปกรณ์เซนเซอร์

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University



ภาพที่ 32 ภายในตู้ควบคุมอุปกรณ์เซนเซอร์

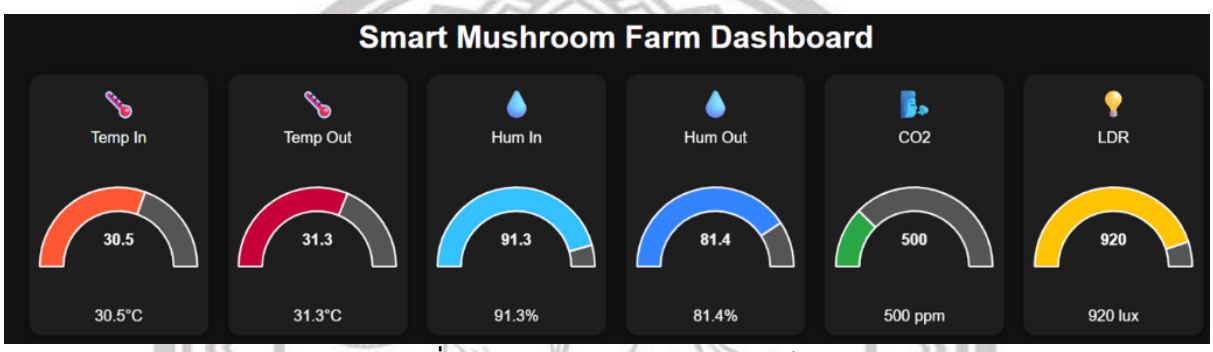
ภาพที่ 33 ผลการเชื่อมต่อ ESP32 ไปยัง Arduino MEGA

4.2 ผลการพัฒนา Web Application

เป็นส่วนแสดงผลข้อมูลและควบคุมอุปกรณ์ผ่านเครือข่ายอินเทอร์เน็ต ผู้ใช้สามารถตรวจสอบค่าจากเซ็นเซอร์แบบเรียลไทม์ สั่งเปิด–ปิดอุปกรณ์ได้ รวมถึงดูแนวโน้มข้อมูลย้อนหลังที่จัดเก็บในฐานข้อมูล PostgreSQL

การแสดงผลข้อมูลเซ็นเซอร์แบบเรียลไทม์

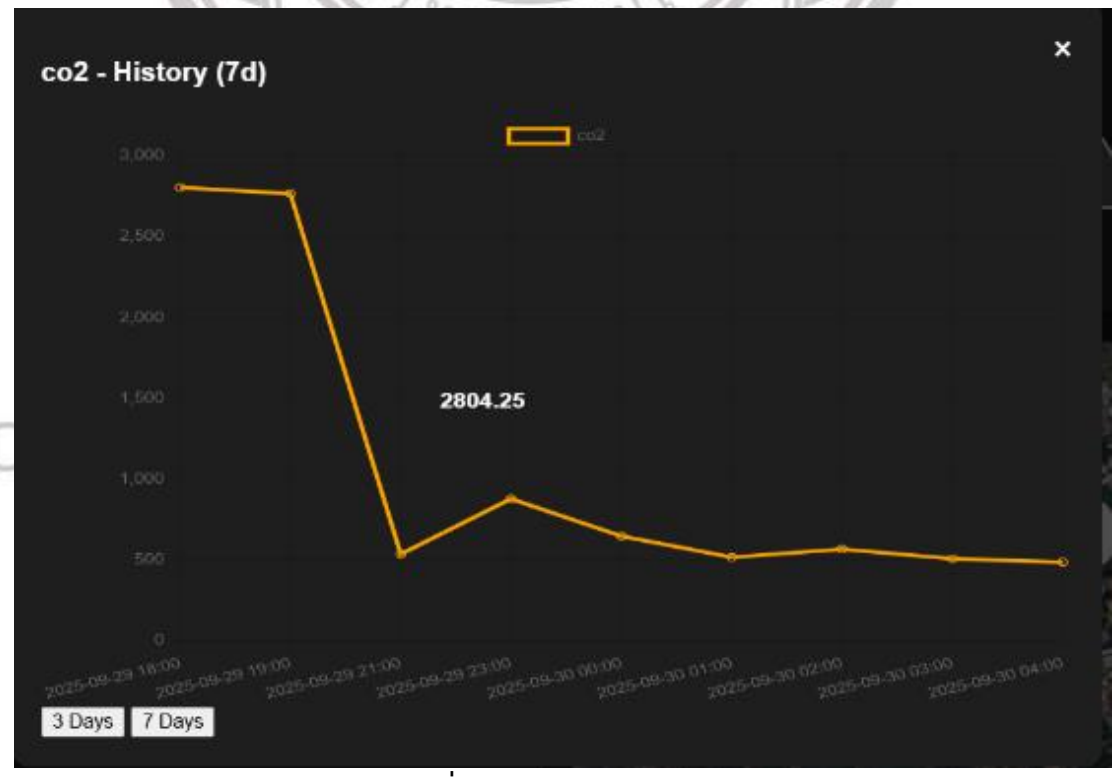
ข้อมูลจากเซ็นเซอร์ เช่น อุณหภูมิ, ความชื้น, CO_2 และแสง ถูกส่งผ่าน MQTT Broker ไปยังเว็บแอปพลิเคชัน โดย Dashboard แสดงผลแบบเรียลไทม์ทั้งในรูปกราฟวงกลมและการ์ดค่าตัวเลข ทำให้ผู้ใช้สามารถติดตามสถานะโรงเรือนได้ทันที



ภาพที่ 34 แสดงผลเซนเซอร์แบบเรียลไทม์

การเรียกดูข้อมูลย้อนหลัง

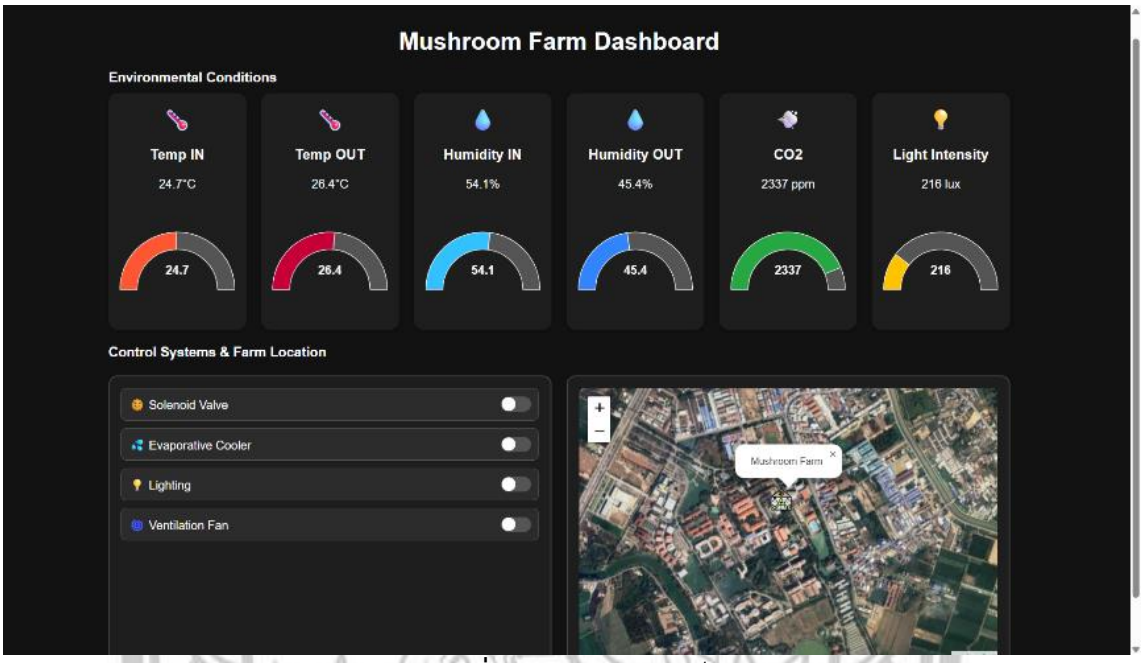
ระบบสามารถเลือกดูข้อมูลย้อนหลัง 3 วัน หรือ 7 วัน โดยกราฟแสดงค่าเฉลี่ยรายชั่วโมง ช่วยลดความผันผวนของข้อมูลดิบและแสดงแนวโน้มของสภาพแวดล้อมได้อย่างชัดเจน



ภาพที่ 35 กราฟข้อมูลย้อนหลัง

การแสดงผลบนแผนที่ (GIS)

ใช้ Leaflet Map ร่วมกับ Google Satellite แสดงตำแหน่งโรงเรือนเพาะเห็ดบนแผนที่ ช่วยให้ผู้ใช้สามารถมองเห็นภาพรวมของพื้นที่ฟาร์มได้



ภาพที่ 36 ภาพรวมหน้าเว็บ

ผลการจัดเก็บข้อมูลลงฐานข้อมูล (PostgreSQL)

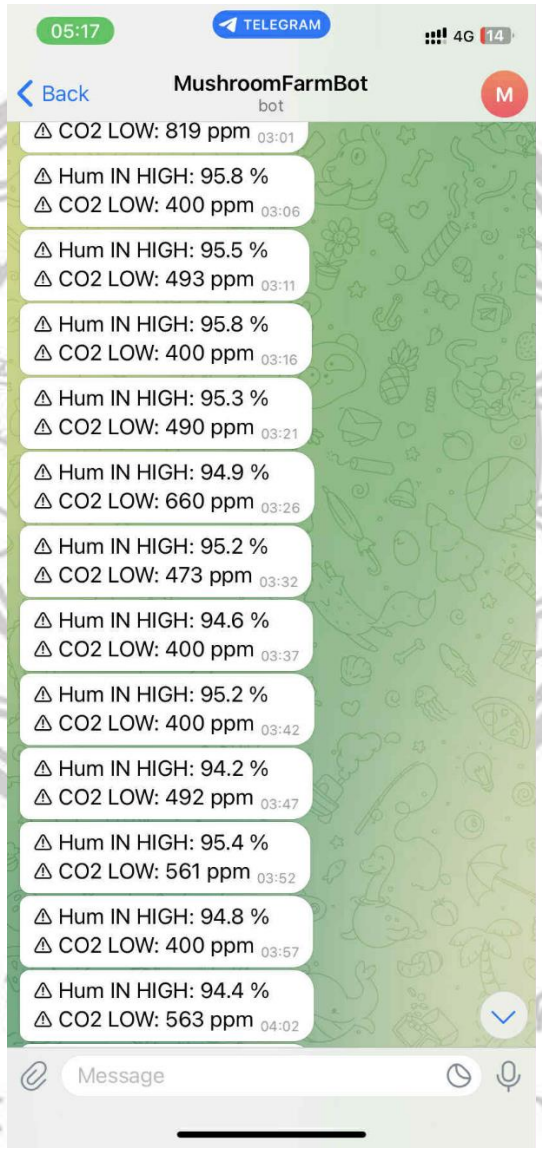
ข้อมูลจาก ESP32 ถูกจัดเก็บลงตาราง mushroomdata โดยมี timestamp กำกับทุกครั้งที่บันทึก ทำให้สามารถย้อนกลับไปวิเคราะห์ข้อมูลเชิงเวลา (time series) ได้ เช่น การเปลี่ยนแปลงอุณหภูมิหรือความชื้นในแต่ละช่วง

	id [PK] integer	temp_in numeric	temp_out numeric	hum_in numeric	hum_out numeric	co2 numeric	ldr numeric	timestamp timestamp without time zone
1	1	23.9	26.1	52.3	43.2	2813	246	2025-09-29 18:57:22.654443
2	2	23.9	26.1	52.6	43.4	2811	215	2025-09-29 18:57:33.106017
3	3	23.9	26.1	52.5	43.7	2814	214	2025-09-29 18:57:42.724063
4	4	23.9	26.1	52.9	43.9	2810	214	2025-09-29 18:57:52.65928
5	5	23.9	26.1	52.8	43.7	2802	231	2025-09-29 18:58:02.634509
6	6	23.9	26.1	52.4	43.8	2796	215	2025-09-29 18:58:12.728941
7	7	23.9	26.1	52.8	44.3	2798	214	2025-09-29 18:58:22.640895
8	8	23.9	26.1	52.8	44.3	2798	216	2025-09-29 18:58:32.632929
9	9	23.9	26.1	52.9	44.2	2798	215	2025-09-29 18:58:42.731495
10	10	24	26.2	52.5	43.7	2802	217	2025-09-29 18:58:52.664401
11	11	24	26.1	52.1	43.3	2802	216	2025-09-29 18:59:02.699392
12	12	24	26.2	52	43	2810	216	2025-09-29 18:59:12.733554
13	13	24	26.2	52.2	43	2807	216	2025-09-29 18:59:22.773562
14	14	24	26.2	52.4	43.2	2803	215	2025-09-29 18:59:32.646857
15	15	23.9	26.2	52.5	43.4	2803	217	2025-09-29 18:59:44.448259
16	16	23.9	26.2	52.8	43.7	2801	215	2025-09-29 18:59:52.669083
17	17	23.9	26.2	53	43.9	2800	214	2025-09-29 19:00:02.810805
18	18	23.9	26.2	53	43.7	2795	218	2025-09-29 19:00:12.638973
19	19	24	26.2	52.6	43.8	2790	215	2025-09-29 19:00:22.674071
20	20	23.9	26.2	52.8	43.6	2782	216	2025-09-29 19:00:32.711199

ภาพที่ 37 ผลการจัดเก็บข้อมูลใน PostgresSQL

ผลการแจ้งเตือนผ่าน Telegram

ผลการแจ้งเตือนผ่านระบบ Telegram Bot เป็นฟีเจอร์สำคัญที่ช่วยให้ผู้ใช้สามารถติดตามสภาพแวดล้อมภายในโรงเรือนเห็ดได้แบบเรียลไทม์ โดยระบบจะทำหน้าที่ตรวจสอบค่าของปัจจัยต่าง ๆ เช่น ความชื้นภายใน (Hum IN) และระดับก๊าซคาร์บอนไดออกไซด์ (CO_2) หากค่าที่วัดได้มีแนวโน้มเกินค่ามาตรฐานที่ตั้งไว้ ระบบจะส่ง ข้อความแจ้งเตือนอัตโนมัติ ไปยังผู้ใช้ทันทีผ่านแอปพลิเคชัน Telegram



ภาพที่ 38 ผลการแจ้งเตือนผ่าน Telegram

ลิขสิทธิ์มหาวิทยาลัยเกษตรศาสตร์
Copyright© by Kasetsart University
All rights reserved

โดยสรุป ระบบแจ้งเตือนผ่าน Telegram Bot เป็นเครื่องมือที่ช่วยเพิ่มความสะดวกและความมั่นใจให้กับผู้ดูแลโรงเรือน เห็นสถานะจริงของโรงเรือนได้ทุกเวลา สามารถตอบสนองต่อปัญหาได้ทันที และทำให้การเพาะเห็ดเป็นไปอย่างมีประสิทธิภาพและปลอดภัยมากยิ่งขึ้น

บทที่ 5

สรุปผลการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาระบบฟาร์มเห็ดอัจฉริยะที่สามารถควบคุมปัจจัยสำคัญของโรงเรือนได้โดยอัตโนมัติ จัดเก็บข้อมูลเข้าสู่ฐานข้อมูล และแสดงผลข้อมูลแบบเรียลไทม์ผ่าน Web Application ผลการดำเนินงานยืนยันว่าระบบที่พัฒนาขึ้นสามารถบรรลุเป้าหมายที่ตั้งไว้ได้อย่างครบถ้วน ในด้านการควบคุมระบบฮาร์ดแวร์และสภาพแวดล้อม ระบบใช้ ESP32 เป็นไมโครคอนโทรลเลอร์หลักเชื่อมต่อกับเซ็นเซอร์หลายชนิด ได้แก่ DHT22 สำหรับวัดอุณหภูมิและความชื้น MH-Z14A สำหรับวัดระดับก๊าซคาร์บอนไดออกไซด์ และ LDR สำหรับวัดความเข้มของแสง ข้อมูลจากเซ็นเซอร์เหล่านี้จะถูกประมวลผลเพื่อนำไปควบคุมอุปกรณ์ผ่าน Relay Module เช่น พัดลมระบายความร้อน หัวพ่นหมอก ไฟ LED และปั๊มน้ำ เพื่อรักษาสภาพแวดล้อมภายในโรงเรือนให้อยู่ในช่วงที่เหมาะสมต่อการเพาะเห็ดนางฟ้าภูฐาน

นอกจากนี้ ระบบยังติดตั้ง Arduino Mega 2560 และ TFT Monitor เพื่อแสดงผลค่าเซ็นเซอร์และสถานะอุปกรณ์ให้ผู้ดูแลสามารถตรวจสอบได้โดยตรง และมีการใช้พลังงานแสงอาทิตย์เป็นแหล่งจ่ายไฟฟ้าหลักหรือสำรอง ทำให้ระบบมีความยืดหยุ่นในการใช้งานในพื้นที่ห่างไกล

ในด้านการจัดการข้อมูลและการแสดงผล ระบบใช้สถาปัตยกรรม Web-Based IoT โดยมีการจัดการข้อมูลสองรูปแบบหลัก ข้อมูลเรียลไทม์ถูกส่งจาก ESP32 ผ่านโปรโตคอล MQTT ไปยัง Web Dashboard เพื่อให้ผู้ใช้งานสามารถดูสถานะปัจจุบันของโรงเรือนและสั่งควบคุมอุปกรณ์จากระยะไกลได้ทันที ข้อมูลย้อนหลังถูกส่งผ่านโปรโตคอล HTTP ไปยัง Node-RED เพื่อจัดเก็บลงในฐานข้อมูล PostgreSQL Web Application ที่พัฒนาขึ้นด้วย PHP, HTML และ JavaScript สามารถแสดงผลค่าเซ็นเซอร์แบบเรียลไทม์ และดึงข้อมูลย้อนหลังสามวันหรือเจ็ดวันมาแสดงในรูปกราฟแนวโน้มได้อย่างชัดเจน ระบบยังสามารถแสดงตำแหน่งโรงเรือนบนแผนที่โดยใช้ Leaflet Map ร่วมกับ Google Satellite ซึ่งเป็นพื้นฐานของระบบสารสนเทศภูมิศาสตร์ (GIS)

ที่สำคัญที่สุดคือระบบแจ้งเตือน เมื่อค่าที่วัดได้เกินขีดจำกัดที่ตั้งไว้ เช่น อุณหภูมิหรือระดับ CO_2 สูงเกินไป ระบบจะส่งข้อความแจ้งเตือนอัตโนมัติไปยังผู้ดูแลผ่าน Telegram Bot API ทำให้ผู้ใช้งานสามารถตอบสนองต่อปัญหาและปรับการทำงานของอุปกรณ์ได้ทันเวลา ลดความเสียหายต่อผลผลิตและเพิ่มประสิทธิภาพการเพาะเห็ดอย่างมีนัยสำคัญ

ลิขสิทธิ์มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

การอภิปรายผล

การอภิปรายผลการวิจัยนี้ตอกย้ำถึงประสิทธิภาพของการบูรณาการเทคโนโลยี IoT ในการยกระดับภาคเกษตรกรรมไทยให้เข้าสู่ยุค Smart Farming ประการแรกคือความแม่นยำในการควบคุมสภาพแวดล้อมและความเชื่อมั่นของเกษตรกร การใช้ระบบอัตโนมัติและเซ็นเซอร์ที่ละเอียดอ่อนช่วยให้สามารถรักษาปัจจัยที่สำคัญต่อการเพาะเห็ด เช่น อุณหภูมิและความชื้น ให้อยู่ในช่วงที่เหมาะสมและคงที่อย่างสม่ำเสมอ ซึ่งส่งผลโดยตรงต่อคุณภาพและปริมาณของผลผลิต การควบคุมที่อาศัยการตรวจวัดระดับ CO_2 ด้วยเซ็นเซอร์ MH-

Z14A และการสั่งการพัดลมระบายอากาศอัตโนมัติถือเป็นจุดเด่นสำคัญ เนื่องจากระดับ CO_2 ที่สูงเกินไปเป็นสาเหตุหลักที่ทำให้ดอกเห็ดมีลักษณะผิดรูปและคุณภาพลดลง การที่ระบบสามารถแจ้งเตือนผ่าน Telegram ได้ทันทีเมื่อค่าที่วัดได้เกินขีดจำกัดช่วยเพิ่มความมั่นใจให้กับผู้ดูแลโรงเรือน ทำให้ไม่จำเป็นต้องเฝ้าหน้างานตลอดเวลา และสามารถตอบสนองต่อเหตุการณ์ฉุกเฉินได้อย่างรวดเร็ว

ประการที่สองคือคุณค่าของการจัดการข้อมูลเชิงเวลา การตัดสินใจในการเพาะปลูกแบบดั้งเดิมมักอาศัยประสบการณ์และความรู้สึก แต่ระบบนี้เปลี่ยนกระบวนการเป็นการตัดสินใจบนพื้นฐานของข้อมูล (Data-Driven Decisions) การจัดเก็บข้อมูลอย่างต่อเนื่องในฐานข้อมูล PostgreSQL และการนำเสนอข้อมูลในรูปกราฟแนวโน้มบน Web Dashboard ช่วยให้เกษตรกรสามารถวิเคราะห์และเปรียบเทียบค่าของแต่ละวันหรือแต่ละช่วงเวลา ซึ่งสามารถใช้ในการปรับปรุงกลยุทธ์การควบคุมในรอบการเพาะปลูกถัดไปได้อย่างเป็นระบบ ข้อมูลย้อนหลังยังสามารถใช้วิเคราะห์ประสิทธิภาพของอุปกรณ์ที่ใช้ในโรงเรือน ทำให้สามารถวางแผนบำรุงรักษาและปรับปรุงระบบได้อย่างมีประสิทธิภาพ

ประการที่สามคือความยืดหยุ่นในการใช้งาน การที่ระบบสามารถใช้พลังงานแสงอาทิตย์และมีการแสดงผลทั้งในรูปแบบออนไลน์ (Web Dashboard) และออฟไลน์ (TFT Monitor) ทำให้ระบบมีความยืดหยุ่นและมีเสถียรภาพในการทำงานสูง แม้ในพื้นที่ที่โครงสร้างพื้นฐานด้านพลังงานหรืออินเทอร์เน็ตอาจไม่เสถียร

ดังนั้น ระบบฟาร์มเห็ดอัจฉริยะที่พัฒนาขึ้นนี้จึงไม่เพียงแต่เป็นเครื่องมือควบคุมสภาพแวดล้อมเท่านั้น แต่ยังเป็นต้นแบบสำคัญที่ช่วยลดภาระแรงงาน เพิ่มความสม่ำเสมอและคุณภาพของผลผลิต และยกระดับความสามารถในการแข่งขันของภาคเกษตรไทยให้เทียบเท่ามาตรฐานสากลได้อย่างยั่งยืน

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

บรรณานุกรม

นัฐพันธ์ พูนวิวัฒน์, มามีน นิลอ่อน, และ อนาวิล เชื้อรอด. (2564). *การพัฒนาโรงเพาะเห็ดควบคุมด้วยระบบอัตโนมัติโดยใช้พลังงานจากเซลล์แสงอาทิตย์.* การประชุมสัมมนาวิชาการเครือข่ายบัณฑิตศึกษามหาวิทยาลัยราชภัฏภาคเหนือ ครั้งที่ 22, มหาวิทยาลัยราชภัฏนครสวรรค์.
แหล่งที่มา: https://gs2.nsru.ac.th/gnru2022/proceeding/group3/3-มามีน%20นิลอ่อน.pdf

วีรศักดิ์ ฟองเงิน, สุรพงษ์ เพ็ชร์หาญ, และ รัฐสิทธิ์ ยะจ่อ. (2561). *การประยุกต์ใช้เทคโนโลยีไอโอทีควบคุมฟาร์มอัจฉริยะในโรงเรือนเพาะเห็ดนางฟ้า.* วารสารวิชาการการจัดการเทคโนโลยี มหาวิทยาลัยราชภัฏมหาสารคาม, 5(1), 172-182.
แหล่งที่มา: https://ph02.tci-thaijo.org/index.php/itm-journal/article/view/140258

ธีรพงศ์ เพ็ญเนตร และคณะ. (2566). *การพัฒนาระบบควบคุมสภาพแวดล้อมโรงเรือนเพาะเห็ดนางรมภูฐานโดยใช้ระบบสมองกลฝังตัวไอโอที.* สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคล.
แหล่งที่มา: (ยังไม่พบแหล่งเผยแพร่บนฐานข้อมูลออนไลน์ — ควรตรวจสอบจากรายงานหรือวิทยานิพนธ์ต้นฉบับของผู้วิจัย)

Kukkong Kaewkorakot, Chanon Traiwan, Thitipat Weeplian, Pimsiri Tiyayon, Khemissara Thanatthirapich, Wanwisa Pansak, และ Suwit Kiravittaya. (2024). *Smart Mushroom Cultivation House: Engineering Development and Data Analysis.* Engineering Journal, 28(4), 11-20.
แหล่งที่มา: https://engj.org/index.php/ej/article/view/4545

Kuntapon Mahamad, Praphan Mansuriwong, และ Wattana Petlamul. (2021). *Application of Smart Farm System to Enhance Phoenix Oyster Mushroom Production.* ASEAN Journal of Scientific and Technological Reports, 24(3), 47-57.
แหล่งที่มา: https://ph02.tci-thaijo.org/index.php/tsujournal/article/view/243604

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

ภาคผนวก

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

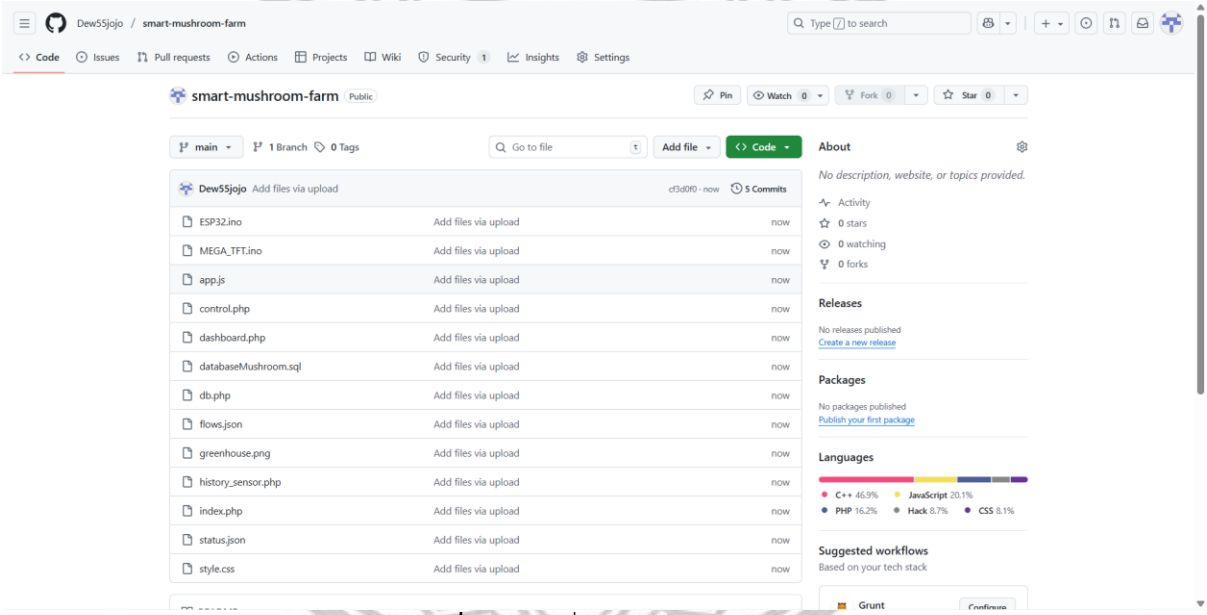
ภาคผนวก

ภาคผนวกโค้ดที่ใช้ในการพัฒนาระบบ

ในภาคผนวกนี้ ผู้วิจัยได้รวบรวมซอร์สโค้ดทั้งหมดที่ใช้ในการพัฒนาระบบฟาร์มเห็ดอัจฉริยะ เพื่อให้ผู้ที่สนใจสามารถตรวจสอบ ศึกษาต่อยอด หรือนำไปประยุกต์ใช้ได้

ซอร์สโค้ดและไฟล์โครงการต่าง ๆ ถูกจัดเก็บไว้ในแพลตฟอร์ม GitHub ซึ่งสามารถเข้าถึงได้ตามลิงก์ที่ระบุไว้ด้านล่างนี้ โดยซอร์สโค้ดจะแบ่งออกเป็นส่วนของไมโครคอนโทรลเลอร์ (ESP32), ระบบ Node-RED Flow, และ Web Application

ที่อยู่ URL https://github.com/Dew55jojo/smart-mushroom-farm.git



ภาพที่ 39 โค้ดที่ใช้ในการพัฒนาระบบ

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved

ประวัติผู้วิจัย

ชื่อ – ชื่อสกุล สรวิศ ปัญญา

วัน เดือน ปี เกิด 11 กันยายน 2546

ที่อยู่ปัจจุบัน บ้านเลขที่ 19 หมู่ที่14 ตำบล ดู่ใต้ อำเภอ เมืองน่าน จังหวัด น่าน รหัสไปรษณีย์ 55000

ประวัติการศึกษา

พ.ศ. 2565 - ปัจจุบัน ระดับอุดมศึกษา วท.บ. สาขาภูมิศาสตร์ มหาวิทยาลัยนเรศวร อำเภอ เมืองพิษณุโลก จังหวัดพิษณุโลก 65000

พ.ศ. 2562 - 2564 ระดับมัธยมศึกษาตอนปลาย (ศิลป์-คำนวณ) โรงเรียนศรีสวัสดิ์วิทยาคารจังหวัดน่าน อำเภอ เมืองน่าน จังหวัดน่าน 55000

พ.ศ. 2559 - 2561 ระดับมัธยมศึกษาตอนต้น โรงเรียนศรีสวัสดิ์วิทยาคารจังหวัดน่าน อำเภอ เมืองน่าน จังหวัดน่าน 55000

กิจกรรมที่เข้าร่วม

1.เข้าร่วมงาน GeoHackathon ณ หมู่บ้านนครชุม อำเภอนครไทย จังหวัดพิษณุโลก

2.ลงพื้นที่สำรวจการเรียนรู้นอกห้องเรียนด้วยการออกภาคสนามจริง สถานีปฏิบัติการบึงราชนก โดยใช้ Trimble GNSS RTK ทำการปักพิกัด GCPs และถ่ายภาพจากโดรน Multispectral Camera อำเภอวังทอง จังหวัดพิษณุโลก

3.เข้าร่วมงานประชุมสมาคมภูมิศาสตร์แห่งประเทศไทย

4.ข้ารับฟัง Special Seminar Theme: From Climate Change to Sustainable Futures: Geoinformatics in Action for SDGs ณ มหาวิทยาลัยนเรศวร

5.เข้ารับฟังบรรยายพิเศษหัวข้อ “Leveraging Geospatial Technologies Using Free & Open Source Software and Open Data” และพูดคุยแลกเปลี่ยนความร่วมมือทางวิชาการระหว่างมหาวิทยาลัยนเรศวรและ Osaka Metropolitan University (OMU) ณ มหาวิทยาลัยนเรศวร

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
Copyright by Naresuan University
All rights reserved