

เว็บแอปพลิเคชัน และ ฟาร์มไฮโดรโปนิกส์

นายจิรเมธ แก้วคำ

นางสาวณัฏฐนิชา เจริมย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ (คอมพิวเตอร์)

ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์

วิทยาลัยเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

พ.ศ.2567

Web Application and Hydroponic Farm

Mr. Jiramet kaewchum

Miss Natthanicha Jewaram

Project Report Submitted in Partial Fulfillment of the Requirements
for the Bachelor's Degree of Engineering in
Electronics Engineering Technology (Computer)
Department of Electronics Engineering Technology
College of Industrial Technology
King Mongkut's University of Technology North Bangkok
2024

หัวข้อปริญญานิพนธ์ : เว็บแอปพลิเคชัน และ ฟาร์มไฮโดรโปนิคส์
โดย : นายจิรเมธ แก้วคำ
นางสาวณัฏฐณิชา เจริมย์
ที่ปรึกษาปริญญานิพนธ์ : ผู้ช่วยศาสตราจารย์ ดร. เลอสรณ์ กิรสมุทรานนท์
สาขาวิชา : เทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ (คอมพิวเตอร์)
ภาควิชา : เทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา : 2567

วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ อนุมัติให้
นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

..... คณบดีวิทยาลัยเทคโนโลยีอุตสาหกรรม
(รองศาสตราจารย์ ดร. สมิตร์ ส่งพิริยะกิจ)

คณะกรรมการสอบปริญญานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. พิสิทธิ์ วิสุทธีเมธีกร)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. เลอสรณ์ กิรสมุทรานนท์)

..... กรรมการ
(อาจารย์ ดำรงเกียรติ แซ่ลิ่ม)

Project Title : Web Application and Hydroponic Farm
By : Mr. Jiramet kaewchum
Miss Natthanicha Jewaram
Project Advisor : Asst. Prof. Dr. Lerson Kirasamuthranon
Major Field : Electronics Engineering Technology (Computer)
Department : Electronics Engineering Technology
Academic Year : 2024

Accepted by the College of Industrial Technology, King Mongkut's University of Technology North Bangkok in Partial Fulfillment of the Requirements for the Bachelor's Degree of Engineering.

..... Dean of College of Industrial Technology
(Assoc. Prof. Dr. Smith Songpiriyakij)

Project Committee

..... Chairperson
(Asst. Prof. Dr. Pisit Wisutmetheekorn)

Lerson Kirasamuthranon Member

(Asst. Prof. Dr. Lerson Kirasamuthranon)

..... Member
(Mr. Damrongkiat Lim)

กิตติกรรมประกาศ

โครงการปริญญานิพนธ์เรื่องเว็บแอปพลิเคชัน และ ฟาร์มไฮโดรโปนิคส์ สำเร็จลุล่วงไปได้ด้วยดี เนื่องด้วยได้รับความช่วยเหลือเป็นอย่างดีจาก ผู้ช่วยศาสตราจารย์ ดร. เลอสรณ์ กิรสมุทธานนท์ อาจารย์ที่ปรึกษาโครงการปริญญานิพนธ์ ที่ได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ ของการจัดทำโครงการปริญญานิพนธ์ และช่วยเหลือในการแก้ไขปัญหาต่าง ๆ มาโดยตลอด

ขอกราบขอบพระคุณนุพพการีและมารดาเป็นอย่างสูง ซึ่งให้การสนับสนุนในทุก ๆ ด้าน เป็นแรงผลักดัน และให้กำลังใจแก่ผู้จัดทำเสมอจนสำเร็จการศึกษา ขอขอบพระคุณคณะอาจารย์สาขาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ แขนงคอมพิวเตอร์ทุกท่านที่ได้ประสิทธิ์ประสาทความรู้ และให้ความช่วยเหลือในด้านเทคนิคหลาย ๆ อย่างเป็นอย่างดี และขอขอบพระคุณทุกท่านผู้มีส่วนเกี่ยวข้องกับความสำเร็จแต่มิได้เอ่ยนามทุกท่าน มา ณ ที่นี้ด้วย

สุดท้ายนี้ทางคณะผู้จัดทำต้องขอขอบพระคุณท่านกรรมการสอบโครงการปริญญานิพนธ์ทุกท่านเป็นอย่างสูงที่ได้ช่วยพิจารณาและให้คำแนะนำในการตรวจทานแก้ไข อนุมัติจนโครงการปริญญานิพนธ์นี้สำเร็จเป็นไปตามวัตถุประสงค์และขอบเขตที่ตั้งไว้ทุกประการ ซึ่งผู้จัดทำหวังว่าโครงการปริญญานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้ที่ทำการปลูกผักไฮโดรโปนิคส์

คณะผู้จัดทำ

เว็บแอปพลิเคชัน และ ฟาร์มไฮโดรโปนิคส์

จิรเมธ แก้วคำ¹, ณัฐธินิชา เจริญชัย¹ และ เลอสรณ์ กิรสมุทรานนท์²

บทคัดย่อ

การเกษตรมีความสำคัญกับมนุษย์เป็นอย่างมาก โดยเฉพาะการเป็นแหล่งสำหรับผลิตอาหารที่สามารถรองรับประชากรทั่วโลก ซึ่งมีบทบาทสำคัญในการจัดหาอาหารที่มีคุณภาพและมีประโยชน์แก่มนุษย์ จึงมีผู้คนจำนวนหนึ่งที่มีความสนใจในการทำเกษตรในพื้นที่เล็ก ๆ บริเวณบ้านหรือระเบียงของตนเอง เพื่อปลูกพืชผักสวนครัวที่จำเป็นต่อการดำรงชีวิต แต่การดำรงชีวิตของมนุษย์ในปัจจุบัน มักมีปัญหาในเรื่องของเวลาที่ใช้ในการดูแลพื้นที่การเกษตร เนื่องจากการดูแลพื้นที่การเกษตรนั้น ต้องมีความละเอียดและซับซ้อน เพื่อให้ผลผลิตออกมาได้อย่างมีคุณภาพ

โครงการปริญญานิพนธ์นี้เน้นการออกแบบและพัฒนาอุปกรณ์การปลูกฟาร์มไฮโดรโปนิคส์ที่สามารถแสดงค่าสถานะต่าง ๆ ผ่านเว็บไซต์ที่พัฒนาด้วย Angular เพื่อให้บริการในระบบฟาร์มไฮโดรโปนิคส์ โดยมุ่งเน้นการบริหารจัดการส่วนหลัก ๆ ได้แก่ การแสดงผลค่าความชื้น, ค่าอุณหภูมิ, ปริมาณน้ำในถัง, และค่า pH ในน้ำ รวมถึงการส่งเปิด-ปิดอุปกรณ์ผ่านหน้าเว็บ ในส่วนการแจ้งเตือนค่า ผ่านทาง Line Notify และการเก็บข้อมูลใน Firebase สำหรับค่าที่ได้รับจากเซ็นเซอร์ รวมถึงสถานะเปิด-ปิดของอุปกรณ์

จากการพัฒนาเว็บแอปพลิเคชันร่วมกับการทำฟาร์มไฮโดรโปนิคส์ โดยมีการสร้างแบบจำลองฟาร์มไฮโดรโปนิคส์ที่ใช้บอร์ด ESP32 ในการรับค่าและควบคุมอุปกรณ์ต่าง ๆ ภายในฟาร์ม เช่น ไฟ พัดลม ปั๊ม สปริงเกอร์ เซ็นเซอร์อุณหภูมิ-ความชื้น และเซ็นเซอร์วัดความเป็นกรด-เบส นอกจากนี้ยังได้พัฒนาเว็บแอปพลิเคชันด้วย Angular และ Node.js เพื่อแสดงผลและให้ผู้ใช้สามารถควบคุมอุปกรณ์ในฟาร์ม โดยใช้ระบบฐานข้อมูลแบบเรียลไทม์เพื่อให้เว็บและ ESP32 อ่านค่าจากฐานข้อมูลมาแสดงผลและประมวลผล อีกทั้งยังมีการแจ้งเตือนผ่าน Line Notify เมื่อพบว่าค่าต่าง ๆ ผิดปกติหรือตามเวลาที่ตั้งค่าไว้ จากการทดสอบพบว่าเว็บแอปพลิเคชันสามารถควบคุมอุปกรณ์ในฟาร์มไฮโดรโปนิคส์ได้จริง ช่วยประหยัดเวลาและเพิ่มความสะดวกในการดูแลฟาร์มของผู้ใช้

คำสำคัญ : Angular, Line Notify, Firebase, Node.js, ESP32

¹นักศึกษา, ²อาจารย์ที่ปรึกษาภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์, วิทยาลัยเทคโนโลยีอุตสาหกรรม, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

Web Application and Hydroponic Farm

Jiramet kaewchum ¹ , Natthanicha Jewaram ¹ and Lerson Kirasamuthranon ²

Abstract

Agriculture is crucial for humanity, especially as a source of food production to support the global population. It plays a vital role in providing quality and nutritious food for humans. Consequently, there is a growing interest among some individuals in engaging in agriculture, even in small spaces such as home gardens or balconies, to cultivate essential vegetables necessary for sustenance. However, in today's world, people often face challenges in dedicating time to care for agricultural spaces due to the detailed and complex nature of agricultural maintenance required to ensure high-quality yields.

This thesis project focuses on designing and developing a hydroponic farm equipment that can display various status parameters through a website built with Angular. The aim is to provide agricultural services, emphasizing key functionalities such as displaying moisture levels, temperature, water levels in tanks, and pH levels in water. Additionally, it includes the ability to control equipment remotely via the web interface, notifications through Line Notify, and data storage in Firebase for sensor readings and equipment status.

The project involves developing a web application integrated with hydroponic farming, using an ESP₃₂ board to receive and control various equipment within the farm, such as lights, fans, pumps, sprinklers, temperature-humidity sensors, and pH sensors. Additionally, the web application was developed with Angular and Node.js to display and allow users to control farm equipment. A real-time database system is utilized so that the web and ESP₃₂ can fetch data from the database for display and processing. Notifications are sent through Line Notify when values are abnormal or at scheduled times. Testing revealed that the web application effectively controls equipment in the hydroponic farm, saving time and increasing convenience for users in managing their farms.

Keywords Angular, Line Notify, Firebase, Node.js, ESP32.

¹Student, ²Lecturer Department of Electronics Engineering Technology, College of Industrial Technology King Mongkut's University of Technology North Bangkok.

1. บทนำ

การเกษตรมีความสำคัญกับมนุษย์เป็นอย่างมาก โดยเฉพาะการเป็นแหล่งสำหรับผลิตอาหารที่สามารถรองรับประชากรทั่วโลก ซึ่งมีบทบาทสำคัญในการจัดหาอาหารที่มีคุณภาพและมีประโยชน์แก่มนุษย์ จึงมีผู้คนจำนวนหนึ่งที่มีความสนใจในการทำเกษตรในพื้นที่เล็กๆ บริเวณบ้านหรือระเบียงของตนเอง เพื่อปลูกพืชผักสวนครัวที่จำเป็นต่อการดำรงชีวิต แต่การดำรงชีวิตของมนุษย์ในปัจจุบัน มักมีปัญหาในเรื่องของเวลาที่ใช้ในการดูแลพื้นที่การเกษตร เนื่องจากการดูแลพื้นที่การเกษตรนั้น ต้องมีความละเอียดและซับซ้อน เพื่อให้ผลผลิตออกมาได้อย่างมีคุณภาพ

โครงการปริญญานิพนธ์นี้เน้นการออกแบบและพัฒนาอุปกรณ์การปลูกฟาร์มไฮโดรโปนิคส์ที่สามารถแสดงค่าสถานะต่าง ๆ ผ่านเว็บไซต์ที่พัฒนาด้วย Angular เพื่อให้บริการในระบบฟาร์มไฮโดรโปนิคส์ โดยมุ่งเน้นการบริหารจัดการส่วนหลัก ได้แก่ การแสดงผลค่าความชื้น, ค่าอุณหภูมิ, ปริมาณน้ำในถัง, และค่า pH ในน้ำ รวมถึงการสั่งเปิด-ปิดอุปกรณ์ผ่านหน้าเว็บ ในส่วนการแจ้งเตือนค่า ผ่านทาง Line Notify และการเก็บข้อมูลใน Firebase สำหรับค่าที่ได้รับจากเซ็นเซอร์ รวมถึงสถานะเปิด-ปิดของอุปกรณ์

ฟาร์มไฮโดรโปนิคส์นำเทคโนโลยีใหม่ เข้ามาใช้ในการเกษตรเพื่อเพิ่มประสิทธิภาพและการควบคุมจากระยะไกลในการเพาะปลูกที่แม่นยำและอัตโนมัติมากยิ่งขึ้น การนำ Angular มาใช้ในการพัฒนาเว็บไซต์จะช่วยให้มีประสิทธิภาพ เข้าถึงข้อมูลได้ง่ายขึ้นและความสามารถในการปรับแต่ง ส่งผลให้ผู้ใช้งานสามารถใช้งานได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้นในการปฏิบัติงานทาง

การเกษตรในทุกด้าน เช่น การตรวจสอบ ความชื้นและอุณหภูมิในโรงเรือน และการควบคุมการให้น้ำและปุ๋ยในระบบการเกษตรได้อย่างแม่นยำและมีประสิทธิภาพสูงสุด

2. ทฤษฎีที่เกี่ยวข้อง

2.1 Angular

Angular [1] คือ Frontend Framework ที่ถูกพัฒนาโดย Google ซึ่งใช้สำหรับการพัฒนาเว็บแอปพลิเคชันบนฝั่งของไคลเอนต์ โดย Angular มีเป้าหมายในการช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันที่มีประสิทธิภาพโดยมีคุณสมบัติที่หลากหลาย เช่น การจัดการสถานะของแอปพลิเคชันแบบเรียลไทม์ (Real-time), การเปลี่ยนแปลงสถานะของข้อมูลโดยไม่ต้องรีเฟรชหน้าเว็บ (Reactive programming), การจัดการเหตุการณ์ (Event handling), และการเชื่อมต่อกับเซิร์ฟเวอร์เพื่อรับข้อมูล (API integration) ซึ่งทำให้ Angular เป็นที่นิยมในการพัฒนาเว็บแอปพลิเคชันในปัจจุบัน

Angular เป็นส่วนหนึ่งของ MEAN Stack ซึ่งเป็นชุดของเฟรมเวิร์กที่มีประสิทธิภาพสำหรับการพัฒนาแอปพลิเคชัน โดย MEAN คือ แอครอนิม-เอ็กซ์เพรส-แก็มมา-แองกูลาร์ โดยมีคุณสมบัติที่ช่วยให้การพัฒนาแอปพลิเคชันเป็นไปอย่างรวดเร็วและมีประสิทธิภาพ แต่ละส่วนประกอบมีความสัมพันธ์กันอย่างเป็นระบบ โดยที่ Angular จะใช้สำหรับฝั่งของไคลเอนต์ เวิร์กเฟรมเวิร์กแบบอื่น ๆ ที่เป็นส่วนหนึ่งของ MEAN Stack ได้แก่ MongoDB เป็นฐานข้อมูลที่ใช้เก็บข้อมูล, Express.js เป็นเฟรมเวิร์กที่ใช้สำหรับพัฒนาเว็บเซิร์ฟเวอร์ด้วย Node.js และ Node.js เป็นเฟรมเวิร์กสำหรับเขียนโค้ดฝั่งเซิร์ฟเวอร์ด้วย JavaScript

2.2 Firebase

Firebase [2] เป็นแพลตฟอร์มที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วน Backend หรือ Server side ซึ่งช่วยให้สามารถพัฒนาแอปพลิเคชันมือถือได้อย่างมีประสิทธิภาพ โดยลดเวลาและค่าใช้จ่ายของการทำ Server side หรือการวิเคราะห์ข้อมูลได้ด้วยเช่นกัน บริการที่ Firebase มีให้บริการได้แก่

2.2.1 Cloud Firestore เป็นบริการฐานข้อมูลที่เป็นลักษณะ NoSQL ที่ใช้สำหรับเก็บข้อมูลแบบ Realtime Database ซึ่งมีความยืดหยุ่นและประสิทธิภาพสูง

2.2.2 Authentication บริการที่ช่วยจัดการการรับรองตัวตน (Auth) โดยรองรับหลากหลายวิธีการเข้าสู่ระบบ เช่น email-password, phone, และ social media อื่น ๆ

2.2.3 Hosting บริการให้โฮสต์สำหรับเว็บไซต์แบบ single-page หรือ landing page ซึ่งช่วยให้ง่ายต่อการจัดการ การ Deploy และมีระบบ Custom Domain รวมถึงการติดตั้ง SSL ให้ด้วย

2.2.4 Cloud Functions บริการที่ช่วยให้สร้างและทำงานกับฟังก์ชันบนเซิร์ฟเวอร์ได้อย่างง่ายดาย โดยสามารถทำงานตามเหตุการณ์ที่เกิดขึ้นในระบบ Firebase ได้

2.2.5 Storage บริการที่ให้การจัดการเกี่ยวกับการเก็บข้อมูลในรูปแบบของไฟล์ เช่น รูปภาพ, วิดีโอ, หรือไฟล์อื่น ๆ ซึ่งสามารถเข้าถึงได้ผ่าน API หรือ Console

2.2.6 Analytics บริการที่ช่วยให้เข้าใจพฤติกรรมของผู้ใช้ และประสิทธิภาพของแอปพลิเคชัน โดยให้ข้อมูลการใช้งาน และการทำธุรกรรมต่าง ๆ

2.2.7 Remote Config บริการที่ช่วยให้ปรับแต่งแอปพลิเคชันของคุณได้โดยไม่ต้องปล่อยเวอร์ชันใหม่ โดยสามารถ

ปรับแต่งค่าต่าง ๆ เช่น รูปแบบ UI หรือฟีเจอร์ใหม่ ๆ ให้กับผู้ใช้ได้ผ่านทางคลาวด์

โดย Firebase ยังมีบริการอื่น ๆ อีกมากมายที่ช่วยให้นักพัฒนาสามารถพัฒนาและบริหารจัดการแอปพลิเคชันได้อย่างมีประสิทธิภาพและสะดวกสบาย

2.3 Line Notify

LINE Notify [3] เป็นบริการที่ช่วยให้ผู้ใช้รับข้อความแจ้งเตือนจากเว็บเซอร์วิสต่าง ๆ ผ่านแอปพลิเคชัน LINE โดยหลังจากทำการเชื่อมต่อกับเว็บเซอร์วิสแล้วผู้ใช้จะได้รับการแจ้งเตือนผ่านบัญชี LINE Notify ซึ่งสามารถเชื่อมต่อกับบริการต่าง ๆ

ผู้ใช้งานสามารถรับการแจ้งเตือนเกี่ยวกับสถานะหรือเหตุการณ์ต่าง ๆ ที่เกิดขึ้นในบริการที่เชื่อมต่อ ทำให้สามารถติดตามสถานะหรือข้อมูลที่สำคัญได้อย่างรวดเร็ว สะดวกสบาย และยังสามารถรับการแจ้งเตือนทางกลุ่มได้อีกด้วย ทำให้การสื่อสารและการติดตามสถานะเป็นไปอย่างมีประสิทธิภาพ

2.4 REST API

REST (Representational State Transfer) API [4] คือการสร้าง API ประเภท RESTful web services ซึ่งจัดเป็น Web Service รูปแบบหนึ่งที่ทำงานอยู่บนพื้นฐานของโปรโตคอล HTTP และ HTTPS ประกอบด้วย Request และ Response ตามรูปแบบของ HTTP ที่รับส่งข้อมูลหรือเนื้อหาในรูปแบบของ XML , SOAP , JSON

REST API นั้นทำงานโดยใช้พื้นฐานของโปรโตคอล HTTP ดังนั้นแต่ละ Method ของ HTTP จึงนำมาใช้งานใน

REST API โดยนักพัฒนา API จะเขียนโปรแกรมให้ API นั้นประมวลผลกับข้อมูลตามความหมายของ HTTP Method

- GET หมายถึง อ่านข้อมูล
- POST หมายถึง สร้างข้อมูลใหม่
- PUT หมายถึง การแก้ไขข้อมูลทั้งหมด
- PATCH หมายถึง การแก้ไขข้อมูลบางส่วน
- DELETE หมายถึง ลบข้อมูล

API (Application Programming Interface) เป็นช่องทางในการเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลกันระหว่างแอปพลิเคชัน การพัฒนา API ที่ได้รับความนิยมในปัจจุบันจะพัฒนาโดยทำงานในรูปแบบที่เรียกว่า “REST API” (RESTful web services)

JSON (JavaScript Object Notation) เป็นรูปแบบการแลกเปลี่ยนหรือรับส่งข้อมูลในระบบคอมพิวเตอร์หรือแอปพลิเคชัน ในอดีตการแลกเปลี่ยนหรือรับส่งข้อมูลนั้นจะใช้รูปแบบ XML แต่เนื่องจาก XML มีโครงสร้างที่ซับซ้อนและมีขนาดใหญ่จึงมีการเปลี่ยนมาใช้ JSON แทน คุณสมบัติของ JSON เป็นไฟล์ประเภทข้อความ (Text) มีโครงสร้างคำสั่งที่มนุษย์สามารถอ่าน-เขียนแล้วเข้าใจได้เลย อีกทั้งยังมีขนาดเล็ก น้ำหนักเบา เป็นมาตรฐานกลางของทุกภาษา สำหรับการแลกเปลี่ยนข้อมูลข้ามแพลตฟอร์มบนระบบปฏิบัติการที่แตกต่างกัน

2.5 Cron-Job

Cron-Job [5] เป็นคุณสมบัติพื้นฐานของระบบปฏิบัติการ Linux จัดว่าเป็นระบบปฏิบัติการยูนิกซ์ประเภทหนึ่งสามารถใช้คุณสมบัติ Cron-Job นี้ เพื่อกำหนดช่วงเวลาการทำงานของภาษา PHP หรือ script ใด ๆ โดยอัตโนมัติได้โดยปกติแล้วการที่ภาษาที่เป็น Server Side ต่าง ๆ จะสามารถเริ่ม

ทำงานได้ จะต้องอาศัย User ในการเริ่มทำงานซึ่งต้องเปิดเบราว์เซอร์หน้าเว็บไซต์นั้น ๆ ก่อน ถึงจะสามารถทำงานคำสั่งต่าง ๆ ภายใน script file นั้นได้โดยเราสามารถใช้ความสามารถของ Cron-Job ในการสั่งให้ไฟล์ทำงานได้โดยไม่ต้องอาศัยการเปิดหน้าเว็บไซต์

2.6 Microsoft Azure

Microsoft Azure [6] คือ บริการคลาวด์ที่ถูกสร้างขึ้นโดย Microsoft ใช้สำหรับสร้าง จัดการ หรือปรับใช้แอปพลิเคชันและบริการต่าง ๆ ให้ตอบสนองกับความต้องการของธุรกิจได้อย่างรวดเร็วผ่านเครือข่ายทั่วโลกของ Microsoft รวมถึงการบริการและเครื่องมือที่หลากหลายตั้งแต่การประมวลผลข้อมูล การจัดเก็บข้อมูล ตลอดจนความสามารถในการวิเคราะห์ข้อมูล

Microsoft Azure ให้บริการต่าง ๆ รวมถึงเครื่องมือที่เก็บข้อมูล และฐานข้อมูล ซึ่งสามารถใช้ในการสร้างและเรียกใช้แอปพลิเคชัน บริการเหล่านี้สามารถเข้าถึงได้ผ่านพอร์ทัล Azure อินเทอร์เฟซบนเว็บ หรือผ่าน Azure API และเครื่องมือบรรทัดคำสั่ง ปรับใช้แอปพลิเคชันกับ Azure ได้โดยใช้วิธีการต่าง ๆ รวมถึงการอัปโหลดโค้ด การบรรจุคอนเทนเนอร์ หรือเทมเพลตที่สร้างไว้ล่วงหน้า

2.7 ESP32

ESP32 [7] เป็นบอร์ดไมโครคอนโทรลเลอร์ที่พัฒนาโดย Espressif Systems ซึ่งมีความสามารถหลากหลายที่เหมาะสมสำหรับการพัฒนาโปรเจกต์ IoT และโปรเจกต์ที่ต้องการการเชื่อมต่อไร้สายและการควบคุมที่ยืดหยุ่นได้ดี มีคุณสมบัติดังนี้

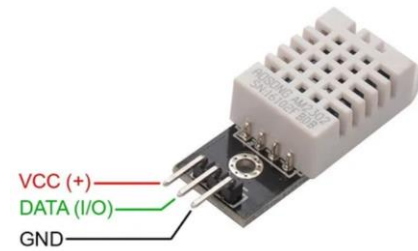
- CPU Dual-core Tensilica LX6 microprocessor ที่ความเร็ว 240 MHz
- ไฟเลี้ยง 3.3 V (ตัวบอร์ได้รับแรงดัน 5 V ได้ มีเรกูเลเตอร์)
- รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB
- ทำงานได้ที่อุณหภูมิ -40°C ถึง 125°C
- พอร์ตดิจิทัล มีทั้งหมด 36 พอร์ต GPIO
- รองรับการเชื่อมต่อกับเซ็นเซอร์และอุปกรณ์ต่างๆ ผ่าน SPI, I2C, UART, ADC, DAC
- การเชื่อมต่อไร้สาย รองรับ Wi-Fi 802.11 b/g/n/e/i, Bluetooth v4.2 BR/EDR และ BLE (Bluetooth Low Energy)

2.8 เซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ DHT22

DHT22 เป็น เซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ในอากาศ [8] มีความแม่นยำใช้งานง่าย มีคุณสมบัติ ดังนี้

- ย่านการวัดอุณหภูมิ -40 ถึง 80 องศาเซลเซียส
- ความแม่นยำในการวัดอุณหภูมิ ± 0.5 องศาเซลเซียส
- ย่านการวัดความชื้นสัมพัทธ์ 0% ถึง 100%
- ความแม่นยำในการวัดความชื้นสัมพัทธ์ $\pm 2\%$ ถึง $\pm 5\%$ RH
- ย่านไฟเลี้ยง 3.3V - 6V กระแสสูงสุด 2.5mA
- คาบเวลาในการวัด (ต่ำสุด) ทุกๆ 2 วินาที
- เซนเซอร์ให้สัญญาณเอาต์พุตแบบดิจิทัล และต้องต่อตัวต้านทาน Pull-up

รูปที่ 1 เป็นรูปร่างภายนอกและขาสัญญาณของเซนเซอร์ DHT22 โดย +VCC , GND เป็นขาไฟเลี้ยง และขา Data ต่อเข้ากับพอร์ตอนพุตแบบดิจิทัลของไมโครคอนโทรลเลอร์



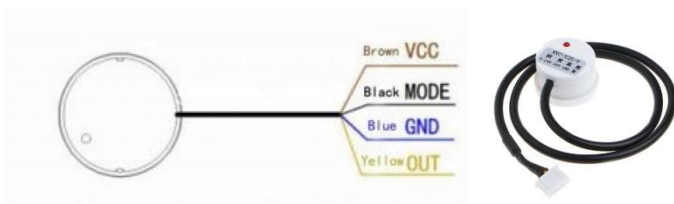
รูปที่ 1 รูปร่าง และสายสัญญาณของเซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์

2.9 เซนเซอร์ระดับน้ำ

XKC-Y25 [9] เป็นเซนเซอร์ตรวจจับของเหลวแบบไม่สัมผัส ภาชนะที่ติดตั้งต้องไม่ใช่โลหะ มีความไวสูง และมีความทนทาน สามารถใช้งานได้ในสภาพแวดล้อมที่หลากหลาย มีคุณสมบัติ ดังนี้

- ไฟเลี้ยง 5-24 V DC ที่กระแสสูงสุด 5 mA
- อุณหภูมิในการทำงาน -20 ถึง 100 องศาเซลเซียส
- ความถี่ในการตอบสนอง น้อยกว่า 500 มิลลิวินาที
- ขาเอาต์พุตเซนเซอร์เป็นแบบ NPN ให้สัญญาณลอจิก 1 เมื่อเซนเซอร์ตรวจจับไม่พบของเหลว และเกิดลอจิก 0 เมื่อตรวจพบเจอของเหลว

รูปที่ 2 เป็นรูปร่างภายนอกและสายสัญญาณของเซนเซอร์ XKC-Y25 โดย VCC และ GND เป็นขาต่อไฟเลี้ยง ส่วนขา OUT เป็นเอาต์พุตต่อดิจิทัลอินพุตของไมโครคอนโทรลเลอร์



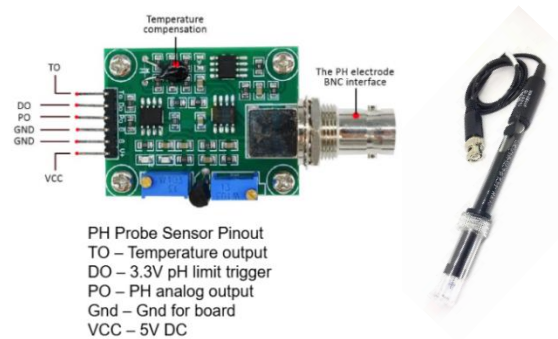
รูปที่ 2 เซนเซอร์ระดับน้ำ XKC-Y25

2.10 โมดูลเซนเซอร์วัดค่า pH

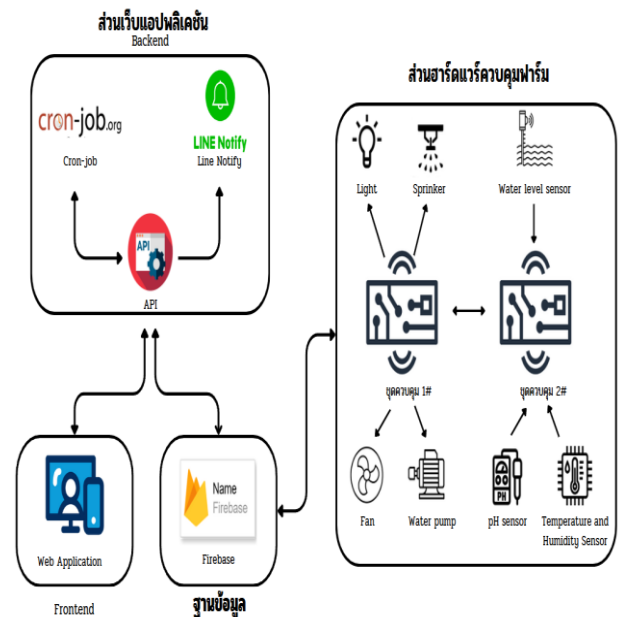
โมดูลเซนเซอร์วัดค่า pH [10] ใช้วัดค่าความเป็นกรด-ด่างของสารละลาย ซึ่งมักใช้ในงานเกษตรกรรม การควบคุมคุณภาพน้ำ การทดลองทางเคมี และการทำระบบไฮโดรโปนิคส์ โมดูลนี้ให้สัญญาณอนาล็อกซึ่งจะเชื่อมต่อกับอนาล็อกอินพุตของไมโครคอนโทรลเลอร์โดยเซนเซอร์วัดค่า pH มีคุณสมบัติ ดังนี้

- ย่านการวัดค่า pH 0 ถึง 14
- ความแม่นยำ ± 0.1 pH
- ความละเอียด 0.01 pH
- แหล่งจ่าย 5V DC ที่กระแสไฟฟ้าสูงสุด 10mA
- เวลาตอบสนอง น้อยกว่า 1 นาที (ขึ้นอยู่กับสภาพแวดล้อม)
- ขา PO เชื่อมต่อกับพอร์ตอนาล็อกของไมโครคอนโทรลเลอร์ ขา PO จะให้แรงดัน 0-5V และแปรตามแบบเชิงเส้นกับค่า pH (0-14)

รูปที่ 3 เป็นโมดูลวัดค่า pH ประกอบด้วยโพรบและวงจรปรับสภาพสัญญาณ โดยขา VCC และ GND คือไฟเลี้ยงเซนเซอร์และวงจร ส่วนขา PO เป็นเอาต์พุต



รูปที่ 3 รูปร่าง และสายสัญญาณของเซนเซอร์ pH



รูปที่ 4 โครงสร้างรวมของระบบ

3. วิธีการดำเนินงาน

การพัฒนาบบควบคุมฟาร์มไฮโดรโปนิคส์ ประกอบไปด้วย 2 ส่วนหลัก คือ ระบบไมโครคอนโทรลเลอร์ควบคุม ฟาร์มไฮโดรโปนิคส์ และเว็บแอปพลิเคชัน เพื่อการทำงาน และแสดงสถานะของฟาร์มไฮโดรโปนิคส์ โดยมีรายละเอียดในแต่ละส่วนดังนี้

3.1 หลักการทำงานโดยรวมของระบบ

จากโครงสร้างระบบรูปที่ 4 ฮาร์ดแวร์ควบคุมฟาร์ม ประกอบด้วย บอร์ดไมโครคอนโทรลเลอร์ ESP32 2 ชุด ชุดควบคุม 1# ทำหน้าที่สั่งงานควบคุมการเปิด-ปิด อุปกรณ์ในฟาร์ม ได้แก่ สปริงเกอร์, หลอดไฟ, พัดลม และมอเตอร์ปั้มน้ำ นอกจากนั้นจะนำค่าสถานะเซนเซอร์ในฟาร์มที่ได้รับจากชุดควบคุม 2# ไปบันทึกบนฐานข้อมูล Firebase ในส่วนของการสั่งเปิดปิดอุปกรณ์ของชุดควบคุม 1# จะเป็นไปตามการสั่งงานผ่านเว็บแอปพลิเคชันของผู้ใช้งานระบบ โดยคำสั่งของผู้ใช้งานจะถูกบันทึกลงบน Firebase เพื่อเป็นเงื่อนไขการควบคุมของชุดควบคุม 1# ชุดควบคุม 2# จะเชื่อมต่อกับเซนเซอร์วัดระดับน้ำ เซนเซอร์วัดค่า pH และเซนเซอร์วัดค่าอุณหภูมิ-ค่าความชื้น โดยค่าที่อ่านได้จากเซนเซอร์นั้นชุดควบคุม 2# จะส่งต่อไปยังชุดควบคุม 1# ดังที่กล่าวมาข้างต้น

เว็บแอปพลิเคชันนอกจากจะถูกใช้เพื่อการสั่งงานระบบโดยผู้ใช้งานแล้ว ยังมีหน้าจอสำหรับการแสดงสถานะต่างๆ ของอุปกรณ์ในฟาร์ม และมีการแสดงสถานะฟาร์มตามค่าที่เซนเซอร์วัดค่าได้ซึ่งถูกบันทึกบนฐานข้อมูล Firebase สำหรับรายละเอียดการออกแบบและการทำงานของทั้ง 3 ส่วนที่กล่าวมาจะมีรายละเอียดดังต่อไปนี้

3.2 อุปกรณ์ในโรงเรือนและหลักการทำงาน

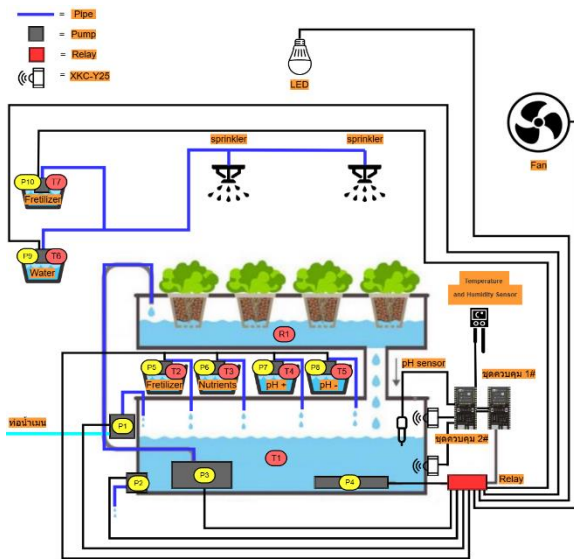
ระบบปลูกผักในปริญญานิพนธ์นี้เป็นระบบแบบน้ำวน มีแผนผังโรงเรือนดังรูปที่ 5 มีส่วนประกอบสำคัญ คือ

- R1 รางน้ำที่ใช้ปลูกผัก
- T2 ถังบรรจุสารอาหารและมีปั้ม P5 เพื่อนำสารอาหารไปเติมในถังพักน้ำ T1

- T3 ถังบรรจุปุ๋ยและมีปั้ม P6 เพื่อนำปุ๋ยไปเติมในถังพักน้ำ T1
- T4 ถังบรรจุสารเพิ่มค่า pH และมีปั้ม P7 เพื่อนำสารเพิ่มค่า pH ไปเติมในถังพักน้ำ T1
- T5 ถังบรรจุสารลดค่า pH และมีปั้ม P8 เพื่อนำสารลดค่า pH ไปเติมในถังพักน้ำ T1
- T6 ถังบรรจุน้ำ และมีปั้ม P9 เพื่อนำน้ำไปยังสปริงเกอร์เพื่อรดน้ำพืชภายในโรงเรือน
- T7 ถังบรรจุชีวภัณฑ์ และมีปั้ม P10 เพื่อนำชีวภัณฑ์ไปสปริงเกอร์เพื่อรดน้ำพืชภายในโรงเรือน
- T1 ถังพักน้ำที่บรรจุน้ำที่มีการปรับค่า pH และผสมสารอาหารเพื่อนำไปเลี้ยงผักในฟาร์มและวนกลับมาในถังนี้ใหม่ ในถังนี้มีปั้ม P1 ปั้มสำหรับเติมน้ำเข้าถัง P2 เป็นปั้มเพื่อระบายน้ำออกจากถัง P3 เป็นปั้มที่นำน้ำในถังพักไปเลี้ยงผัก P4 เป็นปั้มเพื่อกวนน้ำและสารอาหารต่างๆ ในถังให้เข้ากัน

ตั้งแต่ถัง T2 ถึง T7 เราจะเติมสารต่าง ๆ ได้โดยการควบคุมการจ่ายผ่านหน้าเว็บไซต์ ซึ่งมีตัวเลือกให้เลือกเติมสารในปริมาณ 250 หรือ 500 มิลลิตร

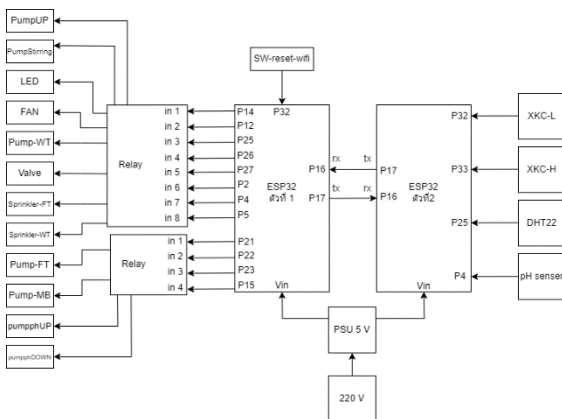
ภายในโรงเรือนมีการติดตั้งอุปกรณ์ตรวจวัดต่างๆ ได้แก่ เซ็นเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์บริเวณข้างพัดลม เซ็นเซอร์วัดระดับน้ำ 2 ตัว บริเวณถังใหญ่ และโมดูล เซ็นเซอร์วัดค่า pH ภายในถัง T1 เพื่อใช้ในการตรวจสอบสภาพแวดล้อมภายในโรงเรือน อุณหภูมิ ความชื้น ระดับน้ำ และค่า pH ของน้ำในถัง



รูปที่ 5 แผนผังโรงเรือน

3.3 การทำงานส่วนฮาร์ดแวร์ของระบบ

ระบบทำงานโดยใช้บอร์ด ESP32 เป็นตัวกลางในการเชื่อมต่อกับ Firebase เพื่อควบคุมการทำงานของปั๊มน้ำผ่านเว็บแอปพลิเคชัน บอร์ด ESP32 จะทำการเก็บรวบรวมข้อมูลจากเซ็นเซอร์ต่างๆ เช่น อุณหภูมิ ความชื้น และสถานะของรีเลย์ แล้วส่งข้อมูลเหล่านี้ไปยัง Firebase เมื่อมีการเปลี่ยนแปลงข้อมูลบน Firebase เช่น การสั่งเปิด-ปิดปั๊มน้ำ บอร์ด ESP32 จะรับคำสั่งนั้นมาและทำการควบคุมรีเลย์ให้ทำงานตามที่กำหนด

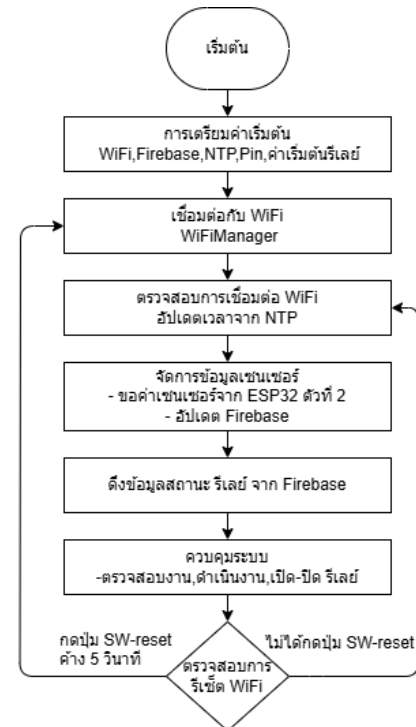


รูปที่ 6 ไลอะแกรมของอุปกรณ์

จากรูปที่ 6 ใช้บอร์ด ESP32 จำนวนสองตัวในการทำงานร่วมกัน โดยทั้งสองบอร์ดจะสื่อสารกันผ่านพอร์ตอนุกรมเพื่อแบ่งหน้าที่ในการทำงาน ดังนี้

3.3.1 ESP32 ตัวที่ 1

ผังงานการทำงานของ ESP32 ตัวที่ 1 แสดงดังรูปที่ 7 การทำงานเริ่มต้นจากไมโครคอนโทรลเลอร์ ESP32 เชื่อมต่อเครือข่าย Wi-Fi เพื่อเชื่อมต่ออินเทอร์เน็ตและเชื่อมต่อกับ Firebase จากนั้นอ่านข้อมูลเวลาจาก NTP และอ่านข้อมูลเซ็นเซอร์ต่าง ๆ จาก ESP32 ตัวที่ 2 เพื่ออัปเดตข้อมูลเซ็นเซอร์ใน Firebase และข้อมูลจาก Firebase จะถูกนำมาใช้สั่งการเปิด-ปิดรีเลย์ อีกทั้งยังมีการตรวจสอบว่ามีการกดปุ่มรีเซ็ต WiFi เป็นเวลา 5 วินาทีหรือไม่ เพื่อทำการเชื่อมต่อ WiFi ใหม่ หากไม่มีการกดปุ่มรีเซ็ต WiFi ระบบจะวนกลับไปอ่านข้อมูลเซ็นเซอร์ต่าง ๆ จาก ESP32 ตัวที่ 2 ต่อไป

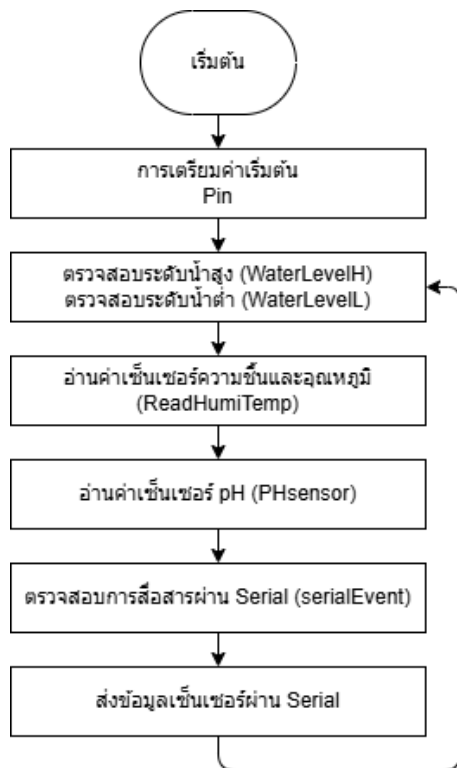


รูปที่ 7 ผังงานการทำงานของ ESP32 ตัวที่ 1

```
void Wifi_Reset() {
  if (digitalRead(SWrf) == LOW) {
    Serial.println("Wifi Reset? waiting 5S..");
    delay(5000);
    if (digitalRead(SWrf) == LOW) {
      delay(10);
      Serial.println("wifi Reset Setting ..OK");
      wm.resetSettings();
      ESP.restart();
    }
  }
}
```

รูปที่ 8 Code ตรวจสอบการรีเซ็ต WiFi

3.3.2 ESP32 ตัวที่ 2



รูปที่ 9 ผังงานการทำงานของ ESP32 ตัวที่ 2

ผังงานการทำงานของ ESP32 ตัวที่ 2 แสดงดังรูปที่ 9 มีหน้าที่อ่านค่าเซ็นเซอร์ต่าง ๆ ได้แก่ เซ็นเซอร์ระดับน้ำ 2 ตัว, เซ็นเซอร์ความชื้นและอุณหภูมิ, และเซ็นเซอร์ pH เมื่ออ่านค่าเซ็นเซอร์เสร็จแล้ว จะตรวจสอบว่ามีการร้องขอ

ข้อมูลจาก ESP32 ตัวที่ 1 หรือไม่ หากมีการร้องขอ จะทำการส่งข้อมูลค่าเซ็นเซอร์ต่าง ๆ ไปให้ ESP32 ตัวที่ 1 ดังรูปที่ 10 จากนั้นจึงวนกลับไปอ่านค่าเซ็นเซอร์ต่าง ๆ ใหม่ต่อไป

```
serialEvent();
if (sendFlag) {
  String data = "";

  data += waterstatehigh ? "1" : "0";
  data += "|";
  data += waterstatelow ? "1" : "0";
  data += "|";
  data += h;
  data += "|";
  data += t;
  data += "|";
  data += pHValue;
  Serial.println(data);
  Serial2.println(data);
  sendFlag = false;
}
```

รูปที่ 10 Code ESP32 ตัวที่ 2 ส่งค่าเซ็นเซอร์ต่าง ๆ ให้ ESP32 ตัวที่ 1

3.4 การทำงานของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันจะแบ่งออกเป็นทั้งหมด 2 ส่วน ได้แก่ ด้านการทำงานของผู้ใช้งาน และด้านการทำงานของเซิร์ฟเวอร์

3.4.1 ด้านการทำงานของผู้ใช้งาน

3.4.1.1 ผู้ใช้งานต้องทำการลงทะเบียน และ Login เข้าสู่ระบบ เพื่อเข้าใช้งานเว็บแอปพลิเคชัน

3.4.1.2 เมื่อเข้ามาที่หน้าแรก ระบบจะแสดงความชื้นอุณหภูมิ ค่าของ pH และระดับน้ำในถังของฟาร์มไฮโดรโปนิคส์ ซึ่งค่าต่าง ๆ นี้เป็นการอ่านข้อมูลที่ได้จาก ESP32 นั้นส่งมาเก็บไว้ที่ Realtime Database

3.4.1.3 การเปิด-ปิดการใช้งานอุปกรณ์ ผู้ใช้สามารถสั่งเปิด-ปิดการใช้งานของอุปกรณ์ได้ตามระยะเวลาหรือ

ปริมาณที่ต้องการได้ผ่านทางหน้าเว็บแอปพลิเคชัน สถานะการทำงานจะถูกส่งไปเก็บไว้ที่ฐานข้อมูล เพื่อให้ ESP32 สามารถอ่านข้อมูลไปใช้ในการสั่งการไปยังอุปกรณ์ต่าง ๆ ภายในโรงเรือน อุปกรณ์ที่สามารถเปิด-ปิดการใช้งานได้ ได้แก่ พัดลม, ไฟ, ปั๊มน้ำเข้าถังใหญ่, ปั๊มน้ำทิ้งในถังใหญ่, ปั๊มถั่วงู, ปั๊มถั่วงูอาหาร, ปั๊มถั่วงูค่า pH, ปั๊มถั่วงูลดค่า pH, ปั๊มน้ำ และปั๊มน้ำชีวภัณฑ์

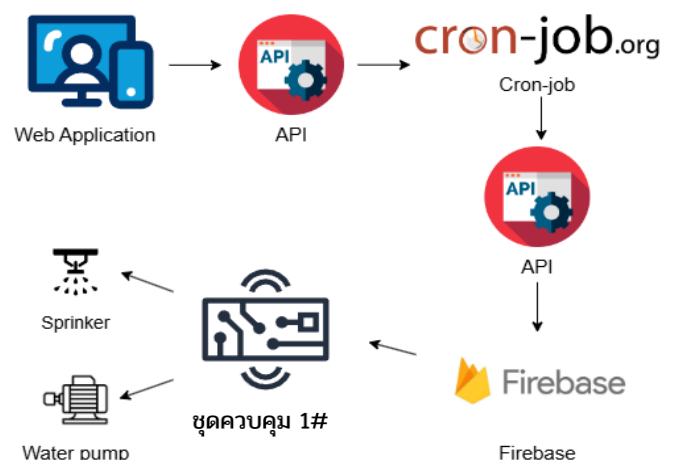
3.4.1.4 การเลือกเติมสารตามปริมาณ ผู้ใช้งานสามารถเลือกเติมสารตามปริมาณที่กำหนดได้ โดยปริมาณที่เติมได้มี 2 ปริมาณ คือ 250 มิลลิกรัม และ 500 มิลลิกรัม เมื่อผู้ใช้ทำการเลือกปริมาณที่ต้องการเติม ข้อมูลจะถูกส่งไปยังฐานข้อมูล เมื่อ Esp32 ได้รับข้อมูล จะทำการสั่งให้อุปกรณ์ที่ต้องการใช้งานทำงาน เมื่อทำงานครบตามปริมาณที่เลือกไว้ อุปกรณ์ก็จะหยุดทำงานโดยอัตโนมัติ อุปกรณ์ที่สามารถเติมตามปริมาณที่กำหนดได้ ได้แก่ ปั๊มถั่วงู, ปั๊มถั่วงูอาหาร, ปั๊มถั่วงูค่า pH, ปั๊มถั่วงูลดค่า pH, ปั๊มน้ำ และปั๊มน้ำ

ชีวภัณฑ์ โดยมีข้อจำกัดคือสามารถเติมได้สูงสุดทีละ 3 อย่างพร้อมกัน เพราะ ESP32 มีข้อจำกัดในการเรียกใช้รัน tasks หลายๆอันพร้อมกัน

3.4.1.5 การตั้งเวลาการทำงานของอุปกรณ์ต่าง ๆ ผู้ใช้สามารถกำหนดเวลาการทำงานของอุปกรณ์ต่าง ๆ ได้ ซึ่งเวลาที่ผู้ใช้ตั้งค่านี้นี้จะถูกส่งไปยังฝั่งเซิร์ฟเวอร์เพื่อนำไปประมวลผลต่อไป อุปกรณ์ที่สามารถตั้งเวลาในการทำงานได้ คือ ถั่วงู, ถั่วงูอาหาร, ถั่วงูค่า pH, ถั่วงูลดค่า pH, ที่พ่นน้ำ และที่พ่นชีวภัณฑ์

3.4.2 ด้านการทำงานของเซิร์ฟเวอร์ แบ่งออกเป็น 2 ส่วน คือ

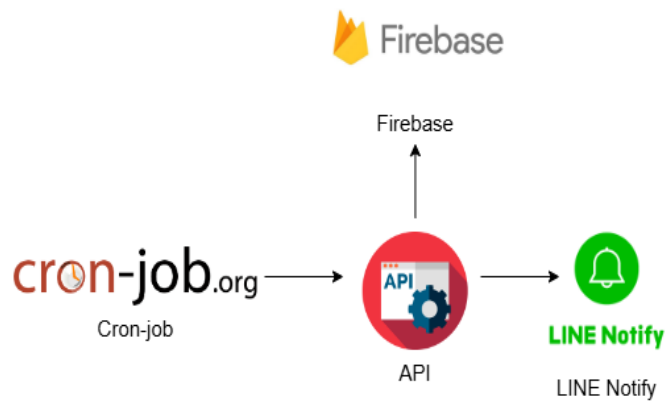
3.4.2.1 การตั้งค่าเวลาการทำงานของอุปกรณ์ สามารถทำได้ผ่านทางเว็บแอปพลิเคชัน โดยผู้ใช้สามารถกำหนดเวลาได้ตามความต้องการ ข้อมูลเวลาที่ตั้งค่าถูกส่งผ่าน API และแปลงเป็นรูปแบบ Cron-Job เมื่อถึงเวลาที่กำหนดใน Cron-Job ระบบจะเรียกใช้งาน API อีกตัวหนึ่งเพื่อเปลี่ยนสถานะการทำงานของอุปกรณ์ตามที่ผู้ใช้ตั้งไว้ นอกจากนี้ผู้ใช้งานสามารถยกเลิกการทำซ้ำของ Cron-Job ได้ผ่านทางหน้าเว็บแอปพลิเคชัน



รูปที่ 11 การตั้งเวลาเปิดใช้งานอุปกรณ์

3.4.2.2. การแจ้งเตือนโดยใช้ Line Notify สามารถทำได้โดยการให้ผู้เข้าร่วมกลุ่มไลน์ จากนั้นทุก ๆ 5 นาที Cron-Job จะทำการเรียกเส้น API ให้เช็คค่าและแจ้งเตือนเมื่อมีความผิดปกติ ดังนี้

- อุณหภูมิสูงกว่า 30 องศา
- ความชื้นสูงกว่า 80% หรือ ความชื้นต่ำกว่า 40%
- ค่า pH สูงกว่า 8 หรือ ค่า pH ต่ำกว่า 3



รูปที่ 12 การแจ้งเตือนโดยใช้ Line Notify

3.4 ระบบฐานข้อมูล

ระบบฐานข้อมูลใช้ Realtime Database ของ Firebase ที่เก็บข้อมูลในรูปแบบ JSON เพื่อให้สามารถรับส่งค่าได้แบบเรียลไทม์ โดยทางฝั่งของหน้าเว็บและฝั่งของ ESP 32 จะสามารถเรียกอ่านข้อมูลมาแสดงและประมวลผลได้ทันที ภายในฐานข้อมูลจะเก็บสถานะการทำงานของอุปกรณ์ เวลาที่ถูกตั้งค่า รวมถึงปริมาณต่าง ๆ ที่จะถูกอ่านเพื่อไปใช้ในการสั่งงานผ่านอุปกรณ์ ESP32 และนำไปประมวลผลอื่น ๆ ต่อไป

ดังรูปที่ 13 เป็นการเก็บข้อมูลในรูปแบบ JSON ซึ่งจะเก็บค่าต่าง ๆ สำหรับแสดงผล ได้แก่ อุณหภูมิ ความชื้น และความเป็นกรด-เบส สำหรับค่า Quantity เป็นการเก็บค่าปริมาณของสารที่ผู้ใช้งานเลือกในหน้าเว็บ เพื่อให้ EPS32 สามารถอ่านค่านี้ไปใช้ในการจับเวลาให้อุปกรณ์ทำการเติมสารลงในถังตามปริมาณที่ผู้ใช้งานตั้งค่าไว้ การเก็บค่าสถานะการทำงานของอุปกรณ์จะถูกเก็บไว้ใน Relay State นอกจากนี้มีการเก็บค่าเวลาไว้สำหรับแสดงผลบนหน้าเว็บ

และ Water State ซึ่งเป็นการเก็บระดับของน้ำในถังเพื่อให้ผู้ใช้งานสามารถรู้ปริมาณของน้ำในถังได้

นอกจากนี้มีการเก็บค่าที่วัดได้จากเซ็นเซอร์วัดความชื้นและอุณหภูมิ รวมถึงค่าที่วัดได้จากเซ็นเซอร์วัด pH โดยในทุก ๆ 1 ชั่วโมง ค่าต่าง ๆ จะถูกเก็บไว้ในฐานข้อมูล หลังจากนั้นจะนำข้อมูลเหล่านี้มาสร้างเป็นกราฟเส้น เพื่อให้เห็นถึงการเปลี่ยนแปลงของอุณหภูมิ ความชื้น และค่า pH บนหน้าเว็บแอปพลิเคชัน ดังรูปที่ 14

```

{
  "Humidity": 63,
  "Temperature": 30,
  "pHValue": 6.78,
  "quantityFT": 0,
  "quantityMB": 0,
  "quantityphdown": 0,
  "quantityphup": 0,
  "quantitiesprinklerfertilizers": 0,
  "quantitiesprinklerwater": 0,
  "relaystate": {
    "fan": true,
    "fertilizers": false,
    "led": true,
    "microbial": false,
    "pumpStirring": false,
    "pumpUP": true,
    "pumpphDown": false,
    "pumpphUP": false,
    "pumpwater": false,
    "sprinklerfertilizers": false,
    "sprinklerwater": false,
    "valve": false
  },
  "timeFT": "",
  "timeMB": "",
  "timePHD": "",
  "timePHU": "",
  "timeSFT": "",
  "timeSWT": "",
  "waterstatehigh": false,
  "waterstatelow": true
}
  
```

รูปที่ 13 ข้อมูลที่ถูกเก็บในรูปแบบ JSON

```

"logSensor": [
  null,
  {
    "humidity": 70,
    "ph": 6.538,
    "temperature": 25,
    "timestamp": "2024-6-26 7:21:12"
  },
  {
    "humidity": 69,
    "ph": 6.653,
    "temperature": 26,
    "timestamp": "2024-6-26 8:21:12"
  }
]
  
```

รูปที่ 14 ข้อมูลที่วัดได้จากเซ็นเซอร์

4. ผลการดำเนินงาน

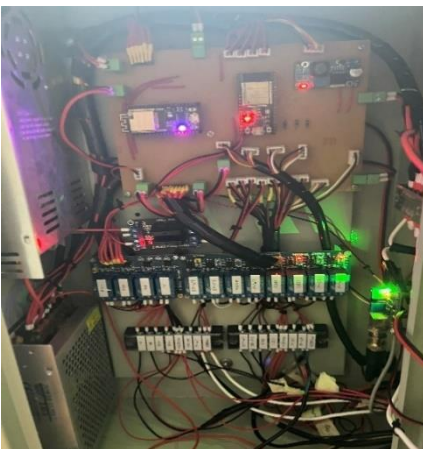
จากการดำเนินการออกแบบและพัฒนาการเว็บแอปพลิเคชันฟาร์มไฮโดรโปนิคส์ ระบบสามารถทำงานได้ตามขอบเขตที่กำหนด โดยมีผลการทดสอบ ส่วนต่างๆ ดังต่อไปนี้

4.1 การทำงานของตัวอุปกรณ์

จากการออกแบบส่วนต่าง ๆ ของระบบทั้งซอฟต์แวร์และฮาร์ดแวร์ เราได้สร้างเครื่องต้นแบบชุดฟาร์มไฮโดรโปนิคส์ แสดงดังรูปที่ 15 และฮาร์ดแวร์ส่วนควบคุมแสดงดังรูปที่ 16



รูปที่ 15 เครื่องต้นแบบชุดฟาร์มไฮโดรโปนิคส์



รูปที่ 16 ฮาร์ดแวร์ส่วนควบคุม

ซึ่งจะทำการทดสอบในส่วนต่าง ๆ ดังต่อไปนี้

4.1.1 การวัดระดับน้ำในถังเก็บน้ำ

การทดลองนี้จะทำการใช้ เซ็นเซอร์วัดระดับน้ำ XKC-Y25 จำนวน 2 ตัวเข้ากับถังเก็บน้ำ เพื่อทำการทดสอบว่าสามารถตรวจจับน้ำในถังได้หรือไม่



รูปที่ 17 การติดตั้งเซ็นเซอร์ตรวจจับระดับน้ำ

ขั้นตอนการทดลอง

ขั้นตอนที่ 1 อุปกรณ์ที่ใช้ในการทดสอบ โดยมีรายละเอียดดังนี้

- จำนวนเซ็นเซอร์ที่ใช้ : 2 ตัว
- ถังเก็บน้ำ ขนาด 5 ลิตร : 1 ถัง
- ระยะห่างจากกันถึงของเซ็นเซอร์ตัวที่ 1 : 0 เซนติเมตร
- ระยะห่างจากกันถึงของเซ็นเซอร์ตัวที่ 2 : 10 เซนติเมตร

ขั้นตอนที่ 2 ติดตั้งเซ็นเซอร์ตามรูปที่ 17 เพื่อการวัดระดับน้ำในถัง จากนั้นเขียนโปรแกรมเพื่ออ่านค่าการเปลี่ยนแปลงของเซ็นเซอร์และแสดงผลบน Serial Monitor ของโปรแกรม Arduino IDE ซึ่งจะแสดงผลหน้าจอตัวอย่างดัง

รูปที่ 18 จากนั้นเติมน้ำทีละ 1 ลิตร แล้วบันทึกค่าที่เซ็นเซอร์วัดได้ พร้อมทั้งบันทึกผลลงในตารางที่ 1

```
13:20:43.660 -> waterstatehigh = low
13:20:43.660 -> waterstatelow = low 0 ลิตร
-----
13:20:53.690 -> waterstatehigh = low
13:20:53.690 -> waterstatelow = high 1 ลิตร
-----
13:21:03.688 -> waterstatehigh = low
13:21:03.688 -> waterstatelow = high 2 ลิตร
-----
13:21:13.697 -> waterstatehigh = low
13:21:13.697 -> waterstatelow = high 3 ลิตร
-----
13:21:23.680 -> waterstatehigh = high
13:21:23.680 -> waterstatelow = high 4 ลิตร
-----
13:21:33.682 -> waterstatehigh = high
13:21:33.682 -> waterstatelow = high 5 ลิตร
```

รูปที่ 18 ค่าการวัดระดับน้ำในถังจากเซ็นเซอร์

จากผลการทดลองในตารางที่ 1 สังเกตได้ว่าค่าเซ็นเซอร์ทั้ง 2 ตัวสามารถตรวจจับน้ำได้ เมื่อน้ำอยู่ระดับเดียวกับขอบล่างของเซ็นเซอร์ จากนั้นนำข้อมูลทั้ง 2 ค่ามาแสดงผลที่หน้าเว็บตัวอย่างดังรูปที่ 19

ตารางที่ 1 ผลการวัดค่าระดับน้ำในถัง

น้ำที่ เติม	เซ็นเซอร์ ระดับน้ำ (low)	เซ็นเซอร์ ระดับน้ำ (high)	แสดงผลหน้า เว็บ
0	low	low	ต่ำกว่า 10 %
1	high	low	ต่ำกว่า 80 %
2	high	low	ต่ำกว่า 80 %
3	high	low	ต่ำกว่า 80 %
4	high	high	มากกว่า 80 %
5	high	high	มากกว่า 80 %



รูปที่ 19 ผลระดับน้ำในถัง

4.1.2 การวัดความชื้นและอุณหภูมิภายในโรงเรือน

เซ็นเซอร์วัดความชื้นและอุณหภูมิ DHT22 เทียบกับ มิเตอร์วัดความชื้นและอุณหภูมิที่มีจำหน่ายในท้องตลาดเพื่อ ตรวจสอบว่ามีค่าใกล้เคียงกันหรือไม่ รวมถึงดูการเปลี่ยนแปลงของค่าความชื้นและอุณหภูมิเมื่อมีการเปลี่ยนแปลง



รูปที่ 20 เซนเซอร์วัดความชื้นและอุณหภูมิ

ขั้นตอนการทดลอง

ขั้นตอนที่ 1 อุปกรณ์ที่ใช้ในการทดสอบ โดยมี รายละเอียดดังนี้

- จำนวนเซ็นเซอร์ที่ใช้ : 1 ตัว
- จำนวนมิเตอร์ที่ใช้ในการวัดเทียบค่า : 1 เครื่อง
- กระบอกฉีดละอองน้ำ : 1 กระบอก
- ไดรฟ์เป่าลม : 1 เครื่อง

ขั้นตอนที่ 2 ติดตั้งเซ็นเซอร์ตามรูปที่ 20 เพื่อทดสอบวัดความชื้นและอุณหภูมิภายในโรงเรือน จากนั้นเขียนโปรแกรมเพื่ออ่านค่าเซ็นเซอร์แสดงผลบน Serial Monitor ของโปรแกรม Arduino IDE การทดสอบจะใช้กระบอกฉีดละอองน้ำแล้วดูค่าที่วัดได้จากเซ็นเซอร์บนหน้าจอคอมพิวเตอร์ดังรูปที่ 21 แล้วนำค่าบันทึกลงในตารางที่ 2

15:19:48.245 -> Humidity: 63.00 % Temperature: 32.80 °C ครั้งที่ 1
 15:19:58.258 -> Humidity: 91.00 % Temperature: 30.10 °C
 15:20:08.249 -> Humidity: 67.00 % Temperature: 31.30 °C ครั้งที่ 2
 15:20:18.280 -> Humidity: 71.00 % Temperature: 30.60 °C
 15:20:28.265 -> Humidity: 56.00 % Temperature: 33.50 °C ครั้งที่ 3
 15:20:38.235 -> Humidity: 64.00 % Temperature: 32.40 °C

รูปที่ 21 ค่าที่เซ็นเซอร์วัดความชื้นและอุณหภูมิอ่านค่าได้

ตารางที่ 2 ผลการทดลอง DHT22 โดยการฉีดละอองน้ำ

การทดลอง (ครั้ง)	ก่อนฉีดละอองน้ำ		หลังฉีดละอองน้ำ	
	ความชื้น (%RH)	อุณหภูมิ (°C)	ความชื้น (%RH)	อุณหภูมิ (°C)
1	90	29.0	91	30.1
2	69	30.0	71	30.6
3	63	31.0	64	32.4

ส่วนอีกทดสอบจะใช้ไดรฟ์เป่าลมเป่าลมร้อนให้กับเซ็นเซอร์ DHT22 แล้วดูผลการวัดบนจอคอมพิวเตอร์ดังรูปที่ 22 แล้วทำการบันทึกค่าที่อ่านได้ลงในตารางที่ 3

19:32:47.467 -> Humidity: 63.00 % Temperature: 31.80 °C ครั้งที่ 1
 19:32:57.468 -> Humidity: 49.00 % Temperature: 36.70 °C
 19:33:07.455 -> Humidity: 64.00 % Temperature: 32.40 °C ครั้งที่ 2
 19:33:17.456 -> Humidity: 37.00 % Temperature: 41.60 °C
 19:33:27.461 -> Humidity: 71.00 % Temperature: 38.00 °C
 19:33:37.434 -> Humidity: 43.00 % Temperature: 39.10 °C ครั้งที่ 3

รูปที่ 22 ค่าที่เซ็นเซอร์วัดความชื้นและอุณหภูมิอ่านค่าได้

ตารางที่ 3 ผลการทดลอง DHT22 โดยไดรฟ์เป่าลม

การทดลอง (ครั้ง)	ก่อนใช้ไดรฟ์เป่า		หลังใช้ไดรฟ์เป่า	
	ความชื้น (%RH)	อุณหภูมิ (°C)	ความชื้น (%RH)	อุณหภูมิ (°C)
1	63	31.8	49	36.7
2	64	32.4	37	41.6
3	71	38.0	43	39.1

จากผลการทดลองในตารางที่ 2 และ 3 พบว่า เมื่อเซ็นเซอร์สัมผัสกับละอองน้ำ ค่าความชื้นจะเพิ่มขึ้น และเมื่อได้รับความร้อนจากไดรฟ์เป่าลม ค่าอุณหภูมิของเซ็นเซอร์ก็สูงขึ้น

ขั้นตอนที่ 3 ทำการทดสอบความถูกต้องของเซ็นเซอร์ โดยการ ทดสอบเซ็นเซอร์วัดความชื้นและอุณหภูมิเทียบกับ มิเตอร์ภายในโรงเรือน จากนั้นเขียนโปรแกรมเพื่ออ่านค่า การเปลี่ยนแปลงของเซ็นเซอร์และแสดงผลบน Serial Monitor ของโปรแกรม Arduino IDE ซึ่งจะแสดงผลหน้าจอตัวอย่างดังรูปที่ 23 จากนั้นบันทึกค่าความชื้นและอุณหภูมิที่อ่านได้จากเซ็นเซอร์เทียบกับมิเตอร์ดังรูปที่ 24 แล้วบันทึกผลลงในตารางที่ 4 และพล็อตเป็นกราฟดังรูปที่ 25 และ 26

19:57:51.070 -> Humidity: 64.00 % Temperature: 32.80 °C ครั้งที่ 1
19:58:01.086 -> Humidity: 68.00 % Temperature: 33.30 °C ครั้งที่ 2
19:58:11.095 -> Humidity: 59.00 % Temperature: 36.50 °C ครั้งที่ 3
19:58:21.107 -> Humidity: 60.00 % Temperature: 35.00 °C ครั้งที่ 4
19:58:31.098 -> Humidity: 61.00 % Temperature: 34.80 °C ครั้งที่ 5
19:58:41.101 -> Humidity: 69.00 % Temperature: 39.50 °C ครั้งที่ 6

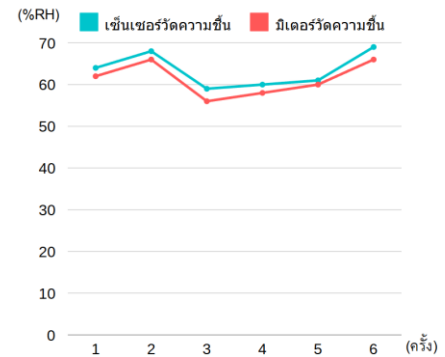
รูปที่ 23 ค่าที่เซ็นเซอร์วัดความชื้นและอุณหภูมิอ่านค่าได้



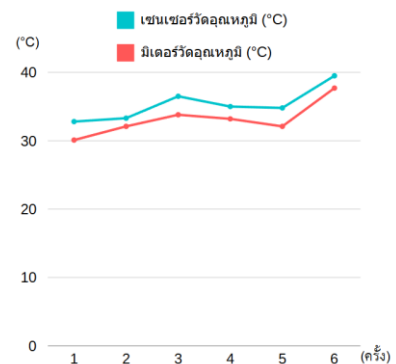
รูปที่ 24 มิเตอร์วัดค่าความชื้นและอุณหภูมิ

ตารางที่ 4 ผลการทดสอบเซ็นเซอร์วัดความชื้นและอุณหภูมิ

การทดลอง	เซ็นเซอร์วัดความชื้นและอุณหภูมิ		มิเตอร์วัดความชื้นและอุณหภูมิ	
	ความชื้น (%RH)	อุณหภูมิ (°C)	ความชื้น (%RH)	อุณหภูมิ (°C)
1	64	32.80	62	30.1
2	68	33.3	66	32.1
3	59	36.5	56	33.8
4	60	35	58	33.2
5	61	34.8	60	32.1
6	69	39.5	66	37.7



รูปที่ 25 กราฟเซ็นเซอร์วัดความชื้น เทียบกับมิเตอร์วัดความชื้น



รูปที่ 26 กราฟเซ็นเซอร์วัดอุณหภูมิ เทียบกับอุปกรณ์วัดอุณหภูมิ

ในผลการทดลองจากตารางที่ 4 สังเกตได้ว่าค่าเซ็นเซอร์วัดความชื้นและอุณหภูมิเมื่อเทียบกับมิเตอร์มีเปอร์เซ็นต์ค่าผิดพลาดสัมบูรณ์เฉลี่ย (MAPE) อยู่ที่ประมาณ 3.41% ซึ่งแสดงให้เห็นว่าค่าที่วัดได้จากเซ็นเซอร์มีความใกล้เคียงกับค่าจริงที่วัดได้จากมิเตอร์ ค่าผิดพลาดนี้อยู่ในระดับที่ยอมรับได้สำหรับการวัด

4.1.3 การวัดค่า PH ในถังเก็บน้ำ

เซ็นเซอร์วัดค่า pH เทียบกับมิเตอร์วัดค่า pH ที่มีจำหน่ายในท้องตลาดเพื่อตรวจสอบว่ามีค่าใกล้เคียงกันหรือไม่



รูปที่ 27 ทดสอบเซนเซอร์วัดค่า pH



รูปที่ 29 ทดสอบมิเตอร์วัดค่า pH

ขั้นตอนการทดลอง

ขั้นตอนที่ 1 อุปกรณ์ที่ใช้ในการทดสอบ โดยมีรายละเอียดดังนี้

- จำนวนเซนเซอร์ที่ใช้ : 1 ตัว
- จำนวนมิเตอร์ที่ใช้ในการวัดเทียบค่า : 1 เครื่อง
- ถ้วยน้ำ : 1 ถ้วย

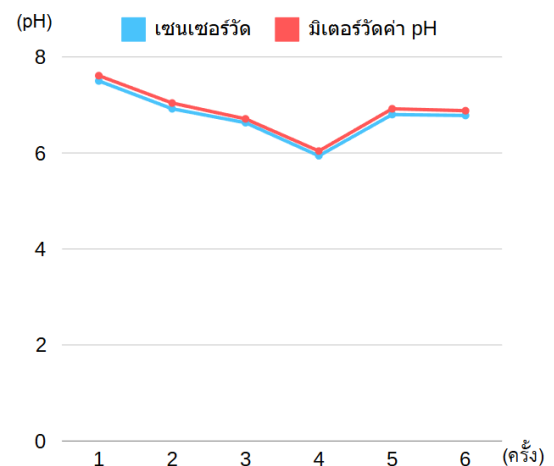
ขั้นตอนที่ 2 เติมน้ำ 200 มิลลิลิตรลงในถ้วยแล้วเติมสารปรับ pH 20 มิลลิลิตรลงในถ้วย จุ่มหัววัดค่าลงไปใต้น้ำ ดังรูปที่ 27 จากนั้นเขียนโปรแกรมเพื่ออ่านค่าการเปลี่ยนแปลงของเซนเซอร์และแสดงผลบน Serial Monitor ของโปรแกรม Arduino IDE ซึ่งจะแสดงผลหน้าจอตัวอย่างดังรูปที่ 28 จากนั้นบันทึกค่า pH ที่อ่านได้จากเซนเซอร์ แล้วเทียบกับมิเตอร์วัดค่า pH ดังรูปที่ 29 แล้วบันทึกผลลงในตารางที่ 5 และพล็อตเป็นกราฟดังรูปที่ 30

```
20:20:20.047 -> phvoltage = 1.70 ph = 7.21
20:20:24.289 -> phvoltage = 1.69 ph = 7.17
20:20:29.524 -> phvoltage = 1.70 ph = 7.21
20:20:31.679 -> phvoltage = 1.69 ph = 7.17
20:20:34.195 -> phvoltage = 1.68 ph = 7.13
20:20:36.298 -> phvoltage = 1.70 ph = 7.21
20:20:39.287 -> phvoltage = 1.70 ph = 7.21
```

รูปที่ 28 ค่าที่อุปกรณ์วัดค่า pH อ่านค่าได้

ตารางที่ 5 ผลการทดสอบเซนเซอร์วัด pH และ มิเตอร์วัดค่า pH

ตัวอย่างของเหลว	ค่า pH ที่วัดจากเซนเซอร์	ค่า pH ที่อ่านได้จากมิเตอร์วัดค่า pH
1	7.50	7.61
2	6.92	7.04
3	6.63	6.71
4	5.94	6.04
5	6.80	6.92
6	6.78	6.88



ภาพที่ 30 กราฟผลการวัดค่า pH ของเซนเซอร์และมิเตอร์

จากผลการทดลองในตารางที่ 5 สังเกตได้ว่าค่าเซ็นเซอร์วัดค่า pH เมื่อเทียบกับมิเตอร์มีเปอร์เซ็นต์ค่าผิดพลาดสัมบูรณ์เฉลี่ย (MAPE) อยู่ที่ประมาณ 1.56% ซึ่งแสดงให้เห็นว่าค่าที่วัดได้จากเซ็นเซอร์มีความใกล้เคียงกับค่าจริงที่วัดได้จากมิเตอร์ ค่าผิดพลาดนี้อยู่ในระดับที่ยอมรับได้สำหรับการวัดทำให้เซ็นเซอร์นี้มีความน่าเชื่อถือในการใช้งาน

4.1.4 การควบคุมอุปกรณ์ต่าง ๆ ผ่าน รีเลย์

อุปกรณ์ภายในโรงเรือนทั้งหมด 12 อย่างถูกควบคุมผ่านรีเลย์ โดยมีรายละเอียดดังนี้

1. หลอดไฟ
2. พัดลม
3. ปั๊มน้ำวน
4. ปั๊มน้ำทิ้ง
5. ปั๊มน้ำเข้า
6. ปั๊มปุ๋ย
7. ปั๊มสารอาหาร
8. ปั๊ม pH+
9. ปั๊ม pH-
10. ปั๊มสปริงเกอร์น้ำ
11. ปั๊มสปริงเกอร์ชีวภัณฑ์
12. ปั๊มกวนน้ำ

ESP32 ตัวที่ 1 จะทำหน้าที่อ่านค่าจาก Firebase แล้วสั่งการเปิด-ปิดอุปกรณ์ต่าง ๆ ตามค่าที่ตั้งไว้ ดังรูปที่ 31



รูปที่ 31 รีเลย์ ที่ใช้ควบคุมอุปกรณ์

ทดลองสั่งเปิด-ปิด พัดลม และ หลอดไฟ ดังรูปที่ 32



รูปที่ 32 พัดลม และ หลอดไฟ

ทดลองสั่งเปิด-ปิด ปั๊มน้ำต่าง ๆ ดังรูปที่ 33



รูปที่ 33 ปั๊มต่าง ๆ

4.1.5 สวิตช์รีเซตการเชื่อมต่อ WiFi

ผู้ใช้งานสามารถเปลี่ยนการเชื่อมต่อ WiFi ได้โดยการกดค้างสวิตช์สีแดงตามที่แสดงในรูปที่ 34 เป็นระยะเวลา 5 วินาที การกดสวิตช์นี้จะทำให้ระบบเริ่มต้นการค้นหาสัญญาณ WiFi ใหม่และเชื่อมต่อกับเครือข่ายที่ต้องการ



รูปที่ 34 สวิตช์รีเซตการเชื่อมต่อ WiFi

4.1.6 สวิตช์เปิด-ปิดการทำงานของ ไฟในวงจร และรีเลย์

ผู้ใช้งานสามารถเปิด-ปิดสวิตช์ 2 ตัวตามที่แสดงในรูปที่ 35 โดยการหมุนสวิตช์เพื่อควบคุมการจ่ายไฟให้กับ รีเลย์ และไฟในวงจร ซึ่งการกระทำนี้จะตัดการจ่ายไฟหรือเปิดการจ่ายไฟให้กับอุปกรณ์ดังกล่าวตามต้องการ



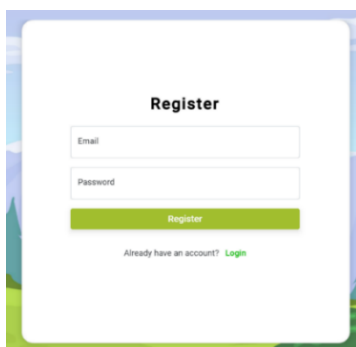
รูปที่ 35 สวิตช์เปิด-ปิดการทำงานของ ไฟในวงจร และ รีเลย์

4.2 การทำงานของเว็บแอปพลิเคชัน

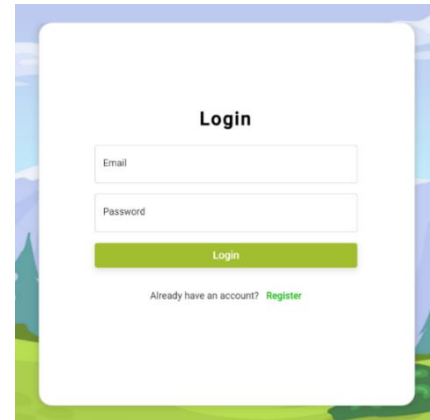
การทำงานของเว็บแอปพลิเคชันสามารถแบ่งการทำงานได้ ดังนี้

4.2.1 หน้าเข้าสู่ระบบและสมัครสมาชิก

ก่อนเริ่มต้นเข้าใช้งานเว็บแอปพลิเคชัน ผู้ใช้งานจะต้องสมัครสมาชิกเพื่อเข้าใช้งานดังรูปที่ 36 เมื่อสมัครสมาชิกเรียบร้อยแล้ว จะสามารถเข้าสู่ระบบได้ ดังรูปที่ 37



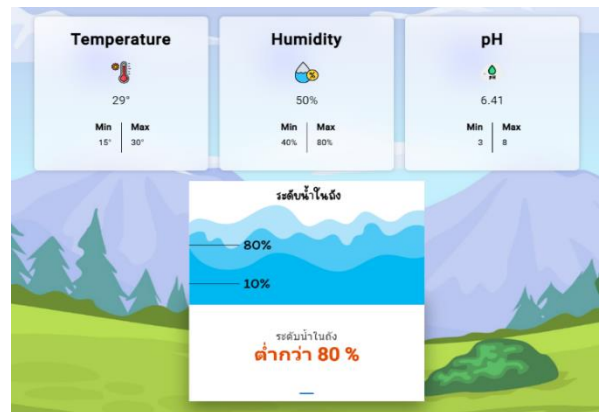
รูปที่ 36 หน้าลงทะเบียน



รูปที่ 37 หน้าเข้าสู่ระบบ

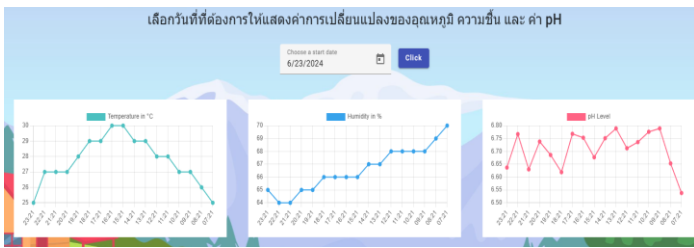
4.2.2 การแสดงผลข้อมูลต่าง ๆ ภายในโรงเรือน

หลังจากที่ผู้ใช้งานเข้าสู่ระบบเรียบร้อยแล้ว จะถูกนำไปที่หน้า Home โดยหน้านี้จะเป็นการแสดงผลข้อมูลต่าง ๆ ได้แก่ อุณหภูมิ ความชื้น ค่า pH และระดับน้ำถัง ดังรูปที่ 38 รวมถึงจะมี QR Code ให้เข้าร่วมกลุ่มไลน์เพื่อรับการแจ้งเตือนการทำงานของอุปกรณ์ต่าง ๆ ในโรงเรือน

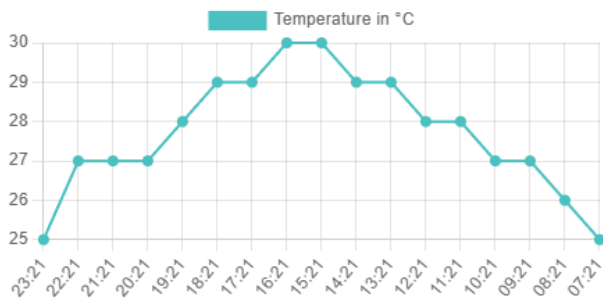


รูปที่ 38 หน้าการแสดงผลข้อมูลต่าง ๆ ภายในโรงเรือน

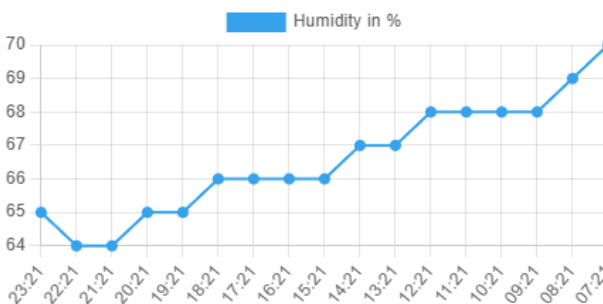
นอกจากนี้มีการแสดงผลของกราฟที่แสดงให้เห็นถึงการเปลี่ยนแปลงของอุณหภูมิ ความชื้น และค่า pH ทุก ๆ 1 ชั่วโมง โดยผู้ใช้งานสามารถเลือกวันที่ที่ต้องการดูกราฟได้ ดังรูปที่ 39



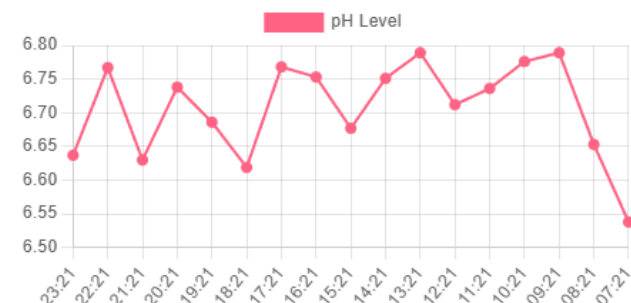
รูปที่ 39 การแสดงผลข้อมูลการเปลี่ยนแปลงของอุณหภูมิ
ความชื้น และค่า pH



รูปที่ 40 กราฟแสดงผลการเปลี่ยนแปลงของอุณหภูมิ



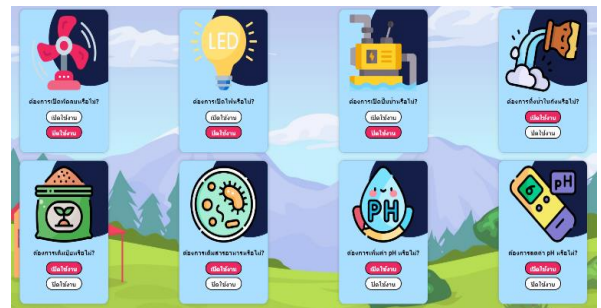
รูปที่ 41 กราฟแสดงผลการเปลี่ยนแปลงของความชื้น



รูปที่ 42 กราฟแสดงผลการเปลี่ยนแปลงของค่า pH

4.2.3 การเปิดปิดการใช้งานอุปกรณ์

ผู้ใช้งานสามารถกดเปิดปิดอุปกรณ์ต่าง ๆ ผ่านทางเว็บแอปพลิเคชันนี้ได้ ดังรูปที่ 43



รูปที่ 43 หน้าเปิดปิดการใช้งานอุปกรณ์

4.2.4 การเติมสารต่าง ๆ ตามปริมาณ

ผู้ใช้งานสามารถเลือกเติมน้ำ ปุ๋ย สารอาหาร สารเพิ่มและลดค่า pH ตามปริมาณดังรูปที่ 44 เพื่อสั่งงานให้เติมสารตามปริมาณที่เลือกระหว่าง 250 ml และ 500 ml



รูปที่ 44 หน้าเติมสารต่าง ๆ ตามปริมาณ

4.2.5 การตั้งค่าเวลาในการเปิดใช้งานอุปกรณ์

ในหน้าตั้งค่าเวลาดังรูปที่ 45 นี้ ไว้สำหรับผู้ใช้งานที่ต้องการตั้งค่าเวลาในการเปิดใช้งานอุปกรณ์ล่วงหน้าหรือต้องการให้ทำซ้ำในช่วงเวลานี้เป็นประจำ โดยผู้ใช้งานสามารถยกเลิกการตั้งค่าเวลานี้ได้

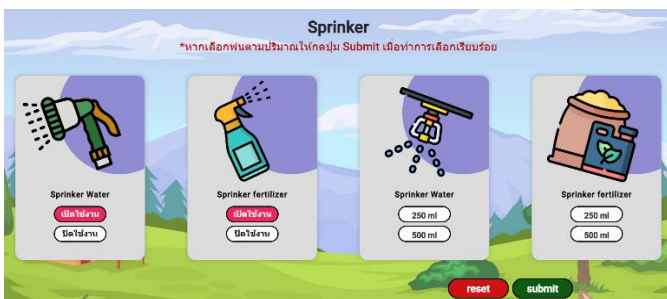


รูปที่ 45 หน้าตั้งค่าเวลาในการเปิดใช้งานอุปกรณ์

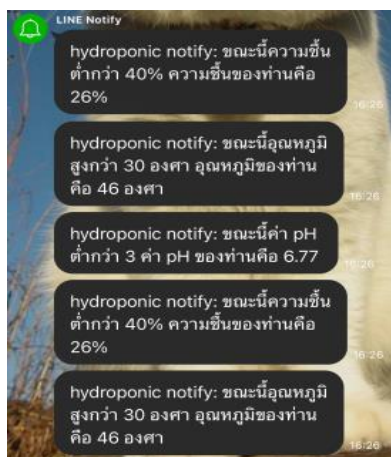
4.2.6 การใช้งานสปริงเกอร์

ในหน้านี้ ดังรูปที่ 46 และ 47 จะเป็นการสั่งการทำงานของสปริงเกอร์ที่พ่นน้ำและสารอาหาร ซึ่งจะเป็นการรวมการทำงานตั้งแต่ข้อที่ 4.2.3 - 4.2.5 คือ

- เปิด-ปิดสปริงเกอร์ได้ตามที่ต้องการ
- เปิดสปริงเกอร์ได้ตามปริมาณที่ต้องการเปิด
- กำหนดเวลาในการเปิดล่วงหน้าหรือเปิดเป็นประจำ



รูปที่ 46 หน้าการใช้งานสปริงเกอร์



รูปที่ 47 หน้าการใช้งานสปริงเกอร์

4.2.7 การแจ้งเตือน

เมื่อผู้ใช้งานเข้าร่วมกลุ่มไลน์ดังรูปที่ 48 จะมีการแจ้งเตือนจากระบบดังรูปที่ 49 ดังนี้

- ถึงเวลาเริ่มเปิดใช้พ่นน้ำ
- ถึงเวลาเริ่มเปิดใช้พ่นปุ๋ย
- ถึงเวลาเริ่มเติมสารเพิ่มค่า pH
- ถึงเวลาเริ่มเติมสารลดค่า pH
- ถึงเวลาเริ่มเติมปุ๋ย
- ถึงเวลาเริ่มเติมสารอาหาร
- อุณหภูมิสูงกว่า 30 องศา
- ความชื้นสูงกว่า 80% หรือ ความชื้นต่ำกว่า 40%
- ค่า pH สูงกว่า 8 หรือ ค่า pH ต่ำกว่า 3



รูปที่ 48 QR Code ของไลน์เพื่อรับการแจ้งเตือน



รูปที่ 49 ตัวอย่างการตั้งค่าแจ้งเตือน

5. สรุปผล

โครงการปริญญานิพนธ์ฉบับนี้ได้นำเสนอการพัฒนาเว็บแอปพลิเคชันร่วมกับการทำฟาร์มไฮโดรโปนิคส์ โดยการสร้างแบบจำลองฟาร์มไฮโดรโปนิคส์ที่มีบอร์ด ESP32 ช่วยในการรับค่าและควบคุมการทำงานอุปกรณ์ต่าง ๆ ภายในฟาร์ม ได้แก่ ไฟ พัดลม ปั๊ม สปริงเกอร์ เซ็นเซอร์อุณหภูมิ-ความชื้น และเซ็นเซอร์วัดความเป็นกรด-เบส รวมถึงมีการพัฒนาเว็บแอปพลิเคชันด้วย Angular และ Node.js เพื่อใช้ในการแสดงผลต่างๆ และ ให้ผู้ใช้งานสามารถสั่งการทำงานของอุปกรณ์ภายในฟาร์ม นอกจากนี้มีการใช้งานระบบฐานข้อมูลแบบเรียลไทม์ เพื่อให้หน้าเว็บและ ESP32 สามารถอ่านค่าจากฐานข้อมูลไปใช้แสดงผลและประมวลผลต่อ รวมถึงสามารถแจ้งเตือนเมื่อพบว่าอุณหภูมิ-ความชื้น ความเป็นกรดเบส มีความผิดปกติและแจ้งเตือนผ่าน Line Notify เมื่อถึงเวลาการเปิดใช้งานอุปกรณ์ต่างๆ ภายในฟาร์มที่ผู้ใช้งานตั้งค่าไว้จากการทดสอบการใช้เว็บแอปพลิเคชันร่วมกับฟาร์มไฮโดรโปนิคส์พบว่าสามารถควบคุมอุปกรณ์ต่างๆ ภายในฟาร์มไฮโดรโปนิคส์ได้ ซึ่งช่วยให้ประหยัดเวลาและช่วยในการอำนวยความสะดวกให้กับผู้ใช้งานในการดูแลฟาร์มไฮโดรโปนิคส์

6. เอกสารอ้างอิง

- [1] Angular. (2567). [ออนไลน์]. [สืบค้นเมื่อ 17 กุมภาพันธ์ 2567]. จาก <https://perjerz.medium.com/angular>
- [2] Firebase. (2567). [ออนไลน์]. [สืบค้นเมื่อ 17 กุมภาพันธ์ 2567]. จาก <https://medium.com/jed-ng/firebase>
- [3] Line Notify. (2567). [ออนไลน์]. [สืบค้นเมื่อ 22 กุมภาพันธ์ 2567]. จาก <https://notify-bot.line.me/th>
- [4] REST API. (2567). [ออนไลน์]. [สืบค้นเมื่อ 22 กุมภาพันธ์ 2567]. จาก <https://kongruksiam.medium.com/>
- [5] Cron Job. (2559). [ออนไลน์]. [สืบค้นเมื่อ 28 กุมภาพันธ์ 2567]. จาก <https://www.codebee.co.th/>
- [6] Microsoft Azure. (2566).] [ออนไลน์]. [สืบค้นเมื่อ 28 กุมภาพันธ์ 2567]. จาก <https://appmaster.io/>
- [7] ESP32. (2567). [ออนไลน์]. [สืบค้นเมื่อ 28 กุมภาพันธ์ 2567]. จาก <https://www.artronshop.co.th/article/51/esp32>
- [8] เซ็นเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ DHT22 (2567). [ออนไลน์]. [สืบค้นเมื่อ 28 กุมภาพันธ์ 2567]. จาก <https://www.analogread.com/product/169/>
- [9] เซ็นเซอร์ระดับน้ำ XKC-Y25. (2567). [ออนไลน์]. [สืบค้นเมื่อ 28 กุมภาพันธ์ 2567]. จาก <https://www.allnewstep.com/product/5/>
- [10] โมดูลเซ็นเซอร์ pH (2567). [ออนไลน์]. [สืบค้นเมื่อ 28 กุมภาพันธ์ 2567] จาก <https://www.cybertice.com/product/3449/>

ประวัติผู้จัดทำ



ชื่อ-นามสกุล : นายจิรเมธ แก้วคำ

อีเมล : s6303051623063@email.kmutnb.ac.th

ประวัติการศึกษา

พ.ศ. 2563 หลักสูตรประกาศนียบัตรวิชาชีพ สาขางานอิเล็กทรอนิกส์
วิทยาลัยเทคนิคเชียงราย

ปัจจุบัน เทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ แขนงวิชาคอมพิวเตอร์
วิทยาลัยเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ประวัติผู้จัดทำ



ชื่อ-นามสกุล : นางสาวณัฏฐณิชา เจวรัมย์

อีเมล : s6303051623161@email.kmutnb.ac.th

ประวัติการศึกษา

พ.ศ. 2563 ระดับมัธยมศึกษาตอนปลาย แผนการเรียนวิทยาศาสตร์-คณิตศาสตร์
โรงเรียนเทพศิรินทร์ นนทบุรี

ปัจจุบัน เทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ แขนงวิชาคอมพิวเตอร์
วิทยาลัยเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ