

เว็บแอปพลิเคชัน ฟาร์มไฮโดรโปนิกส์

นายจิรเมธ แก้วคำ

นางสาวณัฏฐนิชา เจริมย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ (คอมพิวเตอร์)

ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์

วิทยาลัยเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

พ.ศ.2567

Hydroponic Farm Web Application

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 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 : Hydroponic Farm Web Application
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)

..... Member
(Asst. Prof. Dr. Lerson Kirasamuthranon)

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

กิตติกรรมประกาศ

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

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

สุดท้ายนี้ทางคณะผู้จัดทำต้องขอขอบพระคุณท่านกรรมการสอบโครงการปริญญานิพนธ์ทุกท่านเป็นอย่างสูงที่ได้ช่วยพิจารณาและให้คำแนะนำในการตรวจทานแก้ไข อนุมัติจนโครงการปริญญานิพนธ์นี้สำเร็จเป็นไปตามวัตถุประสงค์และขอบเขตที่ตั้งไว้ทุกประการ ซึ่งผู้จัดทำหวังว่าโครงการปริญญานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้ที่ทำการปลูกผักไฮโดรโปนิคส์

คณะผู้จัดทำ

เว็บแอปพลิเคชัน ฟาร์มไฮโดรโปนิคส์

จิรเมธ แก้วคำ¹, ณัฏฐณิชา เจริมย์¹ 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 ในส่วนการแจ้งเตือนค่าผ่านทาง 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 ดึงค่าจากฐานข้อมูลมาแสดงผลและประมวลผล อีกทั้งยังมีการแจ้งเตือนผ่าน Line Notify เมื่อพบว่าค่าต่าง ๆ ผิดปกติหรือตามเวลาที่ตั้งค่าไว้ จากการทดสอบพบว่าเว็บแอปพลิเคชันสามารถควบคุมอุปกรณ์ในฟาร์มไฮโดรโปนิคส์ได้จริง ช่วยประหยัดเวลาและเพิ่มความสะดวกในการดูแลฟาร์มของผู้ใช้

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

¹นักศึกษา, ²อาจารย์ที่ปรึกษาภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์, วิทยาลัยเทคโนโลยีอุตสาหกรรม, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

Hydroponic Farm Web Application

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 จึงมีผู้คนจำนวนหนึ่งที่มีความสนใจในการทำเกษตรในพื้นที่เล็ก ๆ บริเวณบ้านหรือระเบียงของตนเอง เพื่อปลูกพืชผักสวนครัวที่จำเป็นต่อการดำรงชีวิต แต่การดำรงชีวิตของมนุษย์ในปัจจุบัน มักมีปัญหาในเรื่องของเวลาที่ใช้ในการดูแลพื้นที่การเกษตร เนื่องจากการดูแลพื้นที่การเกษตรนั้น ต้องมีความละเอียดและซับซ้อน เพื่อให้ผลผลิตออกมาได้อย่างมีคุณภาพ

โครงงานปริญญานิพนธ์นี้เน้นการออกแบบและพัฒนาอุปกรณ์การปลูกฟาร์มไฮโดรโปนิคส์ที่สามารถแสดงค่าสถานะต่าง 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 และมีประสิทธิภาพมากยิ่งขึ้น ในการปฏิบัติงานทาง

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-แอมมา-แองกูลาร์ โดยมีคุณสมบัติที่ช่วยให้การพัฒนาแอปพลิเคชันเป็นไปอย่างรวดเร็วและมีประสิทธิภาพ แต่ละส่วนประกอบมีความสัมพันธ์กันอย่างเป็นระบบ โดยที่ Angular จะใช้สำหรับฝั่งของไคลเอนต์ เฟรมเวิร์กแบบอื่น 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 อีกมากมายที่ช่วยให้ นักพัฒนาสามารถพัฒนาและบริหารจัดการแอปพลิเคชันได้อย่างมีประสิทธิภาพและสะดวกสบาย

2.3 Line Notify

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

ผู้ใช้สามารถรับการแจ้งเตือนเกี่ยวกับสถานะหรือเหตุการณ์ต่าง 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 ดึงข้อมูล
- 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

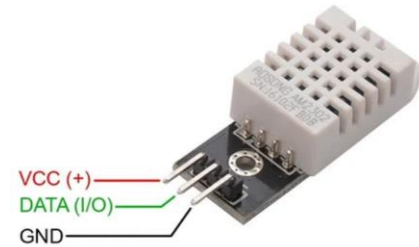
- CPU Dual-core Tensilica LX6 microprocessor ที่ความเร็ว 240 MHz
- ไฟเลี้ยง 3.3V (ตัวบอร์ดรับแรงดัน 5V ได้ มีเรกูเลเตอร์)
- รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB
- ทำงานได้ที่อุณหภูมิ -40°C ถึง 125°C
- พอร์ตดิจิทัล มีทั้งหมด 36 พอร์ต GPIO
- รองรับการเชื่อมต่อกับเซ็นเซอร์และอุปกรณ์ต่างๆ ผ่าน SPI, I2C, 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] มีความแม่นยำสูงและใช้งานง่าย มีคุณสมบัติดังนี้

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

ภาพที่ 1 เป็นรูปร่างภายนอกและขาสัญญาณของเซนเซอร์ DHT22 โดย +VCC และ GND คือขาไฟเลี้ยงย่าน 3.3-6V และขา Data ต่อเข้ากับพอร์ตอินพุตแบบดิจิทัลของไมโครคอนโทรลเลอร์



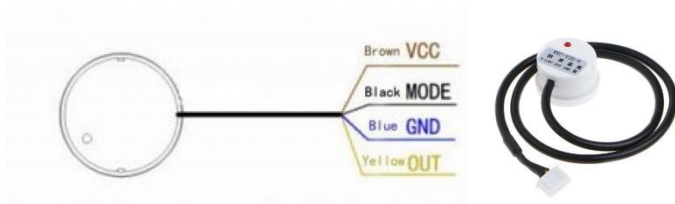
ภาพที่ 1 รูปร่าง และสายสัญญาณของเซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์

2.9 เซนเซอร์ระดับน้ำ

XKC-Y25 [9] เป็นเซนเซอร์สำหรับตรวจจับวัดระดับของเหลวแบบไม่สัมผัสของเหลว ภาชนะที่ติดตั้งต้องไม่ใช่โลหะ มีความไวสูง และมีความทนทาน สามารถใช้งานได้ ในสภาพแวดล้อมที่หลากหลาย 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

ภาพที่ 2 เป็นรูปร่างภายนอกและสายสัญญาณของเซนเซอร์ XKC-Y25 โดยที่ขา VCC และ GND สำหรับต่อไฟเลี้ยง ส่วนขา OUT เป็นขาเอาต์พุตเซนเซอร์ที่ใช้ต่อกับดิจิทัลอินพุตของไมโครคอนโทรลเลอร์



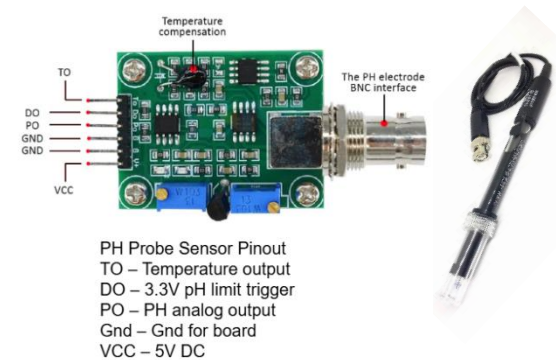
ภาพที่ 2 เซนเซอร์วัดระดับน้ำ XKC-Y25

2.10 โมดูลเซนเซอร์ pH

โมดูลเซนเซอร์ pH [10] เป็นอุปกรณ์ที่ใช้สำหรับวัดค่าความเป็นกรด-ด่างของสารละลายต่าง ๆ ซึ่งมักใช้ในงานเกษตรกรรม การควบคุมคุณภาพน้ำ การทดลองทางเคมี และการทำระบบไฮโดรโปนิคส์ โมดูลนี้สามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์เพื่อการบันทึกข้อมูลและการควบคุมระบบได้อย่างมีประสิทธิภาพ มีคุณสมบัติ ดังนี้

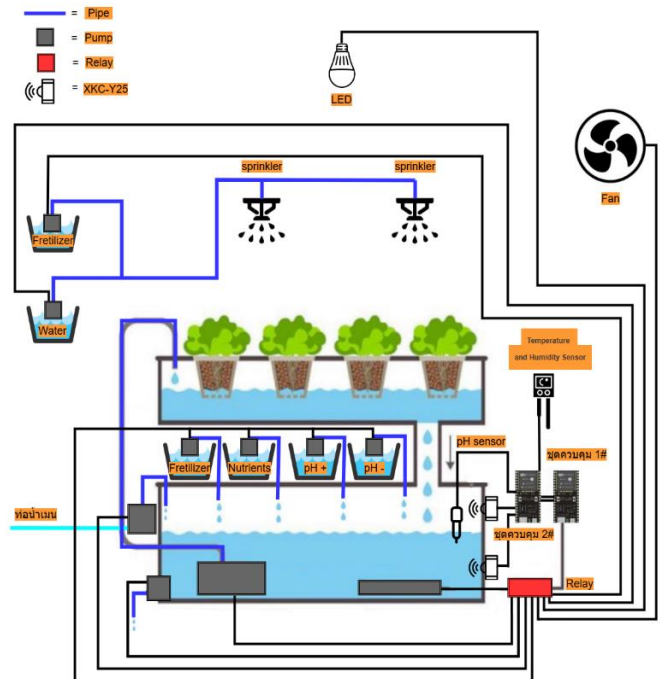
- ช่วงการวัดค่า pH 0 ถึง 14
- ความแม่นยำ ± 0.1 pH
- ความละเอียด 0.01 pH
- แหล่งจ่าย 5V DC ที่กระแสไฟฟ้าสูงสุด 10mA
- เวลาตอบสนอง น้อยกว่า 1 นาที (ขึ้นอยู่กับสภาพแวดล้อม)
- ขา PO เชื่อมต่อกับพอร์ตอนาล็อกของไมโครคอนโทรลเลอร์ ขา PO จะให้แรงดัน 0-5V และแปรตามแบบเชิงเส้นกับค่า pH (0-14)

ภาพที่ 3 เป็นโมดูลวัดค่า pH ประกอบด้วยโพรบและวงจรปรับสภาพสัญญาณ โดยขา VCC และ GND คือส่วนของไฟเลี้ยงเซนเซอร์และวงจร ส่วนขา PO เป็นเอาต์พุตอนาล็อก 0-5V ซึ่งจะนำไปคำนวณ



ภาพที่ 3 รูปร่าง และสายสัญญาณของเซนเซอร์ pH

3. วิธีการดำเนินงาน



ภาพที่ 4 แผนผังโรงเรือน

จากภาพที่ 4 แบ่งถึงเก็บสารเป็นทั้งหมด 7 ถัง โดยมีถังใหญ่ 1 ถัง ที่มีปั๊มทั้งหมด 4 ตัว ได้แก่ ปั๊มน้ำเข้าถังใหญ่, ปั๊มน้ำทิ้งในถังใหญ่, ปั๊มน้ำวน, ปั๊มน้ำวน และถังเล็กอีก 6 ถัง โดยแต่ละถังมีปั๊มเป็นของตัวเอง ได้แก่ ถังสารอาหาร, ถังปุ๋ย, ถังสารเพิ่มค่า pH, ถังสารลดค่า pH, ถังน้ำ, ถังชีวภัณฑ์

การพัฒนาระบบควบคุมฟาร์มไฮโดรโปนิกส์ ประกอบไปด้วย 2 ส่วนหลัก คือ ระบบไมโครคอนโทรลเลอร์ควบคุม ฟาร์มไฮโดรโปนิกส์ และเว็บแอปพลิเคชัน เพื่อการสั่งงาน และแสดงสถานะของฟาร์มไฮโดรโปนิกส์ โดยมีรายละเอียดในแต่ละส่วนดังนี้

3.1 หลักการทำงานโดยรวมของระบบ

จากภาพที่ 5 ส่วนของฮาร์ดแวร์ควบคุมฟาร์มประกอบด้วย บอร์ดไมโครคอนโทรลเลอร์ ESP32 2 ชุด คือ ชุดควบคุม 1# และชุดควบคุม 2#

ชุดควบคุม 2# เชื่อมต่อกับเซนเซอร์ 3 ชนิด ได้แก่ เซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ จำนวน 1 ตัว เซนเซอร์ตรวจจับน้ำในถังเก็บน้ำจำนวน 2 ตัว และโมดูลเซนเซอร์ pH จำนวน 1 ตัว เพื่อวัดค่า pH ในถังเก็บน้ำโดยค่าที่อ่านได้จากเซนเซอร์จะส่งผ่านพอร์ตอนุกรม (Rx และ Tx) ไปยังชุดควบคุม 1# เพื่อที่ชุดควบคุม 1# จะใช้ค่าที่ได้จากเซนเซอร์ที่ชุดควบคุม 2# ส่งมาให้เป็นทีกับบนฐานข้อมูล Firebase แล้วสั่งการเปิด-ปิดรีเลย์ตามข้อมูลจาก Firebase และอีกส่วนที่เป็นเว็บแอปพลิเคชัน ออกแบบมาเพื่อให้สามารถติดต่อสื่อสารกับผู้ใช้โดยตรง โดยแบ่งการทำงานออกเป็น 3 ส่วนหลัก ดังนี้ คือ

การรับ-ส่งข้อมูลกับ Firebase ผ่าน API เพื่อให้ข้อมูลอัปเดตแบบเรียลไทม์ ใช้ในการแสดงผล, การเปิด-ปิดการใช้งานอุปกรณ์, การเลือกเติมสารตามปริมาณ ระบบนี้ช่วยให้ผู้ใช้งานสามารถรับค่าสถานะต่าง ๆ และควบคุมการทำงานของอุปกรณ์ได้แบบเรียลไทม์ผ่านหน้าเว็บแอปพลิเคชัน

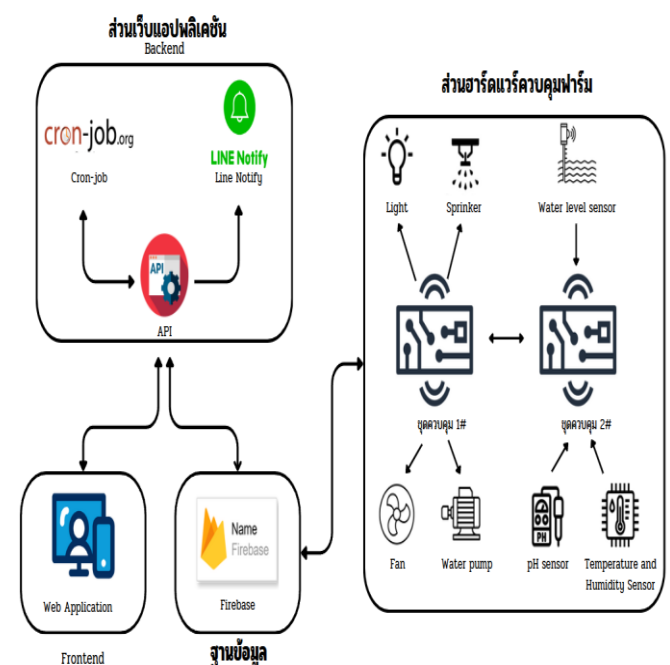
การใช้ API และ Cron-job ในการตั้งเวลาเปิดใช้งานอุปกรณ์ ผู้ใช้ตั้งค่ากำหนดเวลาผ่านหน้าเว็บ ซึ่งข้อมูลที่ได้รับจะถูกแปลงให้อยู่ในรูปแบบของ Cron-job โดยใช้ API

เส้นที่ 1 ในการติดต่อสื่อสารกับ Server เมื่อถึงเวลาที่กำหนดไว้ Cron-job จะถูกเรียกใช้เพื่อเรียกใช้งาน API เส้นที่ 2 ซึ่งจะติดต่อสื่อสารกับ Firebase เพื่อเปลี่ยนสถานะของอุปกรณ์ การเปลี่ยนสถานะนี้จะเกิดขึ้นโดยอัตโนมัติตามเวลาที่ตั้งไว้ใน Cron-job

การตรวจสอบสถานะและแจ้งเตือน การตั้งเวลาใน Cron-job ให้ทำงานทุก ๆ 5 นาที เพื่อเรียกใช้ API ในการตรวจสอบข้อมูลใน Firebase ว่ามีการทำงานที่ผิดปกติหรือไม่ หากพบความผิดปกติเช่น อุณหภูมิสูงกว่า 30 องศา, ความชื้นสูงกว่า 80% หรือ ความชื้นต่ำกว่า 40%, ค่า pH สูงกว่า 8 หรือ ค่า pH ต่ำกว่า 3 ระบบจะทำการแจ้งเตือนไปยังผู้ใช้งานทันที

3.2 การทำงานของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันจะแบ่งออกเป็นทั้งหมด 2 ส่วน ได้แก่ ด้านการทำงานของผู้ใช้งาน และด้านการทำงานของเซิร์ฟเวอร์



ภาพที่ 5 โครงสร้างรวมของระบบ

3.2.1 ด้านการทำงานของผู้ใช้งาน

3.2.1.1 ผู้ใช้งานต้องทำการลงทะเบียน และ Login เข้าสู่ระบบ เพื่อเข้าใช้งานเว็บแอปพลิเคชัน

3.2.1.2 เมื่อเข้ามาที่หน้าแรก ระบบจะแสดงค่าความชื้น อุณหภูมิ ค่าของ pH และระดับน้ำในถังของฟาร์มไฮโดรโปนิคส์ ซึ่งค่าต่าง ๆ นี้เป็นการดึงข้อมูลที่จาก ESP32 นั้นส่งมาเก็บไว้ที่ Realtime Database

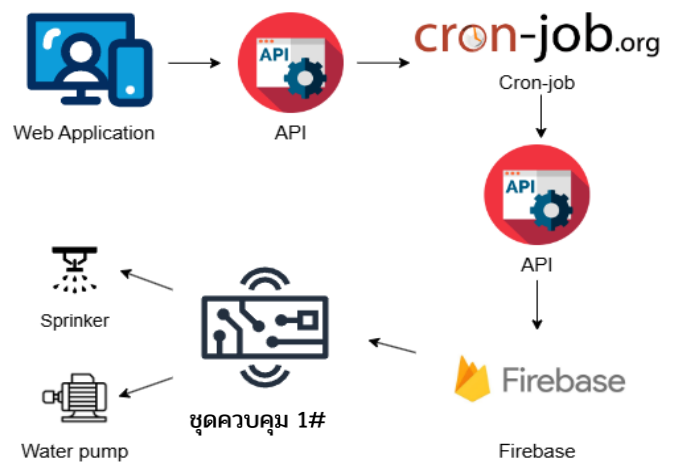
3.2.1.3 การเปิด-ปิดการใช้งานอุปกรณ์ ผู้ใช้สามารถตั้งเปิด-ปิดการใช้งานของอุปกรณ์ได้ตามระยะเวลาหรือปริมาณที่ต้องการได้ผ่านทางหน้าเว็บแอปพลิเคชัน สถานะการทำงานจะถูกส่งไปเก็บไว้ที่ฐานข้อมูล เพื่อให้ ESP32 สามารถดึงข้อมูลไปใช้ในการสั่งการไปยังอุปกรณ์ต่าง ๆ ภายในโรงเรือน อุปกรณ์ที่สามารถเปิด-ปิดการใช้งานได้ ได้แก่ พัดลม, ไฟ, ปั๊มน้ำเข้าถังใหญ่, ปั๊มน้ำทิ้งในถังใหญ่, ปั๊มถังปุ๋ย, ปั๊มถังสารอาหาร, ปั๊มถังเพิ่มค่า pH, ปั๊มถังลดค่า pH, ปั๊มน้ำ และปั๊มน้ำชีวภัณฑ์ ดังภาพที่ 4

3.2.1.4 การเลือกเติมสารตามปริมาณ ผู้ใช้งานสามารถเลือกเติมสารตามปริมาณที่กำหนดได้ โดยปริมาณที่เติมได้มี 2 ปริมาณ คือ 250 มิลลิลิตร และ 500 มิลลิลิตร เมื่อผู้ใช้ทำการเลือกปริมาณที่ต้องการเติม ข้อมูลจะถูกส่งไปยังฐานข้อมูล เมื่อ Esp32 ได้รับข้อมูล จะทำการสั่งให้อุปกรณ์ที่ต้องการใช้งานทำงาน เมื่อทำงานครบตามปริมาณที่เลือกไว้ อุปกรณ์ก็จะหยุดทำงานโดยอัตโนมัติ อุปกรณ์ที่สามารถเติมตามปริมาณที่กำหนดได้ ได้แก่ ปั๊มถังปุ๋ย, ปั๊มถังสารอาหาร, ปั๊มถังเพิ่มค่า pH, ปั๊มถังลดค่า pH, ปั๊มน้ำ และปั๊มน้ำชีวภัณฑ์ โดยมีข้อจำกัดคือสามารถเติมได้สูงสุดทีละ 3 อย่างพร้อมกัน เพราะ ESP32 มีข้อจำกัดในการเรียกใช้รัน tasks หลายๆอันพร้อมกัน

3.2.1.5 การตั้งเวลาการทำงานของอุปกรณ์ต่าง ๆ ผู้ใช้สามารถกำหนดเวลาการทำงานของอุปกรณ์ต่าง ๆ ได้ ซึ่งเวลาที่ผู้ใช้ตั้งค่านี้อาจถูกส่งไปยังฝั่งเซิร์ฟเวอร์เพื่อนำไปประมวลผลต่อไป อุปกรณ์ที่สามารถตั้งเวลาในการทำงานได้ คือ ถังปุ๋ย, ถังสารอาหาร, ถังเพิ่มค่า pH, ถังลดค่า pH, ที่พ่นน้ำ และที่พ่นชีวภัณฑ์

3.2.2 ด้านการทำงานของเซิร์ฟเวอร์ แบ่งออกเป็น 2 ส่วน คือ

3.2.2.1 การตั้งค่าเวลาการทำงานของอุปกรณ์ สามารถทำได้ผ่านทางเว็บแอปพลิเคชัน โดยผู้ใช้สามารถกำหนดเวลาได้ตามความต้องการ ข้อมูลเวลาที่ตั้งค่าถูกส่งผ่าน API และแปลงเป็นรูปแบบ Cron-Job เมื่อถึงเวลาที่กำหนดใน Cron-Job ระบบจะเรียกใช้งาน API อีกตัวหนึ่งเพื่อเปลี่ยนสถานะการทำงานของอุปกรณ์ตามที่ผู้ใช้ตั้งไว้ นอกจากนี้ ผู้ใช้ยังสามารถยกเลิกการทำซ้ำของ Cron-Job ได้ผ่านทางหน้าเว็บแอปพลิเคชัน

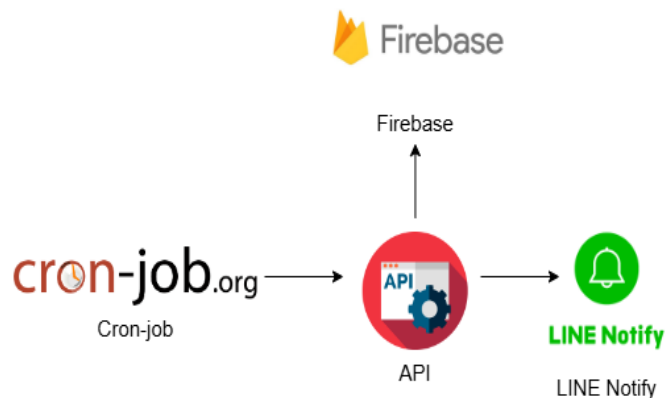


ภาพที่ 6 การตั้งเวลาเปิดใช้งานอุปกรณ์

3.2.2.2. การแจ้งเตือนโดยใช้ Line Notify สามารถทำได้ โดยการให้ผู้เข้าร่วมกลุ่มไลน์ 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



ภาพที่ 7 การแจ้งเตือนโดยใช้ Line Notify

3.3 ระบบฐานข้อมูล

ระบบฐานข้อมูลใช้ Realtime Database ของ Firebase ที่เก็บข้อมูลในรูปแบบ JSON เพื่อให้สามารถรับส่งค่าได้แบบเรียลไทม์ โดยทางฝั่งของหน้าเว็บและฝั่งของ ESP 32 จะสามารถเรียกดึงข้อมูลมาแสดงและประมวลผลได้ทันทีภายในฐานข้อมูลจะเก็บสถานะการทำงานของอุปกรณ์ เวลาที่ถูกตั้งค่า รวมถึงปริมาณต่าง ๆ ที่จะถูกดึงเพื่อไปใช้ในการสั่งงานผ่านอุปกรณ์ ESP32 และนำไปประมวลผลอื่น ๆ ต่อไป

ดังภาพที่ 7 เป็นการเก็บข้อมูลในรูปแบบ JSON ซึ่งจะเก็บค่าต่าง ๆ สำหรับแสดงผล ได้แก่ อุณหภูมิ ความชื้น และความเป็นกรด-เบส สำหรับค่า Quantity เป็นการเก็บค่าปริมาณของสารที่ผู้ใช้งานเลือกในหน้าเว็บ เพื่อให้ EPS32

สามารถดึงค่านี้ไปใช้ในการจับเวลาให้อุปกรณ์ทำการเติมสารลงในถังตามปริมาณที่ผู้ใช้งานตั้งค่าไว้ การเก็บค่าสถานะการทำงานของอุปกรณ์จะถูกเก็บไว้ใน Relay State นอกจากนี้มีการเก็บค่าเวลาไว้สำหรับแสดงผลบนหน้าเว็บและ Water State ซึ่งเป็นการเก็บระดับของน้ำในถังเพื่อให้ผู้ใช้งานสามารถรู้ปริมาณของในถังได้

นอกจากนี้มีการเก็บค่าที่วัดได้จากเซ็นเซอร์วัดความชื้นและอุณหภูมิ รวมถึงค่าที่วัดได้จากเซ็นเซอร์วัด pH โดยในทุก ๆ 1 ชั่วโมง ค่าต่าง ๆ จะถูกเก็บไว้ในฐานข้อมูล หลังจากนั้นจะนำข้อมูลเหล่านี้มาสร้างเป็นกราฟเส้น เพื่อให้เห็นถึงการเปลี่ยนแปลงของอุณหภูมิ ความชื้น และค่า pH บนหน้าเว็บแอปพลิเคชัน ดังภาพที่ 9

```
"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
}
```

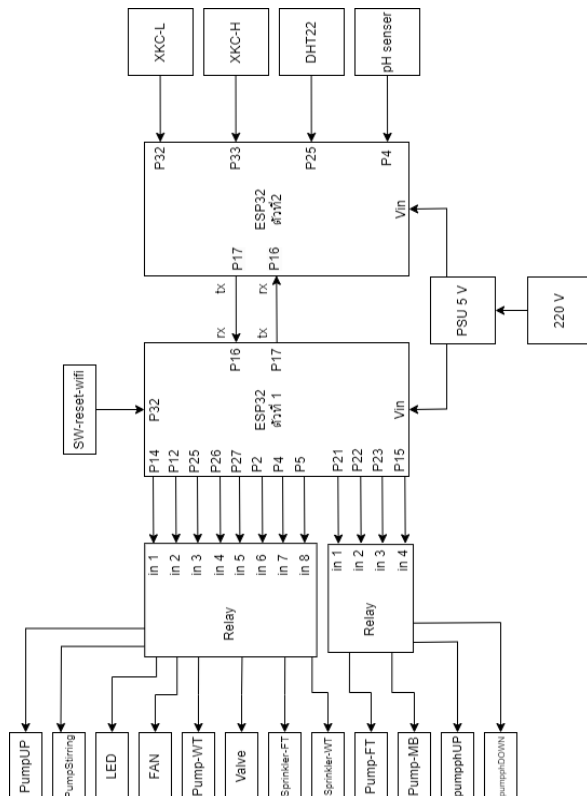
ภาพที่ 8 ข้อมูลที่ถูกเก็บในรูปแบบ JSON

```
"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"
  },
]
```

ภาพที่ 9 ข้อมูลที่ถูกวัดค่าจากเซ็นเซอร์

3.4 การทำงานของไมโครคอนโทรลเลอร์

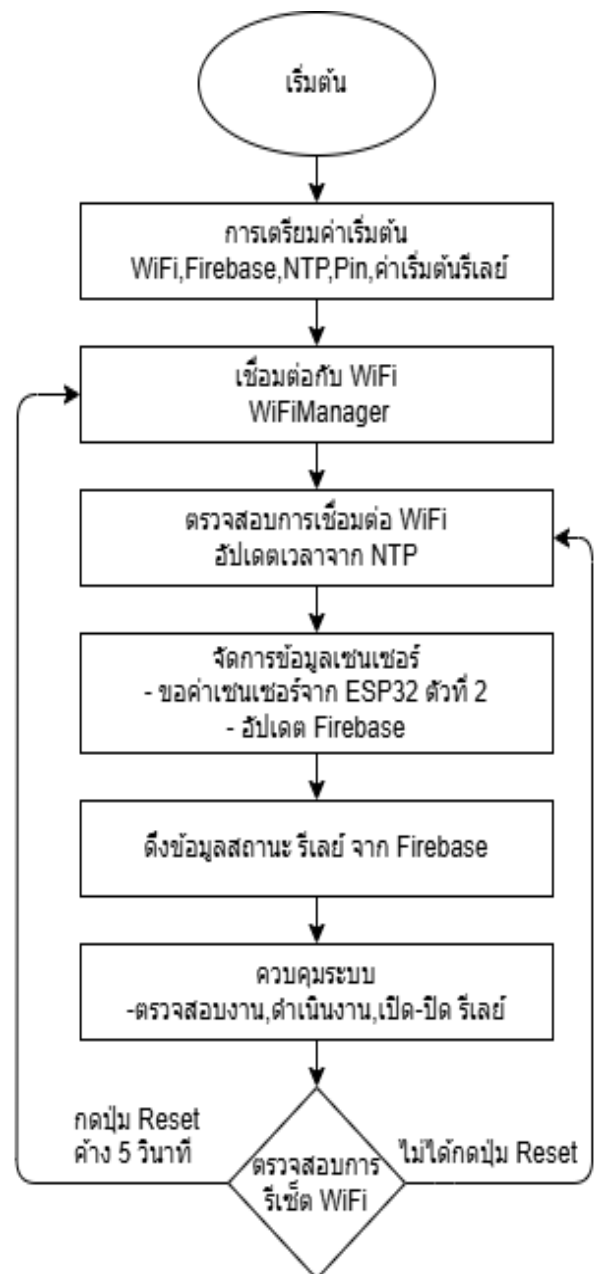
ใช้บอร์ด ESP32 ที่มีการเขียนโปรแกรมขึ้นมารองรับการสื่อสารกับ Firebase โดยส่งข้อมูลเซ็นเซอร์และสถานะต่าง ๆ ของ รีเลย์ให้ Firebase เพื่อให้สามารถสั่งการเปิด-ปิดวาล์วน้ำผ่านแอปพลิเคชัน



ภาพที่ 10 โครงสร้างของอุปกรณ์

การเขียนรับส่งค่าระหว่างบอร์ด จะใช้บอร์ด ESP32 ทั้งหมด 2 ตัว การรับส่งค่าจะเขียนไฟล์ส่ง ไว้ที่ตัวบอร์ด ESP32 ตัวที่ 1 และนำ ESP32 ตัวที่ 2 ทำหน้าที่รับค่าเซ็นเซอร์ต่าง ๆ ที่วัดได้ โดยแบ่งเป็นออกเป็น 2 ฟังก์ชัน

3.4.1 ESP32 ตัวที่ 1



ภาพที่ 11 ฟังก์ชันการทำงานของ ESP32 ตัวที่ 1

มีหน้าที่เชื่อมต่อกับ Firebase โดยใช้ API และ Database URL เพื่อให้สามารถติดต่อสื่อสารกับ Firebase ได้ ดังภาพที่ 12

```
#include <WiFi.h>
#include <FirebaseESP32.h>

#define API_KEY "XXXXXXXXXXXXXXXXXXXXXdbh2xk"
#define DATABASE_URL "https://test-esp32-14872-default-rtdb.firebaseio.com/"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

void setup() {
  Serial.printf("FireBase Client v%s\n", FIREBASE_CLIENT_VERSION);

  config.api_key = API_KEY;
  config.database_url = DATABASE_URL;
  Firebase.begin(DATABASE_URL, API_KEY);
  Firebase.setDoubleDigits(5);
}
```

ภาพที่ 12 Code ESP32 ตัวที่ 1 เชื่อมต่อกับ Firebase

และเรียกค่าเรียกค่าเซ็นเซอร์ต่าง ๆ จาก ESP32 ตัวที่ 2 ดังภาพที่ 13

```
serialEvent();
String data = Serial2.readStringUntil('\n');
String values[5];
int nusValues = split(data, '|', values, 5);
if (numValues == 5) {
  waterstatehigh = values[0].equals("1");
  waterstatelow = values[1].equals("1");
  h = values[2].toFloat();
  t = values[3].toFloat();
  pHvalue = values[4].toFloat();
  Serial2.flush();
}
```

ภาพที่ 13 Code ESP32 ตัวที่ 1 เรียกค่าเซ็นเซอร์ต่าง ๆ จาก ESP32 ตัวที่ 2

เพื่อส่งข้อมูลเซ็นเซอร์ไปอัปเดตใน Firebase ดังภาพที่ 14

```
if (Firebase.ready()) {
  if (t == 0 && h == 0 && pHValue == 0) {
    getBoolFromFirebase(fbdo, "/waterstatehigh", waterstatehigh);
    getBoolFromFirebase(fbdo, "/waterstatelow", waterstatelow);
    getFloatFromFirebase(fbdo, "/Temperature", t);
    getFloatFromFirebase(fbdo, "/Humidity", h);
    getFloatFromFirebase(fbdo, "/pHValue", pHValue);
  }
  setFloatToFirebase(fbdo, "/pHValue", pHValue);
  setFloatToFirebase(fbdo, "/Humidity", h);
  setFloatToFirebase(fbdo, "/Temperature", t);
  setBoolToFirebase(fbdo, "/waterstatehigh", waterstatehigh);
  setBoolToFirebase(fbdo, "/waterstatelow", waterstatelow);
}
```

ภาพที่ 14 Code ESP32 ตัวที่ 1 ส่งข้อมูลเซ็นเซอร์ให้

Firebase

แล้วนำข้อมูลจาก Firebase มาสั่งเปิด-ปิด รีเลย์ ดังภาพที่ 15

```
if (Firebase.ready()) {
  getBoolFromFirebase(fbdo, "/relaystate/valve", valve);
  NFRE(valve, re6);
  getBoolFromFirebase(fbdo, "/relaystate/led", led);
  NFRE(led, re3);
  getBoolFromFirebase(fbdo, "/relaystate/pumpwater", pumpwater);
  NFRE(pumpwater, re5);
  getBoolFromFirebase(fbdo, "/relaystate/fan", fan);
  NFRE(fan, re4);
  getBoolFromFirebase(fbdo, "/relaystate/pumpphUP", pumpphUP);
  getBoolFromFirebase(fbdo, "/relaystate/pumpphDown", pumpphDown);
  getBoolFromFirebase(fbdo, "/relaystate/sprinklerfertilizers", sprinklerfertilizers);
  getBoolFromFirebase(fbdo, "/relaystate/sprinklerwater", sprinklerwater);
  getBoolFromFirebase(fbdo, "/relaystate/fertilizers", fertilizers);
  getBoolFromFirebase(fbdo, "/relaystate/microbial", microbial);
  getBoolFromFirebase(fbdo, "/relaystate/pumpStirring", pumpStirring);
  getBoolFromFirebase(fbdo, "/relaystate/pumpUP", pumpUP);
}
```

ภาพที่ 15 Code ESP32 ตัวที่ 1 นำข้อมูลจาก Firebase

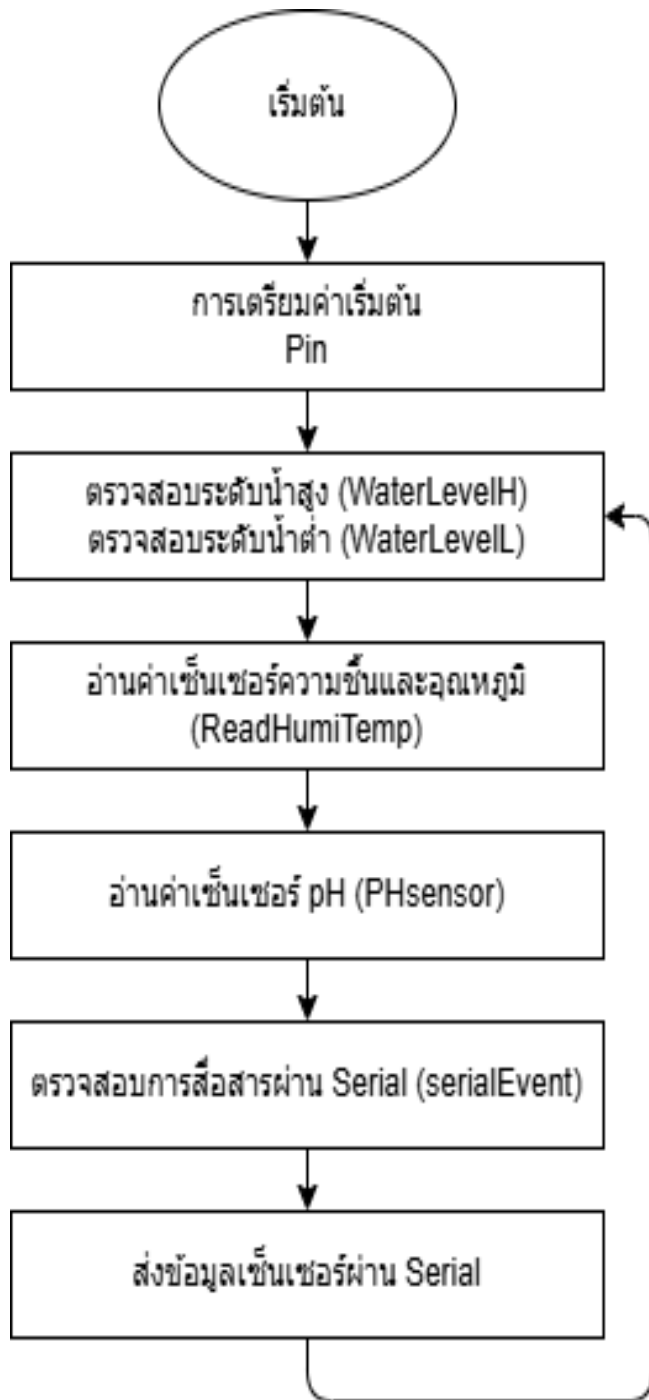
มาสั่งเปิด-ปิด รีเลย์

ตรวจสอบการกดปุ่มรีเซ็ต WiFi ดังภาพที่ 15

```
void Wifi_Reset() {
  if (digitalRead(SWrwf) == LOW) {
    Serial.println("Wifi Reset? waiting 2S..");
    delay(2000);
    if (digitalRead(SWrwf) == LOW) {
      delay(10);
      Serial.println("wifi Reset Setting ..OK");
      wm.resetSettings();
      ESP.restart();
    }
  }
}
```

ภาพที่ 16 Code ESP32 ตัวที่ 1 เช็การกดปุ่มรีเซ็ต WiFi

3.4.2 ESP32 ตัวที่ 2



ภาพที่ 17 ผังงานการทำงานของ ESP32 ตัวที่ 2

มีหน้าที่อ่านค่าเซ็นเซอร์ต่าง ๆ ดังภาพที่ 18

```

void loop() {
  WaterLevelH();
  delay(500);
  WaterLevelL();
  delay(500);
  ReadHumiTemp();
  delay(500);
  PHsensor();
  delay(500);
}
  
```

ภาพที่ 18 Code ESP32 ตัวที่ 2 อ่านค่าเซ็นเซอร์ต่าง ๆ

เพื่อนำค่าเซ็นเซอร์ต่าง ๆ ให้กับ ESP32 ตัวที่ 1 ดังภาพที่ 19

```

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

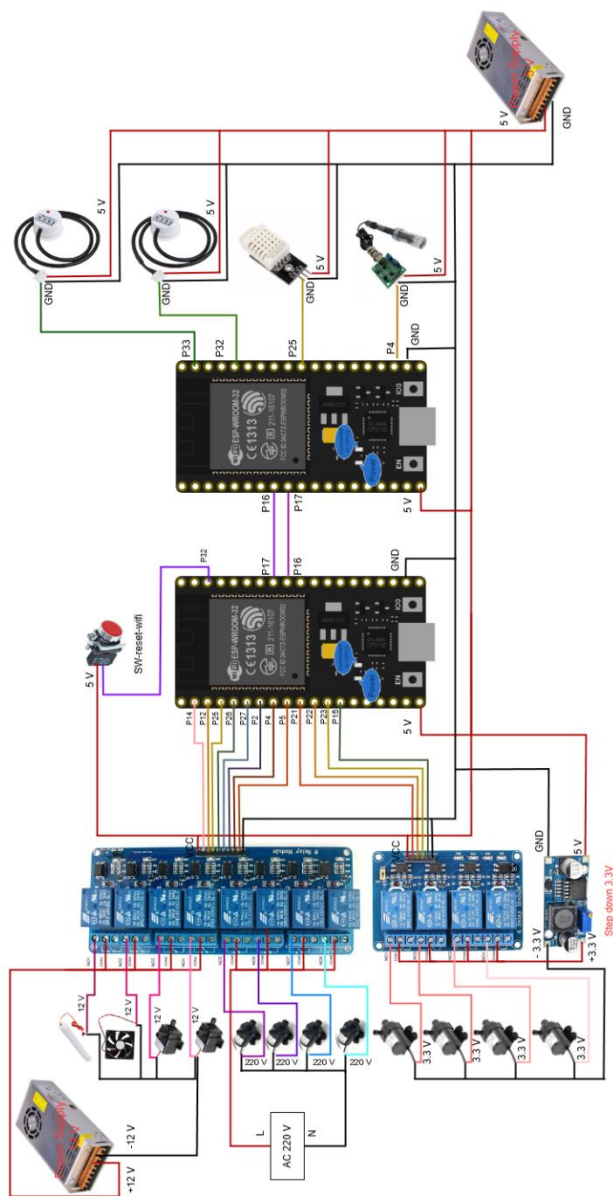
  data += waterstatehigh ? "1" : "0";
  data += "|";
  data += waterstatelow ? "1" : "0";
  data += "|";
  data += h;
  data += "|";
  data += t;
  data += "|";
  data += pHValue;
  Serial.println(data);
  Serial2.println(data);
  sendFlag = false;
}
  
```

ภาพที่ 19 Code ESP32 ตัวที่ 2 ส่งค่าเซ็นเซอร์ต่าง ๆ ให้ ESP32 ตัวที่ 1

4. ผลการดำเนินงาน

จากการดำเนินการออกแบบและพัฒนาการเว็บแอปพลิเคชันฟาร์มไฮโดรโปนิคส์ เป็นไปตามขอบเขตที่ได้วางแผนไว้ระบบมีการใช้งาน ดังนี้

4.1 การทำงานของตัวอุปกรณ์



ภาพที่ 20 วงจรของอุปกรณ์

4.1.1 การวัดระดับน้ำในถังเก็บน้ำ

เซ็นเซอร์วัดระดับน้ำ XKC-Y25 จำนวน 2 ตัวเข้ากับถังเก็บน้ำ โดยตัวแรกจะติดตั้งไว้ที่ระดับ 10% ของความจุถังเก็บน้ำ และตัวที่สองจะติดตั้งไว้ที่ระดับ 80% ตามที่แสดงดังภาพที่ 21 เมื่อระดับน้ำในถังถึงตำแหน่งที่กำหนด เซ็นเซอร์จะส่งค่าสัญญาณ HIGH ไปยัง ESP32 ตัวที่ 2 เพื่อส่งค่าสถานะระดับน้ำในถังเก็บน้ำไปอัปเดตใน Firebase



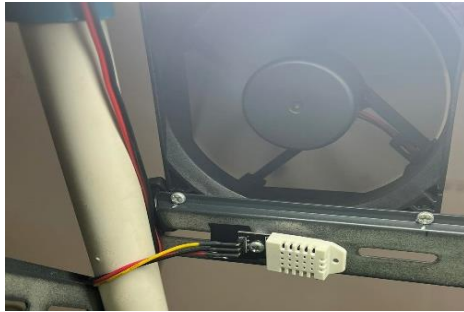
ภาพที่ 21 เซ็นเซอร์ระดับน้ำ

4.1.2 การวัดความชื้นและอุณหภูมิภายในโรงเรือน

เซ็นเซอร์วัดความชื้นและอุณหภูมิ DHT22 ถูกติดตั้งภายในโรงเรือนตามที่แสดงในภาพที่ 22 เพื่อวัดและตรวจสอบค่าความชื้นและอุณหภูมิภายในโรงเรือน เซ็นเซอร์นี้ใช้แรงดันไฟฟ้า 5 โวลต์ สามารถวัดอุณหภูมิในช่วง -40 องศาเซลเซียสถึง 80 องศาเซลเซียส โดยมีความละเอียดในการวัดอุณหภูมิอยู่ที่ 0.5 องศาเซลเซียส และความละเอียดในการวัดความชื้นอยู่ที่ 0.1% RH ความแม่นยำของการวัดอุณหภูมิอยู่ที่ ± 3 องศาเซลเซียส และความแม่นยำของการวัดความชื้นอยู่ที่ $\pm 3\%$ RH

ในแต่ละครั้งที่ทำการอ่านข้อมูลจากเซ็นเซอร์ จะได้ข้อมูลทั้งหมด 40 บิต ซึ่งแบ่งเป็น 16 บิตสำหรับค่าความชื้น 16 บิตสำหรับค่าอุณหภูมิ และ 8 บิตสำหรับการตรวจสอบผลรวมเพื่อยืนยันความถูกต้องของข้อมูลที่อ่านได้ การอ่าน

ค่าจากเซ็นเซอร์นี้จะดำเนินการโดยใช้ ESP32 ตัวที่ 2 ซึ่งจะนำข้อมูลที่ได้อัปโหลดใน Firebase เพื่อนำไปแสดงผลผ่านเว็บไซต์



ภาพที่ 22 เซนเซอร์วัดความชื้นและอุณหภูมิ

ผลการทดลองค่าเซนเซอร์วัดความชื้นและอุณหภูมิจาก serial monitor เพื่อใช้ทดสอบความถูกต้อง ดังภาพที่ 23

```
20:51:19.995 -> Humidity: 64.71 % Temperature: 30.81 C
20:51:24.104 -> Humidity: 64.72 % Temperature: 30.82 C
20:51:25.628 -> Humidity: 64.71 % Temperature: 30.81 C
20:51:28.320 -> Humidity: 64.70 % Temperature: 30.80 C
20:51:30.484 -> Humidity: 64.71 % Temperature: 30.81 C
20:51:32.527 -> Humidity: 64.71 % Temperature: 30.81 C
20:51:34.640 -> Humidity: 64.72 % Temperature: 30.82 C
```

ภาพที่ 23 ค่าที่เซนเซอร์วัดความชื้นและอุณหภูมิอ่านค่าได้

ค่าที่ อุปกรณ์วัดความชื้นและอุณหภูมิอ่านค่าได้เพื่อใช้ทดสอบความถูกต้อง ดังตารางที่ 1

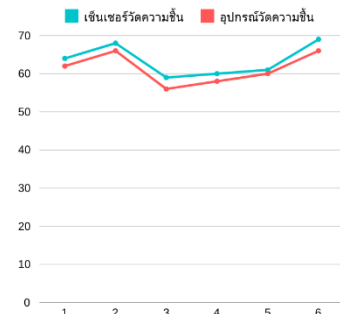


ภาพที่ 24 อุปกรณ์วัดค่าความชื้นและอุณหภูมิ

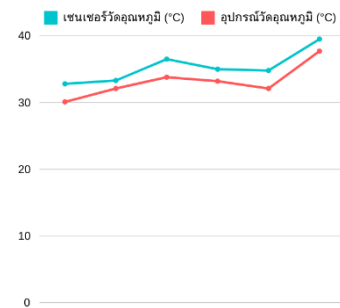
ตารางที่ 1 ตารางการทดลอง เซนเซอร์วัดความชื้น และ

อุณหภูมิ เทียบกับอุปกรณ์วัดความชื้นและอุณหภูมิ

| การทดลอง | เซนเซอร์วัดความชื้นและอุณหภูมิ | | อุปกรณ์วัดความชื้นและอุณหภูมิ | |
|----------|--------------------------------|---------------|-------------------------------|---------------|
| | 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 |



ภาพที่ 25 กราฟเซนเซอร์วัดความชื้น เทียบกับอุปกรณ์วัดความชื้น



ภาพที่ 26 กราฟเซนเซอร์วัดอุณหภูมิ เทียบกับอุปกรณ์วัดอุณหภูมิ

4.1.3 การวัดค่า PH ในถังเก็บน้ำ

เซ็นเซอร์วัดค่า pH ถูกติดตั้งไว้บนถังเก็บน้ำตามที่แสดงในภาพที่ 27 เพื่อวัดค่าความเป็นกรด-ด่าง (pH) ของน้ำในถังเก็บน้ำ เซ็นเซอร์นี้จะส่งสัญญาณไฟฟ้าไปยัง ESP32 ตัวที่สอง ซึ่งจะทำหน้าที่ส่งค่าพีเอช (pH) ของน้ำในถังไปอัปเดตใน Firebase



ภาพที่ 27 เซ็นเซอร์วัดค่า pH

ผลการทดลองค่าอุปกรณ์วัดค่า pH อ่านค่าได้ เพื่อใช้ทดสอบความถูกต้อง ดังตารางที่ 2

```
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
```

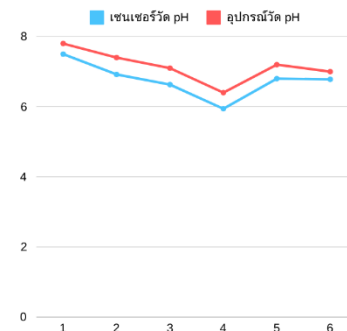
ภาพที่ 28 ค่าที่อุปกรณ์วัดค่า pH อ่านค่าได้



ภาพที่ 29 อุปกรณ์วัดค่า pH

ตารางที่ 2 ตารางการทดลอง เซ็นเซอร์วัด pH เทียบกับอุปกรณ์วัด pH

| ตัวอย่างของเหลว | ค่า pH ที่วัดจากเซ็นเซอร์ | ค่า pH ที่อ่านได้จากอุปกรณ์วัดค่า pH ในของเหลว |
|-----------------|---------------------------|--|
| 1 | 7.50 | 7.8 |
| 2 | 6.92 | 7.4 |
| 3 | 6.63 | 7.1 |
| 4 | 5.94 | 6.4 |
| 5 | 6.80 | 7.2 |
| 6 | 6.78 | 7 |



ภาพที่ 30 กราฟเซ็นเซอร์วัด pH เทียบกับอุปกรณ์วัด pH

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 ดังภาพที่ 31



ภาพที่ 31 รีเลย์ ที่ใช้ควบคุมอุปกรณ์

ทดลองสั่งเปิด-ปิด พัดลม และ หลอดไฟ ดังภาพที่ 31



ภาพที่ 32 พัดลม และ หลอดไฟ

ทดลองสั่งเปิด-ปิด ปั้มน้ำต่าง ๆ ดังภาพที่ 32



ภาพที่ 33 ปั้มน้ำต่าง ๆ

ตารางที่ 3 ผลการทดลอง เซนเซอร์วัด pH ในการเติมสาร

ลด-เพิ่มค่า pH ปริมาณ 250 ml ลงในถัง 5000 ml

| การทดลอง(ครั้ง) | ชนิดสารที่เติม (pH+,pH-) | ค่า pH ที่วัดจาก เซ็นเซอร์ |
|-----------------|-----------------------------|-------------------------------|
| 1 | pH+ | 6.52 |
| 2 | pH+ | 7.10 |
| 3 | pH- | 6.63 |
| 4 | pH- | 6.12 |
| 5 | pH+ | 6.80 |
| 6 | pH- | 6.22 |

4.1.5 สวิตช์รีเซตการเชื่อมต่อ WiFi

ผู้ใช้งานสามารถเปลี่ยนการเชื่อมต่อ WiFi ได้โดยการกดค้างสวิตช์สีแดงตามที่แสดงในภาพที่ 34 เป็นระยะเวลา 5 วินาที การกดสวิตช์นี้จะทำให้ระบบเริ่มต้นการค้นหาสัญญาณ WiFi ใหม่และเชื่อมต่อกับเครือข่ายที่ต้องการ



ภาพที่ 34 สวิตช์รีเซตการเชื่อมต่อ WiFi

4.1.6 สวิตช์เปิด-ปิดการทำงานของ ไฟในวงจร และรีเลย์

ผู้ใช้งานสามารถเปิด-ปิดสวิตช์ 2 ตัวตามที่แสดงในภาพที่ 35 โดยการหมุนสวิตช์เพื่อควบคุมการจ่ายไฟให้กับ รีเลย์ และ

ไฟในวงจร ซึ่งการกระทำนี้จะตัดการจ่ายไฟหรือเปิดการจ่ายไฟให้กับอุปกรณ์ดังกล่าวตามต้องการ



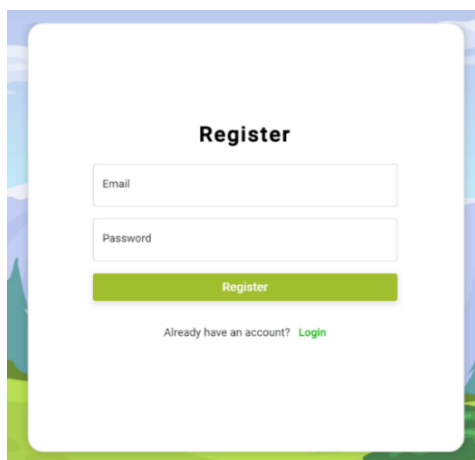
ภาพที่ 35 สวิตช์เปิด-ปิดการทำงานของ ไฟในวงจร และ รีเลย์

4.2 การทำงานของเว็บแอปพลิเคชัน

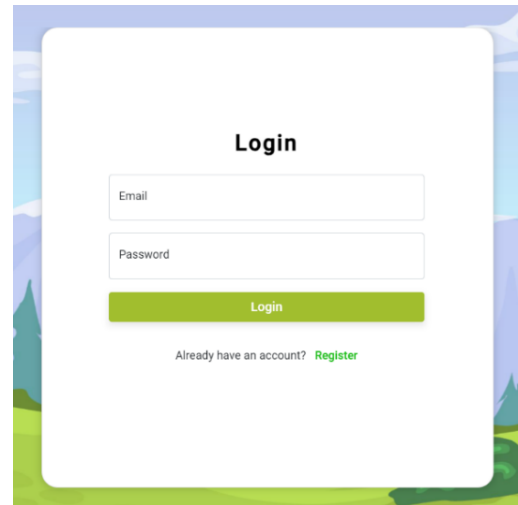
การทำงานของเว็บแอปพลิเคชันสามารถแบ่งการทำงานได้ ดังนี้

4.2.1 หน้าเข้าสู่ระบบและสมัครสมาชิก

ก่อนเริ่มต้นเข้าใช้งานเว็บแอปพลิเคชัน ผู้ใช้งานจะต้องสมัครสมาชิกเพื่อเข้าใช้งานดังภาพที่ 36 เมื่อสมัครสมาชิกเรียบร้อยแล้ว จะสามารถเข้าสู่ระบบได้ ดังภาพที่ 37



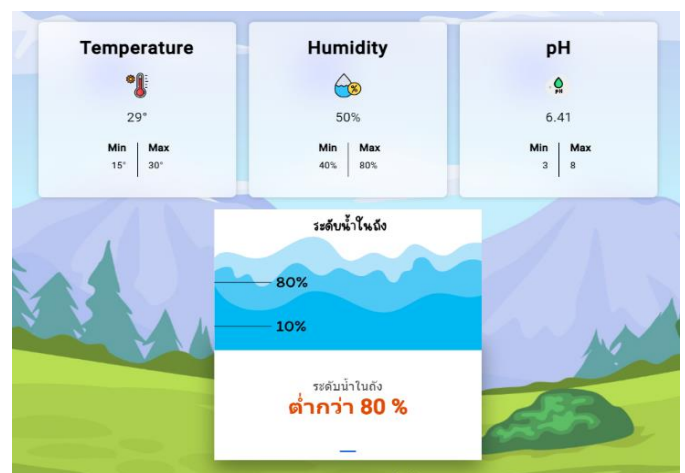
ภาพที่ 36 หน้าลงทะเบียน



ภาพที่ 37 หน้าเข้าสู่ระบบ

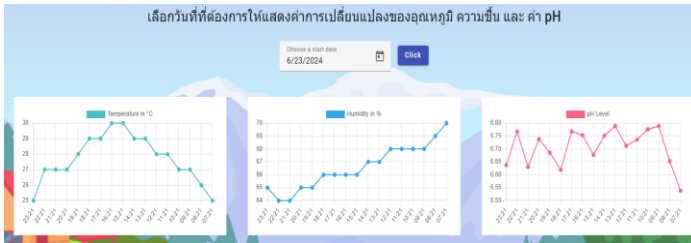
4.2.2 การแสดงผลข้อมูลต่าง ๆ ภายในโรงเรือน

หลังจากที่ผู้ใช้งานเข้าสู่ระบบเรียบร้อยแล้ว จะถูกนำมายังหน้า Home โดยหน้านี้จะเป็นการแสดงผลข้อมูลต่าง ๆ ได้แก่ อุณหภูมิ ความชื้น ค่า pH และระดับน้ำถึง ดังภาพที่ 38 รวมถึงจะมี QR Code ให้เข้าร่วมกลุ่มไลน์เพื่อรับการแจ้งเตือนการทำงานของอุปกรณ์ต่าง ๆ ในโรงเรือน

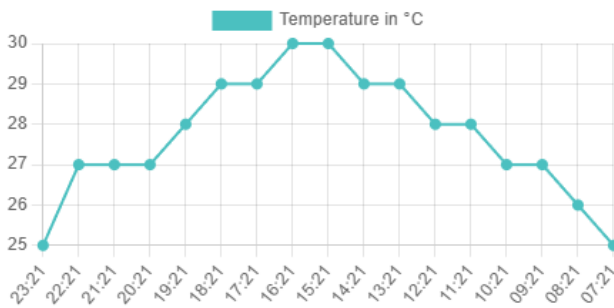


ภาพที่ 38 หน้าการแสดงผลข้อมูลต่าง ๆ ภายในโรงเรือน

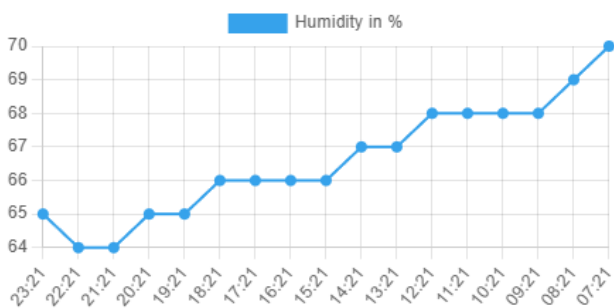
นอกจากนี้มีการแสดงผลของกราฟที่แสดงให้เห็นถึงการเปลี่ยนแปลงของอุณหภูมิ ความชื้น และค่า pH ทุก ๆ 1 ชั่วโมง โดยผู้ใช้งานสามารถเลือกวันที่ที่ต้องการดูกราฟได้ ดังภาพที่ 39



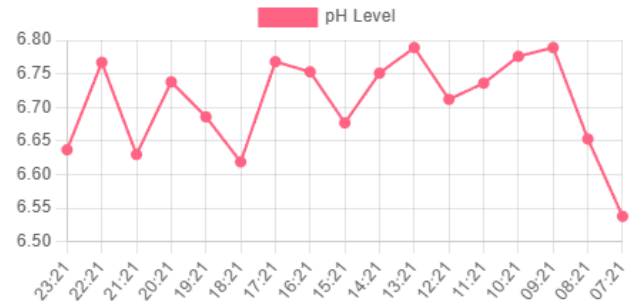
ภาพที่ 39 การแสดงผลข้อมูลการเปลี่ยนแปลงของอุณหภูมิ ความชื้น และค่า pH



ภาพที่ 40 กราฟแสดงผลการเปลี่ยนแปลงของอุณหภูมิ



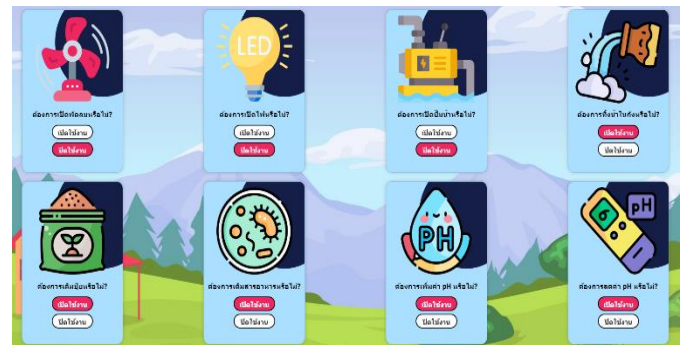
ภาพที่ 41 กราฟแสดงผลการเปลี่ยนแปลงของความชื้น



ภาพที่ 42 กราฟแสดงผลการเปลี่ยนแปลงของค่า pH

4.2.3 การเปิดปิดการใช้งานอุปกรณ์

ผู้ใช้งานสามารถกดเปิดปิดอุปกรณ์ต่าง ๆ ผ่านทางเว็บแอปพลิเคชันนี้ได้ ดังภาพที่ 43



ภาพที่ 43 หน้าเปิดปิดการใช้งานอุปกรณ์

4.2.4 การเติมสารต่าง ๆ ตามปริมาณ

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



ภาพที่ 44 หน้าเติมสารต่าง ๆ ตามปริมาณ

4.2.5 การตั้งค่าเวลาในการเปิดใช้งานอุปกรณ์

ในหน้าตั้งค่าเวลาดังภาพที่ 45 นี้ ไว้สำหรับผู้ใช้งานที่ต้องการตั้งค่าเวลาในการเปิดใช้งานอุปกรณ์ล่วงหน้าหรือต้องการให้ทำซ้ำในช่วงเวลานี้เป็นประจำ โดยผู้ใช้งานสามารถยกเลิกการตั้งค่าเวลานี้ได้



ภาพที่ 45 หน้าตั้งค่าเวลาในการเปิดใช้งานอุปกรณ์

4.2.6 การใช้งานสปริงเกอร์

ในหน้านี้ ดังภาพที่ 46 และ 47 จะเป็นการสั่งการทำงานของสปริงเกอร์ที่พ่นน้ำและสารอาหาร ซึ่งจะเป็นการรวมการทำงานตั้งแต่ข้อที่ 4.2.3 - 4.2.5 คือ

- เปิด-ปิดสปริงเกอร์ได้ตามที่ต้องการ
- เปิดสปริงเกอร์ได้ตามปริมาณที่ต้องการเปิด
- กำหนดเวลาในการเปิดล่วงหน้าหรือเปิดเป็นประจำ



ภาพที่ 46 หน้าการใช้งานสปริงเกอร์



ภาพที่ 47 หน้าการใช้งานสปริงเกอร์

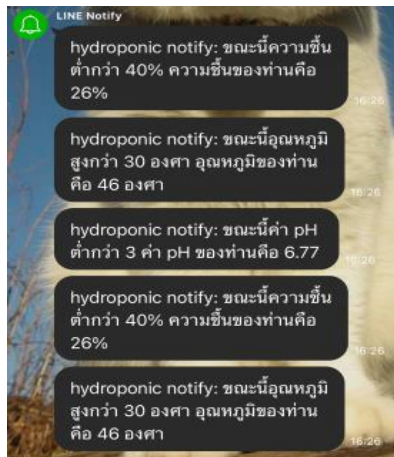
4.2.3 การแจ้งเตือน

เมื่อผู้ใช้งานเข้าร่วมกลุ่มไลน์ดังภาพที่ 48 จะมีการแจ้งเตือนจากระบบ 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



ภาพที่ 48 QR Code ของไลน์เพื่อรับการแจ้งเตือน



ภาพที่ 49 ตัวอย่างการแจ้งเตือน

5. สรุปผล

โครงการปริญญานิพนธ์ฉบับนี้ได้นำเสนอการพัฒนาเว็บแอปพลิเคชันร่วมกับการทำฟาร์มไฮโดรโปนิคส์ โดยการสร้างแบบจำลองฟาร์มไฮโดรโปนิคส์ที่มีบอร์ด ESP32 ช่วยในการรับค่าและควบคุมการทำงานอุปกรณ์ต่างๆ ภายในฟาร์ม ได้แก่ ไฟ พัดลม ปั๊ม สปริงเกอร์ เซ็นเซอร์อุณหภูมิ-ความชื้น และเซ็นเซอร์วัดความเป็นกรด-เบส รวมถึงมีการพัฒนาเว็บแอปพลิเคชันด้วย Angular และ Node.js เพื่อใช้ในการแสดงผลต่างๆ และให้ผู้ใช้งานสามารถสั่งการทำงานของอุปกรณ์ภายในฟาร์ม นอกจากนี้มีการใช้งานระบบฐานข้อมูลแบบเรียลไทม์ เพื่อให้หน้าเว็บและ ESP32 สามารถดึงค่าจากฐานข้อมูลไปใช้แสดงผลและประมวลผลต่อ รวมถึงสามารถแจ้งเตือนเมื่อพบว่าอุณหภูมิ ความชื้น ความเป็นกรดเบส มีความผิดปกติและแจ้งเตือนผ่าน Line Notify เมื่อถึงเวลาการเปิดใช้งานอุปกรณ์ต่างๆ ภายในฟาร์ม ที่ผู้ใช้งานตั้งค่าไว้จากการทดสอบการใช้เว็บแอปพลิเคชันร่วมกับฟาร์มไฮโดรโปนิคส์พบว่าสามารถควบคุมอุปกรณ์ต่างๆ ภายในฟาร์มไฮโดรโปนิคส์ได้ ซึ่งช่วยให้

ประหยัดเวลาและช่วยในการอำนวยความสะดวกให้กับผู้ใช้งานในการดูแลฟาร์มไฮโดรโปนิคส์

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] เซนเซอร์ระดับน้ำ 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 เจอรัมย์

อีเมล : s6303051623161@email.kmutnb.ac.th

ประวัติการศึกษา

พ.ศ. 2563 ระดับมัธยมศึกษาตอนปลาย แผนการเรียนวิทยาศาสตร์-คณิตศาสตร์
โรงเรียนเทพศิรินทร์ นนทบุรี

ปัจจุบัน เทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ แขนงวิชาคอมพิวเตอร์
วิทยาลัยเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ