

ระบบให้น้ำอัตโนมัติสำหรับสวนทุเรียน

ธนกร สุขแสงี่ยม¹ อธิปไตย คำเคน² และ พิสิทธิ วิสุทธิเมธีกร³

บทคัดย่อ

การให้น้ำสวนทุเรียนมีการให้น้ำแบบอัตโนมัติ และแบบรดน้ำปกติ เมื่อผู้ใช้งานทำการรดน้ำต้องกะระยะเวลาในการรดน้ำในแต่ละครั้งซึ่งแต่ละครั้งในการรดน้ำ จะไม่ได้ควบคุมปริมาณการให้น้ำและความชื้นในดินให้ดี ดังนั้นถ้ามีระบบให้น้ำอัตโนมัติสำหรับสวนทุเรียน จะช่วยให้ผู้ใช้งานสามารถควบคุมปริมาณการรดน้ำต้นทุเรียนตามค่าความชื้นในดินได้ เป็นระบบจัดการดูแลสวนที่เข้าถึงได้ยากสำหรับอินเทอร์เน็ต ทำให้สามารถตั้งค่าการรดน้ำในดินได้แบบอัตโนมัติ แล้วยังสามารถควบคุมการทำงานของระบบนี้ได้จากระยะไกลด้วยเทคโนโลยีคลาวด์ สร้างความสะดวกสบาย และง่ายต่อการดูแล

โครงงานปริญญานิพนธ์นี้นำเสนอการออกแบบและพัฒนาระบบให้น้ำอัตโนมัติสำหรับสวนทุเรียนผ่านแอปพลิเคชันเพื่อควบคุมการให้น้ำตามความชื้นในดินในแต่ละจุดที่ติดตั้งชุดควบคุมการให้น้ำ โดยนำอินเทอร์เน็ตกับทุกสิ่งมาใช้งานร่วมกับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ โดยการใช้โหนดเอ็มชียู เป็นตัวควบคุมการทำงานของระบบ เขียนคำสั่งผ่านซอฟต์แวร์ที่มีชื่อว่า อาร์ดูโอโน ไอเอสอี ที่เป็นโอเพ่นซอร์ส โดยควบคุมการทำงานของเซ็นเซอร์วัดค่าความชื้นในดินและโซลินอยวาล์ว สามารถบันทึกค่าความชื้นที่ได้รับจากเซ็นเซอร์และควบคุมการทำงานของระบบนี้จากระยะไกลได้ผ่านแอปพลิเคชัน เป็นระบบที่นำเสนอพัฒนาโดย Dart และ Flutter มีฟรีเบส ถูกใช้เป็นฐานข้อมูลระบบเพื่อเก็บข้อมูลการเพิ่มชุดควบคุมและเป็นตัวกลางในการรับหรือส่งข้อมูลระหว่างแอปพลิเคชัน ไปยังคอนโวลเลอร์ เป็นต้น

จากการทดสอบระบบให้น้ำอัตโนมัติสำหรับสวนทุเรียน พบว่าระบบสามารถทำงานได้ตามวัตถุประสงค์ และขอบเขตของปริญญานิพนธ์

คำสำคัญ : โหนดเอ็มชียู , อินเทอร์เน็ตกับทุกสิ่ง , คลาวด์ , ฟรีเบส

Automatic Watering System for Durian Garden

Thanakon Sugsangaim¹, Atipatai Khamkhen² and Pisit Wisutmetheekorn³

Abstract

Watering the durian garden has automatic watering, and normal watering. When users water, they must estimate the duration of watering each time, each time of watering. Will not control the amount of water and soil moisture well. Therefore, if there is an automatic watering system for the durian orchard. It allows the user to control the amount of watering the durian tree according to the soil moisture value. It is a garden management system that is difficult to access on the internet. Makes it possible to set automatic watering in the soil. And can also Remotely control this system using LoRa technology. create comfort and easy to care for

This thesis project presents the design and development of an automatic watering system for durian orchards through an application to control watering according to soil moisture at each point where the watering control unit is installed. By bringing the Internet and everything to use with various electronic devices. By using the MCU node. It controls the operation of the system. Write commands through the open source Arduino IDE software. By controlling the operation of the soil moisture sensor and solenoid valve. The humidity values received from the sensors can be recorded and the operation of this system can be controlled remotely through the application. It is a proposed system developed by Dart and Flutter. Firebase is used as the system database. To store additional control unit information and act as a medium for receiving or sending data between applications to the controller, etc.

From testing of the automatic watering system for durian orchards. It was found that the system was able to work according to the objectives and scope of the thesis.

Keywords : Node MCU, Internet of Things, LoRa, Firebase

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

* Corresponding author, E-mail: s6403052412146@email.kmutnb.ac.th

1. บทนำ

ประเทศไทยเป็นประเทศเกษตรกรรม ประชากรส่วนใหญ่ประกอบอาชีพเกษตรกร ซึ่งทุเรียนพันธุ์หมอนทองก็เป็นไม้ผลชนิดหนึ่งทางการเกษตร ผลลัพธ์ของไม้ผลดีหรือไม่ดีนั้นขึ้นอยู่กับ หลายๆปัจจัยประกอบด้วย ความสมบูรณ์ของดิน แร่ธาตุ และสารอาหารที่พืชต้องการในการเจริญเติบโต สภาพภูมิอากาศ แสงสว่าง และแหล่งน้ำที่อุดมสมบูรณ์

ในปัจจุบันได้มีการนำเทคโนโลยีคลาวด์ ซึ่งเป็นโครงข่ายสื่อสารไร้สาย กินพลังงานต่ำ มีพื้นที่ครอบคลุมเป็นวงกว้าง เหมาะกับการใช้ร่วมกับเซ็นเซอร์และไมโครคอนโทรลเลอร์ผ่านการส่งสัญญาณคลื่นวิทยุ ที่ต้องมีการรับคำสั่งและส่งค่าต่างๆ โดยสามารถประยุกต์ใช้งานได้หลากหลายรูปแบบ และการให้น้ำสวนทุเรียนส่วนใหญ่เป็นแบบเจ้าของสวนเดินเปิดสปริงเกอร์ จับเวลา ประมาณ 20-30 นาที ต่อต้น แล้วปิด

ด้วยเหตุผลข้างต้นที่กล่าวมาผู้จัดทำ ได้จัดสร้างระบบให้น้ำอัตโนมัติสำหรับสวนทุเรียน เพื่อที่จะนำเทคโนโลยีเข้ามาแก้ปัญหา Internet เข้าไม่ถึงสวนทุเรียนเข้ามาช่วยในการติดต่อสื่อสารระหว่างอุปกรณ์ เช่น การวัดค่าความชื้นในดิน มีหน่วยเป็น % RH และการควบคุมเปิด-ปิดอุปกรณ์แบบ IOT และ ระบบอัตโนมัติ นำเข้ามาประยุกต์ใช้เพื่อเป็นระบบที่คอยอำนวยความสะดวก และแก้ปัญหาให้กับเกษตรกรมากขึ้น

2. ทฤษฎีที่สำคัญ

2.1 Firebase

Firebase [1] เป็นหนึ่งในผลิตภัณฑ์ของ Google โดยไฟร์เบส คือ แพลตฟอร์มที่รวบรวมเครื่องมือต่าง ๆ เช่น การจัดการในส่วนของ Realtime Database เป็นบริการฐานข้อมูลแบบ NoSQL และไฟร์เบสโฮสติ้งเป็นบริการจัดการเว็บไซต์และเนื้อหาเว็บไซต์ ซึ่งช่วยให้นักพัฒนาสามารถโฮสต์และเผยแพร่เว็บไซต์หรือแอปพลิเคชันของตนได้อย่างง่ายดายและรวดเร็ว

2.2 Flutter

Flutter [2] เป็น Framework ที่ใช้สร้าง UI (User Interface) สำหรับโมบายแอปพลิเคชันที่สามารถทำงานข้ามแพลตฟอร์มได้ทั้ง iOS และ Android ในเวลาเดียวกัน โดยภาษาที่ใช้ใน Flutter นั้นก็คือภาษา Dart ถูกพัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อให้เป็นภาษาสำหรับการพัฒนาโปรแกรมประยุกต์บนเว็บ ซึ่งเป็นภาษาที่ถูกพัฒนาขึ้นมาใหม่โดยทีมพัฒนาจาก Google

2.3 Node MCU

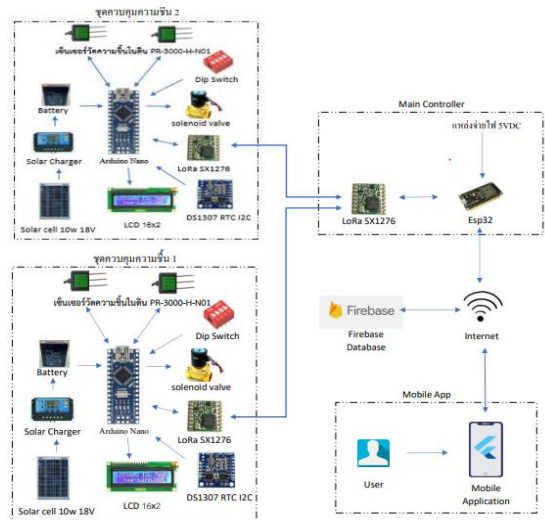
Node MCU [3] เป็นไมโครคอนโทรลเลอร์ที่มาพร้อมกับ Wi-Fi และ Bluetooth ในตัวเดียวกัน และมีความสามารถในการประมวลผลสูง ซึ่งทำให้มันเป็นที่นิยมในการทำโปรเจก IoT หรือโปรเจกที่ต้องการการเชื่อมต่อไร้สาย ในการทำโปรเจกนี้จะต้องใช้บอร์ด ESP32 ในการส่งข้อมูลขึ้นไฟร์เบส รวมถึงดึงข้อมูลมาใช้ในกระบวนการทำงานของระบบอีกด้วย

2.4 LoRa

LoRa [4] เป็นเทคโนโลยีการเชื่อมต่อไร้สายเหมาะสำหรับงาน Internet of Things (IoT) มีจุดเด่นในเรื่องของระยะทางในการสื่อสาร สามารถรับ-ส่งข้อมูลได้ในระยะไกล 5-15 กิโลเมตร และเนื่องจากพลังงานที่ใช้ในช่วงรับ-ส่งข้อมูลค่อนข้างต่ำ อีกทั้งยังสามารถกำหนดให้อุปกรณ์เข้าสู่โหมดประหยัดพลังงานในกรณีที่ไม่ได้ส่งข้อมูล จึงทำให้ถือว่าเป็นเทคโนโลยีที่ช่วยประหยัดพลังงานและค่าใช้จ่ายในการใช้งาน

2.5 Soil Moisture Sensor

Soil Moisture Sensor [5] เป็นตัวเซ็นเซอร์ตรวจวัดความชื้นในดินแบบใช้งานกลางแจ้ง โดยตัวแกนเซ็นเซอร์สามารถจมอยู่ในดินได้หรือฝังอยู่ในดิน 20 เซนติเมตรช่วงในการวัดความชื้นในดินอยู่ในช่วง 0-100% ซึ่งมีค่าเอาต์พุตแบบอนาล็อกเป็นแรงดันแบบโวลต์เทียบกับแรงดันแหล่งจ่าย 3.6 - 30 VDC ค่าความผิดพลาดและแม่นยำ $\pm 3\%$ ในการต่อใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ จะใช้ RS485 เป็นตัวกลางในการติดต่อสื่อสารกับเซ็นเซอร์เหมาะสำหรับการทดลองวิจัยเพื่อปลูกพืชในแบบโรงเรือนและระบบสมาร์ทฟาร์มต่างๆ



รูปที่ 1 โครงสร้างระบบให้น้ำอัตโนมัติสำหรับสวนทุเรียน

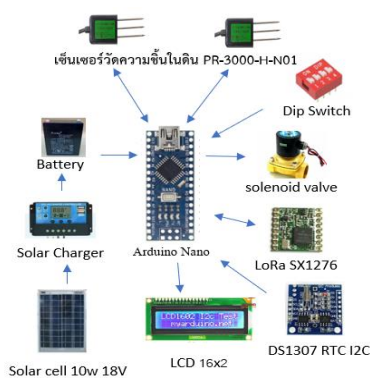
3. วิธีดำเนินการ

จากโครงสร้างระบบในรูปที่ 1 ประกอบด้วย ชุดควบคุมความชื้น, ชุดควบคุมหลัก (Main controller) และโมบายแอปพลิเคชัน ในชุดควบคุมความชื้น 1 ชุด จะมีเซ็นเซอร์วัดความชื้น 2 ตัวและมี 1 เอาต์พุต เพื่อควบคุมวาล์วเปิด-ปิดการให้น้ำต้นทุเรียนที่ชุดควบคุมความชื้นจะมีชุดรับส่งสัญญาณผ่านคลื่นวิทยุด้วยเทคโนโลยี LoRa เพื่อรับส่งข้อมูลกับชุดควบคุมหลัก โดยที่ชุดควบคุมหลักจะทำหน้าที่นำข้อมูลจากชุดควบคุมความชื้นบันทึกลงบนฐานข้อมูลไฟร์เบส และนำค่าคำสั่งที่ผู้ใช้งานสั่งผ่านแอปพลิเคชันและบันทึกลงบนฐานข้อมูล ในส่วนของระบบจะแบ่งออกเป็น 2 ส่วนคือระบบแบบ Auto ผู้ใช้งานจะเป็นคนตั้งค่าเวลาการเปิด

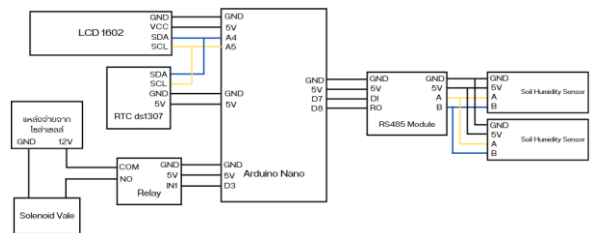
น้ำและกำหนดค่าความชื้นในดินเป็นการปิดน้ำและระบบแบบ Manual ผู้ใช้งานจะเป็นคนสั่งการใช้งานการเปิดปิดวาล์วผ่านแอปพลิเคชัน โดยแต่ละส่วนมีรายละเอียดดังต่อไปนี้

3.1 ชุดควบคุมความชื้น

ชุดควบคุมความชื้นมีส่วนประกอบดังรูปที่ 2 ใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano เป็นตัวควบคุม มีเซ็นเซอร์วัดความชื้นในดิน PR-3000-H จำนวน 2 ตัวและใช้วงจรขับรีเลย์เพื่อสั่งงานให้โซลินอยด์วาล์วเปิด-ปิดน้ำ มีจอแสดงผล LCD เพื่อแสดงสถานะการทำงานของเครื่อง และมีโมดูล DS1307 เป็นส่วนของการเก็บค่าเวลาจริง และมีโมดูล LoRa SX1276 สำหรับการรับส่งข้อมูลผ่านคลื่นวิทยุกับชุดควบคุมหลัก นอกจากนั้นมีการใช้เซลล์แสงอาทิตย์ขนาด 10 W และชุดชาร์ตแบตเตอรี่ (Solar Charger) เพื่อชาร์ตแบตเตอรี่ที่เป็นไฟเลี้ยงระบบของชุดควบคุมความชื้น



รูปที่ 2 อุปกรณ์ภายในชุดควบคุมความชื้น



รูปที่ 3 วงจรการเชื่อมต่อชุดควบคุมความชื้น

การเชื่อมต่อเซ็นเซอร์, โซลินอยด์วาล์ว, จอแสดงผล LCD, และโมดูล RTC จากรูปที่ 3 มีอุปกรณ์และหน้าที่การทำงานดังนี้

RS485 เป็น physical layer ในการสื่อสารระหว่างบอร์ดไมโครคอนโทรลเลอร์และเซ็นเซอร์วัดความชื้นในดิน โดยมีการต่อเซ็นเซอร์ 2 ตัว ต่อขนานกับ RS485 module และเชื่อมต่อ RS485 module เข้ากับบอร์ดไมโครคอนโทรลเลอร์ การสื่อสารผ่าน RS485 ในโปรเจกต์นี้ใช้โปรโตคอล Modbus RTU เป็นตัวกลางในการสื่อสารระหว่างบอร์ดไมโครคอนโทรลเลอร์และเซ็นเซอร์ โดย Modbus RTU เป็นโปรโตคอลที่มักใช้ในงานควบคุมและติดตามข้อมูลในระบบอุตสาหกรรม โดยสามารถระบุที่อยู่ (address) ของแต่ละอุปกรณ์เพื่อสื่อสารได้อย่างถูกต้องและเป็นระบบ การต่อเซ็นเซอร์เป็นตัวขนานกับ RS485 module จะต้องระบุที่อยู่ของแต่ละเซ็นเซอร์เพื่อให้บอร์ดไมโครคอนโทรลเลอร์สามารถระบุและสื่อสารกับแต่ละเซ็นเซอร์ได้อย่างถูกต้อง

เซ็นเซอร์วัดความชื้นในดิน ที่ใช้จะเป็นรุ่น PR-3000-H-N01 ซึ่งสามารถวัดค่าความชื้นในดินได้อย่าง

เดียว มีการเปลี่ยนแอดเดรสเซ็นเซอร์ค่าจากโรงงานผ่านโปรแกรม Modbus pull ในการใช้งานจะใช้เซ็นเซอร์วัดความชื้นในดิน 2 ตัวใน 1 ชุดควบคุมความชื้นเพื่อให้สามารถวัดค่าความชื้นได้หลายจุด การอ่านข้อมูลของเซ็นเซอร์โดยรหัสคำสั่งนี้เป็นคำสั่ง Modbus RTU ที่ใช้สื่อสารกับอุปกรณ์ที่ติดต่อผ่าน RS485 จะเริ่มการเช็คของค่าแอดเดรสเซ็นเซอร์, Function code, ตำแหน่งเริ่มต้นของ Register, จำนวน Register ที่ต้องการอ่าน และ Checksum สำหรับความถูกต้องของข้อมูลที่ส่งโดยค่าที่ของเซ็นเซอร์ 2 ตัวจะมีการเรียกใช้งานดังนี้

ตัวที่ 1 0x01, 0x03, 0x00, 0x00, 0x00, 0x01, 0x84, 0x0A

ตัวที่ 2 0x02, 0x03, 0x00, 0x00, 0x00, 0x01, 0x84, 0x39

ค่าของแอดเดรสเซ็นเซอร์ ปกตินั้นจะเป็น 0x01 แต่เมื่อมีการใช้งาน 2 ตัวจึงต้องเปลี่ยนค่าแอดเดรสผ่าน Modbus ถ้าค่าที่อ่านมาได้จะมีการคืนข้อมูลความยาว 5 bytes ซึ่งประกอบด้วย Byte ต่าง ๆ ดังนี้

Byte 1: แอดเดรสของอุปกรณ์ที่ตอบกลับ

Byte 2: Function code (ควรเป็น 0x03)

Byte 3: จำนวน bytes ข้อมูลที่ตอบกลับ

Byte 4-5: ค่าข้อมูลที่ถูกอ่านจาก Register

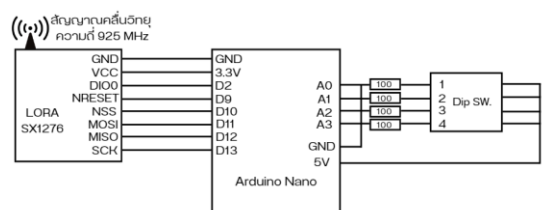
การคำนวณค่าของความชื้น จาก Byte 4-5 จะแปลงเป็นสูตรคำนวณได้คือ $(\text{Byte } 3 * 256 + \text{Byte } 4) / 10.0$ ซึ่งนำค่าที่ได้มาแบ่งด้วย 10.0 เนื่องจากหน่วยของความชื้นอยู่ในรูปทศนิยม

รีเลย์จะใช้เป็นตัว 1 ช่อง 5V การต่อจะต่อเข้ากับโซลินอยด์วาล์ว และรีเลย์ต่อเข้ากับบอร์ดคอนโทรลเลอร์เพื่อใช้ในการควบคุมการเปิด-ปิดวาล์วน้ำ จะใช้งานในการควบคุมโซลินอยด์วาล์วในการควบคุมเปิดและปิดวาล์วน้ำของสวนทุเรียน ตัวที่ใช้งานจะทำงานก็ต่อเมื่อส่งลอจิก low จะเป็นการเปิดวาล์ว และส่ง high เพื่อปิดวาล์วน้ำ

โซลินอยด์วาล์ว ในตัวงานที่ใช้มีขนาด $\frac{3}{4}$ นิ้ว ใช้งานในการเปิด-ปิดการให้น้ำของสวนทุเรียนโดยตัวสั่งใช้งานจะเป็นรีเลย์ที่เป็นตัวควบคุม

จอ LCD 1602 การต่อใช้งานเป็นแบบ i2c ในการใช้งานเพื่อแสดงผลของค่าความชื้นจากเซ็นเซอร์ 2 ตัว , ระบบการทำงาน Auto หรือ Manual , ค่าสถานะวาล์วเปิด/ปิด และค่าเวลาของเครื่อง

RTC ds1307 การต่อใช้งานเป็นแบบ i2c ในการใช้งานเพื่อตั้งข้อมูลเวลาใช้เปิด/ปิด วาล์วในขณะที่เมื่อไม่มีการรับส่งข้อมูลของระบบผ่านการส่งจาก Main Controller



รูปที่ 4 วงจรการเชื่อมต่อชุดควบคุมความชื้นกับลอรา

การเชื่อมต่อกับ โมดูล LoRa และดิฟสวิตช์เพื่อกำหนดแอดเดรสของชุดควบคุมจากรูปที่ 4 มีการทำงานดังนี้ ชุดควบคุมความชื้นต่อใช้งานร่วมกับลอราโดยการกำหนดแอดเดรสของตัวเครื่องนั้นจะใช้ดิฟสวิตช์โดยจะกำหนดตั้งแต่ 0x01 – 0x0F โดยที่อยู่รับของ Main controller เป็น 0x00 การรับข้อมูลนั้นจะเริ่มจากการเชื่อมต่อลอรา สำเร็จจะมีการส่งข้อมูลเป็นระบบอย่าง Auto หรือ Manual และมีการเช็คค่าข้อมูลที่ส่งมานั้นเป็นข้อมูลที่จะส่งให้หมายเลขเครื่องนี้จริงไหม เมื่อเช็คค่าใช่ จะทำการปรับระบบเครื่องของชุดควบคุมค่าความชื้นให้เป็นโหมดนั้นที่รับมาและทำตามคำสั่งการเปิด-ปิด วาล์วน้ำตามที่รับมา

Controller Arduino nano เป็น บอร์ด ที่ ใช้ ในติดต่อสื่อสารที่เชื่อมต่อกับ ลอราโมดูล เพื่อใช้ส่งข้อมูลของค่าความชื้นในดินที่ได้รับจากเซ็นเซอร์วัดความชื้นในตัวบอร์ดนี้จะมีหน้าที่ทำงานอยู่ 2 ระบบ คือ Auto และ Manual ซึ่งคำสั่งของระบบนั้นจะขึ้นอยู่กับผู้ใช้งานส่งผ่านใช้งานจาก Mobile Application ส่งต่อมาให้ Main Controller และส่งต่อให้ตัวบอร์ด Controller Arduino nano หรือชุดควบคุมความชื้น รับคำสั่งการใช้งานของระบบ

ลอราโมดูล ที่ใช้จะเป็นตัว SX1276 การเชื่อมต่อกันกับตัว Controller Arduino nano การต่อจะเป็นในรูปแบบของ SPI สัญญาณในการติดต่อสื่อสารในคลื่นความถี่ที่ใช้จะใช้ 925Mhz. เนื่องจาก กสทช. อนุญาต

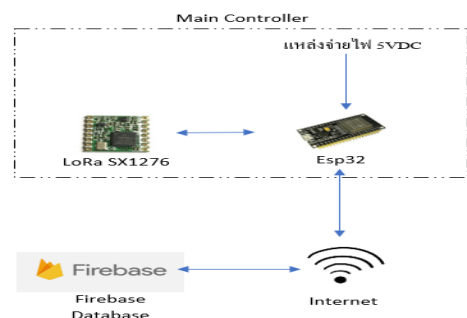
การใช้งานลอรา ในคลื่นความถี่ 920 – 925 MHz กำลังส่งสูงสุดไม่เกิน 4 วัตต์ได้แล้ว

ดิฟสวิตช์ ในรูปแบบการเชื่อมต่อจะใช้แบบ 4 สวิตช์ในการใช้งานจะเอาไปใช้เป็นตัวกำหนดค่าแอดเดรสเพื่อบอกที่อยู่ของตัวเครื่องควบคุมความชื้นกำหนดค่าได้ตั้งแต่ 0x01-0x0F

แหล่งจ่ายไฟของระบบแบตเตอรี่ที่มีขนาด 12 V ได้รับแหล่งจ่ายมาจากแผงโซล่าเซลล์ที่ต่อเข้ากับโซล่าชาร์ต โซล่าชาร์ต รุ่นที่มีขนาด 10A มีหน้าที่ในการรับพลังงานไฟฟ้าจากโซล่าเซลล์และจ่ายพลังงานให้กับอุปกรณ์อื่นๆภายในชุดควบคุมความชื้น แผงโซล่าเซลล์ขนาดแผงที่ใช้ 12V. 20W. 1.86A มีหน้าที่ในการแปลงพลังงานแสงอาทิตย์เป็นพลังงานไฟฟ้าโดยส่งผ่านโซล่าชาร์ตไปให้แบตเตอรี่

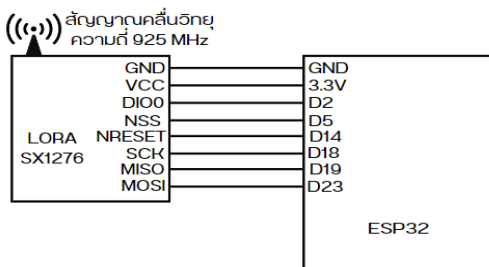
3.2 Main controller

ภายใน ตัว Main controller จะประกอบไปด้วย Node MCU Esp32 และ ลอราโมดูล ดังรูปที่ 5



รูปที่ 5 Main Controller

จากรูปที่ 5 เป็นส่วนประกอบอุปกรณ์ภายในชุด Main Controller ซึ่ง Esp32 รับแหล่งจ่ายไฟฟ้าที่ 5 VDC เพื่อใช้งาน มีหน้าที่ในการรับและส่งข้อมูลไปให้ลอรา เพื่อให้ตัวลอราติดต่อกับชุดควบคุมความชื้นต่อไป และรับส่งข้อมูลไปยังไฟร์เบส โดยผ่านอินเทอร์เน็ต



รูปที่ 6 วงจรการเชื่อมต่อ ESP32 กับลอรา

จากรูปที่ 6 เป็นการเชื่อมต่อวงจรระหว่างลอราและ Esp32 ในแต่ละขา และลอรา มีการเชื่อมต่อสัญญาณคลื่นวิทยุที่มีความถี่ 925 MHz อธิบายการทำงานของอุปกรณ์ดังนี้

Node MCU Esp32 เป็นบอร์ดที่เชื่อมต่อกับอินเทอร์เน็ต ใช้ในการส่งข้อมูลระบบควบคุมที่ได้รับจากผู้ใช้งานผ่าน Mobile App และอัปเดตข้อมูลที่ได้จาก Controller nano ไปเก็บบนไฟร์เบสอีกด้วย

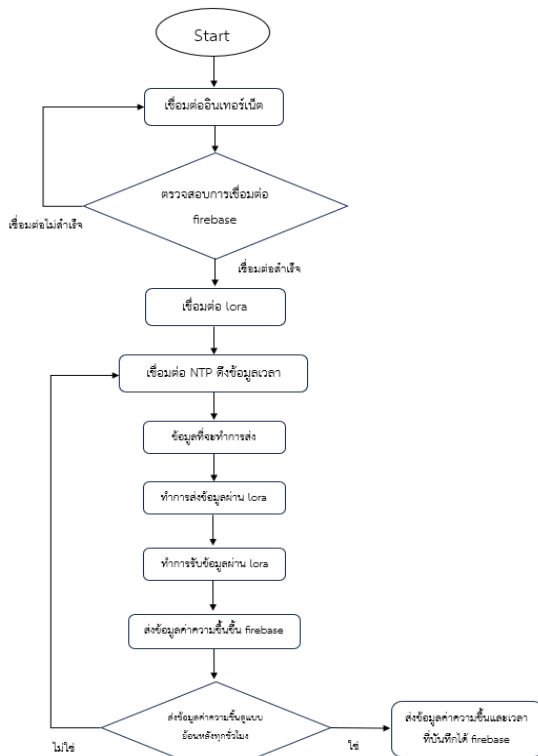
ลอราโมดูล ที่ใช้จะเป็นตัว SX1276 โดยใช้โปรโตคอลการสื่อสารแบบ SPI ลอราโมดูลนั้นเป็นสื่อกลางการสื่อสารระหว่าง Esp32 กับ Controller Arduino nano ให้สามารถส่งข้อมูลกันได้ การทำแบบนี้จะเป็นการทดแทนพื้นที่ในส่วนทุเรียนที่อินเทอร์เน็ตเข้า

ไม่ถึงในการติดต่อตั้งใช้งานระบบ จึงใช้ลอราโมดูลเป็นตัวสื่อสารแทน

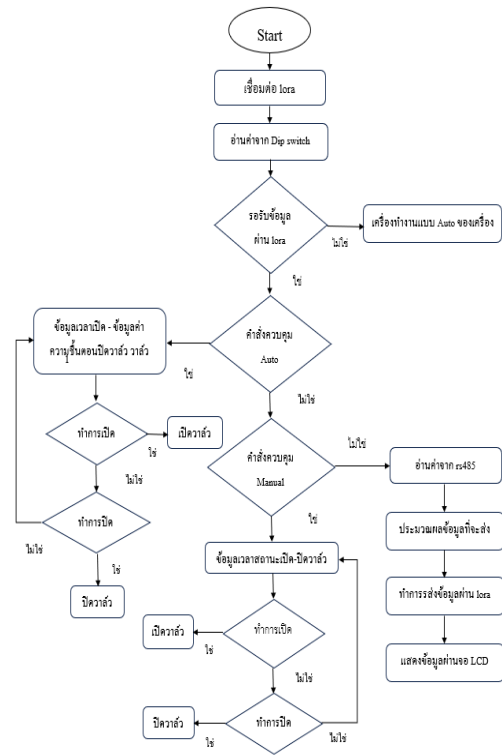
3.3 หลักการทำงานของชุดควบคุมและ Main Controller

หลักการทำงานในโปรเจกต์นี้จะมี Main controller ที่เป็นตัวตั้งการหรือการส่งข้อมูลต่างๆจากไฟร์เบสให้กับชุดควบคุมค่าความชื้น โดยผู้ใช้งานจะทำการตั้งการผ่านโมบายแอปพลิเคชัน โดยรายละเอียดของการทำงานแต่ละขั้นจะอธิบายได้แต่ละส่วนดังนี้

Main Controller จะมีบอร์ดไมโครคอนโทรลเลอร์ ESP32 เป็นตัวที่เชื่อมต่อกับไฟร์เบส เพื่อนำข้อมูลมาใช้ในการส่งข้อมูล การเชื่อมต่อไฟร์เบส มีการใส่ตัว HOST ที่เป็น URL ที่บอกถึงที่อยู่ของไฟร์เบส และ AUTH ที่เป็น Authentication Token ที่ใช้ในการรับรองตัวตนเมื่อทำการเชื่อมต่อกับไฟร์เบส เมื่อมีการเชื่อมต่อแล้วจะใช้ในส่วนของ ลอรา SX1276 เพื่อทำการส่งข้อมูลที่ได้จากไฟร์เบส และข้อมูลเวลา NTP เพื่อใช้ในการเปิดวาล์วน้ำของชุดควบคุมค่าความชื้น ในกรณีรับข้อมูลจะรับข้อมูลค่าความชื้นในดิน สถานะเชื่อมต่อของเซ็นเซอร์ สถานะวาล์วเปิด-ปิด ครบแล้ว ถ้าตัวนับเวลาเมื่อครบชั่วโมงจะทำการส่งข้อมูลค่าความชื้นล่าสุดที่ได้รับจากชุดควบคุมค่าความชื้นไปเก็บเป็นข้อมูลกราฟเพื่อดูย้อนหลังบน แอปพลิเคชัน โดยรายละเอียดการทำงานจะเป็นดังในรูปที่ 7



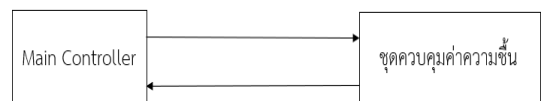
รูปที่ 7 ฟังก์การทำงานของ Main controller



รูปที่ 8 ฟังก์การทำงานของชุดควบคุมความชื้น

ชุดควบคุมค่าความชื้นเมื่อเริ่มทำงานจะทำการเชื่อมต่อลอราในการส่งข้อมูลและทำการอ่านค่าจากเซ็นเซอร์ที่วัดความชื้นในดินได้เก็บลงตัวแปรเก็บข้อมูลเพื่อนำส่งต่อไปให้กับ Main Controller โดยการส่งนั้นจะส่งได้ต่อเมื่อ Main Controller จะถามค่าความชื้นกลับมาหรือจะเป็นการส่งข้อมูลระบบของตัวเครื่องไม่ว่าจะเป็นระบบแบบ Auto ระบบแบบ Manual จากนั้นจึงจะสามารถส่งข้อมูลกลับไปหา Main controller ได้ โดยรายละเอียดการทำงานจะเป็นดังในรูปที่ 8

3.4 การรับส่งข้อมูล Main Controller และชุดควบคุมค่าความชื้น



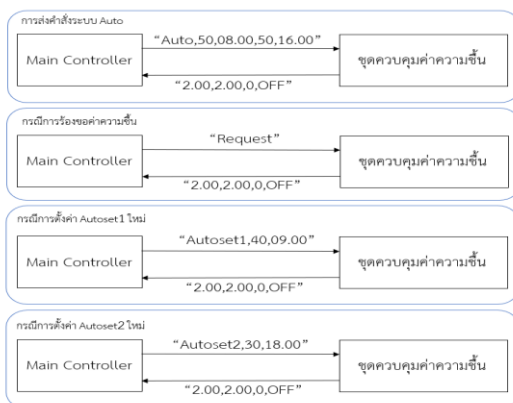
- Address (ค่าของที่อยู่ผู้ส่ง)
- Receiver Address (ค่าของที่อยู่ผู้รับ)
- Message (ข้อมูลหรือข้อความที่ส่ง)

รูปที่ 9 รูปแบบการส่ง-รับข้อมูลผ่านลอราโมดูล

จากรูปที่ 9 จะเป็นการอธิบายการรับ การส่งข้อมูลของลอรา การส่งข้อมูลนั้นจะมีการส่งแอดเดรสคือที่อยู่ผู้ส่ง Receiver Address คือที่อยู่ของผู้รับ และ Message คือข้อความหรือข้อมูลที่จะส่งให้ผู้รับ โดยแอดเดรสของ Main Controller จะ ถูกกำหนดให้อยู่ที่ 0x00 และแอดเดรสของชุดควบคุมความชื้นจะถูกกำหนดผ่านผู้ใช้ในการปรับดิฟฟิวติวในช่วง 0x01 - 0x0F

การสั่งงานชุดควบคุมค่าความชื้น จะมีลักษณะที่แบ่งได้ 2 รูปแบบคือ การสั่งใช้งานระบบแบบ Auto และการสั่งใช้งานระบบแบบ Manual ผู้ใช้สามารถเลือกใช้งานใหม่ตามความต้องการผ่าน Mobile Application ซึ่งคำสั่งจะถูกเก็บไว้ในโฟลเดอร์ไฟร์เบส จากนั้น Main Controller จะทำการเชื่อมต่อกับไฟร์เบสเพื่อส่งข้อมูลที่ได้นั้นไปสั่งใช้งานของชุดควบคุมความชื้น โดยมีรายละเอียดการสั่งใช้งานดังนี้

3.4.1 การสั่งใช้งานแบบ Auto



รูปที่ 10 รูปแบบการส่ง-รับข้อมูลแบบ Auto

จากรูปที่ 10 การส่งคำสั่งระบบ Auto นั้นเครื่องจะมีความทำงานคือเมื่อเปิดวาล์วและปิดวาล์ว ในการเปิดวาล์วนั้นจะทำงานก็ต่อเมื่อค่าเวลาที่สั่งเปิดวาล์วตรงกับค่าเวลาปัจจุบันและการปิดวาล์วจะปิดก็ต่อเมื่อค่าความชื้นที่วัดได้จากเซ็นเซอร์ตรงกับค่าความชื้นที่อ่านค่าได้จาก Rs485 โมดูล

การส่งคำสั่งระบบของ Main Controller จากข้อมูลที่ส่งอธิบายได้ดังนี้

- Auto ระบบการทำงานของเครื่อง
- 50 ค่าความชื้นที่ใช้ปิดโซลินอยด์วาล์วของ Autoset1
- 08.00 ค่าเวลาใช้ในการเปิดโซลินอยด์วาล์วของ Autoset1
- 50 ค่าความชื้นที่ใช้ปิดโซลินอยด์วาล์วของ Autoset2
- 16.00 ค่าเวลาใช้ในการเปิดโซลินอยด์วาล์วของ Autoset2

การส่งคำสั่งระบบของชุดควบคุมค่าความชื้นจากข้อมูลที่ส่งอธิบายได้ดังนี้

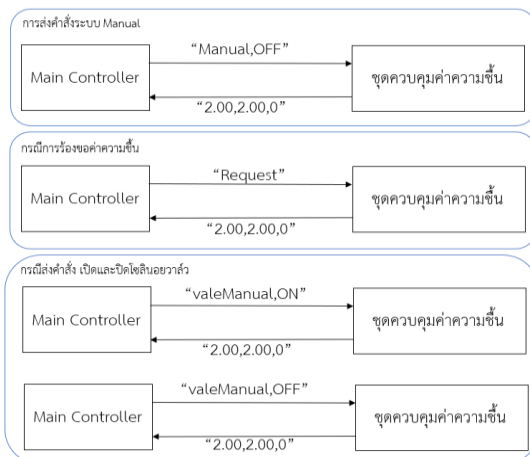
- 2.00 ค่าความชื้นของเซ็นเซอร์ตัวที่ 1
- 2.00 ค่าความชื้นของเซ็นเซอร์ตัวที่ 2
- 0 ค่าความผิดปกติของเซ็นเซอร์ทั้งสองตัว
- OFF สถานะโซลินอยด์วาล์วมีการเปิดหรือปิด

กรณีการร้องขอค่าความชื้น เมื่อไม่มีการเปลี่ยนแปลงการตั้งค่าใหม่ใน Autoset1 และ Autoset2 Main Controller จะส่งข้อความ Request เพื่อร้องขอค่า

ความขึ้น ข้อมูลความผิดปกติของเซ็นเซอร์ทั้งสองและสถานะโซลินอยด์ว่าลั่วส่วนชุดควบคุมค่าความขึ้นจะทำการตอบกลับค่าความขึ้นของเซ็นเซอร์ทั้งสองตัว ค่าความผิดปกติของเซ็นเซอร์ทั้งสองตัวและสถานะโซลินอยด์ว่าลั่ว

กรณีการตั้งค่า Autoset1 และ Autoset2 ใหม่ นั้น Main Controller จะส่งข้อความ Autoset1 หรือ Autoset2 เพื่อให้ค่าความขึ้นที่ใช้ปิดโซลินอยด์ว่าลั่วและค่าเวลาใช้ในการเปิดโซลินอยด์ว่าลั่ว แทนข้อมูลเก่าที่มีอยู่เพื่อใช้ในการเปิดหรือปิดโซลินอยด์ว่าลั่วนั่นเอง

3.4.2 การสั่งใช้งานแบบ Manual



รูปที่ 11 รูปแบบการส่ง-รับข้อมูลแบบ Manual

จากในรูปที่ 11 การส่งคำสั่งระบบ Manual จะทำงานเปิดปิดโซลินอยด์ว่าลั่วตามคำสั่งจากผู้ใช้งาน โดยผู้ใช้งานจะเป็นคนกำหนดการเปิดปิดโซลินอยด์

ว่าลั่วผ่านทาง Mobile Application ที่ผู้ใช้งานใช้งาน มา Main controller นั้นจะส่งคำสั่งระบบ Manual และสถานะการเปิดปิดโซลินอยด์ว่าลั่วมา จากนั้นชุดควบคุมจะตอบกลับด้วยค่าความขึ้นของเซ็นเซอร์ทั้งสองตัวและค่าความผิดปกติของเซ็นเซอร์ทั้งสอง

การส่งคำสั่งระบบของ Main Controller จากข้อมูลที่ส่งอธิบายได้ดังนี้

- Manual ระบบการทำงานของเครื่อง
- OFF สถานการณ์เปิดปิดโซลินอยด์ว่าลั่ว

การส่งคำสั่งระบบของชุดควบคุมค่าความขึ้นจากข้อมูลที่ส่งอธิบายได้ดังนี้

- 2.00 ค่าความขึ้นของเซ็นเซอร์ตัวที่ 1
- 2.00 ค่าความขึ้นของเซ็นเซอร์ตัวที่ 2
- 0 ค่าความผิดปกติของเซ็นเซอร์ทั้งสองตัว

กรณีการร้องขอค่าความขึ้น จะไม่เหมือนของระบบอัตโนมัติคือเมื่อไม่มีการเปลี่ยนแปลงการเปิดปิดว่าลั่ว Main controller จะส่งข้อความ Request เพื่อร้องขอค่าความขึ้นและข้อมูลความผิดปกติของเซ็นเซอร์ทั้งสองส่วนชุดควบคุมค่าความขึ้นจะทำการตอบกลับค่าความขึ้นของเซ็นเซอร์ทั้งสองตัวและความผิดปกติของเซ็นเซอร์ทั้งสองตัว

กรณีการส่งคำสั่ง เปิดปิดโซลินอยด์ว่าลั่วชุดควบคุมจะจำว่าระบบเครื่องตอนนี้เป็นการทำงานแบบ Manual อยู่ Main controller จึงส่งแค่ค่าสถานะว่าลั่วที่เปลี่ยนเหมือนในรูปที่ 11 ที่จะส่ง Vale Manual ตามด้วย ON หรือ OFF คือคำสั่งเปลี่ยนแปลงการเปิดหรือปิดว่าลั่ว

3.5 ฐานข้อมูลและการเก็บข้อมูลในไฟร์เบส

ในระบบนี้ใช้ Firebase Realtime Database ในการจัดการฐานข้อมูลซึ่งเป็นฐานข้อมูลแบบ NoSQL โดยแบ่งเป็น Collection ได้ 3 Collection ใหญ่ๆ ดังนี้

3.5.1 Controller Node

ใน Collection นี้ใช้เก็บข้อมูลต่างๆของชุดควบคุมความชื้นมี 11 Collection ดังนี้

- Humidity1 : เก็บข้อมูลค่าความชื้นจากเซ็นเซอร์ 1 จากชุดควบคุมความชื้น
- Humidity2 : เก็บข้อมูลค่าความชื้นจากเซ็นเซอร์ 2 จากชุดควบคุมความชื้น
- Local : เก็บหมายเลขชุดควบคุมความชื้น
- Nonti-less : เก็บค่าการแจ้งเตือนเมื่อมีค่าน้อย
- Nonti-more : เก็บค่าการแจ้งเตือนเมื่อมีค่ามาก
- StatusSensor : เก็บข้อมูลสถานะของเซ็นเซอร์
- Statusnode : เก็บสถานะการเชื่อมต่อ Node Nano
- System : เก็บสถานการณ์ทำงานของระบบ
- Valve : เก็บสถานะการทำงานของโซลินอยด์วาล์ว

ในระบบ Manual

- ValveAuto : เก็บข้อมูลเงื่อนไขในการเปิดใช้งานโซลินอยด์วาล์วในระบบ Auto

- ValveStatusAuto : เก็บข้อมูลสถานะการทำงานของระบบ Auto

3.5.2 Statistic

ใน Collection นี้ใช้เก็บค่าความชื้นย้อนหลัง 8 ชั่วโมง ภายใน Statistic Node มี 3 Collection ดังนี้

- humi1 : เก็บข้อมูลค่าความชื้นจากเซ็นเซอร์ 1 จากชุดควบคุมความชื้นในปัจจุบัน

- humi2 : เก็บข้อมูลค่าความชื้นจากเซ็นเซอร์ 2 จากชุดควบคุมความชื้นในปัจจุบัน

- time : เก็บข้อมูลค่าเวลาในขณะปัจจุบันที่มีการส่งข้อมูลเข้ามาใน Statistic Node

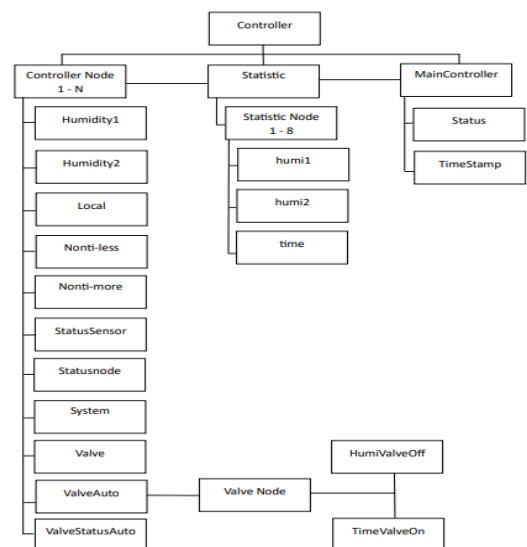
3.5.3 Main Controller

ใน Collection นี้ใช้เก็บสถานะการเชื่อมต่อของของ ESP32 และไฟร์เบส มี 2 Collection ดังนี้

- Status : เก็บข้อมูลค่าสถานะการเชื่อมต่อ ESP32 และไฟร์เบส

- TimeStamp : เก็บข้อมูลค่าเวลาในขณะปัจจุบันที่มีการส่งข้อมูลจาก ESP32 มา

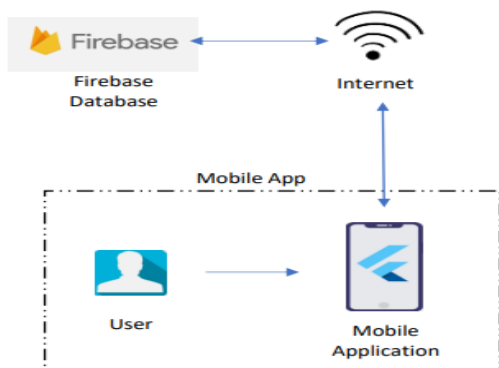
สมาชิกต่างๆจะมีโครงสร้างฐานข้อมูลดังรูปที่ 12



รูปที่ 12 โครงสร้างฐานข้อมูล

3.6 ออกแบบแอปพลิเคชันของระบบ

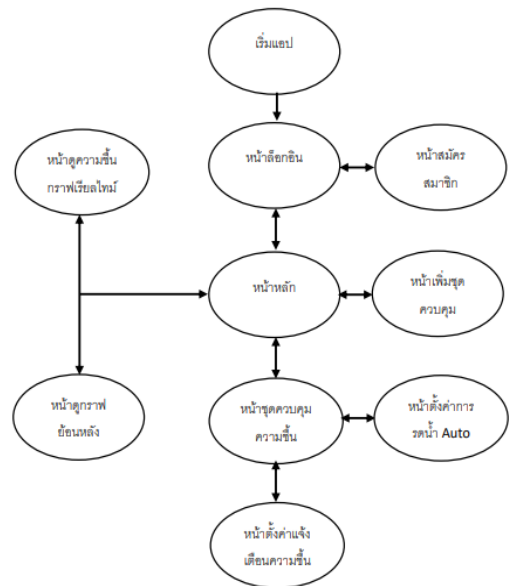
ในตัว Mobile App จะมีหลักการทำงานคือ User สั่งการทำงานผ่าน Mobile Application และรับ-ส่งข้อมูลที่ได้รับไปยังฐานข้อมูลไฟร์เบสโดยผ่านอินเทอร์เน็ต ดังรูปที่ 13



รูปที่ 13 โครงสร้าง Mobile App

แอปพลิเคชันของระบบนั้นถูกพัฒนาขึ้นด้วยภาษา Dart โดยใช้ Cross-Platform Framework Flutter ที่ใช้ในการพัฒนา Native Mobile Application (Android/iOS) ที่พัฒนาโดยบริษัท Google Inc. โดยใช้ภาษา Dart มาช่วยในการสร้าง Widget พื้นฐานเพื่อทำให้การออกแบบ UI มีความง่าย และสะดวก

การทำงานเบื้องต้นของแอปพลิเคชัน ที่พัฒนาขึ้นจะสามารถเขียนเป็นผังขั้นตอน ดังรูปที่ 14 โดยภายในแอปพลิเคชัน จะแบ่งเป็น 2 ส่วนหลัก และ ส่วนควบคุม ชุดควบคุมความชื้นและส่วนแสดงข้อมูล โดยส่วนควบคุมชุดควบคุมความชื้นนั้น คือการสั่งงานเปิด-ปิด ชุดควบคุม



รูปที่ 14 ผังการทำงานของแอปพลิเคชัน

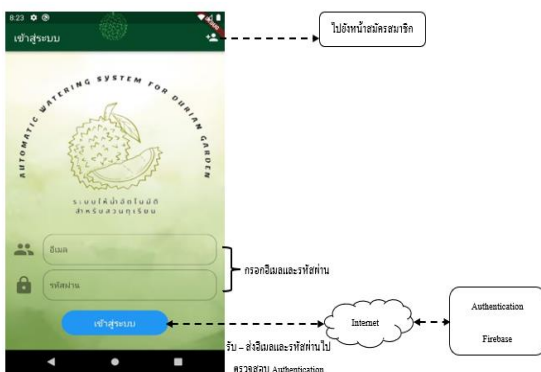
จากรูปที่ 14 เมื่อมีการเริ่มแอปจะไปดังหน้าต่อไปนี้

- หน้าล็อกอิน คือ หน้าจอแสดงผลเพื่อเข้าสู่ระบบ
- หน้าสมัครสมาชิก คือ หน้าจอที่ใช้สมัครสมาชิกเพื่อเข้าถึงการใช้งานบนไฟร์เบส
- หน้าหลัก คือ หน้ากลางสำหรับเลือกเมนูไปยังเมนูต่างๆ
- หน้าเพิ่มชุดควบคุม คือ หน้าที่ใช้เพิ่มชุดควบคุมความชื้นจากแอปพลิเคชันเข้าไปยังฐานข้อมูล
- หน้าดูความชื้นกราฟเรียลไทม์ คือ หน้าที่ใช้ดูค่าความชื้นของโหนดนั้นๆในรูปแบบเรียลไทม์
- หน้าดูกราฟย้อนหลัง คือ หน้าที่ใช้ดูค่าความชื้นย้อนหลัง 8 ชั่วโมงของโหนดนั้นๆ

- หน้าตั้งค่าชุดควบคุมความชื้น คือ หน้าที่ใช้ควบคุมการทำงานของชุดควบคุมความชื้นทั้งหมด ไม่ว่าจะเป็นการการปรับโหมด Auto เป็น Manual หรือแจ้งเตือนต่างก็อยู่ในหน้านี้
- หน้าตั้งค่าการรดน้ำ Auto คือ หน้าที่ใช้เวลาที่ต้องการให้หัวลวระบบ Auto เปิดทำงานและแก้ไขค่าความชื้นที่ต้องการให้หัวลวปิด
- หน้าตั้งค่าการแจ้งเตือนความชื้น คือ หน้าที่ใช้ค่าความชื้นที่ต้องการจะให้ระบบแจ้งเตือนเมื่อมีค่ามากกว่าที่ผู้ใช้งานตั้งหรือน้อยกว่าที่ผู้ใช้งานตั้ง

3.6.1 หน้าล็อกอิน

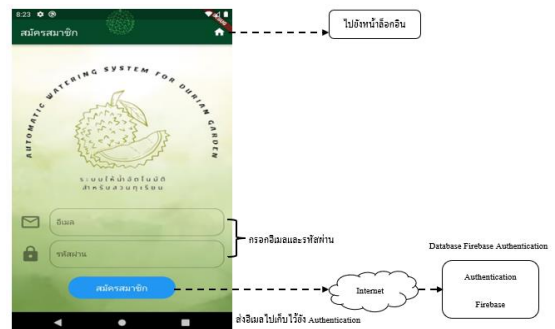
เป็นหน้าที่ผู้ใช้ทำการเข้าสู่ระบบเพื่อเข้าใช้งานระบบจากรูปที่ 15 เมื่อผู้ใช้ทำการกรอกข้อมูลและทำการกดเข้าสู่ระบบ ระบบจะตรวจสอบข้อมูลผู้ใช้งานฐานข้อมูลถ้าหากมีในฐานข้อมูลและรหัสผ่านถูกต้องจึงจะสามารถไปยังหน้าถัดไปได้



รูปที่ 15 หน้าล็อกอินเข้าสู่ระบบ

3.6.2 หน้าสมัครสมาชิก

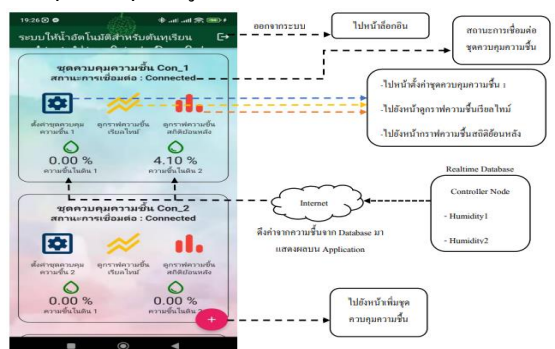
เป็นหน้าที่ผู้ใช้ทำการสมัครสมาชิกเพื่อเข้าใช้งานระบบและบันทึก Email ผู้ใช้ไปยังฐานข้อมูลเพื่อให้ผู้ใช้ทำการล็อกอินต่อไปดังรูปที่ 16



รูปที่ 16 หน้าสมัครสมาชิก

3.6.3 หน้าหลัก

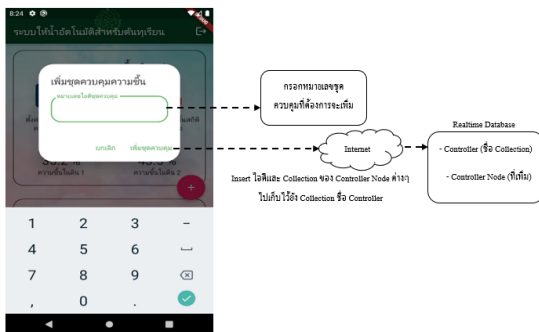
เป็นหน้าที่ผู้ใช้ทำการดูค่าความชื้นของชุดควบคุมต่างๆ ที่ดึงมาจากฐานข้อมูลไฟร์เบส และเลือกเมนูเพื่อไปยังหน้าต่างๆ ในฐานข้อมูลดังนี้ หน้าตั้งค่าชุดควบคุมความชื้น หน้าดูกราฟเรียลไทม์ หน้าดูกราฟสถิติ หน้าเพิ่มชุดควบคุม ดังรูปที่ 17



รูปที่ 17 หน้าหลัก

3.6.4 หน้าเพิ่มชุดควบคุม

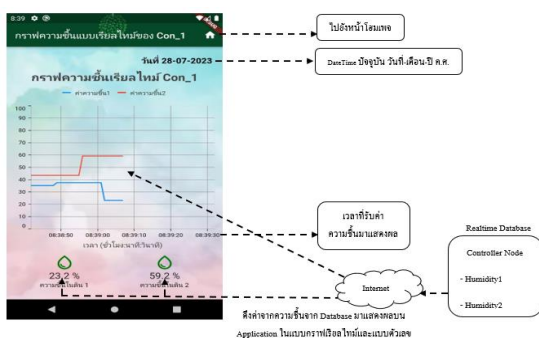
เป็นหน้าที่ผู้ใช้ต้องการเพิ่มชุดควบคุมเข้าสู่ฐานข้อมูลโดยวิธีการคือให้ผู้ใช้กำหนด ID ของชุดควบคุมที่ต้องการจะเพิ่มแล้วกดเพิ่มจากรูปที่ 18



รูปที่ 18 หน้าเพิ่มชุดควบคุมความชื้น

3.6.5 หน้าดูความชื้นกราฟเรียลไทม์

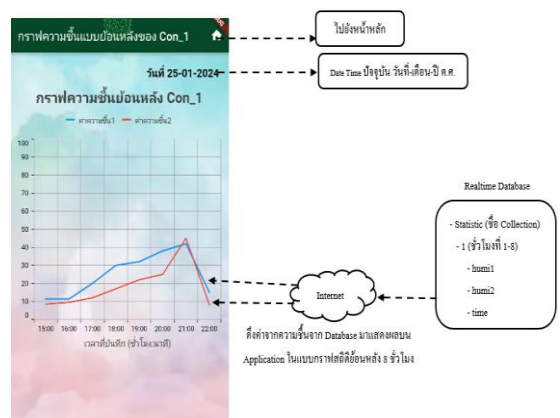
เป็นหน้าที่ผู้ใช้ต้องการดูค่าความชื้นในปัจจุบันของชุดควบคุมความชื้นในแต่ละชุดในรูปแบบของกราฟเรียลไทม์โดยค่าความชื้นมาจากฐานข้อมูลดังรูปที่ 19



รูปที่ 19 หน้าดูความชื้นกราฟเรียลไทม์

3.6.6 หน้าดูกราฟย้อนหลัง

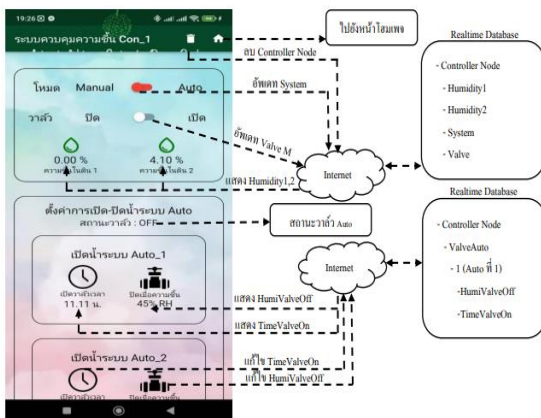
เป็นหน้าที่ผู้ใช้ต้องการดูค่าความชื้นย้อนหลัง 8 ชั่วโมงของชุดควบคุมความชื้นนั้นๆ โดยชุดควบคุมความชื้นจะทำการบันทึกค่าความชื้นในทุกๆ 1 ชั่วโมงไปยังโหนด Statistic ในฐานข้อมูลไฟร์เบส ในรูปที่ 12 แล้วแอปพลิเคชันจะทำการดึงข้อมูลมาแสดงผลในรูปแบบของกราฟดังรูปที่ 20



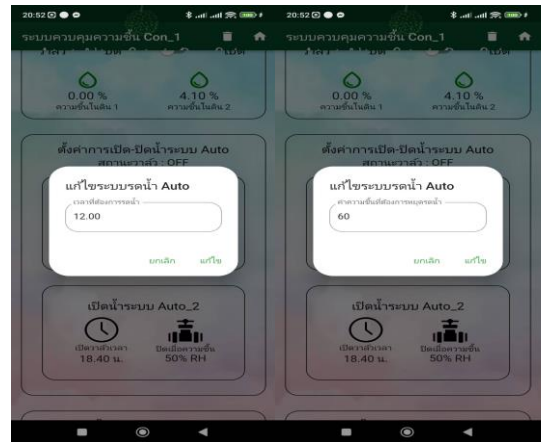
รูปที่ 20 หน้าดูความชื้นย้อนหลัง

3.6.7 หน้าตั้งค่าชุดควบคุมความชื้น

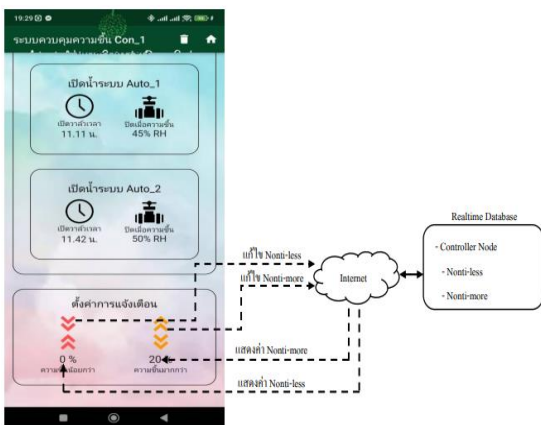
เป็นหน้าที่ผู้ใช้กำหนดการทำงานของชุดควบคุมความชื้นนั้นๆ ประกอบไปด้วย การเลือกโหมดการทำงานเป็น Auto และ Manual การเปิดวาล์วของระบบ Manual ค่าความชื้นจากเซ็นเซอร์ ตั้งค่าการเปิดและปิดวาล์วของระบบ Auto และตั้งค่าแจ้งเตือน ดังรูปที่ 21 และรูปที่ 22



รูปที่ 21 หน้าตั้งค่าชุดควบคุมความชื้น



รูปที่ 23 หน้าตั้งค่าการรดน้ำระบบ Auto



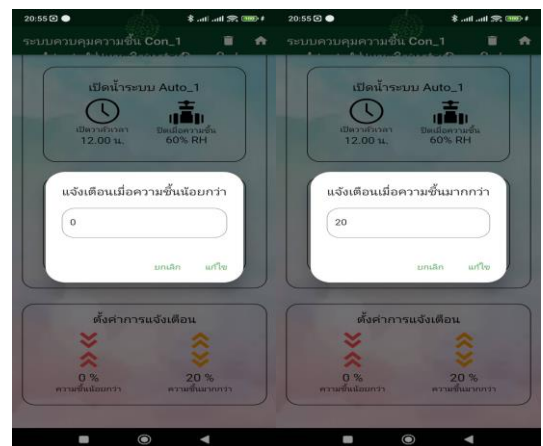
รูปที่ 22 หน้าตั้งค่าชุดควบคุมความชื้น(ต่อ)

3.6.8 หน้าตั้งค่าการรดน้ำระบบ Auto

เป็นหน้าที่ผู้ใช้ทำการแก้ไขเวลาและค่าความชื้นของการเปิด-ปิดวาล์วในระบบ Auto สามารถกำหนดได้สูงสุดจำนวน 2 ช่วงเวลาตามความต้องการของผู้ใช้งาน ดังรูปที่ 23

3.6.9 หน้าตั้งค่าแจ้งเตือนค่าความชื้น

เป็นหน้าที่ผู้ใช้ทำการแก้ไขค่าความชื้นของการแจ้งเตือนประกอบไปด้วย 2 แบบคือ แจ้งเตือนเมื่อค่าความชื้นน้อยกว่าและแจ้งเตือนเมื่อค่าความชื้นสูงกว่าตามความต้องการของผู้ใช้งานดังรูปที่ 24



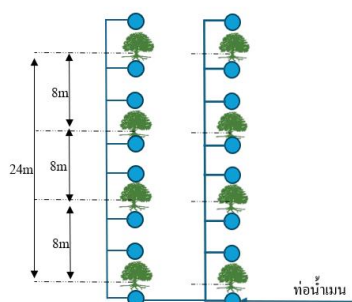
รูปที่ 24 หน้าตั้งค่าแจ้งเตือนค่าความชื้น

4. ผลการทดลอง

ในการให้น้ำต้นทุเรียนในสวนทุเรียนจะมีการติดตั้งสปริงเกอร์ข้างต้นทุเรียนซ้ายและขวาระยะห่าง 75 ซม. เพื่อบริการน้ำให้ทั่วต้นระยะห่างของต้นทุเรียนในแต่ละต้นเท่ากับ 8x8 เมตร ดังรูปที่ 26 จะเป็นรูปแบบของต้นทุเรียนเป็นแถว 4 ต้นและจุดติดตั้งสปริงเกอร์ในพื้นที่จากรูปที่ 25 โดยใช้ชุดควบคุมความชื้นจากรูปที่ 27 เป็นทดลอง



รูปที่ 25 พื้นที่ทำการทดลอง



รูปที่ 26 แบบแปลนตามพื้นที่จริงในรูปที่ 25



รูปที่ 27 ชุดควบคุมความชื้น

จะมีการทดลองต่างๆดังต่อไปนี้

4.1 การวัดความชื้นในดิน

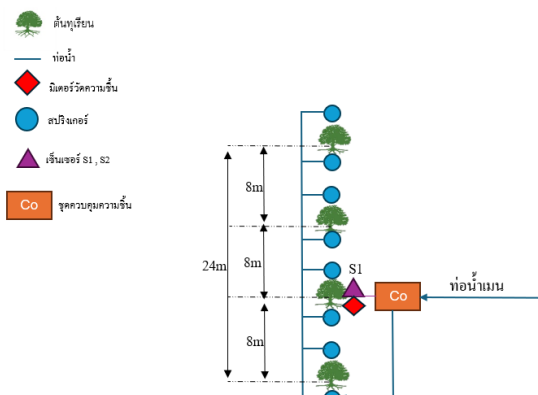
4.1.1 การเปรียบเทียบผลการวัดกับมิเตอร์วัดความชื้น

ทดลองเพื่อเทียบค่าที่วัดได้จากเซ็นเซอร์และมิเตอร์วัดความชื้นในดินมีค่าตรงกันหรือต่างกัน และเซ็นเซอร์วัดความชื้นสามารถใช้งานได้จริงหรือไม่

• ขั้นตอนการทดลอง

- ขั้นตอน ที่ 1 ติดตั้งอุปกรณ์ตามรูปที่ 28 โดยมีรายละเอียดดังนี้

- จำนวนเซ็นเซอร์ที่ใช้ในการทดลอง: 1 เซ็นเซอร์
- จำนวนมิเตอร์ที่ใช้ในการวัดเทียบค่า: 1 เครื่อง
- ความลึกที่ติดตั้งเซ็นเซอร์: 10 เซนติเมตร
- ความลึกที่ติดตั้งมิเตอร์: 10 เซนติเมตร
- สปริงเกอร์: รัศมีการกระจายน้ำ 360 องศา ระยะ 4-5 เมตร



รูปที่ 28 แบบแปลนที่ใช้ทดลองหัวข้อ 4.1.1

- ขั้นตอนที่ 2 หลังจากติดตั้งมิเตอร์วัดความชื้นข้างเซ็นเซอร์ดังรูปที่ 29 และทำการเปิดโปรแกรม Arduino IDE ดังรูปที่ 30 เพื่ออ่านค่าความชื้นจากเซ็นเซอร์ จากนั้นบันทึกค่าความชื้นลงในตารางที่ 1 โดยจะมีการเก็บผลค่าความชื้นจุดที่ทดลองก่อนรดน้ำ

ค่าความชื้นเริ่มต้น S1 เวลา 16:21 = 18

ค่าความชื้นเริ่มต้นมิเตอร์ เวลา 16:21 = 20



รูปที่ 29 ค่าความชื้นที่วัดได้จากมิเตอร์ก่อนรดน้ำ

| Timestamp | Humidity (%) |
|--------------|--------------|
| 16:20:42.348 | 18.00 |
| 16:20:46.354 | 18.00 |
| 16:20:54.415 | 18.00 |
| 16:21:00.474 | 18.00 |
| 16:21:08.539 | 18.00 |
| 16:21:12.560 | 18.00 |
| 16:21:20.634 | 18.00 |
| 16:21:28.672 | 18.00 |
| 16:21:34.694 | 18.00 |
| 16:21:40.750 | 18.00 |
| 16:21:46.799 | 18.00 |
| 16:21:50.856 | 18.00 |
| 16:21:56.870 | 17.90 |
| 16:22:00.928 | 17.90 |
| 16:22:08.997 | 18.00 |
| 16:22:17.032 | 18.00 |

รูปที่ 30 ค่าความชื้นที่วัดได้จากเซ็นเซอร์ก่อนรดน้ำ

จากรูปที่ 30 เป็นรูปในการรับ-ส่งข้อมูลบน โปรแกรม Arduino IDE ก่อนรดน้ำ ข้อมูลประกอบไปด้วย

-สีส้ม 16:20:46.354 : คือเวลาในการอ่านข้อมูล ชั่วโมง/นาที/วินาที/มิลลิวินาที

-สีน้ำเงิน 18.00 :คือ ค่าที่วัดได้จากเซ็นเซอร์วัดความชื้น S1

-สีแดง 0.00 :คือ ค่าที่วัดได้จากเซ็นเซอร์วัดความชื้น S2 มีค่าเป็น 0 เนื่องจากไม่ได้ติดตั้ง

ตารางที่ 1 ผลการวัดค่าความชื้นในดิน (ก่อนรดน้ำ)

| เวลาที่บันทึก (น.) | ค่าที่วัดได้จากเซ็นเซอร์ (% RH) | ค่าที่วัดได้จากมิเตอร์(%) |
|--------------------|---------------------------------|---------------------------|
| 16.21 | 18 | 20 |
| 16.25 | 18.50 | 19 |
| 16.30 | 18.50 | 19 |

จากผลการทดลองก่อนรดน้ำ เมื่อติดตั้งเซ็นเซอร์และสปริงเกอร์ตามจุด แล้วทำการอ่านค่าความชื้นก่อนเริ่มรดน้ำจะได้ค่าจากมิเตอร์ และได้ค่าความชื้นจากเซ็นเซอร์ และบันทึกค่าลงในตาราง ตามเวลาในตารางที่ 1 สังเกตได้ว่าค่าที่วัดจากเซ็นเซอร์และมิเตอร์วัดความชื้นในดินซึ่งค่ายังไม่เปลี่ยนแปลงมากนัก เนื่องจากยังไม่ได้มีการรดน้ำ

- ขั้นตอนที่ 3 หลังจากทดลองวัดค่าความชื้นจุดทดลองแบบก่อนรดน้ำเสร็จ ทำการเปิดสปริงเกอร์แล้วสังเกตผลค่าที่เปลี่ยนไปจากมิเตอร์ดังรูปที่ 31 และค่าความชื้นจากเซ็นเซอร์ดังรูปที่ 32 จากนั้นทำการบันทึกผลลงในตารางที่ 2 โดยจะเป็นการเก็บผลค่าความชื้นสภาพพื้นดินที่ทดลองเมื่อเริ่มรดน้ำ

ค่าความชื้นเมื่อเริ่มรดน้ำ S1 เวลา 16:31 = 19.90

ค่าความชื้นจากมิเตอร์ เวลา 16:31 = 21



รูปที่ 31 ค่าความชื้นที่วัดได้จากมิเตอร์ขณะรดน้ำ

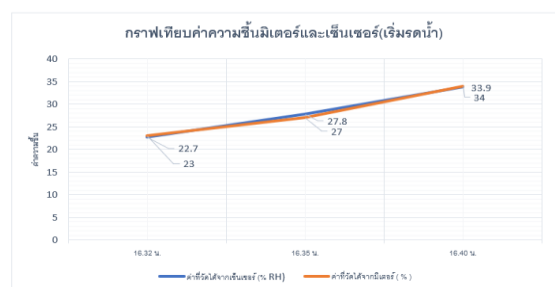
| เวลา | ค่าความชื้น (%) | ค่าความชื้น (%) |
|--------------|-----------------|-----------------|
| 16:31:21.230 | 19.10 | 0.00 |
| 16:31:29.255 | 19.10 | 0.00 |
| 16:31:37.334 | 19.10 | 0.00 |
| 16:31:45.370 | 19.90 | 0.00 |
| 16:31:51.425 | 20.30 | 0.00 |
| 16:31:57.448 | 2.80 | 0.00 |
| 16:32:03.506 | 21.10 | 0.00 |
| 16:32:09.562 | 21.30 | 0.00 |
| 16:32:15.626 | 21.60 | 0.00 |
| 16:32:21.654 | 21.80 | 0.00 |
| 16:32:29.717 | 21.90 | 0.00 |
| 16:32:33.738 | 21.90 | 0.00 |
| 16:32:39.800 | 22.00 | 0.00 |
| 16:32:45.865 | 2.10 | 0.00 |
| 16:32:51.887 | 22.50 | 0.00 |

รูปที่ 32 ค่าความชื้นที่วัดได้จากเซ็นเซอร์ขณะรดน้ำ

จากรูปที่ 32 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ขณะรดน้ำ จะสังเกตได้ว่าเมื่อทำการเปิดน้ำค่าความชื้นจะค่อยๆเพิ่มขึ้น

ตารางที่ 2 ผลการวัดค่าความชื้นในดิน (เริ่มรดน้ำ)

| เวลาที่บันทึก (น.) | ค่าที่วัดได้จากเซ็นเซอร์ (% RH) | ค่าที่วัดได้จากมิเตอร์ (%) |
|--------------------|---------------------------------|----------------------------|
| 16.32 | 22.70 | 23 |
| 16.35 | 27.80 | 27 |
| 16.40 | 33.90 | 34 |



รูปที่ 33 กราฟเทียบค่าความชื้นมิเตอร์และเซ็นเซอร์ ขณะรดน้ำ

จากผลการทดลองเมื่อเริ่มรดน้ำ ค่าความชื้นจะค่อยๆเพิ่มขึ้นและบันทึกผลค่าความชื้นที่บันทึกได้เป็นค่าความชื้นที่วัดได้จากมิเตอร์และจากเซ็นเซอร์ ลงในตารางที่ 2 นำมาเขียนในรูปแบบของกราฟดังรูปที่ 33 สังเกตได้ว่า ในขณะรดน้ำ เซ็นเซอร์มีการรับค่าความชื้นในดินได้ใกล้เคียงกับมิเตอร์วัดความชื้นเมื่อเทียบกับ จึงสรุปได้ว่าเซ็นเซอร์สามารถใช้งานในระบบนี้ได้

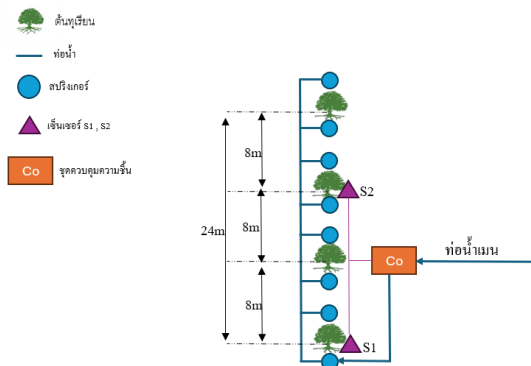
4.1.2 การวัดความชื้นที่จุดต่างๆ

ทดลองเพื่อเทียบค่าที่วัดได้ระหว่างจุดติดตั้งเซ็นเซอร์ S1 ที่อยู่ข้างคันทุเรียนต้นแรก และจุดติดตั้งเซ็นเซอร์ S2 ที่อยู่ข้างคันทุเรียนต้นที่ 3 นำค่ามาเทียบกันเมื่อมีการรดน้ำด้วยสปริงเกอร์ค่าที่ได้จะเท่ากันหรือแตกต่างกันมากหรือไม่

● ขั้นตอนการทดลอง

- ขั้นตอน ที่ 1 คิดตั้งอุปกรณ์ตามรูปที่ 34 โดยมีรายละเอียดดังนี้

- จำนวนเซ็นเซอร์ที่ใช้ในการทดลอง: 2 เซ็นเซอร์
- ความลึกที่ติดตั้งเซ็นเซอร์ S1 : 10 เซนติเมตร
- ความลึกที่ติดตั้งเซ็นเซอร์ S2 : 10 เซนติเมตร
- สปริงเกอร์: รัศมีการกระจายน้ำ 360 องศา ระยะ 4-5 เมตร



รูปที่ 34 แบบแปลนที่ใช้ทดลองหัวข้อ 4.1.2

- ขั้นตอน ที่ 2 หลังจากติดตั้งอุปกรณ์ต่างๆเสร็จแล้วทำการเปิดโปรแกรม Arduino IDE ดังรูปที่ 35 เพื่ออ่านค่าความชื้นจากเซ็นเซอร์ S1 และเซ็นเซอร์ S2 จากนั้นบันทึกค่าความชื้นลงในตารางที่ 3 โดยจะมีการเก็บผลค่าความชื้นจุดที่ทดลองก่อนรดน้ำ

```

COM7
12:56:18.023 -> Sending message to Nano2: wait_reconnect
12:56:28.647 -> Sending message to Nano1: Request
12:56:28.694 -> Receive message from Nano01: 7.70 9.30 0
12:56:32.326 ->
12:56:39.309 -> Sending message to Nano2: wait_reconnect
12:56:48.907 -> Sending message to Nano1: Request
12:56:48.954 -> Receive message from Nano01: 7.80 9.50 0
12:56:52.591 ->
12:56:58.452 -> Sending message to Nano2: wait_reconnect
12:57:09.307 -> Sending message to Nano1: Request
12:57:09.354 -> Receive message from Nano01: 7.60 9.20 0
12:57:12.938 ->
12:57:18.942 -> Sending message to Nano2: wait_reconnect
12:57:28.451 -> Sending message to Nano1: Request
12:57:28.496 -> Receive message from Nano01: 7.60 9.30 0
12:57:32.084 ->
12:57:39.216 -> Sending message to Nano2: wait_reconnect
12:57:48.734 -> Sending message to Nano1: Request
12:57:48.780 -> Receive message from Nano01: 7.60 9.30 0
12:57:52.369 ->
12:57:58.236 -> Sending message to Nano2: wait_reconnect
12:58:07.748 -> Sending message to Nano1: Request
12:58:07.795 -> Receive message from Nano01: 7.60 9.30 0
12:58:11.435 ->
12:58:17.306 -> Sending message to Nano2: wait_reconnect
12:58:26.914 -> Sending message to Nano1: Request
12:58:26.960 -> Receive message from Nano01: 7.60 9.40 0
12:58:30.593 ->
12:58:36.459 -> Sending message to Nano2: wait_reconnect
12:58:47.085 -> Sending message to Nano1: Request
  
```

รูปที่ 35 ค่าความชื้นที่วัดได้จากเซ็นเซอร์ S1 และ S2 ก่อนรดน้ำ

จากรูปที่ 35 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ก่อนรดน้ำ ข้อมูลประกอบไปด้วย

- สี แดง “Sending message to Nano2 : wait_reconnect” คือ ESP32 ส่งข้อความไปยังชุดควบคุมความชื้น 2 ไม่สามารถส่งข้อความกลับได้เมื่อ wait_reconnect

- สี น้ำเงิน “Sending message to Nano1 : Request” คือ ESP32 ส่งข้อความไปยังชุดควบคุมความชื้น 1 สามารถส่งข้อความได้เมื่อมีการ Request ไปยังชุดควบคุมความชื้น 1 และรอตอบกลับ

- สี เขียว “Receive message from Nano01: 7.60 9.40 0” คือ ข้อความที่ตอบกลับจากชุดควบคุมความชื้น Nano1

- 7.60 คือ ค่าความชื้นจากเซ็นเซอร์ S1

- 9.40 คือ ค่าความชื้นจากเซ็นเซอร์ S2

- 0 คือ สถานการณ์เชื่อมต่อเซ็นเซอร์ปกติ หากเป็น 1 คือ ติดต่อเซ็นเซอร์ S1 ไม่ได้ หรือเป็น 2 ติดต่อเซ็นเซอร์ S2 ไม่ได้ และกรณีเป็น 3 คือไม่สามารถติดต่อเซ็นเซอร์ได้ทั้ง 2 ตัว

ตารางที่ 3 ผลการวัดค่าความชื้นเซ็นเซอร์ S1 และ S2 ก่อนรดน้ำ

| เวลาที่บันทึก (น.) | ค่าที่วัดได้จากเซ็นเซอร์ S1 (% RH) | ค่าที่วัดได้จากเซ็นเซอร์ S2 (% RH) |
|--------------------|------------------------------------|------------------------------------|
| 12.54 | 7.50 | 9.30 |
| 12.56 | 7.70 | 9.40 |
| 12.59 | 7.70 | 9.40 |

จากการทดลองก่อนรดน้ำ เมื่อผู้ใช้ทำการติดตั้งเซ็นเซอร์ S1 และ S2 เสร็จแล้วอ่านค่าความชื้นที่ได้จากเซ็นเซอร์ที่แสดงบนโปรแกรม Arduino IDE มาบันทึกผลที่ได้ลงในตารางที่ 3 ตามเวลาต่างๆ ในระยะเวลา 5 นาที สังเกตได้ว่าค่าความชื้นค่อนข้างที่จะนิ่งเมื่อยังไม่มีการรดน้ำ

- ขั้นตอน ที่ 3 หลังจากทดลองวัดค่าความชื้นจนทดลองแบบก่อนรดน้ำเสร็จ ทำการเปิดสปริงเกอร์แล้วสังเกตผลค่าที่เปลี่ยนไปจากรูปที่ 36 จากนั้นทำการบันทึกผลลงในตารางที่ 4 โดยจะเป็นการเก็บผลค่าความชื้นสภาพพื้นดินที่ทดลองเมื่อเริ่มรดน้ำ

ค่าความชื้นเริ่มต้นขณะรดน้ำ S1 เวลา 13.00 = 7.70

ค่าความชื้นเริ่มต้นขณะรดน้ำ S2 เวลา 13.00 = 9.40

```

COMP
12:58:36.459 -> Sending message to Nano2: wait_reconnect
12:58:47.085 -> Sending message to Nano1: Request
12:58:47.131 -> Receive message from Nano1: 7.60 9.40 0
12:58:50.727 ->
12:58:56.597 -> Sending message to Nano2: wait_reconnect
12:59:06.146 -> Sending message to Nano1: Request
12:59:06.192 -> Receive message from Nano1: 7.60 9.40 0
12:59:15.779 -> Sending message to Nano2: wait_reconnect
12:59:25.366 -> Sending message to Nano1: Request
12:59:25.413 -> Receive message from Nano1: 7.70 9.40 0
12:59:29.050 ->
12:59:36.356 -> Sending message to Nano2: wait_reconnect
12:59:45.956 -> Sending message to Nano1: Request
12:59:46.003 -> Receive message from Nano1: 7.70 9.40 0
12:59:49.731 ->
12:59:55.599 -> Sending message to Nano2: wait_reconnect
13:00:05.237 -> Sending message to Nano1: Request
13:00:05.294 -> Receive message from Nano1: 7.60 9.40 0
13:00:12.790 ->
13:00:18.423 -> Sending message to Nano2: wait_reconnect
13:00:29.081 -> Sending message to Nano1: ON
13:00:29.127 -> Receive message from Nano1: 7.70 9.40 0
13:00:33.359 ->
13:00:39.229 -> Sending message to Nano2: wait_reconnect
13:00:49.978 -> Sending message to Nano1: Request
13:00:50.024 -> Receive message from Nano1: 8.10 9.50 0
13:00:53.611 ->
13:00:59.471 -> Sending message to Nano2: wait_reconnect
13:01:09.576 -> Sending message to Nano1: Request
  
```

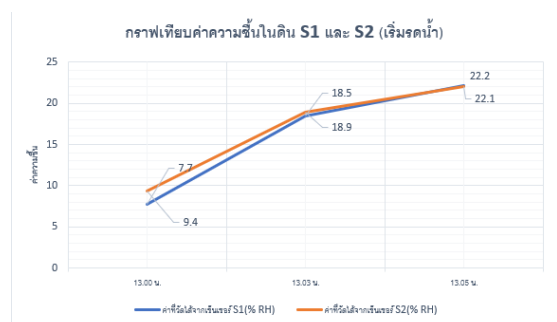
รูปที่ 36 ค่าความชื้นที่วัดได้จากเซ็นเซอร์ S1 และ S2 ขณะรดน้ำ

จากรูปที่ 36 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE เริ่มรดน้ำ ข้อมูลประกอบไปด้วย

- สีแดง ที่เวลา 13.00 น. “Sending message to Nano1 : ON” คือ ESP32 รับข้อความจากไฟร์เบสที่ผู้ใช้ทำการเปลี่ยนผ่านแอปพลิเคชัน และ ESP32 จะส่งข้อความ “ON” ไปยังชุดควบคุมความชื้น1 เพื่อสั่งให้โซลินอยด์ตัวเปิดใช้งาน จะสังเกตได้ว่าเมื่อทำการเปิดสปริงเกอร์ค่าความชื้นจะค่อยๆเพิ่มขึ้น

ตารางที่ 4 ผลการวัดค่าความชื้นเซ็นเซอร์ S1 และ S2 ขณะรดน้ำ

| เวลาที่บันทึก (น.) | เซ็นเซอร์ S1(% RH) | เซ็นเซอร์ S2(% RH) |
|--------------------|--------------------|--------------------|
| 13.00 | 7.70 | 9.40 |
| 13.03 | 18.50 | 18.90 |
| 13.05 | 22.20 | 22.10 |



รูปที่ 37 กราฟเทียบค่าความชื้นเซ็นเซอร์ S1 และ S2 ขณะรดน้ำ

ผลการทดลองขณะรดน้ำ เมื่อผู้ใช้ทำเริ่มรดน้ำและเวลาต่างๆ จากในตารางที่ 4 ที่บันทึกค่ามาจากรโปรแกรม Arduino IDE ค่าความชื้นจะค่อยๆเพิ่มตามกราฟในรูปที่ 37 สังเกตได้ว่าค่าความชื้นจะเพิ่มขึ้นค่อนข้างที่จะใกล้เคียงกันเนื่องจากมีลักษณะการให้น้ำรดน้ำนั้นเหมือนกัน

- ขั้นตอนที่ 4 หลังจากทดลองวัดค่าความชื้นจุดทดลองแบบเริ่มรดน้ำเสร็จ ทำการปิดสปริงเกอร์แล้วสังเกตผลค่าที่เปลี่ยนไปจากรูปที่ 38 จากนั้นทำการบันทึกผลลงในตารางที่ 5 โดยจะเป็นการเก็บผลค่าความชื้นสภาพพื้นดินที่ทดลองหลังรดน้ำ

ค่าความชื้นหลังรดน้ำ S1 เวลา 13.06 = 28.00

ค่าความชื้นหลังรดน้ำ S2 เวลา 13.06 = 25.10

```
COM1
13:04:48.689 -> Receive message form Nano01: 22.80 21.60 0
13:04:52.323 ->
13:04:58.193 -> Sending message to Nano2: wait_reconnect
13:05:07.709 -> Sending message to Nano1: Request
13:05:07.755 -> Receive message form Nano01: 22.00 20.70 0
13:05:11.339 ->
13:05:17.209 -> Sending message to Nano2: wait_reconnect
13:05:26.040 -> Sending message to Nano1: Request
13:05:26.886 -> Receive message form Nano01: 22.20 22.10 0
13:05:30.520 ->
13:05:37.503 -> Sending message to Nano2: wait_reconnect
13:05:47.111 -> Sending message to Nano1: Request
13:05:47.156 -> Receive message form Nano01: 28.60 25.90 0
13:05:50.789 ->
13:05:56.653 -> Sending message to Nano2: wait_reconnect
13:06:06.162 -> Sending message to Nano1: Request
13:06:06.208 -> Receive message form Nano01: 28.00 25.10 0
13:06:09.839 ->
13:06:17.014 -> Sending message to Nano2: wait_reconnect
13:06:26.654 -> Sending message to Nano1: OFF
13:06:26.701 -> Receive message form Nano01: 27.00 24.30 0
13:06:30.283 ->
13:06:36.194 -> Sending message to Nano2: wait_reconnect
13:06:46.906 -> Sending message to Nano1: Request
13:06:46.953 -> Receive message form Nano01: 26.20 23.90 0
13:06:50.592 ->
13:06:56.543 -> Sending message to Nano2: wait_reconnect
13:07:06.082 -> Sending message to Nano1: Request
13:07:06.128 -> Receive message form Nano01: 26.10 23.60 0
13:07:09.708 ->
```

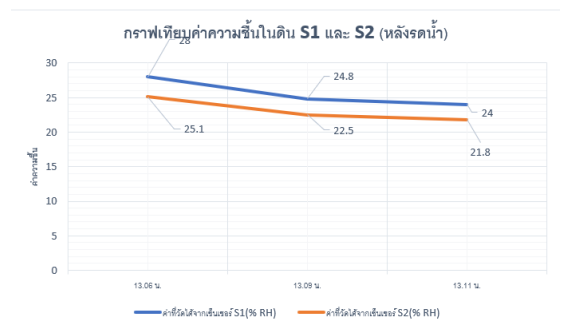
รูปที่ 38 ค่าความชื้นที่วัดได้จากเซ็นเซอร์ S1 และ S2 หลังรดน้ำ

จากรูปที่ 38 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ขณะรดน้ำ ข้อมูลประกอบไปด้วย

-สีแดง ที่เวลา 13.00 น. “Sending message to Nano1 : OFF” คือ ESP32 รับข้อความจากไฟร์เบสที่ผู้ใช้ทำการเปลี่ยนผ่านแอปพลิเคชัน และ ESP32 จะส่งข้อความ “OFF” ไปยังชุดควบคุมความชื้น1 เพื่อสั่งให้โซลินอยด์วาล์วปิดใช้งาน จะสังเกตได้ว่าการทำการปิดสปริงเกอร์ค่าความชื้นจะค่อยๆลดลง

ตารางที่ 5 ผลการวัดค่าความชื้นเซ็นเซอร์ S1 และ S2 หลังรดน้ำ

| เวลาที่บันทึก (น.) | ค่าที่วัดได้จากเซ็นเซอร์ S1 (% RH) | ค่าที่วัดได้จากเซ็นเซอร์ S2 (% RH) |
|--------------------|------------------------------------|------------------------------------|
| 13.06 | 28.00 | 25.10 |
| 13.09 | 24.80 | 22.50 |
| 13.11 | 24.00 | 21.80 |



รูปที่ 39 กราฟเทียบค่าความชื้นเซ็นเซอร์ S1 และ S2 หลังรดน้ำ

จากผลการทดลองหลังรดน้ำ เมื่อผู้ใช้ทำการปิดน้ำแล้วทำการบันทึกค่าความชื้นที่ได้จากเซ็นเซอร์ S1 และ S2 ลงในตารางที่ 5 และค่าความชื้นจะค่อยๆลดลงตามกราฟในรูปที่ 39 จะสังเกตได้ว่าค่าความชื้นจะลดลงเร็วมากในช่วงเวลา 1-3 นาทีเนื่องจากการติดตั้งเซ็นเซอร์เพียง 10 ซม. ไม่สามารถอ่านค่าความชื้นที่น้ำซึมลงดินมากกว่า 10 ซม. ได้

จากการทดลองติดตั้งเซ็นเซอร์จุดต่างๆทั้งก่อนรดน้ำ เริ่มรดน้ำ และหลังรดน้ำ สรุปได้ว่าเมื่อมีการให้น้ำในรูปแบบเดียวกันและหน้าพื้นดินเดียวกันค่าความชื้นจะเพิ่มขึ้นในปริมาณที่เท่ากันนั้นหมายความว่าชุดควบคุมความชื้นหากติดตั้งเซ็นเซอร์แบบต้นเวินต้นค่าความชื้นก็จะไม่ห่างกันมาก

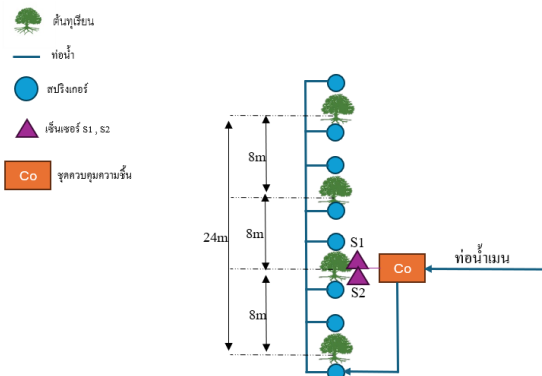
4.1.3 การวัดความชื้นที่ความลึกต่างกัน

ในหัวข้อการทดลองนี้ทดลองเพื่อเทียบค่าที่วัดได้ระหว่างเซ็นเซอร์ S1 และเซ็นเซอร์ S2 ในจุดติดตั้งเดียวกัน แต่ค่าความลึกในการฝังเซ็นเซอร์ต่างกัน นำมาเปรียบเทียบกันเพื่อหาค่าความลึกที่เหมาะสมสำหรับติดตั้งเซ็นเซอร์สำหรับใช้งานระบบ

● ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 ติดตั้งอุปกรณ์ตามรูปที่ 40 โดยมีรายละเอียดดังนี้

- จำนวนเซ็นเซอร์ที่ใช้ในการทดลอง: 2 เซ็นเซอร์
- ความลึกที่ติดตั้งเซ็นเซอร์ S1 : 10 เซนติเมตร
- ความลึกที่ติดตั้งเซ็นเซอร์ S2 : 15 เซนติเมตร
- สปริงเกอร์: รัศมีการกระจายน้ำ 360 องศา ระยะ 4-5 เมตร



รูปที่ 40 แบบแปลนที่ใช้ทดลองหัวข้อ 4.1.3

- ขั้นตอนที่ 2 หลังจากติดตั้งอุปกรณ์ต่างๆเสร็จแล้วทำการเปิดโปรแกรม Arduino IDE ดังรูปที่ 41 เพื่ออ่านค่าความชื้นจากเซ็นเซอร์ S1 และเซ็นเซอร์ S2 จากนั้นบันทึกค่าความชื้นลงในตารางที่ 6 โดยจะมีการเก็บผลค่าความชื้นจุดที่ทดลองก่อนรดน้ำ

ค่าความชื้นเริ่มต้น S1 เวลา 16:28 : 14.60

ค่าความชื้นเริ่มต้น S2 เวลา 16:28 : 23.50

| Time | S1 Moisture (%) | S2 Moisture (%) |
|--------------|-----------------|-----------------|
| 16:28:08.801 | 14.60 | 23.50 |
| 16:28:12.821 | 14.60 | 23.40 |
| 16:28:16.878 | 14.60 | 23.50 |
| 16:28:22.921 | 14.60 | 23.50 |
| 16:28:30.960 | 14.60 | 23.40 |
| 16:28:35.013 | 14.60 | 23.40 |
| 16:28:43.077 | 14.60 | 23.30 |
| 16:28:49.111 | 14.60 | 23.40 |
| 16:28:55.165 | 14.70 | 23.40 |
| 16:29:01.238 | 14.60 | 23.40 |
| 16:29:07.277 | 14.00 | 23.40 |
| 16:29:15.329 | 1.60 | 2.50 |
| 16:29:21.375 | 14.60 | 23.40 |
| 16:29:27.430 | 14.60 | 23.40 |
| 16:29:33.437 | 14.60 | 23.50 |
| 16:29:37.494 | 14.70 | 3.40 |
| 16:29:43.561 | 14.70 | 23.40 |
| 16:29:51.620 | 14.60 | 23.40 |
| 16:29:55.661 | 14.70 | 23.40 |
| 16:29:59.683 | 14.60 | 23.40 |
| 16:30:05.752 | 14.60 | 23.50 |
| 16:30:11.791 | 14.60 | 23.40 |
| 16:30:19.847 | 1.70 | 3.40 |
| 16:30:27.881 | 14.60 | 23.50 |

รูปที่ 41 ค่าความชื้นที่วัดได้จากเซ็นเซอร์ S1 และ S2 ก่อนรดน้ำ

จากรูปที่ 41 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ก่อนรดน้ำ จะสังเกตได้ว่าค่าความชื้นที่วัดได้จากเซ็นเซอร์จะมีค่าที่ค่อนข้างนิ่งคงที่และนำไปค่าที่ได้บันทึกลงในตารางที่ 6



รูปที่ 42 ค่าที่วัดได้จากมิเตอร์จุด S1 และ S2 ก่อนรดน้ำ

ตารางที่ 6 ผลการวัดค่าความชื้นเซ็นเซอร์ S1 และ S2 ก่อนรดน้ำ

| เวลาที่บันทึก (น.) | เซ็นเซอร์ S1 ฟังลึก 10 ซม. (%RH) | เซ็นเซอร์ S2 ฟังลึก 15 ซม. (%RH) |
|-------------------------|-------------------------------------|-------------------------------------|
| 16.28 | 14.60 | 23.40 |
| 16.31 | 14.70 | 23.30 |
| 16.36 | 14.80 | 23.60 |

จากผลการทดลองก่อนรดน้ำ หลังจากติดตั้งเซ็นเซอร์ตามจุดติดตั้ง และสังเกตความลึกให้ต่างกันเสร็จแล้ว ทำการบันทึกค่าความชื้นจากเซ็นเซอร์ที่แสดงบนโปรแกรม Arduino IDE ลงในตารางที่ 6 ตามเวลาต่างๆ และตรวจสอบค่าความถูกต้องด้วยมิเตอร์วัดความชื้น เพื่อนำค่าเริ่มต้นก่อนที่จะเริ่มรดน้ำและนำไปเปรียบเทียบค่าเมื่อเริ่มรดน้ำได้

- ขั้นตอน ที่ 3 หลังจากทดลองวัดค่าความชื้นจุดทดลองแบบก่อนรดน้ำเสร็จ ทำการเปิดสปริงเกอร์แล้วสังเกตผลค่าที่เปลี่ยนไปจากรูปที่ 43 จากนั้นทำการบันทึกผลลงในตารางที่ 7 โดยจะเป็นการเก็บผลค่าความชื้นสภาพพื้นดินที่ทดลองเมื่อเริ่มรดน้ำ

ค่าความชื้นเริ่มต้นขณะรดน้ำ S1 เวลา 16:40 : 14.70

ค่าความชื้นเริ่มต้นขณะรดน้ำ S2 เวลา 16:40 : 23.50

| Time (HH:MM:SS) | Humidity (%) |
|-----------------|--------------|
| 16:39:34.240 | 14.80-23.40 |
| 16:39:38.303 | 14.70-23.40 |
| 16:39:44.364 | 14.80-23.40 |
| 16:39:50.406 | 14.80-23.50 |
| 16:39:56.436 | 14.70-23.40 |
| 16:40:06.479 | 14.70-23.50 |
| 16:40:10.539 | 14.70-23.50 |
| 16:40:18.580 | 14.70-23.50 |
| 16:40:24.655 | 14.90-23.40 |
| 16:40:30.665 | 15.50-23.50 |
| 16:40:36.748 | 17.20-23.40 |
| 16:40:44.799 | 17.40-23.40 |
| 16:40:52.822 | 17.70-23.40 |
| 16:40:58.867 | 17.70-23.40 |
| 16:41:04.912 | 17.70-23.40 |
| 16:41:12.954 | 17.60-23.40 |
| 16:41:17.014 | 17.0-23.40 |
| 16:41:25.074 | 17.40-23.40 |
| 16:41:29.095 | 17.40-23.40 |
| 16:41:35.165 | 17.60-23.50 |
| 16:41:41.217 | 17.50-23.40 |
| 16:41:49.235 | 17.40-23.40 |
| 16:41:53.320 | 17.40-23.60 |
| 16:41:59.347 | 17.70-23.40 |

รูปที่ 43 ค่าความชื้นที่วัดได้จากเซ็นเซอร์ S1 และ S2 ขณะรดน้ำ

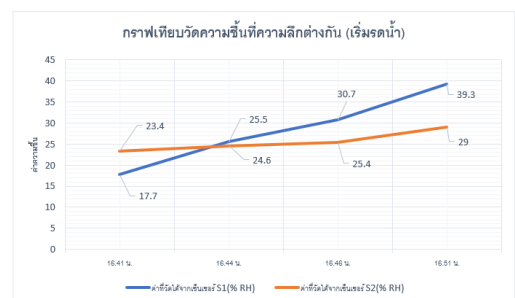
จากรูปที่ 43 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ขณะรดน้ำ จะสังเกตได้ว่าเมื่อทำการเปิดน้ำค่าความชื้นจะค่อยๆ เพิ่มขึ้นและนำไปค่าที่ได้บันทึกลงในตารางที่ 7



รูปที่ 44 ค่าวัดได้จากมิเตอร์ในจุด S1 และ S2 ขณะรดน้ำ

ตารางที่ 7 ค่าความชื้นเซ็นเซอร์ S1 และ S2 ขณะรดน้ำ

| เวลาที่บันทึก (น.) | เซ็นเซอร์ S1 ฟังลึก 10 ซม. (%RH) | เซ็นเซอร์ S2 ฟังลึก 15 ซม. (%RH) |
|-------------------------|-------------------------------------|-------------------------------------|
| 16.41 | 17.70 | 23.40 |
| 16.44 | 25.50 | 24.60 |
| 16.46 | 30.70 | 25.40 |
| 16.51 | 39.30 | 29.00 |



รูปที่ 45 กราฟเทียบวัดความชื้นที่ความลึกต่างกัน ขณะรดน้ำ

จากผลการทดลองในขณะเริ่มรดน้ำ นำค่าความชื้นจากเซ็นเซอร์บนโปรแกรม Arduino IDE มาบันทึกผลลงในตารางที่ 7 นำค่าความชื้นมาแสดงเป็นรูปแบบของกราฟดังรูปที่ 45 สังเกตได้ว่า ค่าความชื้นจะค่อยๆ เพิ่มขึ้นเมื่อเริ่มรดน้ำ และตรวจสอบค่าความถูกต้องด้วยมิเตอร์ จากกราฟสรุปได้ว่าค่าความชื้นจากเซ็นเซอร์ S2 นั้นเพิ่มช้ากว่าค่าความชื้นเซ็นเซอร์ S1 เนื่องจากความลึกของเซ็นเซอร์ S2 มีมากกว่า S1 ซึ่งปริมาณน้ำที่ซึมซับลงมีปริมาณที่พอๆ กัน และในระยะเวลาเท่ากันในช่วงรดน้ำ

จากผลการทดลองความลึกเซ็นเซอร์ สรุปได้ว่ายังติดตั้งเซ็นเซอร์ลึกมาก ค่าความชื้นจากเซ็นเซอร์ก็เพิ่มช้ามากแต่จะทำให้ค่าความชื้นหลังรดน้ำของความลึก 15 ซม. อยู่ยาวนานกว่าความลึก 10 ซม. และอาจทำให้เกิดความผิดพลาดในการให้ปริมาณน้ำมากเกินไป ดังนั้นควรติดตั้งเซ็นเซอร์วัดความชื้นที่ความลึก 10 ซม.

4.2 การทดสอบการควบคุมความชื้นแบบอัตโนมัติ

ในหัวข้อการทดลองนี้ทดลองเพื่อดูว่าเมื่อผู้ใช้งานต้องการที่จะเปิดการรดน้ำในระบบอัตโนมัติผ่านแอปพลิเคชันจะสามารถทำงานได้หรือไม่

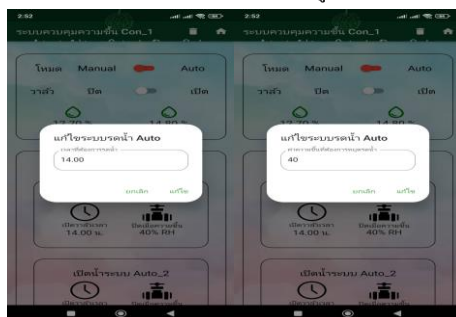
● ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เข้าสู่แอปพลิเคชันหน้าตั้งค่าชุดควบคุมความชื้นและทำการเปลี่ยนโหมดจาก Manual เป็น Auto ดังรูป 46



รูปที่ 46 ตั้งค่าเปิดน้ำระบบ Auto บนแอปพลิเคชัน

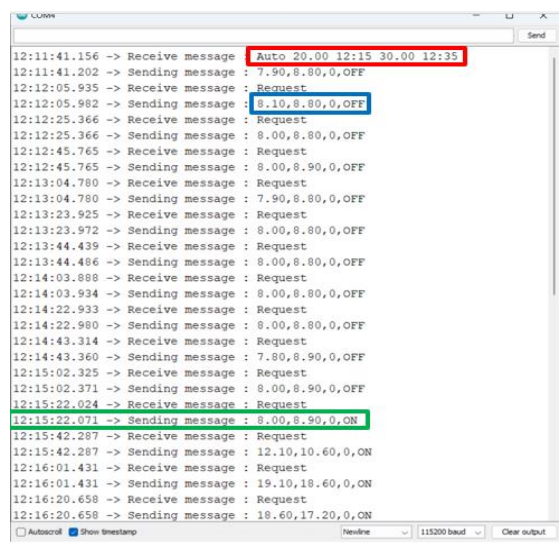
- ขั้นตอนที่ 2 ทำการกำหนดเวลาที่เปิดการทำงานของระบบอัตโนมัติโดยการกดที่ในไอคอนนาฬิกาและกำหนดค่าความชื้นที่ต้องการจะให้ชุดควบคุมหยุดการทำงานบนแอปพลิเคชันโดยการกดที่ในไอคอนวาล์วน้ำ จากรูปที่ 46 หลังจากนั้นทำการกำหนดค่าที่ต้องการให้ระบบทำงานดังรูปที่ 47



รูปที่ 47 กำหนดค่าเวลาเปิดวาล์วและค่าความชื้นปิดวาล์ว

- ขั้นตอนที่ 3 ทำการเปิดโปรแกรม Arduino IDE เพื่อดู

ข้อความ เมื่อถึงเวลาตามที่ผู้ใช้งานตั้งไว้ซึ่งในที่นี้ก็คือ 12.15 น. ระบบจะทำการตรวจค่าความชื้นจากเซ็นเซอร์ 1 และเซ็นเซอร์ 2 ก่อน ถ้ามีค่าความชื้นน้อยกว่าค่าความชื้นที่ต้องการจะให้ปั๊ควาล์วอัตโนมัติจะทำการเปิดวาล์ว แต่ถ้ามีค่าความชื้นมากกว่าค่าความชื้นที่ต้องการจะให้ปั๊ควาล์วอัตโนมัติจะไม่มีการเปิดวาล์ว โดยจากรูปที่ 48 ระบบจะเริ่มทำงาน



รูปที่ 48 หน้าต่าง Arduino IDE ระบบ Auto เริ่มทำงาน

จากรูปที่ 48 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE เมื่อระบบอัตโนมัติทำงาน ประกอบไปด้วย

- แสดง Auto/20.00/12.15/30.00/12.35 อธิบายตามลำดับ

- Auto คือ โหมดการทำงานของระบบแบบอัตโนมัติ
- 20.00 คือ ค่าความชื้นที่ปั๊ควาล์วครั้งที่ 1
- 12:15 คือ เวลาที่ปั๊ควาล์วครั้งที่ 1
- 30.00 คือ ค่าความชื้นที่ปั๊ควาล์วครั้งที่ 2
- 12.35 คือ เวลาที่ปั๊ควาล์วครั้งที่ 2

- สิ้นน้ำเงิน 8.10 / 8.80 / 0 / OFF อธิบายตามลำดับ

- 8.10 คือ ค่าความชื้นจากเซ็นเซอร์ 1
- 8.80 คือ ค่าความชื้นจากเซ็นเซอร์ 2
- 0 คือ ค่าสถานะของเซ็นเซอร์ 0 คือเชื่อมต่อ
- OFF คือ สถานะของวาล์วในระบบอัตโนมัติมีค่าเป็น ONF เมื่อถึงเวลาที่ตั้ง และจะเปลี่ยนเป็น OFF เมื่อค่าความชื้นถึงค่าความชื้นที่ตั้งให้หยุด

- สีเขียว แสดงช่วงเวลา 12.15 ระบบจะแสดงข้อความเมื่อวาล์วมีการเปลี่ยนสถานะเป็น ON ตามเวลาที่ผู้ใช้งานตั้ง



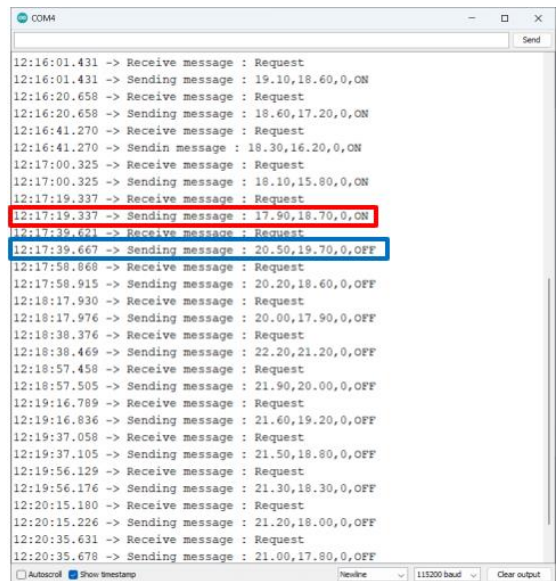
รูปที่ 49 สถานะจอ LCD เมื่อระบบ Auto ยังไม่ทำงาน

จากรูปที่ 49 เป็นรูปหน้าจอ LCD ของชุดควบคุมความชื้นเมื่อระบบยังไม่มีการรดน้ำ ซึ่งผู้ทดลองตั้งให้เริ่มรดน้ำเมื่อ 12.15 น.



รูปที่ 50 สถานะจอ LCD เมื่อระบบ Auto เริ่มทำงาน

จากรูปที่ 50 เป็นรูปหน้าจอ LCD ของชุดควบคุมความชื้นเมื่อถึงเวลาตามที่ผู้ใช้งานตั้งไว้ นั่นก็คือ 12.15 น. ระบบจะทำการสั่งวาล์วเปิดและให้หยุดเมื่อค่าความชื้น S1 และ S2 มีค่ามากกว่าที่ผู้ใช้งานตั้งไว้ต่อไป



รูปที่ 51 หน้าต่าง Arduino IDE ระบบ Auto หยุดทำงาน

จากรูปที่ 51 เป็นรูปในการรับ-ส่งข้อมูลบน โปรแกรม Arduino IDE ขณะรดน้ำ จะสังเกตได้ว่าระบบอัตโนมัติจะส่งสถานะ ON ครั้งสุดท้ายในกรอบสีแดง และหยุดทำงานระบบส่งสถานะ OFF เมื่อค่าความชื้นสูงกว่าที่ตั้งไว้คือ 20 %RH ในกรอบสีน้ำเงิน



รูปที่ 52 สถานะจอ LCD เมื่อระบบ Auto หยุดทำงาน

จากรูปที่ 52 เป็นรูปหน้าจอ LCD ของชุดควบคุมความชื้นเมื่อระบบ Auto หยุดทำงานเนื่องจากค่าความชื้นที่วัดได้จากเซ็นเซอร์ S1 และ S2 มีค่ามากกว่าที่ผู้ทดลองตั้งไว้คือ 20 % RH นั่นเอง

ตารางที่ 8 ผลการทดลองระบบ Auto

| เวลาที่ เริ่มรดน้ำ (น.) | ค่าความชื้นที่ ต้องการหยุด (% RH) | เวลาหยุด รดน้ำ (น.) | ค่าความชื้นเมื่อ หยุด (% RH) | |
|-------------------------------|---|---------------------------|---------------------------------|------|
| | | | S1 | S2 |
| 12.15 | 20 | 12.17 | 20.5 | 20.1 |
| 12.35 | 30 | 12.39 | 31.6 | 33.2 |

จากผลการทดลอง เมื่อถึงเวลาที่ดังตารางที่ 8 ตัวชุดควบคุมความชื้นจะเริ่มรดน้ำ และหยุดการทำงานเมื่อค่าความชื้นของ S1 และ S2 มีค่ามากกว่าหรือเท่ากับค่าความชื้นที่ตั้งให้หยุดดังตารางที่ 8 สรุปได้ว่าเมื่อผู้ใช้งานต้องการจะกำหนดให้มีการรดน้ำอัตโนมัติในแต่ละครั้ง สามารถใช้งานระบบอัตโนมัติได้ตามความต้องการของผู้ใช้งานได้แน่นอน

4.3 การสั่งงานเปิด-ปิดวาล์วผ่านแอปพลิเคชัน

ในหัวข้อการทดลองนี้ทดลองเพื่อดูว่าเมื่อผู้ใช้งานต้องการที่จะเปิด-ปิดวาล์วผ่านแอปพลิเคชันจะสามารถทำงานได้หรือไม่

● ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เข้าสู่แอปพลิเคชันหน้าตั้งค่าชุดควบคุมความชื้นและทำการเปลี่ยนโหมดจาก Auto เป็น Manual แล้วในเมนูวาล์วทำการกดเปลี่ยนสวิตช์จากปิดเป็นเปิด ดังรูป 53



รูปที่ 53 เปิดน้ำระบบ Manual บนแอปพลิเคชัน



รูปที่ 54 สถานะจอ LCD เมื่อระบบ Manual ยังไม่ทำงาน

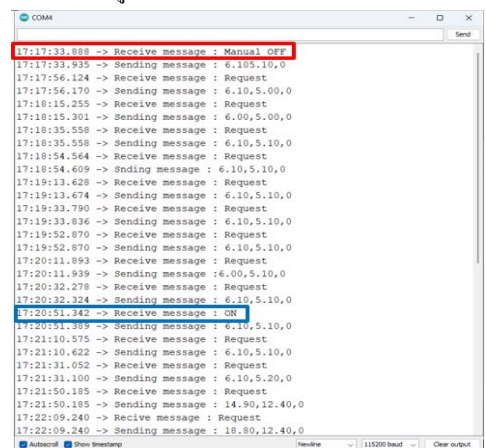
จากรูปที่ 54 เป็นรูปหน้าจอ LCD ของชุดควบคุมความชื้นเมื่อระบบ Manual ยังไม่ทำงาน



รูปที่ 55 สถานะจอ LCD เมื่อระบบ Manual เริ่มทำงาน

จากรูปที่ 55 เป็นรูปหน้าจอ LCD ของชุดควบคุมความชื้นเมื่อระบบ Manual เริ่มทำงาน

- ขั้นตอนที่ 2 ทำการเปิดโปรแกรม Arduino IDE เพื่อดูข้อความ เมื่อผู้ใช้งานทำการเปิดวาล์วผ่านแอปพลิเคชันระบบจะเปลี่ยนสถานะดังรูปที่ 56



รูปที่ 56 หน้าต่าง Arduino IDE ระบบ Manual เปลี่ยนสถานะ

จากรูปที่ 56 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE อธิบายดังนี้

- กรอบสีแดงหมายถึงช่วงเวลาที่สถานะวาล์วเป็น OFF และ
- กรอบสีน้ำเงินคือช่วงเวลาที่ย่ี่ยนสถานะจาก OFF เป็น ON

ตารางที่ 9 ผลการทดลองระบบ Manual

| ลำดับที่ | สถานะของวาล์ว | สถานะบนจอ LCD |
|----------|---------------|---------------|
| 1 | ON | ON |
| 2 | OFF | OFF |

จากผลการทดลอง เมื่อผู้ใช้ทำการเปลี่ยนโหมดการทำงานของระบบเป็นแบบ Manual และเปิดวาล์วเป็น ON ตามตารางที่ 9 จะเห็นได้ว่าระบบจะเริ่มรดน้ำ และเมื่อผู้ใช้ปิดวาล์วสถานะของวาล์วก็เปลี่ยนจาก ON เป็น OFF จึงสรุปได้ว่าเมื่อผู้ใช้งานต้องการจะเปิด-ปิดวาล์วน้ำผ่านแอปพลิเคชัน สามารถใช้งานระบบ Manual ได้ตามความต้องการของผู้ใช้งานได้แน่นอน

4.4 การดูค่าความชื้นแบบกราฟเรียลไทม์

ในหัวข้อการทดลองนี้ทดลองเพื่อดูว่าเมื่อผู้ใช้ต้องการที่จะดูค่าความชื้นในรูปแบบของกราฟเรียลไทม์บนแอปพลิเคชันว่าสามารถแสดงค่าความชื้นได้หรือไม่ในช่วงเวลานั้นๆ

●ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เข้าสู่แอปพลิเคชันหน้าหลัก จากนั้นทำการเลือกเมนูดูกราฟความชื้นเรียลไทม์ของชุดควบคุมความชื้นที่ต้องการจะดูค่าความชื้นเรียลไทม์ในกรอบสีแดง ดังรูปที่ 57



รูปที่ 57 เมนูไปยังหน้าดูกราฟความชื้นเรียลไทม์

- ขั้นตอนที่ 2 หลังจากที่ได้เลือกเมนูเสร็จแล้วจะไปยังหน้าดูกราฟความชื้นรูปแบบเรียลไทม์ของชุดควบคุมความชื้นที่เลือก ดังรูปที่ 58 จากนั้นทำการบันทึกผลการทดลองลงในตารางที่ 10



รูปที่ 58 กราฟแสดงค่าความชื้นแบบเรียลไทม์

ตารางที่ 10 ผลการทดลองดูค่าความชื้นแบบเรียลไทม์

| เวลาที่บันทึก (น.) | ค่าที่วัดได้จาก เซ็นเซอร์ S1(% RH) | ค่าที่วัดได้จาก เซ็นเซอร์ S2(% RH) |
|-----------------------|---------------------------------------|---------------------------------------|
| 16.12 | 17.40 | 15.40 |
| 16.14 | 18.60 | 15.40 |
| 16.20 | 28.10 | 39.00 |

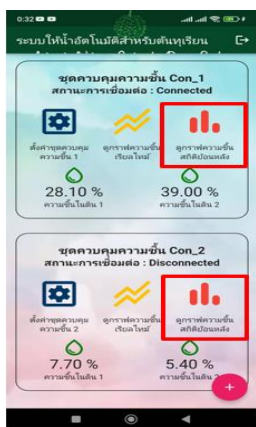
จากผลการทดลองจะเห็นได้ว่าเมื่อค่าความชื้นที่ได้จากเซ็นเซอร์ S1 และ S2 เปลี่ยนแปลงไปตามช่วงเวลาในตารางที่ 10 ทำให้ค่าความชื้นที่แสดงรูปแบบกราฟเรียลไทม์บนแอปพลิเคชันมีการเปลี่ยนแปลงตามเซ็นเซอร์ S1 และ S2 นั้นหมายความว่าผลการค่าความชื้นในรูปแบบกราฟเรียลไทม์บนแอปพลิเคชันสามารถใช้งานได้

4.5 การดูค่าความชื้นย้อนหลัง

ในหัวข้อการทดลองนี้ทดลองเพื่อดูว่าเมื่อผู้ใช้ต้องการที่จะดูค่าความชื้นย้อนหลัง 8 ชั่วโมงของชุดควบคุมนั้นๆ สามารถดูและแสดงค่าได้ถูกต้องหรือไม่

●ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เข้าสู่แอปพลิเคชันหน้าหลัก จากนั้นทำการเลือกเมนูดูกราฟความชื้นสถิติย้อนหลังของชุดควบคุมความชื้นที่ต้องการจะดูค่าความชื้นเรียลไทม์ในกรอบสีแดง ดังรูปที่ 59



รูปที่ 59 เมนูไปยังหน้าดูกราฟความชื้นสถิติย้อนหลัง

- ขั้นตอนที่ 2 หลังจากที่ได้เลือกเมนูเสร็จแล้วจะไปยังหน้าดูกราฟความชื้นสถิติย้อนหลังของชุดควบคุมความชื้นที่เลือกดังรูปที่ 60 จากนั้นทำการบันทึกผลการทดลองลงในตารางที่ 11



รูปที่ 60 กราฟแสดงค่าความชื้นย้อนหลัง 8 ชั่วโมง

ตารางที่ 11 ผลการทดลองดูค่าความชื้นสถิติย้อนหลัง

| เวลาที่บันทึก(น.) | ค่าที่วัดได้จากเซ็นเซอร์ S1(% RH) | ค่าที่วัดได้จากเซ็นเซอร์ S2(% RH) |
|-------------------|-----------------------------------|-----------------------------------|
| 10.00 | 13.90 | 13.00 |
| 11.00 | 13.90 | 13.60 |
| 12.00 | 25.10 | 24.80 |
| 13.00 | 22.10 | 21.70 |
| 14.00 | 17.80 | 18.60 |
| 15.00 | 17.80 | 18.60 |
| 16.00 | 17.40 | 15.40 |
| 17.00 | 23.40 | 23.80 |

จากผลการทดลอง เมื่อเปิดการทำงานของระบบชุดควบคุมจะทำการบันทึกช่วงเวลานั้นๆตามตารางที่ 11 ส่งไปยังฐานข้อมูลเก็บไว้เพื่อให้ตัวแอปพลิเคชันดึงข้อมูลตามช่วงเวลานั้นไปแสดงผลในรูปแบบของกราฟสถิติย้อนหลัง 8 ชั่วโมงมาแสดงได้

4.6 การทดสอบการแจ้งเตือน

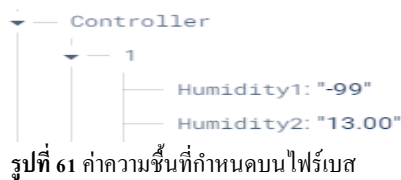
ในหัวข้อการทดลองนี้ทดลองเพื่อดูว่าเมื่อระบบเกิดข้อผิดพลาดไม่ว่าจะเป็นในกรณีใดๆ ตัวแอปพลิเคชันจะส่งแจ้งเตือนไปยังผู้ใช้งานจริงหรือไม่ การทดลองในระบบแจ้งเตือนบนแอปพลิเคชันมี 5 การทดลองดังนี้

4.6.1 ส่งแจ้งเตือนที่ขึ้นอยู่กับค่าความชื้น

ในหัวข้อการทดลองนี้ทดลองให้ระบบจะส่งแจ้งเตือนเมื่อค่าความชื้นจากเซ็นเซอร์ S1 หรือเซ็นเซอร์ S2 มีค่าผิดปกติ หรือมีค่ามากกว่าและน้อยกว่าค่าความชื้นที่ผู้ใช้ตั้งให้แจ้งเตือน

● ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 ทำการกำหนดค่าความชื้นจากเซ็นเซอร์ S1 และเซ็นเซอร์ S2 บนฐานข้อมูลไฟร์เบสที่ไม่ใช่เลข 0.00-99.99 เพื่อทดลองแจ้งเตือนเมื่อค่าความชื้นผิดปกติดังรูปที่ 61



- ขั้นตอนที่ 2 เข้าสู่แอปพลิเคชันหลัก จากนั้นทำการเลือกเมนูตั้งค่าชุดควบคุมความชื้นที่ต้องการจะทดลองในกรอบสีแดง ดังรูปที่ 62



หลังจากเข้ามาหน้าชุดควบคุมความชื้นระบบจะส่งแจ้งเตือนไปแสดงยังแอปพลิเคชันดังรูปที่ 63 จากนั้นทำการเปลี่ยนค่าบนฐานข้อมูลตามตารางที่ 12 เพื่อบันทึกผลการทดลองเมื่อค่าความชื้นผิดปกติต่อไป



ตารางที่ 12 ผลการทดลองส่งแจ้งเตือนค่าความชื้นผิดปกติ

| ค่าที่วัดได้จากเซ็นเซอร์ S1 (% RH) | ค่าที่วัดได้จากเซ็นเซอร์ S2 (% RH) | ผลที่ได้ |
|------------------------------------|------------------------------------|--------------|
| -99 | 13.00 | ส่งแจ้งเตือน |
| 25.10 | 156.25 | ส่งแจ้งเตือน |
| .\$\$\$ | None | ส่งแจ้งเตือน |
| -125 | 101.5 | ส่งแจ้งเตือน |

จากผลการทดลองแจ้งเตือนค่าความชื้นผิดปกติ เมื่อค่าความชื้นจากเซ็นเซอร์ S1 หรือ S2 ดังตารางที่ 12 ที่ผู้ทดลองกำหนดในขณะที่ทำการทดลองเมื่อมีค่าผิดปกติ หรือมากกว่า 99 หรืออักษรแปลกๆที่ไม่ใช่ 0 - 99 ระบบจะส่งแจ้งเตือนไปแสดงบนแอปพลิเคชันว่า “ค่าเตือน ค่าความชื้นผิดปกติ” นั้นหมายความว่าระบบเป็นไปตามความต้องการของผู้ใช้งาน

- ขั้นตอนที่ 3 ทดลองให้ส่งแจ้งเตือนเมื่อค่าความชื้นจากเซ็นเซอร์ S1 หรือ S2 มีค่ามากกว่าค่าความชื้นที่ต้องการให้แจ้งเตือนซึ่งก็คือ 15%RH โดยเริ่มต้นทำการติดตั้งเซ็นเซอร์ S1 และ S2 จากนั้นเข้าแอปพลิเคชันหน้าหลักแล้วเลือกเมนูตั้งค่าชุดควบคุมความชื้นดังรูปที่ 64



รูปที่ 64 เมนูไปยังหน้าตั้งค่าชุดควบคุมความชื้น

- ขั้นตอนที่ 4 ทำการเลือกเมนูกำหนดแจ้งเตือนความขึ้น
มากกว่าในกรอบสีแดง ดังรูปที่ 65



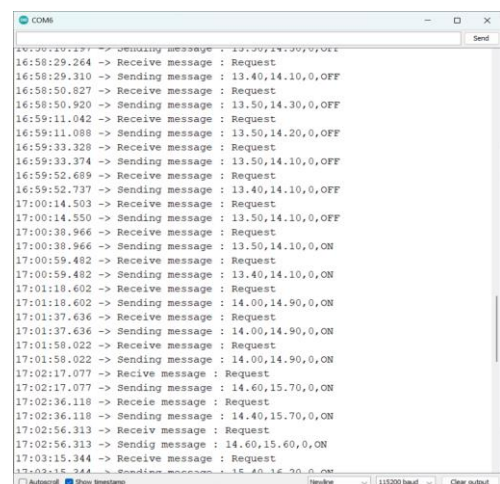
รูปที่ 65 เสนอไปยังหน้ากำหนดแจ้งเตือนค่าความชื้นสูง

- ขั้นตอนที่ 5 กำหนดค่าที่ต้องแจ้งเตือนความชื้นสูงถึง
ค่าที่ 15 %RH ดังรูปที่ 66

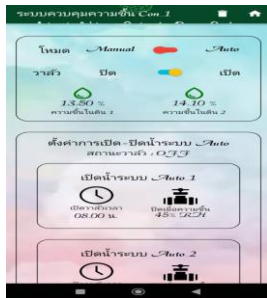


รูปที่ 66 หน้ากำหนดแจ้งเตือนค่าความชื้นสูง

- ขั้นตอนที่ 6 ทำการเข้าสู่โปรแกรม Arduino IDE เพื่อดูค่าความชื้นจากเซ็นเซอร์ ดังรูปที่ 67 จากนั้นเปิดดาวน์โหลดระบบ Manual ผ่านแอปพลิเคชัน ดังรูปที่ 68 แล้วรอแสดงผลพร้อมเมื่อค่าความชื้นจากเซ็นเซอร์สูงกว่า 15%RH จะส่งแจ้งเตือนดังรูปที่ 69



รูปที่ 67 หน้าต่าง Arduino IDE ระบบ Manual เปิดแล้ว



รูปที่ 68 เปิดวาล์วในระบบ Manual



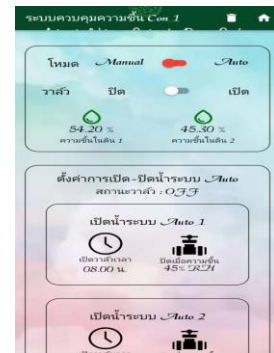
รูปที่ 69 ส่งแจ้งเตือนเมื่อค่าความชื้นมากกว่า 15 %RH

ตารางที่ 13 ผลการทดลองส่งแจ้งเตือนค่าความชื้นสูง

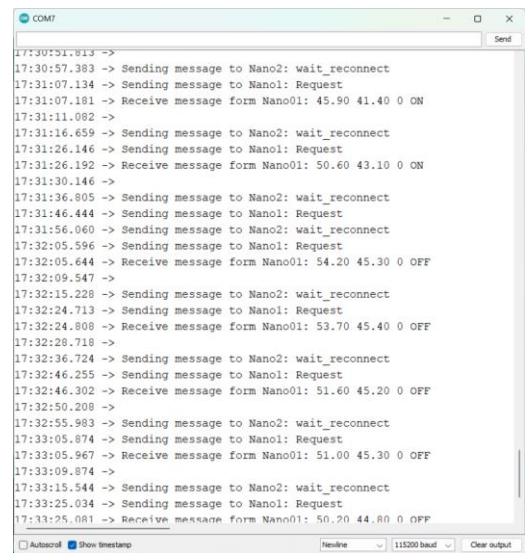
| เวลาที่บันทึก (น.) | ค่าความชื้น เซ็นเซอร์ S1 (% RH) | ค่าความชื้น เซ็นเซอร์ S2 (% RH) | ผลที่ได้ |
|--------------------|---------------------------------|---------------------------------|-----------------|
| 16.58 | 13.40 | 14.10 | ไม่ส่งแจ้งเตือน |
| 17.00 | 13.50 | 14.10 | ไม่ส่งแจ้งเตือน |
| 17.02 | 14.60 | 15.70 | ส่งแจ้งเตือน |

จากผลการทดลองแจ้งเตือนความชื้นสูง เมื่อค่าความชื้นจาก เซ็นเซอร์ S1 หรือเซ็นเซอร์ S2 ในตารางที่ 13 มีมากกว่า 15% RH ซึ่งเป็นค่าที่ผู้ใช้กำหนดบนแอปพลิเคชันระบบจะส่งแจ้งเตือนไปยังแอปพลิเคชันว่า“ค่าเตือน ค่าความชื้นสูง” นั้นหมายความว่าระบบเป็นไปตามความต้องการของผู้ใช้งาน

- ขั้นตอนที่ 7 ทดลองให้ส่งแจ้งเตือนเมื่อค่าความชื้นจาก เซ็นเซอร์ S1 หรือ S2 มีค่าน้อยกว่าค่าความชื้นที่ต้องการให้แจ้งเตือนซึ่งก็คือ 45%RH โดยเริ่มต้นทำการปิดวาล์วบนแอปพลิเคชัน ดังรูปที่ 70 แล้วเข้าโปรแกรม Arduino IDE เพื่อดูค่าความชื้นดังรูปที่ 71

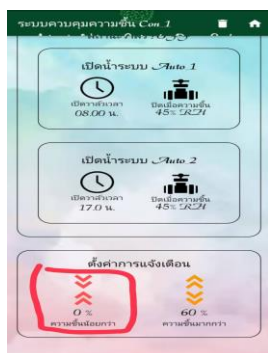


รูปที่ 70 ปิดวาล์วในระบบ Manual



รูปที่ 71 หน้าต่าง Arduino IDE ระบบ Manual ปิดวาล์ว

- ขั้นตอนที่ 8 ทำการเลือกเมนูกำหนดแจ้งเตือนความชื้น
น้อยกว่าในกรอบสีแดง ดังรูปที่ 72



รูปที่ 72 เมนูไปยังหน้ากำหนดแจ้งเตือนค่าความชื้นต่ำ

- ขั้นตอนที่ 9 กำหนดค่าที่ต้องแจ้งเตือนความชื้นต่ำ ตั้ง
ค่าที่ 45 %RH ดังรูปที่ 73



รูปที่ 73 หน้ากำหนดแจ้งเตือนค่าความชื้นต่ำ

หลังจากที่กำหนดค่าที่จะให้แจ้งเตือนเสร็จ รอให้ค่าความชื้น
จากเซ็นเซอร์ S1 หรือ S2 มีค่าน้อยกว่า 45%RH ระบบจะส่งแจ้ง
เตือนไปยังแอปพลิเคชัน ดังรูปที่ 74



รูปที่ 74 ส่งแจ้งเตือนเมื่อค่าความชื้นต่ำกว่า 45 %RH

ตารางที่ 14 ผลการทดลองส่งแจ้งเตือนค่าความชื้นต่ำ

| เวลาที่บันทึก (น.) | ค่าความชื้น เซ็นเซอร์ S1 (% RH) | ค่าความชื้น เซ็นเซอร์ S2 (% RH) | ผลที่ได้ |
|--------------------|---------------------------------|---------------------------------|-----------------|
| 17.32 | 54.20 | 45.30 | ไม่ส่งแจ้งเตือน |
| 17.33 | 51.00 | 45.30 | ไม่ส่งแจ้งเตือน |
| 17.34 | 50.20 | 44.80 | ส่งแจ้งเตือน |

จากผลการทดลองแจ้งเตือนความชื้นต่ำ เมื่อค่าความชื้นจาก
เซ็นเซอร์ S1 หรือเซ็นเซอร์ S2 ในตารางที่ 13 มีค่าน้อยกว่า 45%
RH ซึ่งเป็นค่าที่ผู้ใช้กำหนดบนแอปพลิเคชันระบบจะส่งแจ้งเตือน
ไปยังแอปพลิเคชันว่า“ค่าเตือน ค่าความชื้นต่ำ” นั้นหมายความว่า
ระบบเป็นไปตามความต้องการของผู้ใช้งาน

4.6.2 ส่งแจ้งเตือนเมื่อเปิดระบบแล้วค่าความชื้นไม่เพิ่ม

●ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เข้าสู่แอปพลิเคชันหน้าหลัก จากนั้นทำการ
เลือกเมนูตั้งค่าชุดควบคุมความชื้นที่ต้องการจะดูค่าความชื้น
เรียลไทม์ในกรอบสีแดง ดังรูปที่ 75



รูปที่ 75 เมนู ไปยังหน้าตั้งค่าชุดควบคุมความชื้น

- ขั้นตอนที่ 2 ทำการเปิดวาล์วแล้ว แต่ค่าความชื้นจากเซ็นเซอร์ S1 หรือ S2 มีค่าความชื้นที่ไม่เพิ่มขึ้นเมื่อผ่านไปแล้ว 5 นาที ระบบจะส่งแจ้งเตือนไปยังแอปพลิเคชันดังรูปที่ 76



รูปที่ 76 ส่งแจ้งเตือนเมื่อค่าความชื้นไม่เพิ่ม

ตารางที่ 15 ผลการทดลองส่งแจ้งเตือนเมื่อค่าความชื้นไม่เพิ่ม

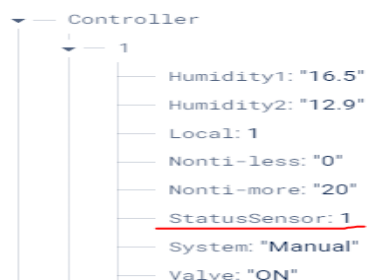
| ระยะเวลา (น.) | ค่าความชื้น เซ็นเซอร์ S1 (% RH) | ค่าความชื้น เซ็นเซอร์ S2 (% RH) | ผลที่ได้ |
|--------------------|---------------------------------------|---------------------------------------|-------------|
| 1 | 17 | 15.40 | ไม่มีการส่ง |
| 3 | 16.6 | 13.2 | ไม่มีการส่ง |
| 5 | 16.5 | 12.9 | มีการส่ง |

จากผลการทดลองแจ้งเตือนความชื้นไม่เพิ่ม เมื่อผู้ใช้ทำการเปิดวาล์วแล้วแล้วแต่ค่าความชื้นที่ไม่เพิ่มระยะเวลาผ่านไปตามช่วงเวลาต่างๆดังในตารางที่ 15 โดยสาเหตุอาจจะมาจากน้ำไม่ไหลหรือน้ำรั่วจึงจำเป็นต้องทดสอบระบบแจ้งเตือนนี้เพื่อเกิดเหตุในกรณีดังกล่าว ระบบจะไม่มีการส่งแจ้งเตือน นอกเสียจากว่าจะผ่านไปแล้ว 5 นาทีเท่านั้น เมื่อเซ็นเซอร์ตัวใดตัวหนึ่งมีค่าความชื้นไม่เพิ่มขึ้น จะมีการส่งแจ้งเตือนไปยังแอปพลิเคชันให้ผู้ใช้ไปตรวจสอบ

4.6.3 ส่งแจ้งเตือนเมื่อติดต่อเซ็นเซอร์ไม่ได้

● ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 ทำการตัดขาดการเชื่อมต่อของเซ็นเซอร์ในชุดควบคุมความชื้นแล้วไปตรวจสอบสถานะเซ็นเซอร์ในไฟร์เบสในเส้นสีแดงดังรูปที่ 71



รูปที่ 77 สถานะเซ็นเซอร์บนไฟร์เบสเมื่อขาดการเชื่อมต่อ

- ขั้นตอนที่ 2 เข้าสู่แอปพลิเคชันเพื่อตรวจสอบว่าเมื่อสถานะเซ็นเซอร์บนไฟร์เบสมีค่าเป็น 1 ระบบจะส่งแจ้งเตือนไปแอปพลิเคชันดังรูปที่ 78 จากนั้นทำการบันทึกผลลงในตารางที่ 16



รูปที่ 78 ส่งแจ้งเตือนเมื่อติดต่อเซ็นเซอร์ไม่ได้

ตารางที่ 16 ผลการทดลองส่งแจ้งเตือนเมื่อติดต่อเซ็นเซอร์ไม่ได้

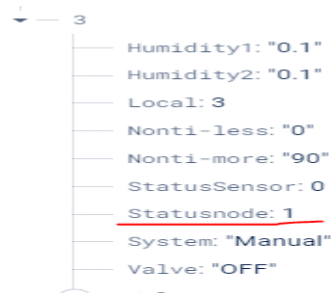
| สถานะเชื่อมต่อ เซ็นเซอร์ S1 | สถานะเชื่อมต่อ เซ็นเซอร์ S2 | ผลการทดลอง |
|--------------------------------|--------------------------------|-----------------|
| 0 | 0 | ไม่ส่งแจ้งเตือน |
| 0 | 1 | ส่งแจ้งเตือน |
| 1 | 0 | ส่งแจ้งเตือน |
| 1 | 1 | ส่งแจ้งเตือน |

จากผลการทดลองแจ้งเตือนติดต่อเซ็นเซอร์ไม่ได้ เมื่อควบคุมค่าความชื้นตัวใดตัวหนึ่งส่งสถานะการเชื่อมต่อมีค่าเท่ากับ 1 ดังตารางที่ 16 โดยสถานะการเชื่อมต่อจะมาจากไฟร์เบส จะมีการส่งแจ้งเตือนไปยังแอปพลิเคชันว่า "ติดต่อเซ็นเซอร์ไม่ได้"

4.6.4 ส่งแจ้งเตือนเมื่อติดต่อชุดควบคุมความชื้นไม่ได้

●ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 ทำการเพิ่มชุดควบคุมความชื้น Con_3 บนแอปพลิเคชันแล้วไปตรวจสอบสถานะชุดควบคุมในไฟร์เบสในเส้นสีแดงดังรูปที่ 79



รูปที่ 79 สถานะชุดควบคุมบนไฟร์เบสเมื่อขาดการเชื่อมต่อ

- ขั้นตอนที่ 2 เข้าสู่แอปพลิเคชันเพื่อตรวจสอบว่าเมื่อสถานะชุดควบคุมความชื้น Con_3 บนไฟร์เบสมีค่าเป็น 1 ระบบจะส่งแจ้งเตือนไปแอปพลิเคชันดังรูปที่ 80 และสถานะการเชื่อมต่อบนแอปพลิเคชันจะเป็น Disconnected จากนั้นทำการบันทึกผลลงในตารางที่ 17



รูปที่ 80 ส่งแจ้งเตือนเมื่อติดต่อชุดควบคุมความชื้นไม่ได้

ตารางที่ 17 ผลการทดลองส่งแจ้งเตือนเมื่อติดต่อชุดควบคุมไม่ได้

| สถานะเชื่อมต่อชุดควบคุม 1 | สถานะเชื่อมต่อชุดควบคุม 2 | ผลการทดลอง |
|---------------------------|---------------------------|-----------------|
| 0 | 0 | ไม่ส่งแจ้งเตือน |
| 0 | 1 | ส่งแจ้งเตือน |
| 1 | 0 | ส่งแจ้งเตือน |
| 1 | 1 | ส่งแจ้งเตือน |

จากผลการทดลองแจ้งเตือนติดต่อชุดควบคุมไม่ได้ เมื่อโหนด Statusnode บนไฟร์เบสของชุดควบคุมความชื้น 1 และ 2 ได้รับค่าเท่ากับ 1 จาก Main Controller ระบบจะทำการส่งแจ้งเตือนไปยังแอปพลิเคชันว่าติดต่อชุดควบคุมไม่ได้ โดยหลักการคือให้ ESP32 ตรวจสอบสถานะของชุดควบคุมความชื้นในตารางที่ 17 แล้วส่งค่าไปยัง Statusnode บนไฟร์เบส

4.6.5 ส่งแจ้งเตือนเมื่อ Main Controller ติดต่อไฟร์เบสไม่ได้

●ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 ทำการปิด Main Controller จากนั้นเข้าไปยังไฟร์เบสใน Collection ของ Main Controller แล้วสังเกตค่าเวลาในโหนด TimeStamp ดังรูปที่ 81 ไม่มีการเปลี่ยนแปลงภายใน 10 นาที ระบบจะส่งแจ้งเตือนไปยังแอปพลิเคชันดังรูปที่ 82 แล้วบันทึกผลลงในตารางที่ 18



รูปที่ 81 สถานะ Main Controller บนไฟร์เบส



รูปที่ 82 ส่งแจ้งเตือนเมื่อ Main Controller ไม่เชื่อมต่อไฟร์เบส

ตารางที่ 18 ผลการทดลองส่งแจ้งเตือนไม่เชื่อมต่อไฟร์เบส

| Time Stamp ที่บันทึก(น.) | สถานะเชื่อมต่อ Main Controller | ผลการทดลอง |
|--------------------------|--------------------------------|-----------------|
| 14.53 | ON | ไม่ส่งแจ้งเตือน |
| 14.53 | OFF | ส่งแจ้งเตือน |

จากผลการทดลอง Main Controller ไม่เชื่อมต่อ เมื่อถึงเวลาในตารางที่ 18 เมื่อ Time Stamp จาก ไฟร์เบส ไม่มีการเปลี่ยนค่าภายใน 10 นาที ระบบจะทำการส่งแจ้งเตือนไปยังแอปพลิเคชัน ซึ่งการทดลองนี้ทำเพื่อแจ้งให้ผู้ใช้งานทราบว่า Main Controller ไม่สามารถได้เชื่อมต่อกับ ไฟร์เบสในขณะนี้ ให้ผู้ใช้ทำการตรวจสอบนั่นเอง

4.7 ทดลองเพิ่มชุดควบคุมในระบบ

ในหัวข้อการเพิ่มชุดควบคุมภายในระบบจะมีการทดลองดังต่อไปนี้

4.7.1 ทดลองเพิ่มชุดควบคุมบนแอปพลิเคชัน

ในหัวข้อการทดลองนี้ทดลองเพิ่มชุดควบคุมภายในแอปพลิเคชันในกรณีต่างๆเพื่อดูว่าสามารถเพิ่มได้สูงสุดเท่าไรและในกรณีที่ตั้งชื่อชุดควบคุมจะเป็นอย่างไรและไม่สามารถเพิ่มได้ในกรณีใดบ้าง

●ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เข้าสู่ระบบแอปพลิเคชันหน้าหลัก จากนั้นทำการเลือกเมนูเพิ่มชุดควบคุมความชื้นในกรอบสีแดง ดังรูปที่ 83



รูปที่ 83 เมนูไปยังหน้าเพิ่มชุดควบคุมความชื้น

- ขั้นตอนที่ 2 ทำการกรอก ID ชุดควบคุมความชื้นที่ต้องการจะเพิ่มดังรูปที่ 84



รูปที่ 84 หน้าเพิ่มชุดควบคุมความชื้น

- ขั้นตอนที่ 3 ตรวจสอบว่ามี ID ชุดควบคุมความชื้นที่เพิ่มขึ้นมาได้หรือไม่ดังรูปที่ 85 จากนั้นทำการบันทึกผลลงในตารางที่ 19



รูปที่ 85 เพิ่มชุดควบคุมความชื้นสำเร็จ

ตารางที่ 19 ผลการทดลองเพิ่มชุดควบคุมความชื้น

| ชื่อหมายเลขที่เพิ่ม | ผลที่ได้ |
|---------------------|-------------|
| 3 | เพิ่มได้ |
| 15 | เพิ่มได้ |
| Atipatai | เพิ่มไม่ได้ |
| 16 | เพิ่มไม่ได้ |

จากผลการทดลองเพิ่มชุดควบคุมความชื้น เมื่อผู้ใช้ทำการเพิ่มชุดควบคุมความชื้นภายในแอปพลิเคชัน ในกรณีที่เพิ่มได้ จะชื่อที่จะต้องตั้งได้มีเพียง 0 – 15 มีการกรอกข้อมูลในรูปแบบของตัวอักษรดังในตารางที่ 19 จะไม่สามารถเพิ่มได้ และได้สูงสุดเพียง 16 ชุด เนื่องจากมีการใช้ คิวสวิตซ์ 4 ขา ซึ่งใช้ในการเปลี่ยนชื่อชุดควบคุมแบบมือในกรณีที่ผู้ใช้ต้องการเปลี่ยนชุดควบคุมให้เป็นนั้นๆ

4.7.2.1 ทดลองเปิดระบบ Auto ในเวลาเดียวกัน

ในหัวข้อการทดลองนี้ทดลองเพื่อการใช้ชุดควบคุมความชื้นจำนวน 2 ชุดให้รับ-ส่งสถานะของวาล์วในระบบ Auto ไปยัง Main Controller ในเวลาเดียวกันได้หรือไม่

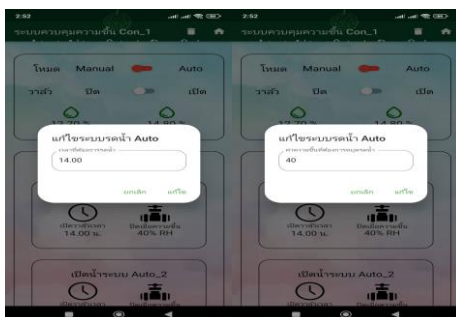
●ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เข้าสู่แอปพลิเคชันหน้าตั้งค่าชุดควบคุมความชื้นและทำการเปลี่ยนโหมดจาก Manual เป็น Auto ดังรูป 86



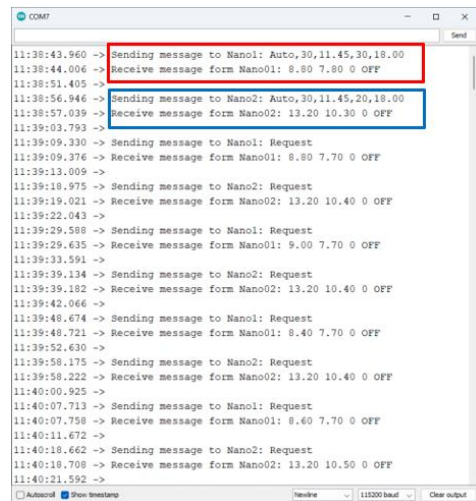
รูปที่ 86 ตั้งค่าเปิดน้ำระบบ Auto บนแอปพลิเคชัน

- ขั้นตอนที่ 2 ทำการกำหนดเวลาที่เปิดการทำงานของระบบอัตโนมัติโดยการกดที่ในไอคอนนาฬิกาและกำหนดค่าความชื้นที่ต้องการจะให้ชุดควบคุมหยุดการทำงานบนแอปพลิเคชัน โดยการกดที่ในไอคอนวาล์วน้ำ จากรูปที่ 86 หลังจากนั้นทำการกำหนดค่าที่ต้องการให้ระบบทำงานดังรูปที่ 87



รูปที่ 87 กำหนดค่าเวลาเปิดวาล์วและค่าความชื้นปิดวาล์ว

- ขั้นตอนที่ 3 ทำการเปิดใช้งาน Main Controller และชุดควบคุมความชื้น 1 และชุดควบคุมความชื้น 2 จากนั้นเข้าโปรแกรม Arduino IDE ของ Main Controller เพื่อดูว่ามีกรส่งข้อมูลหรือไม่ ดังรูปที่ 88



รูปที่ 88 หน้าจอ Arduino IDE โหมด Auto สถานะวาล์วปิด

จากรูปที่ 88 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ของ Main Controller สถานะวาล์วปิด เมื่อมีการรับ-ส่งข้อมูลไปยังชุดควบคุมความชื้นแต่ละชุดควบคุม อธิบายดังนี้

- กรอบแดง คือการส่งข้อมูลของชุดควบคุมความชื้น 1
- กรอบน้ำเงิน คือการส่งข้อมูลของชุดควบคุมความชื้น 2

รูปที่ 89 หน้าจอ Arduino IDE โหมด Auto สถานะวาล์วเปิด

จากรูปที่ 89 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ของ Main Controller เมื่อมีการส่งคำสั่งให้ระบบทำงานในโหมด Auto แล้วหรือเมื่อถึงเวลาที่ผู้ใช้งานตั้ง ตัวชุดควบคุมจะส่งสถานะ ON ไปมาให้ Main Controller อธิบายดังนี้

- กรอบแดง คือการส่งข้อมูลของชุดควบคุมความชื้น 1
- กรอบน้ำเงิน คือการส่งข้อมูลของชุดควบคุมความชื้น 2

จะทำการส่งสถานะ ON ไปยังไฟร์เบสต่อไป

- ขั้นตอนที่ 4 ตรวจสอบ หน้าจอ LCD ของชุดควบคุมความชื้น 1 ดูสถานะของวาล์วดังรูปที่ 90 และ 91 จากนั้นทำการบันทึกผลลงในตารางที่ 20



รูปที่ 90 สถานะจอ LCD ชุดควบคุม 1 ระบบ Auto เมื่อวาล์วปิด



รูปที่ 91 สถานะจอ LCD ชุดควบคุม 1 ระบบ Auto เมื่อวาล์วเปิด

ตารางที่ 20 ผลการทดลองระบบ Auto ของชุดควบคุม 1

| เวลาที่ตั้ง (น.) | ค่าความชื้นที่ตั้ง ให้หยุด(% RH) | สถานะของ วาล์ว | สถานะบน จอ LCD |
|---------------------|-------------------------------------|-------------------|-------------------|
| 14.33 | 20 | OFF | OFF |
| 14.35 | 20 | ON | ON |

จากตารางที่ 20 เมื่อสถานะของวาล์วปิดชุดควบคุมความชื้น 1 จะแสดงบนจอ LCD และเมื่อถึงเวลาที่ตั้งให้ ON ดังตารางที่ 20 ชุดควบคุมจะส่งค่าไปยัง Main Controller เพื่อให้สั่ง ON และนำไปแสดงผลบนจอ LCD

- ขั้นตอนที่ 5 ตรวจสอบ หน้าจอ LCD ของชุดควบคุมความชื้น 2 ดูสถานะของวาล์วดังรูปที่ 92 และ 93 จากนั้นทำการบันทึกผลลงในตารางที่ 21



รูปที่ 92 สถานะจอ LCD ชุดควบคุม 2 ระบบ Auto เมื่อวาล์วปิด



รูปที่ 93 สถานะจอ LCD ชุดควบคุม 2 ระบบ Auto เมื่อวาล์วเปิด

ตารางที่ 21 ผลการทดลองระบบ Auto ของชุดควบคุม 2

| เวลาที่ตั้ง (น.) | ค่าความชื้นที่ตั้ง ให้หยุด(% RH) | สถานะของ วาล์ว | สถานะบน จอ LCD |
|---------------------|-------------------------------------|-------------------|-------------------|
| 14.40 | 20 | OFF | OFF |
| 14.45 | 20 | ON | ON |

จากตารางที่ 21 เมื่อสถานะของวาล์วปิดชุดควบคุมชุดควบคุมความชื้น 2 จะแสดงบนจอ LCD และเมื่อถึงเวลาที่ตั้งให้ ON ชุดควบคุมจะส่งค่าไปยัง Main Controller เพื่อให้สั่ง ON และนำไปแสดงผลบนจอ LCD ต่อไป

จากผลการทดลองในเปิดระบบ Auto 2 ชุดควบคุมพร้อมกันเมื่อมีการเปิดน้ำระบบ Auto จากชุดควบคุมความชื้น 1 และ 2 ในเวลาเดียวกัน Main Controller จะไม่อ่านค่าพร้อมกันจึงสรุปได้ว่าชุดควบคุมความชื้น 1 และ 2 สามารถทำงานระบบ Auto ในเวลาพร้อมกันได้

4.8 การทดสอบระบบชาร์ตแบตเตอรี่และแหล่งจ่ายไฟ

ในหัวข้อทดลองนี้ทดลองระบบไฟฟ้าสำหรับชาร์ตแบตเตอรี่ใช้เวลานานแค่ไหนต่อแบตเตอรี่ 1 ชีต จากเวลา 08.00 - 16.00 น. โดยเริ่มต้น เมื่อแบตเตอรี่เต็ม ค่าแรงดัน เริ่มต้น = 13.07 V (8.00 น. เป็นค่าเวลาเริ่มต้นทดลองขณะที่แบตเตอรี่เต็ม)

● ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 ติดตั้งชุดควบคุมความชื้นและทำการวัดแรงดันไฟฟ้าแบตเตอรี่ 8.00 น. ดังรูปที่ 94 และเปิดวาล์วหลังจากวัด 30 นาที และรอวัดแรงดันในช่วงโมงต่อไป



รูปที่ 94 วัดแรงดันไฟฟ้าจากแบตเตอรี่ 8.00 น

- ขั้นตอนที่ 2 หลังจากวัดแรงดัน 15.00 น. เสร็จดังรูปที่ 95 เริ่มมีฝนตกในการวัดครั้งสุดท้ายที่เวลา 16.00 น. จึงทำให้พลังงานจากโซลาร์เซลล์ผลิตไฟได้น้อยและทำการบันทึกผลลงในตารางที่ 22



รูปที่ 95 วัดแรงดันไฟฟ้าจากแบตเตอรี่ 15.00 น.

ตารางที่ 22 ผลการทดลองระบบชาร์ตแบตเตอรี่และแหล่งจ่ายไฟ

| เวลา (น.) | ค่าแรงดัน (VDC) | หมายเหตุผลการทดลอง |
|--------------|----------------------|--|
| 8:00 น. | 13.07 | สภาพท้องฟ้ามีแดดออกเล็กน้อย กลางวันเมฆเยอะ มีการใช้งานโซลิต นอยวาล์วครึ่งชั่วโมง |
| 9:00 น. | 12.37 | สภาพท้องฟ้าเริ่มมีแดดออกเมฆ น้อยลง |
| 10:00 น. | 12.78 | สภาพท้องฟ้ามีแดดออก |
| 11:00 น. | 13.03 | สภาพท้องฟ้ามีแดดออก |
| 12:00 น. | 12.96 | สภาพท้องฟ้ามีแดดออก |
| 13:00 น. | 12.80 | สภาพท้องฟ้ามีแดดออก |
| 14:00 น. | 12.96 | สภาพท้องฟ้าเริ่มมีแดดออก เล็กน้อย เมฆเริ่มก่อตัวหนาขึ้น |
| 14:00 น. | 12.96 | สภาพท้องฟ้าเริ่มมีแดดออก เล็กน้อย เมฆเริ่มก่อตัวหนาขึ้น |
| 15:00 น. | 12.60 | สภาพท้องฟ้ามีเมฆเยอะขึ้น แดด ออกเล็กน้อย มีฝนตกนิดหน่อย |
| 16:00 น. | 12.47 | สภาพท้องฟ้ามีเมฆมากและฝนตก |

จากผลการทดลองระบบชาร์ตแบตเตอรี่และแหล่งจ่ายไฟ เมื่อทำการวัดแรงดันไฟฟ้าจากแบตเตอรี่ ในเวลา 8.00 น. เราจะได้แรงดันเริ่มต้นในขณะที่แบตเตอรี่เต็มสำหรับทดสอบ และเปิดระบบทำงานของชุดควบคุมทิ้งไว้ 1 ชม. แล้วไปวัด จากตารางที่ 22 สังเกตได้ว่าค่าแรงดันตกลงเนื่องจากการเปิดระบบต่อเนื่องและสภาพของแดดและระบบหยุดทำงานในเวลา 10.00 น.แรงดันก็จะค่อยๆเพิ่มจนถึงเมื่อ 14.00 น. เปิดระบบทำงานอีกครั้งทำให้แรงดันช่วงเวลา 15.00 น. ลดลง

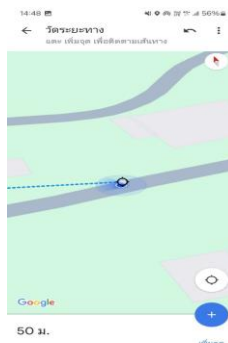
4.9 การทดสอบระยะการส่งสัญญาณล่อว่า

ในหัวข้อทดลองนี้เป็นการทดลองระยะทางในการรับ-ส่งข้อมูลด้วยลอว่า ว่าจะสามารถรับ-ส่งข้อมูลได้ในระยะไกลแค่ไหนและระยะทางมีผลต่อการรับ-ส่งข้อมูลผิดพลาดหรือไม่

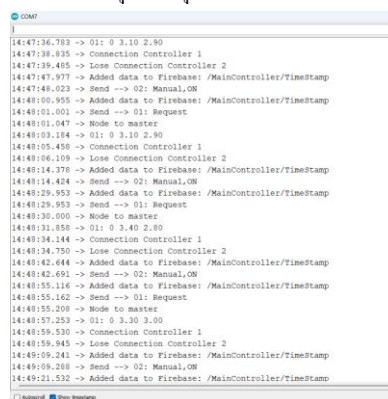
●ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เตรียมอุปกรณ์ในการทดลองมีชุดควบคุมความชื้น 1 ชุด และ Main Controller พื้นที่ในการทดลองเป็นพื้นที่เปิดโล่งและไม่มีสัญญาณรบกวนใดๆขณะทำการทดลอง

- ขั้นตอนที่ 2 เริ่มต้นทดลองที่ระยะ 50 เมตร ดังรูปที่ 96 และเปิดโปรแกรม Arduino IDE เพื่อการรับ-ส่งข้อมูล ดังรูปที่ 97 จากนั้นทำการทดลองในระยะต่อไป



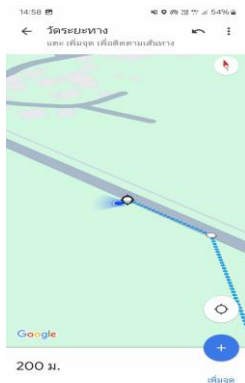
รูปที่ 96 ระยะห่างชุดควบคุมกับ Main Controller 50 เมตร



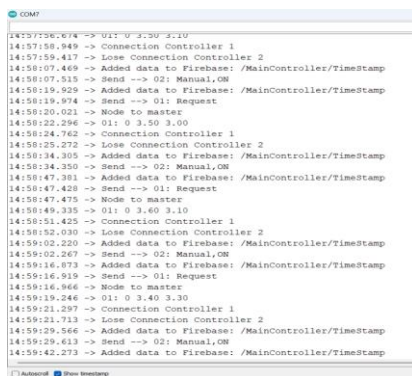
รูปที่ 97 หน้าจอ Arduino IDE รับ-ส่งข้อมูลระยะ 50 เมตร

จากรูปที่ 97 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ของ Main Controller จะสังเกตได้ว่าชุดควบคุมความชื้น 1 ยังสามารถเชื่อมต่อกับ Main Controller ได้ในระยะ 50 เมตร

- ขั้นตอนที่ 3 หลังจากที่ได้ทดลองที่ระยะ 200 เมตร ดังรูปที่ 98 และเปิดโปรแกรม Arduino IDE เพื่อการรับ-ส่งข้อมูล ดังรูปที่ 99 แล้วบันทึกผลที่ได้ลงในตารางที่ 23



รูปที่ 98 ระยะห่างชุดควบคุมกับ Main Controller 200 เมตร



รูปที่ 99 หน้าจอ Arduino IDE รับ-ส่งข้อมูลระยะ 200 เมตร

จากรูปที่ 99 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE ของ Main Controller จะสังเกตได้ว่าชุดควบคุมความชื้น 1 ยังสามารถเชื่อมต่อกับ Main Controller ได้ในระยะ 200 เมตร

ตารางที่ 23 ผลการทดลองระยะการรับ-ส่งสัญญาณล่อว่า

| เวลาที่บันทึก(น.) | ระยะทาง (เมตร) | ผลการทดลอง |
|-------------------|----------------|---------------------------------|
| 14.47 | 50 | รับ-ส่งข้อมูลได้ไม่มีข้อผิดพลาด |
| 14.49 | 100 | รับ-ส่งข้อมูลได้ไม่มีข้อผิดพลาด |
| 14.54 | 150 | รับ-ส่งข้อมูลได้ไม่มีข้อผิดพลาด |
| 14.59 | 200 | รับ-ส่งข้อมูลได้ไม่มีข้อผิดพลาด |

จากผลการทดลองระยะในการรับ-ส่งข้อมูลผู้ใช้ได้ทำการทดลองทั้งหมด 4 ระยะ ดังในตารางที่ 26 และพื้นที่ในการทดลองเป็นแบบพื้นที่โล่งและไม่มีคลื่นสัญญาณรบกวนในขณะที่ทำการทดลอง ผลที่ได้ทั้งหมด 4 ระยะสามารถรับ-ส่งข้อมูลได้ไม่มีข้อผิดพลาด ไม่มีปัญหาใดๆใน 4 ระยะทางนี้

4.10 การทดสอบใช้งานชุดควบคุมความชื้นสำหรับต้นทุเรียน

ในหัวข้อการทดลองนี้ทดลองเพื่อใช้งานชุดควบคุมความชื้นกับสำหรับต้นทุเรียนตามความต้องการของต้นทุเรียนจริงๆ โดยปกติการให้น้ำทุเรียนจะเป็น 2 ครั้งต่อวันคือช่วงเช้าและช่วงเย็น

● ขั้นตอนการทดลอง

- ขั้นตอนที่ 1 เข้าสู่แอปพลิเคชันหน้าตั้งค่าชุดควบคุมความชื้นและทำการเปลี่ยนโหมดจาก Manual เป็น Auto ดังรูปที่ 100



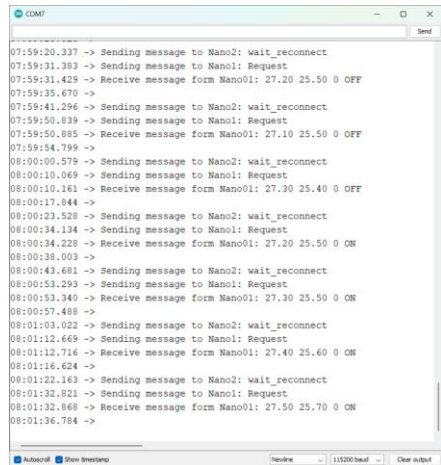
รูปที่ 100 ตั้งค่าเปิดน้ำระบบ Auto บนแอปพลิเคชัน

- ขั้นตอนที่ 2 ทำการกำหนดเวลาที่เปิดการทำงานของระบบอัตโนมัติโดยการกดที่ในไอคอนนาฬิกาให้เป็น 08.00 และ 17.00 กำหนดค่าความชื้นที่ต้องการจะให้ชุดควบคุมหยุดการทำงานเป็น 45%RH ในแอปพลิเคชันโดยการกดที่ในไอคอนวาล์วน้ำ จากรูปที่ 100 หลังจากนั้นทำการกำหนดค่าที่ต้องการให้ระบบทำงานดังรูปที่ 101



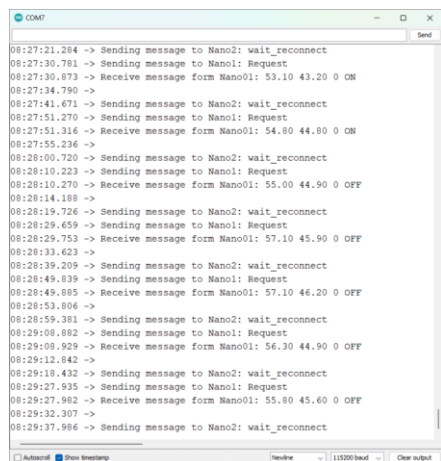
รูปที่ 101 กำหนดค่าเวลาเปิดวาล์วและค่าความชื้นปิดวาล์ว

- ขั้นตอนที่ 3 ทดสอบร่น้ำช่วงเช้าทำการเปิดโปรแกรม Arduino IDE เพื่อดูข้อความ เมื่อถึงเวลาตามที่ผู้ใช้งานตั้งไว้ซึ่งในที่นี้ก็คือ 08.00 น. ระบบจะทำการตรวจค่าความชื้นจากเซ็นเซอร์ 1 และเซ็นเซอร์ 2 ก่อน ถ้ามีค่าความชื้นน้อยกว่าค่าความชื้นที่ต้องการจะให้ปิดวาล์วอัตโนมัติซึ่งก็คือ 45%RH จะทำการเปิดวาล์ว แต่ถ้ามีค่าความชื้นมากกว่าค่าความชื้นที่ต้องการจะให้ปิดวาล์วอัตโนมัติจะไม่มีการเปิดวาล์ว โดยจากรูปที่ 102 ระบบจะเปิดทำงาน



รูปที่ 102 หน้าต่าง Arduino IDE ระบบ Auto เริ่มทำงาน 08.00 น.

จากรูปที่ 102 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE เมื่อระบบอัตโนมัติทำงาน จะสังเกตได้ว่าเมื่อถึงเวลาที่ผู้ใช้งานกำหนด 08.00 น. ระบบส่งสถานะ ON ไปให้ชุดควบคุมสั่งให้โซลินอยด์วาล์วเปิดร่น้ำและร่น้ำให้ค่าความชื้นถึง 45% RH เพื่อสั่งให้โซลินอยด์วาล์วปิด



รูปที่ 103 หน้าต่าง Arduino IDE ระบบ Auto หยุดทำงาน 08.28 น.

จากรูปที่ 103 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE เมื่อระบบอัตโนมัติหยุดทำงาน จะสังเกตได้ว่าเมื่อค่าความชื้นที่วัดได้จากเซ็นเซอร์ S1 และ S2 เกิน 45% RH ตามผู้ใช้งานกำหนดไว้ ที่เวลา 08.28 น. ระบบส่งสถานะ OFF ไปยังชุดควบคุมความชื้นเพื่อสั่งให้โซลินอยด์วาล์วปิดครัน

- ขั้นตอนที่ 4 ทดสอบระบบในช่วงเ็นทำการเปิดโปรแกรม Arduino IDE เพื่อดูข้อความ เมื่อถึงเวลาตามที่ผู้ใช้งานตั้งไว้ซึ่งในที่นี้ก็คือ 17.00 น. ระบบจะทำการตรวจค่าความชื้นเริ่มต้นจากเซ็นเซอร์ 1 และเซ็นเซอร์ 2 ก่อน ถ้ามีค่าความชื้นน้อยกว่าค่าความชื้นที่ต้องการจะให้ปิดวาล์วอัตโนมัติซึ่งก็คือ 45% RH จะทำการเปิดวาล์ว แต่ถ้ามีค่าความชื้นมากกว่าค่าความชื้นที่ต้องการจะให้ปิดวาล์วอัตโนมัติจะไม่มีการเปิดวาล์ว โดยจากรูปที่ 104 ระบบจะเปิดทำงาน

รูปที่ 104 หน้าค่า Arduino IDE ระบบ Auto เริ่มทำงาน 17.00 น.

จากรูปที่ 104 เป็นรูปในการรับ-ส่งข้อมูลบนโปรแกรม Arduino IDE เมื่อระบบอัตโนมัติทำงาน จะสังเกตได้ว่าเมื่อถึงเวลาที่ผู้ใช้งานกำหนด 17.00 น. ระบบส่งสถานะ ON ไปให้ชุดควบคุมสั่งให้โซลินอยด์วาล์วเปิดครันและรดน้ำให้ค่าความชื้นถึง 45% RH เพื่อสั่งให้โซลินอยด์วาล์วปิด

รูปที่ 105 หน้าค่า Arduino IDE ระบบ Auto หยุดทำงาน 17.32น.

จากรูปที่ 105 เป็นรูปในการรับ-ส่งข้อมูลบน โปรแกรม Arduino IDE เมื่อระบบอัตโนมัติหยุดทำงาน จะสังเกตได้ว่าเมื่อค่าความชื้นที่วัดได้จากเซ็นเซอร์ S1 และ S2 เกิน 45% RH ตามผู้ใช้งานกำหนดไว้ ที่เวลา 17.32 น. ระบบส่งสถานะ OFF ไปยังชุดควบคุมความชื้นเพื่อสั่งให้โซลินอยด์วาล์วปิดครัน

ตารางที่ 24 ผลการทดลองใช้ชุดควบคุมสำหรับต้นทุเรียน

| เวลาที่เริ่มรดน้ำ (น.) | เวลาที่หยุดครันน้ำ (น.) | ค่าความชื้นก่อนเริ่มรดน้ำ (% RH) | | ค่าความชื้นเมื่อหยุดครันน้ำ (% RH) | |
|---------------------------|----------------------------|-------------------------------------|-------|---------------------------------------|-------|
| | | S1 | S2 | S1 | S2 |
| 08.00 | 08.28 | 27.30 | 25.40 | 55.00 | 44.90 |
| 17.00 | 17.32 | 13.50 | 14.10 | 54.20 | 45.30 |

จากผลการทดลอง เมื่อถึงเวลาที่ผู้ใช้งานกำหนดระบบจะเริ่มทำการเปิดครันน้ำระยะเวลาเฉลี่ย 25-35 นาทีเพื่อให้ค่าความชื้น S1 และ S2 นั้นมากกว่า 45% RH จากปกติที่เจ้าของสวนเปิดน้ำเองแล้วจับเวลาที่ 20 นาที หลังจากได้มีการใช้งานชุดควบคุมสำหรับต้นทุเรียนค่าความชื้นจะสามารถกำหนดค่าความชื้นที่แม่นยำต่อต้นทุเรียนได้มากขึ้น โดยเจ้าของสามารถกำหนดค่าความชื้นที่เหมาะสมต่อช่วงอายุของต้นทุเรียนที่หลากหลายได้

5. สรุปผล

โครงการปริญญานิพนธ์นี้มุ่งเน้นการพัฒนาระบบให้น้ำอัตโนมัติสำหรับสวนทุเรียน โดยการใช้เทคโนโลยีให้สอดคล้องกับพื้นที่ที่เครือข่ายอินเทอร์เน็ตเข้าไม่ถึงโดยใช้สัญญาณ ลอราในการสื่อสารของระบบเครื่องในการสั่งการใช้งาน ผลลัพธ์ที่คาดหวังคือควบคุมการให้น้ำและอาจเพิ่มประสิทธิภาพในการผลิตทุเรียน

จากการทดสอบระบบให้น้ำอัตโนมัติสำหรับสวนทุเรียนพบว่า สามารถดำเนินงานได้อยู่ในขอบเขตครบถ้วน ผู้ใช้งานหรือชาวสวนสามารถใช้งานได้เป็นอย่างดีทั้งระบบสั่งการทำงานและรายละเอียดในการดูข้อมูล

6. เอกสารอ้างอิง

- [1] Jedsada Saengow. [Firebase] คืออะไร มาดูวิธีสร้าง Project และทำความรู้จักกับFirebase. (2018).[ออนไลน์].สืบค้น เมื่อ [5 พฤศจิกายน 2566].จาก<https://medium.com/jed-ng/Firebase-คืออะไร-มาดูวิธีสร้าง-project-และทำความรู้จักกับ-firebase-d48bfac67b14>
- [2] Hizoka. [Flutter] มาทำความรู้จักกับ Flutter กันเถอะ เมื่อ [7 พฤศจิกายน 2566].จาก <https://medium.com/@hizokaz/มาทำความรู้จักกับ-flutter-กันเถอะ-4dca2ad634bd>
- [3] SPM studio [NodeMCU] การใช้งาน NodeMCU ESP8266 EP.1: ทำความรู้จักเบื้องต้นและทำการติดตั้ง NodeMCU ESP8266.สืบค้น เมื่อ [7 พฤศจิกายน 2566].จาก <https://medium.com/@pattanapong.sriph/การใช้งาน-nodemcu-esp8266-ep-1-ทำความรู้จักเบื้องต้นและทำการติดตั้ง-nodemcu-esp8266-1ff4d1bc0ed1>
- [4] Kritsada Arjchariyaphat [LoRa] LoRA, LoRaWAN คืออะไร มารู้จักกันดีกว่า. สืบค้น เมื่อ [8 พฤศจิกายน 2566].จาก medium.com/deaware/lora-lorawan-คืออะไร-มารู้จักกันดีกว่า-98d20055a4ca
- [5] Sathittham (Phoo) Sangthong [Soil Moisture Sensor] การอ่านค่าความชื้นในดิน (Soil Moisture Sensor).สืบค้นเมื่อ [10 พฤศจิกายน 2566]. medium.com/sathittham/galileo-gen2-getting-started-6-การอ่านค่าความชื้นในดิน-soil-moisture-sensor-6f54e0dd92d0