**บทที่ 3**

**วิธีการดำเนินงาน**

ในบทนี้จะกล่าวถึงขั้นตอนและวิธีการดำเนินงานจัดทำโครงงานตลอดจนการออกแบบโครงสร้างและส่วนต่างๆ ของโครงงาน ซึ่งได้รับการศึกษาข้อมูลและทฤษฎีที่เกี่ยวข้องในบทที่ 2 จึงได้นำข้อมูลที่ได้มาศึกษาและวิเคราะห์ออกแบบเพื่อเลือกใช้วัสดุอุปกรณ์ที่จะนำมาทำโครงงาน ให้มีความถูกต้องเหมาะสมกับลักษณะงานและดำเนินการสร้างโครงงานให้มีประสิทธิภาพและเป็นไปตามวัตถุประสงค์ของโครงงาน

**3.1 โครงสร้างของระบบ**

โครงสร้างระบบของโครงงานนี้ประกอบด้วยส่วนต่างๆดังรูปที่ 3.1 และ รูปที่ 3.2 ซึ่งจะอธิบายโครงสร้างของระบบตามหัวข้อดังต่อไปนี้ 1. โรงเรือนเพาะเห็ด 2. Lora Communication 3.Windows Application 4. Web Application 5. Line Notify

ส่วนที่ 1 โรงเรือเพาะเห็ดส่วนนี้จะประกอบไปด้วยอุปกรณ์ฮาร์ดแวร์ที่ใช้ในระบบโดยแต่ละอุปกรณ์มีวัตถุประสงค์การนำมาใช้งานดังต่อไปนี้

* AM2315 คือเซนเซอร์ที่ใช้วัดอุณหภูมิและความชื้นภายในโรงเรือน สามารถใช้งานกับไมโครคอนโทรลเลอร์ ESP32LoRa ผ่านทาง I2C bus
* ESP32LoRa คือไมโครคอนโทรลเลอร์ที่ใช้ควบคุมภายในระบบ เช่น การอ่านค่าจากเซนเซอร์ การควบคุมการทำงานของอุปณ์ เป็นต้น
* Current Sense Resistors คือตัวต้านทานที่มีค่าความต้านทานต่ำมากใช้สำหรับการสอบการไหลของกระแส เพื่อตรวจสอบสถานะการทำงานของอุปกรณ์
* I2C LCD คือจอที่ใช้แสดงผลข้อมูลต่างๆที่หน้าโรงเรือนเพาะเห็ด เช่น ค่าอุณหภูมิ ค่าความชื้น โหมดการทำงานของระบบ สถานะการทำงานของอุปกรณ์ เป็นต้นสามารถใช้งานกับไมโครคอนโทรลเลอร์ ESP32LoRa ผ่านทาง I2C bus
* Relay คืออุปกรณ์ที่ใช้ในการควบคุมการทำงานของพัดลมและปั๊ม
* Fan พัดลมระบายอากาศใช้ในการระบายอากาศภายในโรงเรือน
* Pump ปั๊มใช้ในการพ่นหมอกเพื่อเพิ่มความชื้นภายในโรงเรือน

ส่วนที่ 2 Lora Communication ในโครงงานนี้ผู้จัดทำได้นำเทคโนโลยีการสื่อสาร Lora มาประยุกต์ใช้เพื่อเป็นตัวกลางการสื่อสารระหว่างโรงเรือนเพาะเห็ด (Mushroom Node) และพื้นที่ที่มีสัญญาณอินเทอร์เน็ต (STA Node) เพื่อแก้ปัญหาที่ตั้งของโรงเรือนไม่มีสัญญาณอินเทอร์เน็ต โดยเป็นการสื่อสารแบบ Point to Point หรือ Node-to-Node Communication

|  |
| --- |
| **รูปที่ 3.1** แสดงภาพ System Diagram |
| **รูปที่ 3.2** แสดงภาพ Block Diagram |

ส่วนที่ 3.Windows Application สามารถดูค่าสถานะและควบคุมระบบภายในโรงเรือนได้ผ่านทาง Serial Port ระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์ ใช้ภาษา Visual Basic ของ .NET Core ในการสร้าง Windows Application สามารถใช้งานได้กับไมโครคอนโทรลเลอร์ที่อยู่โรงเรือนเพาะ

เห็ด (Mushroom Node) และไมโครคอนโทรลเลอร์ที่ทำหน้าที่เป็น Web Server (STA Node) สามารถนำวินโดว์แอปพลิเคชันไปติดตั้งที่คอมพิวเตอร์เครื่องอื่นที่เป็น ระบบปฏิบัติการวินโดว์แล้วนำแอปพลิเคชันมาใช้งานได้ทันที

ส่วนที่ 4 Web Application ใช้ภาษา HTML CSS Javascript และ PHP ในการสร้าง Web Application และใช้แอปพลิเคชัน ngrok ที่ติดไว้บน Raspberry Pi ทำ Port forwarding ของ Web Server เพื่อให้สามารถใช้งานเว็บแอปพลิเคชันได้จากทุกที่ที่สามารถใช้งานอินเทอร์เน็ตได้ Web Application ส่วนที่แสดงค่าสถานะต่างๆและควบคุมการทำงานของระบบจะถูกเก็บอยู่ที่ ESP32LoRa และ Web Application ส่วนที่เกี่ยวกับฐานข้อมูล เช่น บันทึกข้อมูลลงในฐานข้อมูล นำข้อมูลในฐานข้อมูลมาแสดงบน Web Application เป็นต้น จะถูกเก็บอยู่ที่ Raspberry Pi 3 Model B เพราะใช้ Raspberry Pi เป็นฐานข้อมูล

ส่วนที่ 5 Line Notify เป็นบริการของ LINE ที่ให้สามารถส่งข้อความ การแจ้งเตือนต่าง ๆ ไปยังบัญชีหรือกลุ่มต่างๆ ผ่านทาง API ในโครงงานนี้ผู้จัดทำได้นำ Line Notify มาใช้ในการแจ้งเตือนสถานะต่างๆไปยัง Line Application เช่น การแจ้งเตือนเมื่อ Web Server เริ่มทำงานระบบจะส่ง URL ที่ได้จากโปรแกรม Ngrok เพื่อให้สะดวกต่อการเข้าใช้งาน Web Application การแจ้งเตือนเมื่ออุปกรณ์เปลี่ยนสถานะการทำงาน การแจ้งเมื่อมีการบันทึกข้อมูลลงในฐานข้อมูล เป็นต้น

**3.2 การออกแบบและสร้างโรงเรือนสำหรับเพาะเห็ด**

ในโครงงานนี้ผู้จัดทำได้ใช้โปรแกรม Google Sketchup ในการออกแบบโรงเรือนสำหรับเพาะเห็ดขนาด 200x150x210 เซนติเมตร ตามรูปที่ 3.3 โดยได้ออกแบบให้เป็นโรงเรือนแบบปิด ควบคุมอุณหภูมิภายในโรงเรือนด้วยพัดลมระบายอากาศ และควบคุมความชื้นโดยใช้ปั้มพ่นหมอก โครงสร้างของโรงเรือนเพาะเห็ดเป็นเหล็กกล่องขนาด 1x1 นิ้ว ดังรูปที่ 3.4 เพื่อโรงเรือนมีความแข็งแรงและสามารถใช้งานได้ในระยะยาว เมื่อทำการเชื่อมโครงสร้างโรงเรือนเสร็จแล้ว ขั้นตอนต่อไปคือการพ่นสีเพื่อป้องกันการเกิดสนิม ดังรูปที่ 3.5

|  |
| --- |
| รูปที่ 3.3 แสดงภาพการออกแบบโรงเรีอนที่ใช้เพาะเห็ด |

|  |
| --- |
| รูปที่ 3.4 แสดงภาพโครงสร้างของโรงเรือนเพาะเห็ด |
| รูปที่ 3.5 แสดงภาพขั้นตอนการพ่นสีโรงเรือนเพาะเห็ด |

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพขั้นตอนการติดลวดตาข่ายเพื่อป้องกันก้อนเห็ดหล่น |

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพขั้นตอนการคลุมโรงเรือนเพาะเห็ดเพื่อเก็บรักษาความชื้น |

**3.3 การออกแบบวงจรตัวต้านทานตรวจสอบกระแส**

วงจรตัวต้านทานตรวจสอบกระแส มีวัตถุประสงค์เพื่อใช้การตรวจสอบสถานการณ์ทำงาน ของ พัดลมระบายอากาศ ปั้มพ่นหมอก ว่าทำงานจริงตามที่ควบคุมหรือไม่ โดยแสดงการออกแบบวงดังรูปด้านล่าง

|  |
| --- |
| รูปที่ 3.ป แสดงภาพวงจรตัวต้านทานตรวจสอบกระแส |

จากรูปที่ 3.ป แสดงภาพของแสดงภาพวงจรตัวต้านทานตรวจสอบกระแส โดยจะประกอบด้วย LOAD คือ พัดลมระบายอากาศ หรือ ปั้มพ่นหมอก RS คือ ตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) และส่วนของวงจรขยายแบบไม่กลับเฟส (Non-inverting amplifier) ใช้เพื่อขยายแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแส โดยมีอัตราการขยายประมาณ 10 เท่า เพื่อให้ไมโครคอนโทรลเลอร์สมารถแรงดันเอาท์พุทของวงจรได้ โดยสามารถหาอัตราการขยายของวงจรได้จากสมการ

จากสูตร

จะได้

อัตราการขยาย () =

ดังนั้น

อัตราการขยาย () = 10.1 เท่า

**3.3.1 การ****คำนวณแรงดันเอาท์พุทของวงจรเมื่อใช้งานกับพัดลมระบายอากาศ**

ในโครงงานนี้ผู้จัดทำใช้พัดลมระบายอากาศขนาด 12 นิ้ว 12V 6.6A ในการควบคุมอุณหภูมิภายในโรงเรือนเพาะเห็ด ดังนั้นสามารถคำนวณแรงดันเอาท์พุทของวงจรตัวต้านทานตรวจสอบกระแส ได้จากสมการ

จากสมการด้านบน คือแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) ดังนั้นจึงต้องทราบค่า ก่อน

|  |
| --- |
| รูปที่ 3.ป แสดงภาพวงจรตัวต้านทานตรวจสอบกระแสเมื่อใช้งานกับพัดลมระบายอากาศ |

จารรูปที่ 3.ป สามารหาแรงดันตกคร่อมต้านทานตรวจสอบกระแส หรือ ได้จากกฎของโอห์มดังสมการ

I =

*ดังนั้นหาแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแสจากรูปที่ 3.ป กำหนดให้ กระแส (*I*)* = 6.6A *ความต้านทานของ*ต้านทานตรวจสอบกระแส (Rs) = 0.05 Ω

จากสูตร

I =

*จะได้*

Vin = IRs

*แทนค่า*

Vin = (6.6)(0.05)

*ดังนั้นแรงดันตกคร่องตัวต้านทานตรวจสอบกระแส (*Vin*) มีค่าเท่ากับ* 0.33V *เท่ากับ ดังนั้นสามารถคำนวณแรงดันเอาท์พุทของวงจรเมื่อใช้งานกับพัดลมระบายอากาศได้จากสมการ*

*แทนค่า*

*ดังนั้นแรงดันเอาท์พุทของวงจรเมื่อใช้งานกับพัดลมระบายอากาศ* (Vout) *เท่ากับ* 3.33V *สามารถคำนวณ**กำลัง*วัตต์ของตัวต้านทานตรวจสอบกระแส (Prs)

*จากสูตร*

P = VI

*จะได้*

Prs = Vin I

*แทนค่า*

Prs = (0.33)(6.6)

*ดังนั้นกำลัง*วัตต์ของตัวต้านทานตรวจสอบกระแส (Prs) เท่ากับ 2.178W

**3.3.2 การคำนวณแรงดันเอาท์พุทของวงจรเมื่อใช้งานกับปั้มพ่นหมอก**

ในโครงงานนี้ผู้จัดทำใช้ปั้มพ่นหมอกขนาดแรงดัน 4.8 bar / 70 PSI 12V 2A ในการควบคุมความชื้นภายในโรงเรือนเพาะเห็ด ดังนั้นสามารถคำนวณแรงดันเอาท์พุทของวงจรตัวต้านทานตรวจสอบกระแส ได้จากสมการ

จากสมการด้านบน คือแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแส (Current Sense Resistors) ดังนั้นจึงต้องทราบค่า ก่อน

|  |
| --- |
| รูปที่ 3.ป แสดงภาพวงจรตัวต้านทานตรวจสอบกระแสเมื่อใช้งานกับปั้มพ่นหมอก |

จารรูปที่ 3.ป สามารหาแรงดันตกคร่อมต้านทานตรวจสอบกระแส หรือ ได้จากกฎของโอห์มดังสมการ

I =

*ดังนั้นหาแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแสจากรูปที่ 3.ป กำหนดให้ กระแส (*I*)* = 2A *ความต้านทานของ*ต้านทานตรวจสอบกระแส (Rs) = 0.05 Ω

จากสูตร

I =

*จะได้*

Vin = IRs

*แทนค่า*

Vin = (2)(0.05)

*ดังนั้นแรงดันตกคร่อง**ตัวต้านทานตรวจสอบกระแส (*Vin*) มีค่าเท่ากับ* 0.1V *เท่ากับ ดังนั้นสามารถคำนวณแรงดันเอาท์พุทของวงจรเมื่อใช้งานกับปั้มพ่นหมอกได้จากสมการ*

*แทนค่า*

*ดังนั้นแรงดันเอาท์พุทของวงจรเมื่อใช้งานกับปั้มพ่นหมอก (*Vout*) เท่ากับ* 1.01V *สามารถคำนวณ**กำลัง*วัตต์ของตัวต้านทานตรวจสอบกระแส (Prs)

*จากสูตร*

P = VI

*จะได้*

Prs = Vrs I

*แทนค่า*

Prs = (0.1)(2)

*ดังนั้นกำลัง*วัตต์ของตัวต้านทานตรวจสอบกระแส (Prs) เท่ากับ 0.2W

**3.3.3 การใช้ไมโครคอนโทรลเลอร์ ESP32LoRa อ่านค่าแรงดันเอาท์พุทจากวงจรตัวต้านทานตรวจสอบกระแส**

**3.3.3.1 อ่านค่าแรงดันเอาท์พุทเมื่อใช้งานกับปั๊ม 12V 2A**

**3.3.3.2 อ่านค่าแรงดันเอาท์พุทเมื่อใช้งานกับพัดลม 12V 6.6A**

**3.4 การออกแบบฮาร์ดแวร์ระบบควบคุมอุณหภูมิและความชื้นภายในโรงเรือนเพาะเห็ด**

**3.5 การออกแบบการสื่อสารผ่าน Lora**

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

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพการสื่อสารแบบ Node-to-Node |

จากรูปที่ 3.ป Node แต่ละ Node จะมี IP Address เป็นของตัวเองเพื่อใช้ในการระบุตัวตน เช่น Mushroom Node มี IP Address คือ X1 และ STA Node มี IP Address คือ X0

**3.5.1 การออกแบบการสื่อสาร Lora ที่ Mushroom Node**

ในการออกแบบการสื่อสารผ่าน Lora ที่ Mushroom Node ผู้จัดทำจะขออธิบายรายละเอียดต่างๆ โดยใช้ Flowchart ประกอบการอธิบาย

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพ Flowchart การทำงานของ Mushroom Node (1) |

จากรูปที่ 3.ป แสดงการทำงานของ Mushroom Node เริ่มต้นระบบจะตรวจสอบว่าถ้ามีการเปลี่ยนแปลงของข้อมูลก็จะส่งข้อมูลการเปลี่ยนแปลง (C3) ไปยัง STA Node เช่น เมื่อผู้ใช้กดปุ่มเปลี่ยนโหมดการทำงานที่ Mushroom Node จากนั้น Mushroom Node ก็จะส่งข้อมูลการเปลี่ยนแปลงโหมดการทำงานไปยัง STA Node เพื่อให้ STA Node ทราบถึงการเปลี่ยนแปลงของข้อมูลโดยมีรูปแบบของข้อมูล (Data Format) ดังรูปที่ 3.ป

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพรูปแบบของข้อมูล (Data Format) |

จากรูปที่ 3.ป แสดงรูปแบบของข้อมูล (Data Format) ที่ส่งไปยัง STA Node เมื่อข้อมูลมีการเปลี่ยนแปลงโดยมีรายละเอียดดังต่อไปนี้

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

การส่งข้อมูลการเปลี่ยนแปลงไปยัง Node ปลายทางหรือ STA Node จะเรียกใช้ฟังก์ชัน sendUpdateData() ในการส่งข้อมูล ฟังก์ชัน sendUpdateData() เป็นฟังก์ชันที่เขียนด้วยภาษา C++ ดังรูปที่ 3.ป

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพฟังก์ชัน sendUpdateData() |

จากรูปที่ 3.ป แสดงฟังก์ sendUpdateData() ที่ใช้ส่งข้อมูลเมื่อมีการเปลี่ยนแปลงโดยรายละเอียดมีการทำงานดังต่อไปนี้ ก่อนส่งข้อมูลจะตรวจสอบค่าแรงดันตกคร่อมตัวต้านทานตรวจสอบกระแสก่อนเพื่อให้ได้ค่าสถานะการทำงานของอุปกรณ์ที่ตรงกับค่าจริง จากนั้นจะส่งข้อมูลที่มีรูปแบบ (Data Format) เหมือนรูปที่ 3.ป ตัวแปลต่างๆที่ถูกใช้ในฟังก์ชันมีรายละเอียดดังต่อไปนี้

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

จากรายละเอียดด้านบนเป็นการส่งข้อมูลการเปลี่ยนแปลงไปยัง STA Node ในกรณีที่ข้อมูลมีการเปลี่ยนแปลง แต่ถ้าหากข้อมูลไม่มีการเปลี่ยนแปลงก็จะทำการตรวจสอบเงื่อนไขข้อต่อไปคือการตรวจสอบว่าข้อมูลส่งมาจาก STA Node หรือไม่ ดังรูปที่ 3.ป ถ้าไม่มีข้อมูลที่ส่งมาก็จะกลับไปตรวจสอบการเปลี่ยนแปลงของข้อมูล ถ้ามีข้อมูลส่งมาจาก STA Node ก็จะทำการหา IP Address -ของตัวเอง (Mushroom Node IP Address) ว่าข้อมูลที่ส่งมาเป็นข้อมูลของตัวเองหรือไม่ ถ้าใช่ก็จะนำข้อมูลที่ได้มาประมวลผลต่อไป

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมส่วนที่ตรวจสอบข้อมูลที่ส่งมายัง Mushroom Node |

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพ Flowchart การทำงานของ Mushroom Node (2) |
| **รูปที่ 3.ป** แสดงภาพ Flowchart การทำงานของ Mushroom Node (3) |

จากรูปที่ 3.ป และ รูปที่ 3.ป เมื่อได้รับข้อมูลที่ส่งมาจาก STA Node แล้วขั้นตอนต่อไปคือการนำข้อมูลมาประมวลผลโดยตัวอย่างของข้อมูลที่ถูกส่งมาจาก STA Node มีดังนี้ X1M, X1Y20, X1T30, X1H80, X1J85, X1P, X1F, หรือ X1R เป็นต้น มีรายละเอียดการทำงานในแต่ละข้อมูลดังนี้

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (1) |

จากรูปที่ 3.ป ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “M” หมายความว่าต้องการเปลี่ยนโหมดการทำงานของระบบควบคุมอุณหภูมิและความชื้นโดยการทำงานคือจะตรวจสอบว่าถ้าโหมดการทำงานเป็นโหมด AUTO (ctrlMode = false) จะเปลี่ยนเป็นโหมด MANUAL (ctrlMode = true) และถ้าโหมดการทำงานเป็นโหมด MANUAL (ctrlMode = true) ก็จะเปลี่ยนโหมดการทำงานเป็นโหมด AUTO (ctrlMode = false) กล่าวคือเป็นการทำงานแบบสลับ (toggle) โหมดการทำงาน เมื่อเปลี่ยนโหมดการทำงานสำเร็จแล้วก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (2) |

จากรูปที่ 3.ป ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “Y” หมายความว่าต้องการเปลี่ยนแปลงค่าอุณหภูมิต่ำสุดที่ใช้ในการควบคุมอุณหูมิ เช่น ปัจจุบันค่าอุณหภูมิต่ำสุดคือ 20°C ถ้าได้รับ Keyword “Y” ส่งมาจาก STA Node แสดงว่าต้องการเปลี่ยนค่าอุณหภูมิต่ำสุด โดยข้อมูลที่ส่งมาอาจจะเป็น Y21 ค่าตัวเลข 2 หลักต่อจาก Keyword เป็นค่าอุณหภูมิต่ำสุดที่ต้องการเปลี่ยน เมื่อได้รับค่าที่ต้องการเปลี่ยนแล้วก็จะเปลี่ยนค่าแล้วบันที่ค่าที่เปลี่ยนลงใน EEPROM ตำแหน่งที่ 1 จากนั้นก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (3) |

จากรูปที่ 3.ป ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “T” หมายความว่าต้องการเปลี่ยนแปลงค่าอุณหภูมิสูงสุดที่ใช้ในการควบคุมอุณหูมิ เช่น ปัจจุบันค่าอุณหภูมิสูงสุดคือ 32°C ถ้าได้รับ Keyword “T” ส่งมาจาก STA Node แสดงว่าต้องการเปลี่ยนค่าอุณหภูมิสูงสุด โดยข้อมูลที่ส่งมาอาจจะเป็น T30 ค่าตัวเลข 2 หลักต่อจาก Keyword เป็นค่าอุณหภูมิสูงสุดที่ต้องการเปลี่ยน เมื่อได้รับค่าที่ต้องการเปลี่ยนแล้วก็จะเปลี่ยนค่าแล้วบันที่ค่าที่เปลี่ยนลงใน EEPROM ตำแหน่งที่ 0 จากนั้นก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (4) |

จากรูปที่ 3.ป ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “H” หมายความว่าต้องการเปลี่ยนแปลงค่าความชื้นต่ำสุดที่ใช้ในการควบคุมความชื้น เช่น ปัจจุบันค่าความชื้นต่ำสุดคือ 80% ถ้าได้รับ Keyword “H” ส่งมาจาก STA Node แสดงว่าต้องการเปลี่ยนค่าความชื้นต่ำสุด โดยข้อมูลที่ส่งมาอาจจะเป็น H79 ค่าตัวเลข 2 หลักต่อจาก Keyword เป็นค่าความชื้นต่ำสุดที่ต้องการเปลี่ยน เมื่อได้รับค่าที่ต้องการเปลี่ยนแล้วก็จะเปลี่ยนค่าแล้วบันที่ค่าที่เปลี่ยนลงใน EEPROM ตำแหน่งที่ 10 จากนั้นก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (5) |

จากรูปที่ 3.ป ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “J” หมายความว่าต้องการเปลี่ยนแปลงค่าความชื้นสูงสุดที่ใช้ในการควบคุมความชื้น เช่น ปัจจุบันค่าความชื้นสูงสุดคือ 90% ถ้าได้รับ Keyword “H” ส่งมาจาก STA Node แสดงว่าต้องการเปลี่ยนค่าความชื้นสูงสุด โดยข้อมูลที่ส่งมาอาจจะเป็น H85 ค่าตัวเลข 2 หลักต่อจาก Keyword เป็นค่าความชื้นสูงสุดที่ต้องการเปลี่ยน เมื่อได้รับค่าที่ต้องการเปลี่ยนแล้วก็จะเปลี่ยนค่าแล้วบันที่ค่าที่เปลี่ยนลงใน EEPROM ตำแหน่งที่ 9 จากนั้นก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (6) |

จากรูปที่ 3.ป ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “P” หมายความว่าต้องการ เปิด/ปิด การทำงานของปั๊ม (Pump) โดยมีรายละเอียดการทำงานดังต่อไปนี้ ขั้นแรกจะตรวจสอบว่าถ้าค่าควบคุมการทำงานของปั๊ม (pumpState) มีค่าเท่ากับ false (pumpState == false) หรือสั่งให้ปั๊มหยุดทำงาน ก็จะเปลี่ยนค่าควบคุมการทำงานของปั๊ม (pumpState) ให้มีค่าเท่ากับ ture (pumpState = true) เพื่อให้ปั๊มกลับมาทำงาน เช่นเดียวกันถ้าปัจจุบันค่าควบคุมการทำงานของปั๊ม (pumpState) มีค่าเท่ากับ ture ก็จะเปลี่ยนค่าควบคุมการทำงานของปั๊ม (pumpState) ให้มีค่าเท่ากับ false กล่าวคือเป็นการทำงานแบบสลับ (toggle) ค่าควบคุมการทำงานของปั๊ม (pumpState) เมื่อเปลี่ยนสำเร็จแล้วก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมประมวลผลข้อมูลที่ได้รับจาก STA Node (7) |

จากรูปที่ 3.ป ถ้าข้อมูลที่ได้รับจาก STA Node มี Keyword ตัวแรกเท่ากับ “F” หมายความว่าต้องการ เปิด/ปิด การทำงานของพัดลม (Fan) โดยมีรายละเอียดการทำงานดังต่อไปนี้ ขั้นแรกจะตรวจสอบว่าถ้าค่าควบคุมการทำงานของพัดลม (fanState) มีค่าเท่ากับ false (fanState == false) หรือสั่งให้พัดลมหยุดทำงาน ก็จะเปลี่ยนค่าควบคุมการทำงานของพัดลม (fanState) ให้มีค่าเท่ากับ ture (fanState = true) เพื่อให้พัดลมกลับมาทำงาน เช่นเดียวกันถ้าปัจจุบันค่าควบคุมการทำงานของพัดลม (fanState) มีค่าเท่ากับ ture ก็จะเปลี่ยนค่าควบคุมการทำงานของพัดลม (fanState) ให้มีค่าเท่ากับ false กล่าวคือเป็นการทำงานแบบสลับ (toggle) ค่าควบคุมการทำงานของพัดลม (fanState) เมื่อเปลี่ยนสำเร็จแล้วก็จะส่งข้อมูลการเปลี่ยนแปลงกลับไปยัง STA Node โดยใช้ฟังก์ชัน sendUpdateData()

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

**3.5.2 การออกแบบการสื่อสาร Lora ที่ STA Node**

ในการออกแบบการสื่อสารผ่าน Lora ที่ STA Node ผู้จัดทำจะขออธิบายรายละเอียดต่างๆ โดยใช้ Flowchart ประกอบการอธิบาย

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพ Flowchart การทำงานของ STA Node (1) |

จากรูปที่ 3.ป แสดงการทำงานของ STA Node เริ่มต้นจะส่ง Keyword “R” ไปยัง Mushroom Node เพื่อขออัปเดตข้อมูลก่อนเริ่มระบบ โดยข้อมูลที่ส่งไปมีรูปแบบดังนี้ X1R เมื่อ X1 คือ IP Address ของ Mushroom Node เพื่อให้ทราบว่าต้องการส่งข้อมูลไปยัง Mushroom Node และ Keyword “R” คือ Keyword ที่ส่งไปเพื่อให้ Mushroom Node ทราบว่าต้องการอัปเดตข้อมูล

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

* M คือ Keyword ที่ส่งเมื่อกดปุ่มเปลี่ยนโหมด
* P คือ Keyword ที่ส่งเมื่อกดปุ่มเปิดหรือปิดปั๊ม
* F คือ Keyword ที่ส่งเมื่อกดปุ่มเปิดหรือปิดพัดลม
* Y คือ Keyword ที่ส่งเมื่อเปลี่ยนค่าอุณหภูมิต่ำสุดตามดัวยตัวเลขจำนวน 2 หลัก เช่น Y25 เมื่อ 25 คือค่าที่ต้องการเปลี่ยน
* T คือ Keyword ที่ส่งเมื่อเปลี่ยนค่าอุณหภูมิสูงสุดตามดัวยตัวเลขจำนวน 2 หลัก เช่น T32 เมื่อ 32 คือค่าที่ต้องการเปลี่ยน
* H คือ Keyword ที่ส่งเมื่อเปลี่ยนค่าความชื้นต่ำสุดตามดัวยตัวเลขจำนวน 2 หลัก เช่น H80 เมื่อ 80 คือค่าที่ต้องการเปลี่ยน
* J คือ Keyword ที่ส่งเมื่อเปลี่ยนค่าความชื้นสูงสุดตามดัวยตัวเลขจำนวน 2 หลัก เช่น J90 เมื่อ 90 คือค่าที่ต้องการเปลี่ยน

จากรายละเอียดด้านบนเป็นข้อมูลจาก Serial Port เมื่อได้รับข้อมูลจาก Serial Port แล้วก็จะส่งข้อมูลต่อไปยัง Mushroom Node ทันทีดังรูปที่แสดงด้านล่าง

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมส่วนที่รับข้อมูลจาก Serial Port |

จากรูปที่ 3.ป จะรับข้อมูลจาก Serial Port แล้วส่งข้อมูลไปยัง Mushroom Node ด้วยฟังก์ชัน loraSend() โดยข้อมูลที่ส่งจะมีรูปแบบ (Data Format) ดังนี้ X1M เมื่อ X1 คือ IP Address ของ Mushroom Node และ M คือ Keyword ที่ส่งไปเพื่อควบคุมการทำงานของระบบดังรายละเอียดด้านบน จากนั้นจะล้างข้อมูลที่ได้รับจาก Serial Port

เมื่อระบบทำงานผ่านขั้นตอนต่างๆ ด้านบนมาแล้วก็ขั้นตอนต่อไปจะทำการตรวจสอบเงื่อนไขข้อต่อไปคือการตรวจสอบว่าข้อมูลส่งมาจาก Mushroom Node หรือไม่ ดังรูปที่ 3.ป ถ้าไม่มีข้อมูลที่ส่งมาก็จะส่งข้อมูลเพื่อขออัปเดตข้อมูล (C3) ทุกๆ 60 วินาที ถ้ามีข้อมูลส่งมาจาก Mushroom Node ก็จะทำการหา IP Address -ของตัวเอง (STA Node IP Address) ว่าข้อมูลที่ส่งมาเป็นข้อมูลของตัวเองหรือไม่ ถ้าใช่ก็จะนำข้อมูลที่ได้มาประมวลผลต่อไป

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมส่วนที่ตรวจสอบข้อมูลที่ส่งมายัง STA Node |

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพ Flowchart การทำงานของ STA Node (2) |

เมื่อได้รับข้อมูลที่ส่งมาจาก Mushroom Node แล้วก็จะนำข้อมูลที่ได้มาเก็บไว้ในตัวแปลต่างๆ โดยใช้ฟังก์ substring() เพื่อนำข้อมูลที่ได้ไปใช้งานบน Web Application และ Windows Application ตัวอย่างรูปแบบของข้อมูล (Data Format) ที่ส่งมาจาก Mushroom Node ดังรูปที่ 3.ป

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพรูปแบบของข้อมูล (Data Format) ที่ส่งมาจาก Mushroom Node |

จากรูปที่ 3.ป แสดงรูปแบบของข้อมูลที่ส่งมาจาก Mushroom Node เมื่อได้รับข้อมูลก็จะนำข้อมูลที่ได้มา substring() ดังรูปที่ 3.ป

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพการใช้ฟังก์ชัน substring() กับข้อมูลที่ได้รับจาก Mushroom Node |

จากรูปที่ 3.ป แสดงการใช้ฟังก์ชัน substring() เพื่อนำข้อมูลที่ได้รับมาเก็บไว้ในตัวแปลที่ถูกต้องโดยมีรายละเอียดดังต่อไปนี้

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

เมื่อจัดการกลับข้อมูลที่ส่งมาเรียบร้อยแล้วขั้นตอนต่อไปคือการส่ง Keyword เพื่อขออัปเดตข้อมูลทุกๆ 60 วินาที ดังรูปที่ 3.ป

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโปรแกรมที่ใช้ในการส่ง Keyword ทุก 60 วินาที |

จากรูปที่ 3.ป มีหลักการทำงานคือ STA Node จะส่ง Keyword “R” ทุกๆ 60 วินาทีโดยใช้ฟังก์ชัย millis() ในการนับเวลา ซึ่งตัวแปล node1 มีค่าเท่ากับ X1 คือ IP Address ของ Mushroom Node รูปแบบข้อมูล (Data Format) คือ X1R

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

จากรุปที่ 3.ป เมื่อ Keyword เพื่อขออัปเดตข้อมูลทุกๆ 60 วินาทีแล้ว ขั้นตอนต่อไปคือการตรวจสอบว่ามีการควบคุมการทำงานของระบบจากเว็บแอปพลิเคชัน (Web Application) หรือไม่ โดยมีรายละเอียดการทำงานดังต่อไปนี้

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

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพ Flowchart การทำงานของ STA Node (4) |

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

**3.6 การออกแบบและสร้างเว็บแอปพลิเคชัน**

**3.7 การออกแบบและสร้างเวินโดว์แอปพลิเคชัน**

**3.8 การออกแบบฐานข้อมูลและการบันทึกข้อมูล**

**3.8.1 การออกแบบฐานข้อมูล**

ในการออกแบบฐานข้อมูลในโครงงานนี้จะใช้แค่ 1 ตาราง (entity) เท่านั้นเพื่อใช้เก็บข้อมูลต่างๆที่เกี่ยวข้องเช่น ค่าอุณหภูมิ (temp) ค่าความชื้น (humi) ค่าอุณหภูมิต่ำสุด (temp\_limit\_min) ค่าอุณหภูมิสูงสุด (temp\_limit\_max) ค่าความชื้นต่ำสุด (humi\_limit\_min) ค่าความชื้นสูงสุด (humi\_imit\_max) โหมดการทำงาน (ctrl\_mode) สถานะการทำงานของปั๊ม (pump\_state) สถานการณ์ทำงานของพัดลม (fan\_state) และเวลาที่บันทึกข้อมูล (save\_time) ดังรูปที่ 3.ป โดยตารางมีชื่อว่า SensorData

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพโครงสร้างของฐานข้อมูล |

จากรูปที่ 3.ป โครงสร้างของฐานข้อมูลโดยมี Primary key (PK) คือ id สามารถบันทึกข้อมูลเป็นตัวเลขจำนวนเต็มแบบไม่มีเครื่องหมายไม่เกิน 6 หลัก เป็นลำดับของข้อมูลแบบเพิ่มขึ้นอัตโนมัติ (auto increment) มี save\_time attribute ใช้ในการเก็นค่าวันเวลาที่บันทึกข้อมูล และ #temp #humi #temp\_limit\_min #temp\_limit\_max #humi\_limit\_min #humi\_limit\_max #ctrl\_mode #pump\_state #fan\_state attribute เป็น attribute ที่ใช้ในการเก็บข้อมูลต่างๆภายในระบบ สามารถบันทึกข้อมูลประเภทตัวอักษรรวมกันไม่เกิน 10 ตัวอักษร

**ตางรางที่ 3.ป** Data Dictionary SensorData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Example | Key |
| id | ลำดับของข้อมูล | int(6) | 1 | PK |
| temp | อุณหภูมิ | varchar(10) | 32.10 |  |
| humi | ความชื้น | varchar(10) | 80.70 |  |
| temp\_limit\_min | อุณหภูมิต่ำสุด | varchar(10) | 20 |  |
| temp\_limit\_max | อุณหภูมิสูงสุด | varchar(10) | 29 |  |
| humi\_limit\_min | ความชื้นต่ำสุด | varchar(10) | 80 |  |
| humi\_limit\_max | ความชื้นสูงสุด | varchar(10) | 85 |  |
| ctrl\_mode | โหมดการทำงาน | varchar(10) | AUTO |  |
| pump\_state | สถานะการทำงานของปั๊ม | varchar(10) | ON |  |
| fan\_state | สถานะการทำงานของพัดลม | varchar(10) | ON |  |
| save\_time | วันเวลาที่บันทึกข้อมูล | timestamp | 2020-09-08 18:00:00 |  |

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพตัวอย่างข้อมูลเก็บอยู่ภายในฐานข้อมูล |

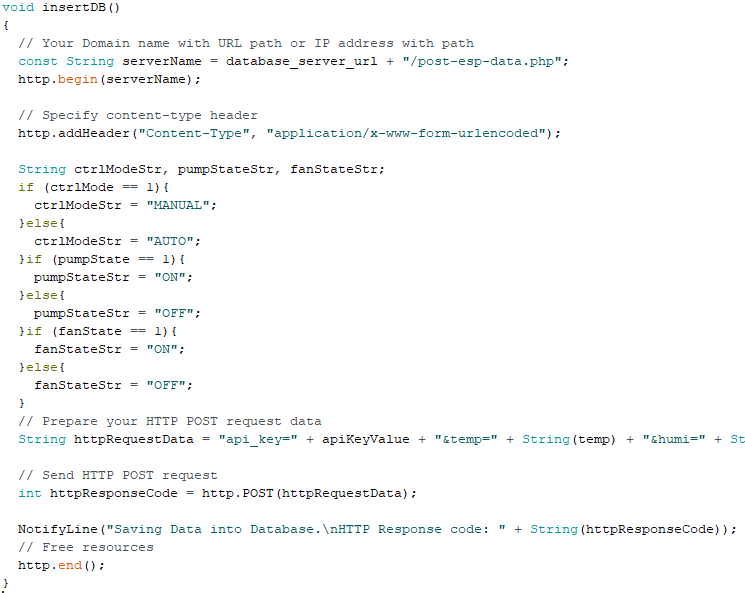
**3.8.2 การบันทึกข้อมูลลงในฐานข้อมูล**

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

|  |
| --- |
| รูปที่ 3.ป แสดงภาพ Flowchart ขั้นตอนการบันทึกข้อมูลลงในฐานข้อมูล |
| รูปที่ 3.ป แสดงภาพการบันทึกข้อมูลลงในฐานข้อมูลทุกชั่วโมง |

จากรูปที่ 3.ป จะเป็นการดึงค่าเวลาจริงมาจากเซิร์ฟเวอร์ NTP (Network Time Protocol) มาใช้ในการตรวจสอบเพื่อบันทึกข้อมูล โดยจะตรวจสอบเวลาในหลักนาทีและวินาที เช่น ถ้าเวลาในหน่วยนาทีมีค่าเท่ากับ 00 และเวลาในหน่วยวินาทีมีค่าเท่ากับ 00 ก็จะทำการบันทึกข้อมูลโดยเรียกใช้ฟังก์ชัน insertDB() ก็จะสามารถบันทึกข้อมูลทุกๆชั่วโมงได้แล้ว

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



**รูปที่ 3.ป** แสดงภาพฟังก์ชันที่ใช้บันทึกข้อมูลลงในฐานข้อมูล

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

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

|  |
| --- |
| **รุปที่ 3.ป** แสดงภาพไฟล์ post-esp-data.php บางส่วน |

จากรูปที่ 3.ป แสดง code บางส่วนของไฟล์ post-esp-data.php โดยมีตัวแปลที่มีชื่อว่า $api\_key\_value ที่มีค่าเท่ากับ “tPmAT5Ab3j7F9” เป็นค่าที่ใช้ตรวจสอบว่าข้อมูลที่ส่งมา มาจากแหล่งที่มีที่ถูกต้อง เช่น ถ้า ESP32 LoRa ต้องการที่จะบันทึกข้อมูลลงในฐานข้อมูลจะต้องส่งค่า $api\_key\_value มาด้วยผ่าน HTTP POST

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

**3.9 การออกแบบ Use Case Diagram**

|  |
| --- |
| **รูปที่ 3.ป** แสดงภาพ Use Case Diagram |

รายละเอียด Use Case Diagram จากตาราง 3.ป ถึงตาราง 3.ป ด้านล่างต่อไปนี้แสดงถึงรายละเอียดการใช้งานของแต่ละ Use Case ตามรูปแบบของ UML Use Case Description

ตาราง 3.ป Login Use Case

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

ตาราง 3.ป Logout Use Case

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

ตาราง 3.ป View Dashboard Use Case

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

ตาราง 3.ป Control Use Case

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

ตาราง 3.ป Setting Use Case

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

ตาราง 3.ป View Data Use Case

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

ตาราง 3.ป View Status Use Case

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