เสาไฟทางเดินตรวจอากาศอัจฉริยะ

SMART AERIAL WALKWAY LIGHTS

ธีรสิทธิ์ โท้ทอง

THIRASIT THOTHONG

พีรพัฒน์ สุขเกษม

PEERAPAT SUKKASEM

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART AERIAL WALKWAY LIGHTS

THIRASIT THOTHONG

PEERAPAT SUKKASEM

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เสาไฟทางเดินตรวจอากาศอัจฉริยะ	
ชื่อนักศึกษา	นายธีรสิทธิ์ โท้ทอง	รหัสนักศึกษา 62010446
	นายพีรพัฒน์ สุขเกษม	รหัสนักศึกษา 62010654
อาจารย์ที่ปรึกษาปริญญานิพนธ์	รศ.ดร.อรรถสิทธิ์ หล่าสกุล	
	ดร.ธนวิชญ์ อนุวงศ์พินิจ	
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	สาขาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	2565	

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....

(รศ.ดร.อรรถสิทธิ์ หล่าสกุล)

.....

(ดร.ธนวิชญ์ อนุวงศ์พินิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เสาไฟทางเดินตรวจอากาศอัจฉริยะ	
ชื่อนักศึกษา	นายธีรสิทธิ์ โท้ทอง	รหัสนักศึกษา 62010446
	นายพีรพัฒน์ สุขเกษม	รหัสนักศึกษา 62010654
อาจารย์ที่ปรึกษาปริญญานิพนธ์	รศ.ดร.อรรถสิทธิ์ หล่าสกุล	
	ดร.ธนวิชญ์ อนุวงศ์พินิจ	
ปริญญา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	สาขาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมคอมพิวเตอร์	
ปีการศึกษา	2565	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการออกแบบระบบและพัฒนาชุดตรวจวัดสภาพอากาศของเครื่องมือตรวจวัด (Sensor) มาตรฐาน RS485 เพื่อทำงานร่วมกับเสากระจายเสียงผ่าน VoIP เดิม โดยทำการอ่านค่าจากเครื่องมือวัด (Sensor) ด้วยอุปกรณ์ควบคุมขนาดเล็ก (Microcontroller) และทำการส่งค่าที่ได้ไปเก็บข้อมูลบนฐานข้อมูล (Database) และแสดงผลได้ ผ่านการสื่อสาร LoRaWAN โดยระบบจะทำการติดตั้งในรูปแบบของเสาไฟทางเดิน เพื่อเป็นการพัฒนาเสาไฟทางเดินทั่วไปให้ดีขึ้น และมีการเลือกใช้ microcontroller และการส่งสัญญาณแบบ LoRaWAN เพื่อให้มีการใช้พลังงานต่ำและลดต้นทุนการผลิต ทำให้มีการเข้าถึงการใช้งานได้ง่ายขึ้นโดยสามารถวัดสภาพอากาศได้หลากหลาย ทั้งปริมาณฝุ่น PM2.5 PM10 ปริมาณน้ำฝน อุณหภูมิ ความชื้น ตำแหน่งที่ตั้งของเสา ความเร็วและทิศทางลมได้ และนำข้อมูลที่วัดได้เหล่านั้นส่งผ่านคลื่นวิทยุ LoRa ไปยัง Gateway เพื่อนำข้อมูลเข้าสู่ระบบเก็บข้อมูลใน AWS และนำข้อมูลไปแสดงผลบน Dashboard ของ Grafana เพื่อให้ผู้ใช้สามารถเข้ามาดูสภาพอากาศในบริเวณเสาผ่านอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	smart aerial walkway lights
Student	Thirasit Thothong
	Peerapat Sukkasem
Thesis Advisor	Assoc.Prof.Dr.Attasit Lasakul
	Dr.Thanavit Anuwongpinit
Degree	Bachelor of Engineering
Program	Information Engineering
Year	2022

ABSTRACT

This thesis is the design and development of an RS485 Sensor for work with VoIP Telecom that we make before. This thesis work by reading Sensor with a microcontroller sending data of value to a database and showing it on a dashboard with LoRaWAN communication protocol. This system is installed on the walkway light to upgrade the walkway light to a better one, use a microcontroller and LoRaWAN for low energy usage and reduce the cost to build, so it will be easier to access. Able to measure a variety of weather conditions including dust, PM2.5, PM10, rainfall, temperature, humidity, location of the poles wind speed and direction These measured data are then sent to the Gateway with LoRa radio waves, and the data is stored in the AWS storage system and displayed on the Grafana Dashboard so that users can view the weather data in the area over the Internet.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงมิอาจสำเร็จได้ ถ้าปราศจากความร่วมมือย่างยิ่งจากทุกฝ่ายที่เกี่ยวข้อง ซึ่งผู้จัดทำใคร่ขอบคุณทุก ๆ ท่านที่ได้มีส่วนช่วยเหลือ แนะนำ ให้คำปรึกษา ในทุก ๆ ด้าน

ขอขอบพระคุณ รศ.ดร.อรรถสิทธิ์ หล่าสกุล และ ดร.ธนวิชญ์ อนุวงศ์พินิจ อาจารย์ประจำภาควิชาวิศวกรรมสารสนเทศ อาจารย์ที่ปรึกษาทั้งสองท่านได้ช่วยเหลือ ให้คำปรึกษา และให้ข้อเสนอแนะที่เป็นประโยชน์ รวมถึงเอื้อเฟื้อข้อมูล และทรัพยากรต่าง ๆ ในการจัดทำโครงงาน จึงทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

คุณประโยชน์อันพึงมีจากโครงงานนี้ ทางผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่านไว้ ณ โอกาสนี้

ธีรสิทธิ์ โท้ทอง

พีรพัฒน์ สุขเกษม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

เนื่องจากปัจจุบัน โลกได้มีการพัฒนาเข้าสู่สังคมระบบ IOT มากขึ้น หลาย ๆ อย่างสามารถทำผ่านโทรศัพท์เครื่องเดียวได้ รวมถึงรอบตัวเรามีสิ่งอำนวยความสะดวกมากมายแต่ใช้ประโยชน์ได้น้อยหรือเข้าถึงได้ยาก อย่าง เช่น เสาไฟทางเดิน ซึ่งเสาไฟในปัจจุบันมีราคาที่สูงแต่ใช้ประโยชน์ได้น้อยเพียงส่องสว่างเท่านั้น หรือจะเป็นเสาวัดสภาพอากาศ หรือสถานีวัดสภาพอากาศที่มีก็เข้าถึงข้อมูลได้ยาก บุคคลทั่วไปไม่สามารถเข้าถึงได้ง่าย ๆ

ทางผู้ดำเนินงานจึงเล็งเห็นการพัฒนาระบบเสาไฟทางเดินเดิม โดยรวมกับสถานีตรวจวัดสภาพอากาศ ให้สามารถตรวจสภาพอากาศ และสภาพแวดล้อมต่าง ๆ ได้ เพื่อให้การตรวจวัดสภาพอากาศ สามารถทำได้ง่ายขึ้น ผู้คนทั่วไปสามารถเข้าถึงง่ายขึ้น และยังมีการบันทึกค่าไว้ แบ่งเป็นข้อมูลเฉพาะของเขตได้ เพื่อไว้ใช้สำหรับการประมวลข้อมูล และคาดการณ์สภาพอากาศในอนาคต โดยติดตั้งในรูปของเสาไฟทางเดินเพื่อให้สามารถใช้ประโยชน์ได้ทั่วไป และเพื่อให้สอดคล้องกับสภาพสังคมที่เข้าสู่ระบบ IOT ที่สามารถเขาถึงข้อมูลเฉพาะพื้นที่ได้ง่าย โดยจะออกแบบให้ต้นทุนต่ำสามารถพัฒนาได้ง่าย ติดตั้งได้ง่าย และสร้างประโยชน์จากเทคโนโลยีเดิมให้มีความคุ้มค่ายิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1. สร้างระบบที่สามารถตรวจสภาพอากาศ และสภาพแวดล้อมต่าง ๆ ได้
2. ให้ผู้คนทั่วไปสามารถเข้าถึงข้อมูลทางสภาพอากาศได้ง่ายขึ้น
3. เพื่อพัฒนาระบบการเก็บข้อมูลและแสดงผลการตรวจอากาศผ่าน Internet

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบที่พัฒนาขึ้น สามารถนำไปใช้งานจริงได้ ในมหาวิทยาลัย
2. สามารถเก็บและแสดงผลข้อมูลที่เก็บได้ อย่างมีคุณภาพ
3. ได้รับความรู้เกี่ยวกับการพัฒนาระบบนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 **ขอบเขตโครงงาน**

1. พัฒนาเสาให้สามารถตรวจอากาศได้
2. ปรับปรุงโครงสร้างเสาเดิมให้ดีขึ้น และรองรับการตรวจอากาศได้
3. พัฒนาระบบการเก็บข้อมูลและการแสดงผลการตรวจอากาศผ่านอินเทอร์เน็ต

1.5 **อุปกรณ์ที่ใช้ในการทำงาน**

5.1 ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์สำหรับพัฒนาโปรแกรม ที่มีการต่อเชื่อมกับเน็ตเวิร์ค จำนวน 1 เครื่อง
- Microcontroller ที่รองรับการทำงานผ่าน RS485 สำหรับติดตั้งที่เสา จำนวน 1 ชิ้น
- Microcontroller ควบคุมการส่งค่าผ่าน LoRa จำนวน 1 ชิ้น
- ตัวส่งสัญญาณ LoRa จำนวน 1 ชิ้น
- เครื่องมือวัดปริมาณน้ำฝน จำนวน 1 ชิ้น
- เครื่องมือวัดมลพิษทางอากาศ จำนวน 1 ชิ้น
- เครื่องมือวัดความเร็วลม จำนวน 1 ชิ้น
- เครื่องมือวัดทิศทางลม จำนวน 1 ชิ้น
- ฐานข้อมูล (สามารถใช้ฐานข้อมูลออนไลน์ได้) จำนวน 1 เครื่อง

5.2 ซอฟต์แวร์

- ซอฟต์แวร์สำหรับการเขียนโปรแกรมใส่ Microcontroller (Arduino)
- ซอฟต์แวร์จัดการฐานข้อมูล (PhpMyAdmin)
- ซอฟต์แวร์ทำหน้าแสดงผล (Dashboard)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 **ขั้นตอนการดำเนินงาน**

1. กำหนดหัวข้อและขอบเขต
2. ศึกษาเนื้อหาที่เกี่ยวข้อง
   - ศึกษาเรียนรู้การทำงานของเครื่องมือวัด (Sensor)
   - ศึกษาเรียนการทำงานของไมโครคอนโทรลเลอร์ (Microcontroller)
   - ศึกษาเรียนรู้การใช้งานฐานข้อมูล (Database)
3. พัฒนาระบบส่วนต่างๆ
   - การอ่านค่าจากเครื่องมือวัด (Sensor)
   - การสื่อสารกับฐานข้อมูล (Database)
   - การทำงานแสดงผล
4. บันทึกผลการทำงานและสรุปผลการทดลอง

1.7 **คำจำกัดความ**

ตาราง 1.1 คำจำกัดความ

คำที่ใช้	**ความหมาย หรือหน้าที่การเรียกใช้**
sensor	เครื่องมือวัด
Microcontroller	อุปกรณ์ควบคุมขนาดเล็ก
TTGO, LoRa32	Microcontroller สำหรับนำมาส่งข้อมูลผ่าน LoRa
Server	เครื่องอุปกรณ์แม่ข่าย นำมาเก็บข้อมูล
Database	ฐานข้อมูล เรียกแทนระบบการเก็บข้อมูล ที่ประกอบด้วย Server, โปรแกรมจัดการฐานข้อมูล
Dashboard	หน้าจอสรุปข้อมูล แสดงผลค่าที่จัดเก็บใน Database
Grafana	โปรแกรมจัดการ Dashboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำที่ใช้	**ความหมาย หรือหน้าที่การเรียกใช้**
Gateway, LoRa gateway	อุปกรณ์รับสัญญาณผ่าน LoRa
voltage regulator, power supply	อุปกรณ์สำหรับจ่ายไฟฟ้า หรือแปลงไฟฟ้า จากกระแสสลับทั่วไป เป็นกระแสตรงที่ใช้สำหรับอุปกรณ์อื่นๆ
Function	ฟังก์ชัน, ชุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะขอกล่าวถึงทฤษฎีที่เกี่ยวข้องในการทำงาน ทั้งการทำงานของระบบเครื่องมือวัด การสื่อสาร และการทำงานในระบบฐานข้อมูล

2.1 ส่วนการทำงานกับเครื่องมือวัด

2.1.1 Serial Communication

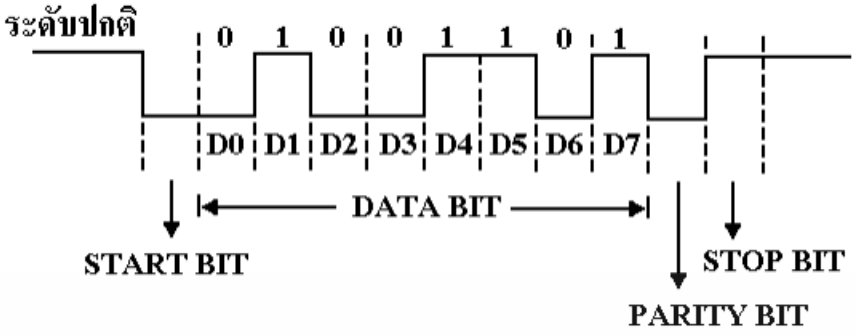
การสื่อสารแบบอนุกรม (Serial Communication) เป็นการสื่อที่รับส่งข้อมูลที่ละบิต แทนการรับส่งแบบขนานที่รับส่งพร้อมกันทุกบิต มีข้อดีคือใช้สายที่น้อยกว่าแบบขนาน แต่มีข้อเสียที่จะใช้เวลามากกว่าแบบบขนาน โดย การสื่อสารมี 3 รูปแบบ

1) Simplex สื่อสารทางเดียว
2) Half-duplex สื่อสารสองทาง แต่จะต้องสลับกันสื่อสารไปกลับ
3) Full-duplex เป็นการสื่อสารสองทางที่สามารถสื่อสารพร้อมกันได้ทั้งสองฝั่ง

สำหรับการสื่อสารแบบอนุกรมมี 2 ลักษณะ

1) Synchronous เป็นการส่งข้อมูลเป็นบล็อก ครั้งละหลายๆ ไบต์ใช้สัญญาณนาฬิกาเป็นตัวช่วยในการทำงานขของตัวรับส่งให้สอดคล้องกัน
2) Asynchronous เป็นการส่งข้อมูลที่ละ 1 ไบต์ 8 บิต โดยรูปแบบ Stream bit มีความหมายดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 bit meaning

Start bit บอกจุดเริ่มต้นของข้อมูล มีขนาด 1 บิต

Data bit แทนค่าข้อมูล มีขนาดได้ 5 ถึง 8 บิต

Parity bit บิตสำหรับใช้ตรวจสอบความผิดพลาดของข้อมูล มีขนาดได้ 0 ถึง 1 บิต

Stop bit บิตบอกจุดสิ้นสุดข้อมูล มีขนาดได้ 1, 1.5 และ 2 บิต

โดยระยะเวลาการส่งข้อมูลและบิตขึ้นอยู่กับการใช้งาน เป็นจำนวน bit/second โดยมีค่ามาตรฐานอยู่หลายช่วง เช่น 2400 4800 9600 19200 115200 เป็นต้น

2.1.2 RS485

Recommended Standard no. 485 เป็นมาตรฐานการรับ/ส่งข้อมูลแบบ Half-duplex สำหรับการรับส่งข้อมูลของ RS485 จะใช้สายเพียงแค่ 2 เส้นคือ A และ B เป็นตัวบอกสัญญาณดิจิตอล โดยใช้ค่าความต่างของแรงดันระหว่าง A และ B

เมื่อแรงดัน Va – Vb ได้แรงดันน้อยกว่า -200 mV คือสัญญาณดิจิตอล 1

เมื่อแรงดัน Va – Vb ได้แรงดันมากกว่า +200 mV คือสัญญาณดิจิตอล 0

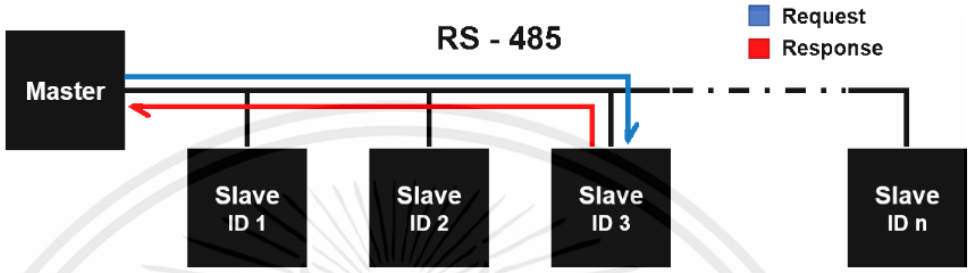
การทำงานแบบระบบ network จะเชื่อมต่อกันได้มากสุดถึง 32 ตัว โดยมี 1 ตัวทำหน้าที่จัดการลำดับกรสื่อสาร เรียกว่า Master ตัวอุกรณ์ที่เหลือ เรียกว่า Slave โดยทุกอุปกรณ์จะมี ID ของตัวเอง โดย master จะทำการส่งคำสั่งพร้อม ID ไปให้ Slave ทุกตัว และถ้าคำสั่งนั้นมี ID ตรงกับตัวเอง slave นั้นจะทำงานตามคำสั่งนั้น

คุณสมบัติของ RS485 สามารถส่งสัญญาณได้ไกลสุดถึง 1200 เมตร และเป็นระยะที่เพียงพอในการใช้ในระบบอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 Modbus RTU

Modbus RTU คือ โพรโทคอลที่ใช้การสื่อสารแบบอนุกรม ด้วยการสื่อสารแบบ Master/Slave หรืออุปกรณ์ Slave จะไม่ส่งข้อมูล (Response) กลับมาจนกว่าจะมีการร้องขอ (Request) จากอุปกรณ์ Master



รูปที่ 2.2 Modbus RTU 1

Modbus RTU โดยทั่วไปจะใช้กับการสื่อสาร แบบ RS-232 หรือ RS-485 ข้อมูลในโพรโทคอล Modbus จะถูกเก็บ 4 รูปแบบ คือ

1) Output coils
2) Input contacts
3) Input registers
4) Holding registers

โดย Output coils และ Input contacts แต่ละ address จะเก็บค่าเพียง 1 บิต หรือมีค่าได้แค่ “0” กับ “1” เปรียบเสมือนค่าการเปิดและปิดของอุปกรณ์รีเลย์และสวิตช์ที่พบได้ในระบบงานอัตโนมัติอุตสาหกรรม

ในขณะที่ Input registers และ Holding registers สามารถเก็บค่าเป็นตัวเลขได้ถึง 16 บิต เปรียบเสมือนค่าที่มาจากอุปกรณ์ตรวจวัดที่ส่งข้อมูลแบบอนาล็อก (Analog)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Modbus RTU จะรับส่งเป็นชุดข้อมูล โดยที่ใน 1 ชุดข้อมูลนั้นจะประกอบด้วยส่วน 6 ส่วนดังตารางที่ 2.1

ตาราง 2.1 ชุดข้อมูลของ Modbus RTU

Field Name	Bit length	Function
Start	28	อ้างอิงถึงการเริ่มต้นชุดข้อมูล
Address	8	Address ของอุปกรณ์
Function	8	ชุดสำหรับ Function Code
Data	N x 8	ข้อมูลที่ต้องการ
CRC	16	ชุดข้อมูลตรวจสอบความผิดพลาด
End	28	อ้างอิงถึงการสิ้นสุดข้อมูล

Function code สามารถแบ่งหน้าที่ได้ตามรหัส มีฟังก์ชันการทำงานอยู่ 2 แบบ คือ การอ่าน (Read) และเขียน (Write) ไปยัง Coils, Contacts หรือ Registers

ตาราง 2. 2 รหัสฟังก์ชันของ RS485

Function code	Action	Data Type	Object Type
01	Read	Single bit	Output Coils
05	Write Single	Single bit	Output Coils
15	Write Multiple	Single bit	Output Coils
02	Read	Single bit	Input Contracts
04	Read	Word (16 bit)	Input Registers
03	Read	Word (16 bit)	Holding Registers
06	Write Single	Word (16 bit)	Holding Registers
16	Write Multiple	Word (16 bit)	Holding Registers

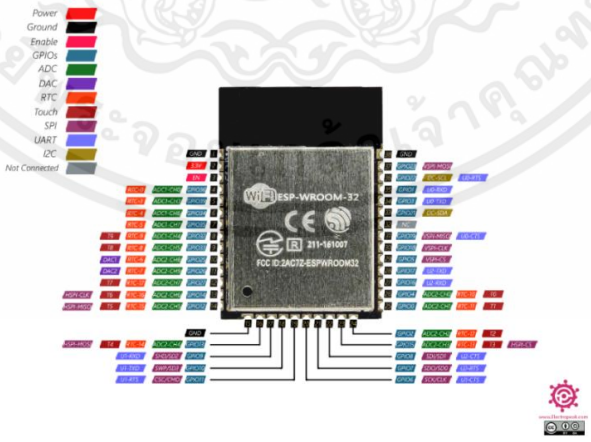
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 Microcontroller

Microcontroller เป็นอุปกรณ์ควบคุมขนาดเล็ก ซึ่งมีความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาหน่วยประมวลผล(ซีพียู), หน่วยความจำ(RAM) และพอร์ตการเชื่อมต่อต่าง ๆ ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในวงจรเดียวกัน

ESP32 เป็น microcontroller ที่รองรับการเชื่อมต่อ Wi-Fi และ Bluetooth 4.2 BLE ขาใช้งานต่าง ๆ ของ ESP32 รองรับการเชื่อมต่อบัสต่าง ๆ ดังนี้

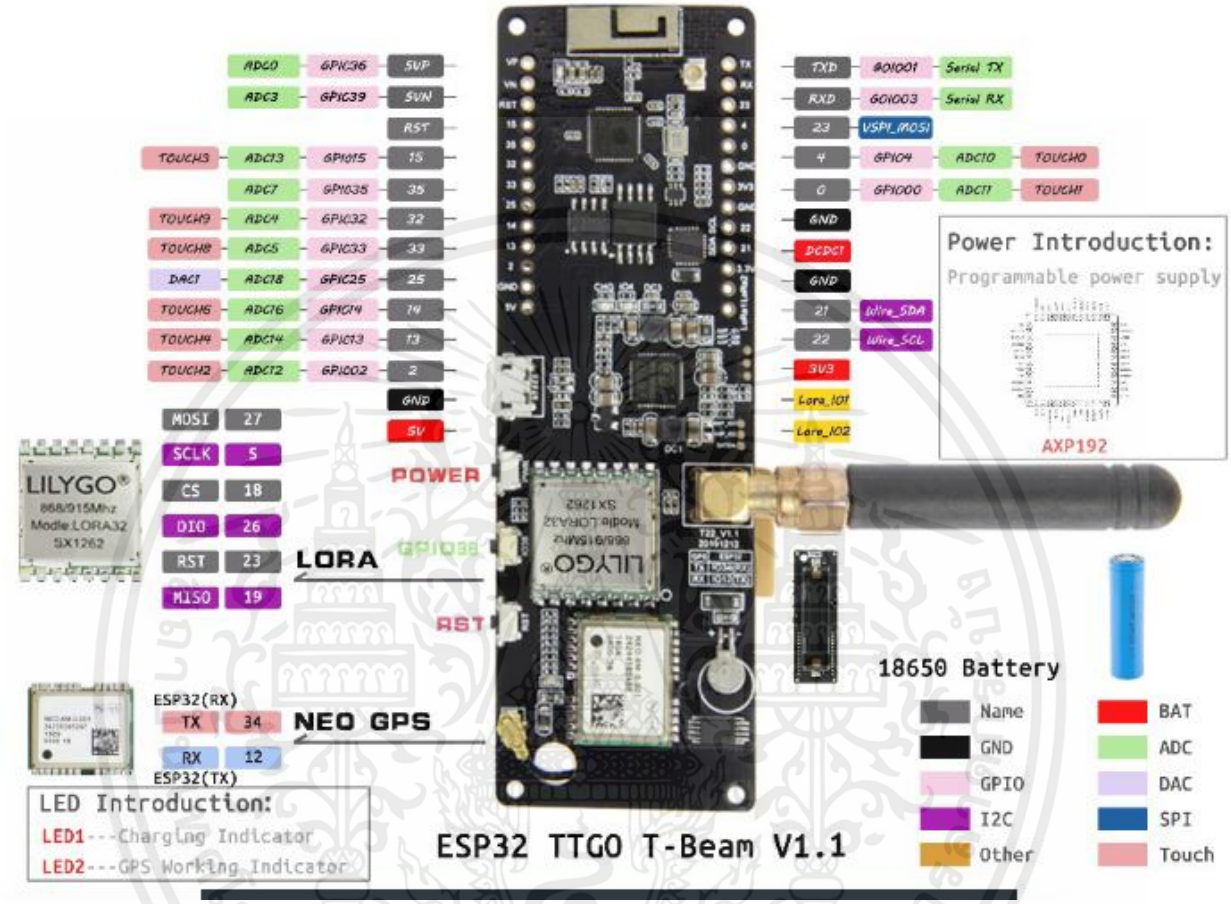
- มี GPIO จำนวน 32 ช่อง
- รองรับ UART จำนวน 3 ช่อง
- รองรับ SPI จำนวน 3 ช่อง
- รองรับ I2C จำนวน 2 ช่อง
- รองรับ ADC จำนวน 12 ช่อง
- รองรับ DAC จำนวน 2 ช่อง
- รองรับ I2S จำนวน 2 ช่อง
- รองรับ PWM / Timer ทุกช่อง
- รองรับการเชื่อมต่อกับ SD-Card



รูปที่ 2.3 TTGO LoRa32 pinout (ที่มา https://electropeak.com)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TTGO LoRa32 แผงความคุมการทำงาน ที่ใช้ ESP32 เป็นตัวจัดการการทำงาน มีโมดูล RFM95 ในตัวทำให้รองรับการส่งสัญญาณ LoRaWAN

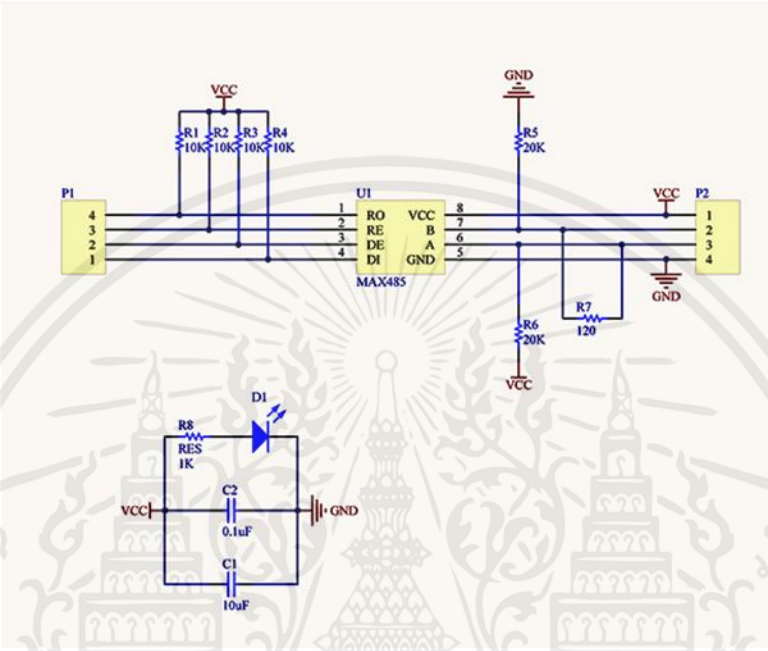


รูปที่ 2.4 TTGO LoRa32 pinout ที่มา (https://doc.riot-os.org)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 MAX485

ตัวกลางในการแปลงสัญญาณ Serial RS485 เป็นสัญญาณ Serial SDA SCL ของ Arduino ตัวโมดูลรองรับการทำงานที่ 5V สำหรับการสื่อสารอนุกรมแบบมีสายที่ใช้พลังงานต่ำ



รูปที่ 2.5 ผังวงจรของ MAX485

รูปที่ 2.6 โมดูล MAX485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

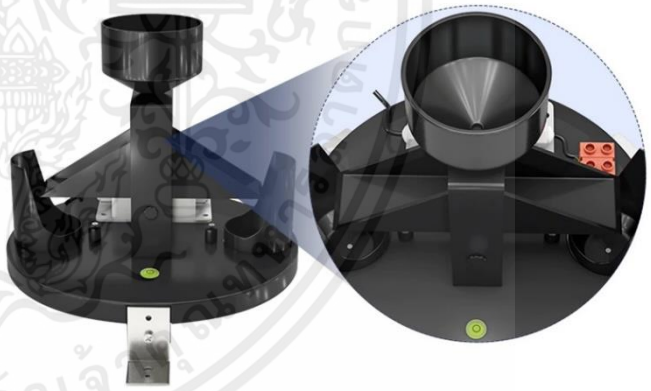
2.1.6 Rain Sensor ABS rs485

ปริมาณน้ำฝนที่วัดได้เป็นการวัด หน่วย ปริมาณ mm (milli-meter) ต่อนาที (minute) โดยการทำงานจะเป็นการกระดกของกรวยรองรับน้ำภายใน ความละเอียดอยู่ที่ 0.5 mm คือ จะมีการเปลี่ยนค่าทุก ๆ 0.5 mm/min ความละเอียดสูงสุดที่สามารถวัดได้ที่ 8 mm/min

ตาราง 2. 3 ข้อมูลเริ่มต้นของ Rain Sensor ABS rs485

Baud rate	2400, 4800 (เริ่มต้น), 9600
Address	0x00 ปริมาณน้ำฝน 0x01 trig count
Power Supply	4.5V – 30VDC

รูปที่ 2.7 Rain sensor dimension



รูปที่ 2.8 Rain sensor inside

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7 Weather Station VMS-300BYH-M

Sensor วัดค่า อุณหภูมิอากาศ, ความชื้นในอากาศ และ ปริมาณฝุ่น PM ในอากาศ ผ่านตัววัดความละเอียดสูง อ่านค่าอุณหภูมิ ได้ในช่วง -40 C ถึง 80 C ความแม่นยำที่ ± 0.5 C อ่านค่าความชื้นที่ 0-100 % ความแม่นยำที่ ± 0.3 %

รูปที่ 2.9 Sensor วัดอุณหภูมิ,ความชื้น

ตาราง 2. 4 ข้อมูลเริ่มต้นของ Weather Station VMS-300BYH-M

Baud rate	2400, 4800, 9600 (เริ่มต้น)
Address	0x00 ความชื้น 0x01 อุณหภูมิ 0x0A ค่า PM
Power Supply	12V – 24VDC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8 Wind Speed VMS-3000-FS-N01

เซ็นเซอร์ใช้ตรวจวัดความเร็วลมแนวนอน ก้านหมุนด้านบนเป็นแบบถ้วย 3 ถ้วยทำการหมุนเมื่อมีลมสัมผัส จะหมุนขั้วทองแดงผ่าน สนามขั้วแม่เหล็ก ความละเอียด 0.1 m/s ช่วงการวัดที่ 0-30 m/s สำหรับติดตั้งภายนอกทนอุณหภูมิได้ -40 C ถึง 80 C

ตาราง 2. 5 Wind Speed VMS-3000-FS-N01

Baud rate	2400, 4800, 9600 (เริ่มต้น)
Address	0x00 ความเร็วการหมุน
Power Supply	12V - 24VDC

2.1.9 Wind Direction VMS-3000-FX-N01

เซ็นเซอร์ใช้ตรวจทิศทางของลม ทำงานผ่าน contact ขั้วแม่เหล็กไฟฟ้า แบ่ง 8 ช่วง สามารถแบ่งได้ 8 ทิศทาง สำหรับติดตั้งภายนอกทนอุณหภูมิได้ -40 C ถึง 80 C

ตาราง 2.6 Wind Direction VMS-3000-FX-N01

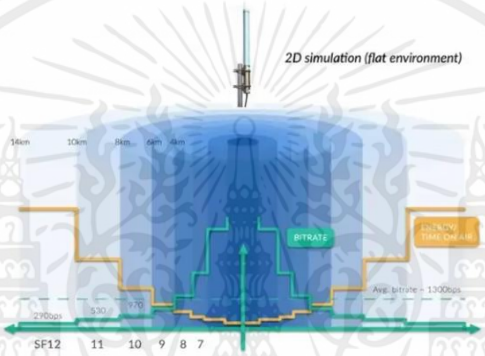
Baud rate	2400, 4800, 9600 (เริ่มต้น)
Address	0x00 ทิศทางการหมุน 0x01 องศาการหมุน
Power Supply	10V - 30VDC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ส่วนการสื่อสาร LoRaWAN

2.2.1 LoRaWAN

LoRa เป็นเทคโนโลยีการสื่อสารไร้สายผ่านคลื่นวิทยุที่เหมาะสมกับการใช้งานด้าน IOT ซึ่งมีจุดเด่นในการส่งสัญญานได้ไกลสูงสุดถึง 20 กิโลเมตร การใช้พลังงานน้อยในการส่งสัญญาณ เป็นการสื่อสารทางตรงจริงไม่จำเป็นต้องมีโครงสร้างระบบที่ซับซ้อน มีค่าใช้จ่ายถูกกว่าการทำโครงสร้างส่งข้อมูลแบบอื่น แต่มีข้อเสียคือการจำกัดข้อมูลในการส่ง ถูกพัฒนาโดย Semtech Corporation กำหนดมาตรฐานโดย LoRa Alliance



รูปที่ 2.8 LoRa Spreading factor ที่มา (https://www.thethingsnetwork.org)

หากต้องการส่งข้อมูลในระยะทางที่ไกลมาก ๆ ทำให้ใช้พลังงานในการส่งข้อมูลสูงและ Data Rate ที่ใช้ได้น้อยลงไปด้วย ทำให้เราสามารถส่งข้อมูลในแต่ละครั้งได้น้อย กล่าวได้ว่ายิ่งอุปกรณ์อยู่ใกล้ Gateway มาก จะทำให้สามารถส่งข้อมูลที่มี Data Rate สูงขึ้นได้และใช้พลังงานน้อยลงด้วย

การนำเทคโนโลยี LoRa มาใช้นั้นจำเป็นต้องเป็นไปตามมาตรฐานที่ถูกกำหนดไว้ ในประเทศไทยสามารถใช้ความถี่ในช่วง 920 – 925 MHz และช่วง 433 MHz ซึ่งเป็นช่วงความถี่ที่ไม่ต้องมีใบอนุญาต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.7 ช่วงความถี่ที่ไม่ต้องมีใบอนุญาตในการใช้งาน LoRa

Region	Code	Frequency (MHz)
Europe	EU868	863-870
North America	US915	902-928
South America	AU915/AS923	915-928
India	IN865	865-867
Asia	AS923	433 and 915-928

การเชื่อมต่อระหว่างอุปกรณ์กับ Gateway ผ่าน LoRaWAN Network โดยมีการเปิดใช้งานโดยการยืนยันด้วย Device Address 32 bits, Network Session Key 128 bits และ Application Session Key 128 bits ซึ่งการเชื่อมต่อของฝั่งอุปกรณ์ไปยัง Gateway มีทั้งสิ้น 3 ประเภทคือ Class A, Class B และ Class C ที่เหมาะสมกับการใช้งานที่ต่างกัน

ตาราง 2.8 ประเภทของรูปแบบการส่งข้อมูลด้วย LoRa

Class	ข้อดี	ข้อเสีย
A	ประหยัดพลังงานที่สุด	ส่งข้อมูลได้ช้า
B	สามารถกำหนดความเร็วการสื่อสารได้	ใช้พลังงานมากกว่าเมื่อเทียบกับ Class A
C	มีการรับส่งข้อมูลตลอดเวลา	ใช้พลังงานมากที่สุดและตลอดเวลา

2.2.2 Lora CLASS-A

การสื่อสารรูปแบบนี้เป็นการสื่อสารที่ให้อุปกรณ์ปลายทางเป็นฝ่ายเริ่มการสื่อสารก่อนเท่านั้น Server หรือ Gateway ไม่สามารถเริ่มต้นการสื่อสารแบบ A ได้ โดยอุปกรณ์จะส่งสัญญาณช่วงความถี่หนึ่งออกไป เมื่อ Server หรือ Gateway ได้รับข้อมูลแล้วจะทำการส่งสัญญาณตอบกลับมายังอุปกรณ์ส่งเพื่อรอการส่งข้อมูลครั้งต่อไป ในขณะที่ไม่มีการส่งข้อมูล อุปกรณ์จะเข้าสู่การจำศีลเพื่อประหยัดพลังงานและจะถูกปลุกให้ตื่นขึ้นเมื่อถึงเวลาที่จะส่งข้อมูลครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 เกตเวย์

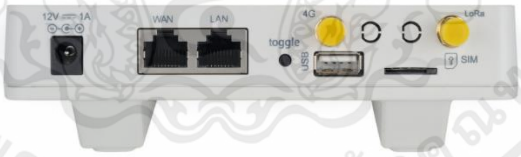
เกตเวย์ (Gateway) คือ อุปกรณ์ฮาร์ดแวร์ที่เชื่อมต่อเครือข่ายต่าง ๆ เข้าด้วยกัน โดยมีความสามารถสูงในเชื่อมต่อเครือข่ายที่ใช้โปรโตคอลต่างกัน และใช้สื่อส่งข้อมูลต่างชนิดกันได้อย่างไม่มีขีดจำกัด เช่น การใช้เกตเวย์ในการเชื่อมต่อเครือข่ายที่เป็นคอมพิวเตอร์ประเภทพีซี (PC) เข้ากับคอมพิวเตอร์ประเภทแมคอินทอช (MAC) หรือการเชื่อมต่ออีเทอร์เน็ตแลน (Ethernet LAN) ที่ใช้สายส่งแบบยูทีพี (UTP) เข้ากับโทคเคน ริง แลน (Token Ring LAN) เป็นต้นในที่นี้ใช้ LoRa Gateway ของ Dragino รุ่น LG308

2.2.4 LG308

LG308 เป็นโอเพ่นซอร์ส LoRaWAN Pico Gateway ช่วยให้เชื่อมต่อเครือข่ายไร้สาย LoRaWAN กับเครือข่าย IP ผ่าน Wi-Fi, อีเธอร์เน็ต หรือ 3G/4G เซลลูลาร์ผ่านโมดูลเสริม LTE LoRa ไร้สายช่วยให้สามารถส่งข้อมูลและเข้าถึงระยะไกลมากด้วยอัตราข้อมูลต่ำ

LG308 ใช้การส่งข้อมูลด้วย semtech packet forwarder และทำงานร่วมกับโปรโตคอล LoRaWAN ได้อย่างสมบูรณ์ ประกอบด้วยเสารับสัญญาณ LoRa SX1301 และ SX1257 2 ตัว รองรับการถอดรหัสสัญญาณแบบคู่ขนานพร้อมกันได้ 10 Channels

LG308 มีการตั้งค่าพื้นฐาน LoRaWAN ตามมาตรฐานที่กำหนดค่าไว้ล่วงหน้าเพื่อใช้สำหรับประเทศต่าง ๆ ผู้ใช้สามารถปรับแต่งคลื่นความถี่เพื่อใช้ในเครือข่าย LoRa ในประเทศของตนเองได้อีกด้วย



รูปที่ 2.9 พอร์ตการเชื่อมต่อของ LoRa Gateway Dragino LG308

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 RFM95

RFM95 คือ โมดูลสื่อสารไร้สายที่ใช้ในการสื่อสารแบบไร้สายที่รองรับการส่งสัญญาณในช่วงความถี่ LoRa ที่มีความถี่ อยู่ในช่วงประมาน 800 MHz ถึง 940 MHz ส่งและรับข้อมูลด้วยระบบการเปลี่ยนแปลงความถี่แบบโมดูลชุดเดียวกัน (FSK - Frequency Shift Keying) และระบบการเปลี่ยนแปลงความถี่แบบขาสี่เหลี่ยม (OOK - On-Off Keying) มีการควบคุมผ่านสัญญาณ SPI (Serial Peripheral Interface) แบบใช้พลังงานต่ำ RFM95 ทำให้เหมาะสำหรับการใช้งานในอุปกรณ์ที่ต้องการพลังงานต่ำและอยู่ในโหมดการส่งข้อมูลเป็นระยะเวลานาน ๆ เช่น ระบบตรวจวัดระยะไกลหรืออุปกรณ์ IOT ที่ต้องส่งข้อมูลจากอุปกรณ์เซนเซอร์ไปยังจุดเชื่อมต่อเครือข่ายกลาง LoRaWAN ในระยะไกล

2.3 ฐานข้อมูล

2.3.1 Amazon Web Services

คลาวด์ (Cloud) เป็นการนำเครื่องเซิร์ฟเวอร์มาเชื่อมต่อทำงานร่วมกันเป็นกลุ่ม ในรูปแบบ Cluster ทำหน้าที่เป็นที่จัดเก็บข้อมูล การนำไปประมวลผล หรือ อื่น ๆ โดยที่ผู้พัฒนาไม่ต้องมีตัวเครื่องเซิร์ฟเวอร์อยู่เลย กล่าวคือ การจำลองเซิร์ฟเวอร์ไว้บนเครื่องเซิร์ฟเวอร์อีกทีหนึ่ง ในงานนี้ ได้เลือกใช้ Firebase ในช่วงทดสอบ และปรับเปลี่ยนมาใช้ AWS

AWS (Amazon Web Services) เป็นแพลตฟอร์มคลาวด์ที่ให้บริการโดย Amazon ที่มีมากกว่า 200 บริการจากดาต้าเซนเตอร์รอบโลก เน้นการให้บริการออนไลน์ที่สามารถขยายระดับได้และมีแนวทางการประมวลผลของคลาวด์ที่คุ้มค่าต่อราคา บริการของ Amazon มีรายประเภท ได้แก่ บริการการประมวลผล ที่เก็บข้อมูล ฐานข้อมูล การทำเครือข่ายและการขนส่งข้อมูล เครื่องมือความปลอดภัย เครื่องมือในการพัฒนา และ เครื่องมือในการจัดการ เป็นต้น ในโปรเจคนี้ ทางผู้จัดทำได้เลือกใช้เครื่องมือดังต่อไปนี้

- บริการการประมวลผล
    - EC2
- ฐานข้อมูล
    - RDS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.1 EC2 (Elastic Compute Cloud)

เป็นบริการโฮสต์เซิร์ฟเวอร์ของ Amazon โดยเซิร์ฟเวอร์ที่ว่านี้ เป็นเซิร์ฟเวอร์เสมือน (virtual machine / virtual server) สามารถเลือกฮาร์ดแวร์และซอฟต์แวร์ต่าง ๆ ได้ มีสถานที่ตั้งของเซิร์ฟเวอร์ที่แตกต่างกัน รองรับหลาย OS ทำงาน เช่น ubuntu, Linux, Window เป็นต้น

2.3.1.2 RDS (Relational Database Service)

ระบบบริการฐานข้อมูลเชิงสัมพันธ์บนคลาวด์ สามารถปรับขนาดฐานข้อมูลตามการใช้งานได้ สามารถสร้างรูปแบบฐานข้อมูลได้อย่างอิสระ และสามารถสร้างได้หลายฐานข้อมูลใน instance รองรับ PostgreSQL, MySQL, Maria DB, Oracle, SQL Server และ Amazon Aurora

2.3.2 SQL

SQL Structured Query Language คือ คำสั่งที่ใช้จัดการ database เป็นภาษาที่ออกแบบมาเพื่อทำการจัดการข้อมูล, ค้นหาข้อมูล, ปรับปรุง, เปลี่ยนแปลง, เพิ่ม และ ลบข้อมูล ซึ่งข้อมูลจะถูกเก็บอยู่ในฐานข้อมูลในรูปแบบตารางที่มีลักษณะเป็นคอลัมน์และแถว โดยมี MySQL เป็นระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL

2.3.3 Grafana

Grafana เป็น Dashboard tool แบบโอเพนซอร์ส เป็นเครื่องมือในการสร้าง dashboard สำหรับMonitoring Database วิธีการใช้งานคือเราต้องเพิ่มทำการเพิ่ม data source (แหล่งข้อมูล) เช่น Prometheus, Elasticsearch, MySQL, PostgreSQL ฯลฯ เป็นต้น ให้ Grafana ไปดึงข้อมูลมาสร้างเป็นกราฟแสดงข้อมูล สามารถแสดงข้อมูลในระดับ real-time และยังกำหนดการแจ้งเตือนไปยังอีเมล, ไลน์ หรือ Slack ได้ ทำให้ใช้งานได้หลากหลาย

โดยผู้พัฒนาเลือกใช้การเก็บข้อมูลแบบ SQL โดยใช้ phpMyAdmin เป็นโปรแกรมที่ใช้วิธีจัดการฐานข้อมูลแบบ MySQL แล้วนำ Grafana มาดึงข้อมูลจากฐานข้อมูลนี้ไปแสดงผลเป็น Real-time Dashboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 Node-RED

Node-RED เป็นเครื่องมือจัดการและจัดการเหตุการณ์ขึ้นอยู่กับ Node.js แอปพลิเคชัน Node-RED มักทำงานเป็นเว็บเซิร์ฟเวอร์และผู้ใช้สามารถปรับแต่งและจัดการการเชื่อมต่อระหว่างฮาร์ดแวร์ต่างๆ และสร้างขั้นตอนการทำงานจากเบราว์เซอร์ของคอมพิวเตอร์เครื่องใดก็ได้ แต่เป็นเรื่องง่ายและทำงานในเว็บเบราว์เซอร์ แอปพลิเคชันเซิร์ฟเวอร์นี้ยังมีความสามารถในการทำงานบนอุปกรณ์ เช่น Gateway

ฟังก์ชันทั้งหมดของ Node-RED ทำงานกับโค้ดที่มีอยู่แล้ว อินเทอร์เฟซผู้ใช้ของ Node-RED ดูเรียบง่ายรูปแบบ Flow Base เหมาะสำหรับงานด้าน IOT สามารถเขียนโปรแกรมบน Web Browser ได้สะดวกลดการเขียน Code โดย Node-Red มี Node Service ต่างๆ ให้เราเลือกนำมาตั้งค่าและใช้ได้ทันที และเปิดเผยคุณแทบไม่มีปัญหาในการพัฒนาโครงการ IOT ด้วย Node-RED ใน GitHub แสดงใน JSON (JavaScript Object Notation) และสามารถส่งออกไปยังคลิปบอร์ดได้อย่างง่ายดายหรือสามารถนำเข้าสู่ Node-RED หรือแชร์ทางออนไลน์

2.4 Internet of Things

ระบบไอโอที (Internet of Things) คือ ระบบที่เชื่อมโยงอุปกรณ์ อิเลกทรอนิกซ์ต่างๆ เข้าไว้ด้วยกัน เช่น เซนเซอร์ต่างๆ โดยผ่านเครือข่ายอินเตอร์เน็ต การเชื่อมโยงนี้จะ ทำให้อุปกรณ์ในเครือข่ายสามารถสื่อสารกันได้ และยังทำให้ผู้ใช้สั่งการควบคุมอุปกรณ์ได้ ซึ่งข้อมูลที่เก็บมาได้จากอุปกรณ์จะนำมาเก็บไว้บนพื้นที่ เช่น บนคลาวด์ เป็นต้น และสามารถนำข้อมูลที่เก็บมาได้เหล่านั้นเพื่อไปวิเคราะห์ในภายหลัง การใช้ระบบไอโอทียังช่วยในเรื่องความแม่นยำของข้อมูล และความเป็นปัจจุบัน (Real-time) ทำให้ผู้ใช้สามารถมองเห็นถึงปัญหาได้ตลอดเวลา แต่ด้วยจำนวนข้อมูลที่มีจำนวนมากจึงมีโอกาสเสี่ยงในเรื่องของความปลอดภัยด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 Protocol

โปรโตคอลคือข้อกำหนดในการสื่อสารกันระหว่างอุปกรณ์ต่างๆ เพื่อให้เกิดการสื่อสารระหว่างกันเปรียบเสมือนกับภาษาที่ใช้ในการสื่อสารที่อุปกรณ์สามารถนำไปใช้ในการติดต่อกันได้ซึ่งในโปรโตคอลแต่ละชนิดก็มีจุดประสงค์ที่ทำงานแตกต่างกันไปในโปรเจคนี้ได้เน้นใช้โปรโตคอลสำหรับการแลกเปลี่ยนข้อมูล ผ่าน MQTT

2.4.2 MQTT

MQTT (ย่อมาจาก MQ Telemetry Transport) เป็นโปรโตคอลที่ใช้ในการส่งข้อความ ซึ่งถูกใช้บ่อยในอุปกรณ์ประเภทไอโอที โปรโตคอลนี้ขับเคลื่อนด้วยเหตุการณ์ และเชื่อมต่ออุปกรณ์ผ่านรูปแบบของการติดตาม-เผยแพร่ (Subscribe-Publish) ผู้ส่งและผู้รับจะสื่อสารกันผ่านหัวข้อที่กำหนดไว้ การเชื่อมต่อระหว่างผู้ใช้งานจะต้องผ่านคนกลางคือ ผู้ให้บริการ MQTT คนกลางนี้จะมีหน้าที่รับข้อความที่เข้ามาจากหัวข้อต่างๆ และกระจายไปยังผู้ที่ติดตามหัวข้อนั้นๆ

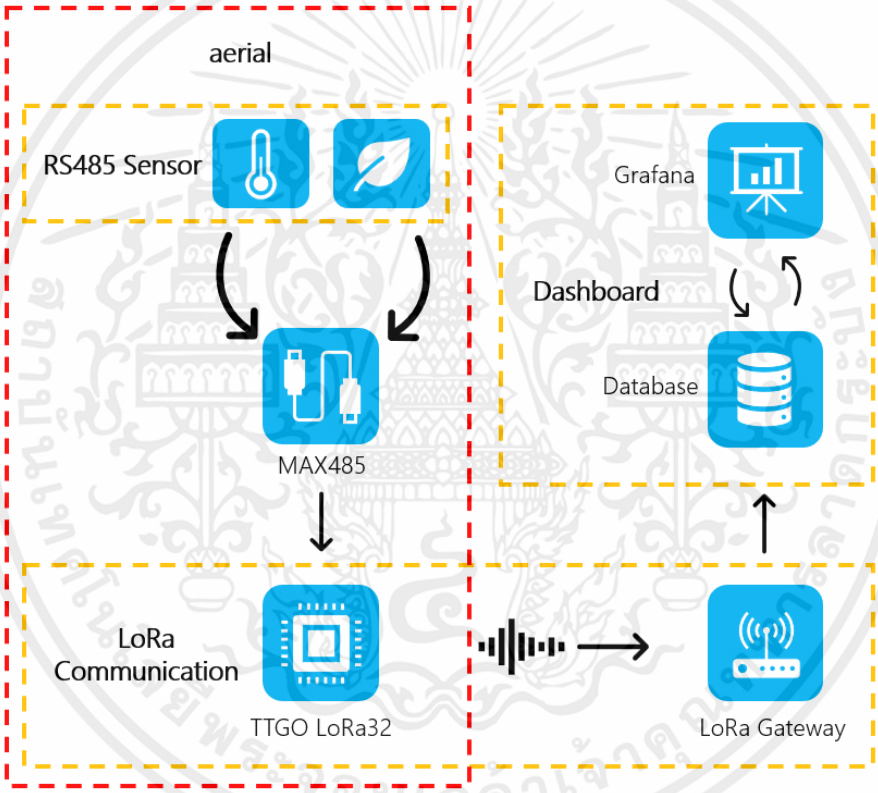
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบโครงงาน

บทนี้จะอธิบายวิธีการและรายละเอียดของการทดลองในส่วนต่างๆ ของโครงงาน

- ส่วนการวัดค่า (Sensor Reading)
- ส่วนการสื่อสารและส่งค่า (Communication)
- ส่วนแสดงผล (Dashboard)



รูปที่ 3.1 Project Diagram 1

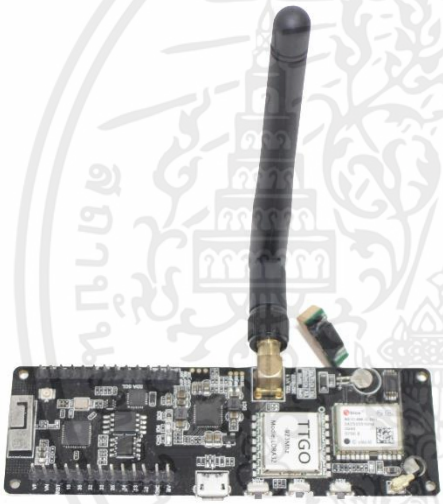
รูปนี้แสดงภาพรวมและการทำงานของระบบทั้งหมด โดยเดินที่การวัดค่าจาก sensor เข้าที่ TTGO LoRa32 ส่งค่าไปที่ Gateway และนำค่านั้นเข้าเก็บที่ Database แล้วให้ Grafana ดึงค่าขึ้นแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ส่วนการวัดค่า

3.1.1 อุปกรณ์ที่ใช้

- บอร์ด TTGO LoRa32
- MAX485
- Sensor วัดปริมาณน้ำฝน
- Sensor วัดอุณหภูมิ, ความชื้น
- voltage regulator or power supply



รูปที่ 3.2 บอร์ด TTGO LoRa32

รูปที่ 3.3 MAX485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 Sensor วัดปริมาณน้ำฝน

รูปที่ 3.5 Sensor วัดคุณภาพอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 Sensor วัดความเร็วลม VMS-3000-FS-N01

รูปที่ 3.7 Sensor วัดทิศทางลม VMS-3000-FX-N01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ขั้นตอนการทดลอง

ประกอบ Module เข้ากับบอร์ด เชื่อมสาย Wiring โดย Sensor RS485 จะมีสาย 4 สาย

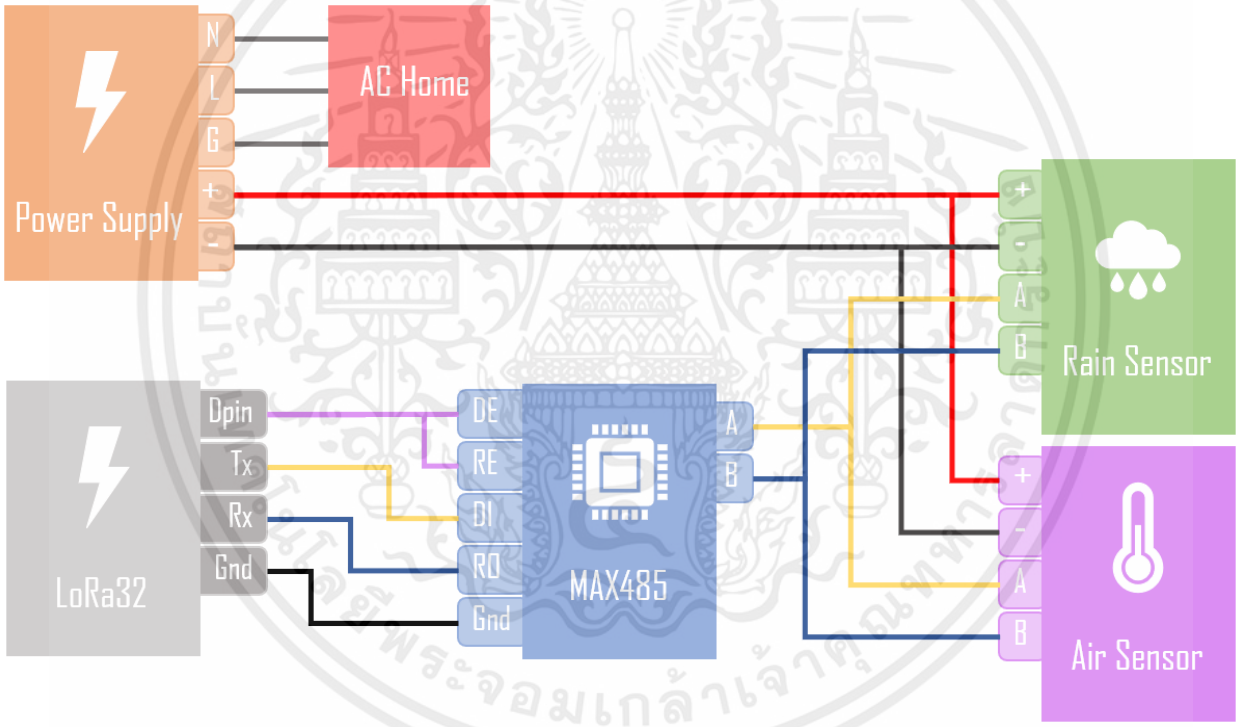
1) สายไฟพวก

2) สายไฟลบ

3) สาย Data A

4) สาย Data B

นำสายไฟบวกและลบ ต่อเข้า voltage regulator ตามขั้ว และ สาย Data A และ Data B ต่อเข้าที่ module MAX485 ตามช่อง

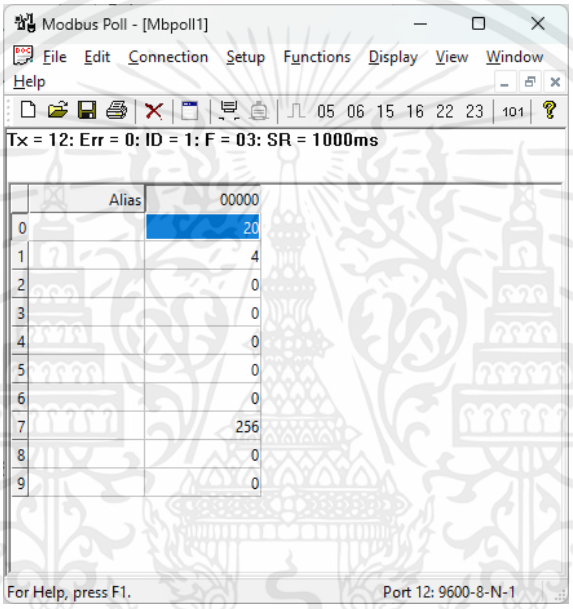


รูปที่ 3.8 Wiring Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนโปรแกรมสำหรับอ่านค่าจาก Sensor โดยผ่าน Module MAX485 กำหนดค่า Slave ID และ Address ของ Sensor ที่เราจะอ่านให้ตรง โดยสามารถเข้าไปดูได้จาก Data Sheet ของ Sensor

โปรแกรมสำหรับการทดลองเกี่ยวกับ Modbus คือ โปรแกรม Modbus Poll สามารถดาวน์โหลดได้ทาง https://www.modbustools.com/download.html โดยการแก้ไข Slave ID และ Baud rate ของ Sensor สามารถทำได้หลายวิธี โดยทางผู้จัดทำได้เลือกการแก้ไขค่าผ่าน โปรแกรม Modbus Poll

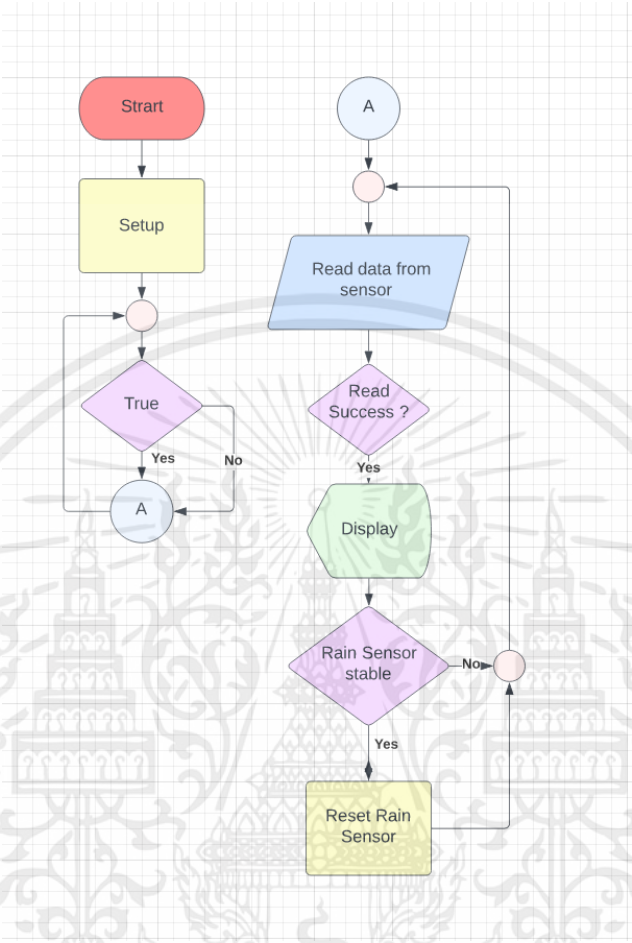


รูปที่ 3.9 Modbus Poll

- ทดลองอ่านค่าและส่งค่าผ่านโปรแกรม Modbus Poll
- ทดลองแก้ไขค่าพื้นฐานที่ต้องใช้
- จากการตั้งค่าข้างต้น นำค่าที่ได้ไปใช้ ทดลองอ่านค่าโดยใช้ TTGO LoRa32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยได้มีการออกแบบ Flow chart การทำงานของระบบไว้ดังนี้



รูปที่ 3.10 Flow Chart การทำงานของโปรแกรมการอ่านค่า

โดยเมื่อเริ่มโปรแกรมจะทำการ เข้าสู่การ setup โดยทำการ setup pin ต่างๆ ที่ใช้ในการอ่านค่า ตั้งค่า Serial ที่ใช้แล้วตัวแปรข้อมูลที่จำเป็นต่างๆ และจะเข้าสู่การวน loop ของโปรแกรม ภายใน loop จะทำการส่งคำสั่งขออ่านค่าจาก sensor ไปแล้วหากได้รับกาตอบรับจาก sensor จะนำค่าที่ได้ไปบันทึกลงตัวแปร และทำการแสดงผล หากอ่านไม่ได้จะข้ามขั้นตอนการบันทึกไป และมีส่วนเพิ่มเติมในการ reset ค่าของ sensor วัดปริมาณน้ำฝน หากไม่มีการเปลี่ยนแปลงเป็นระยะเวลาหนึ่งจะถือว่าฝนหยุดตก จะทำการส่งคำสั่ง reset ไปหา sensor ให้ทำการ set ค่าเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

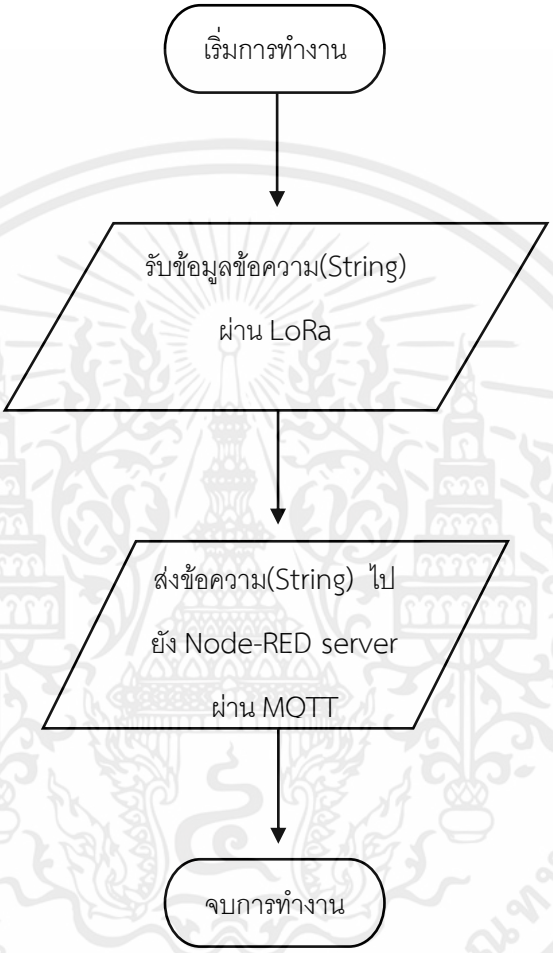
3.2 ส่วนการสื่อสารและส่งค่า

หลังจากสามารถวัดค่าผ่านเซนเซอร์ต่างๆ ด้วยบอร์ด Tony Space แล้ว จะทำการแปลงข้อมูลเป็นรูปแบบ JSON และส่งผ่าน Serial Communications มายังบอร์ด TTGO T-Beam V1.1 lora32 แล้วจะทำการแปลงข้อมูลเป็นข้อความ(String)รูปแบบ JSON ใหม่ เพื่อทำการส่งไปยัง Node-RED Server ผ่าน LoRa โดยมีการทำงานดังภาพที่ 3.9

รูปที่ 3.11 ภาพรวมของโปรแกรมบน TTGO T-Beam V1.1 lora32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

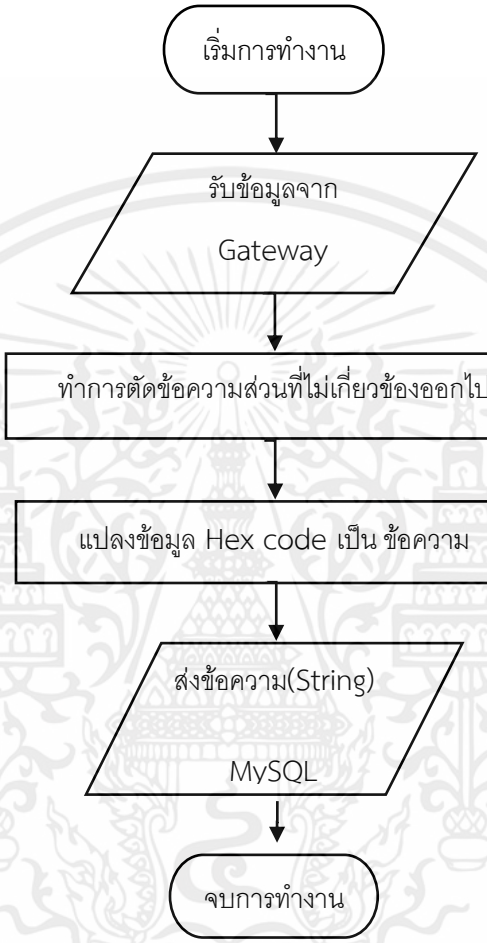
หลังจากข้อมูลถูกส่งออกจากบอร์ด TTGO จะเปิดการทำงานของ Gateway ซึ่งจะทำหน้าที่เป็นทางผ่านเพื่อส่งข้อมูลไปยัง Node-RED server มีการทำงานดังภาพที่ 3.10



รูปที่ 3.12 ภาพหลักการทำงานของ Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นข้อมูลจะถูกส่งไปยัง Node-RED server ผ่านโปรโตคอล MQTT เมื่อ Server ได้รับข้อมูลเข้ามา จะทำการตัดข้อความส่วน Header ออก เพื่อให้เหลือไว้เพียงข้อมูลที่เราต้องการ และนำข้อมูลนั้นเปลี่ยนเป็น JSON เพื่อนำข้อมูลส่งต่อไปยัง MySQL เพื่อเก็บข้อมูลสำหรับแสดงผลบน Grafana dashboard



รูปที่ 3.13 ภาพ Flowchart การแปลงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนแสดงผล

ส่วนนี้จะทำการส่งค่าที่ได้ไปที่ Database โดย ผู้จัดทำได้เลือกการทำงานเป็น local โดยการตั้ง server เองและทำการใช้ PhpMyAdmin เป็นตัวจัดการ Database และข้อมูล เป็นการเก็บข้อมูลและจัดการแบบ SQL ทำให้สามารถนำไปใช้ต่อกับ Grafana ได้

3.3.1 อุปกรณ์ที่ใช้

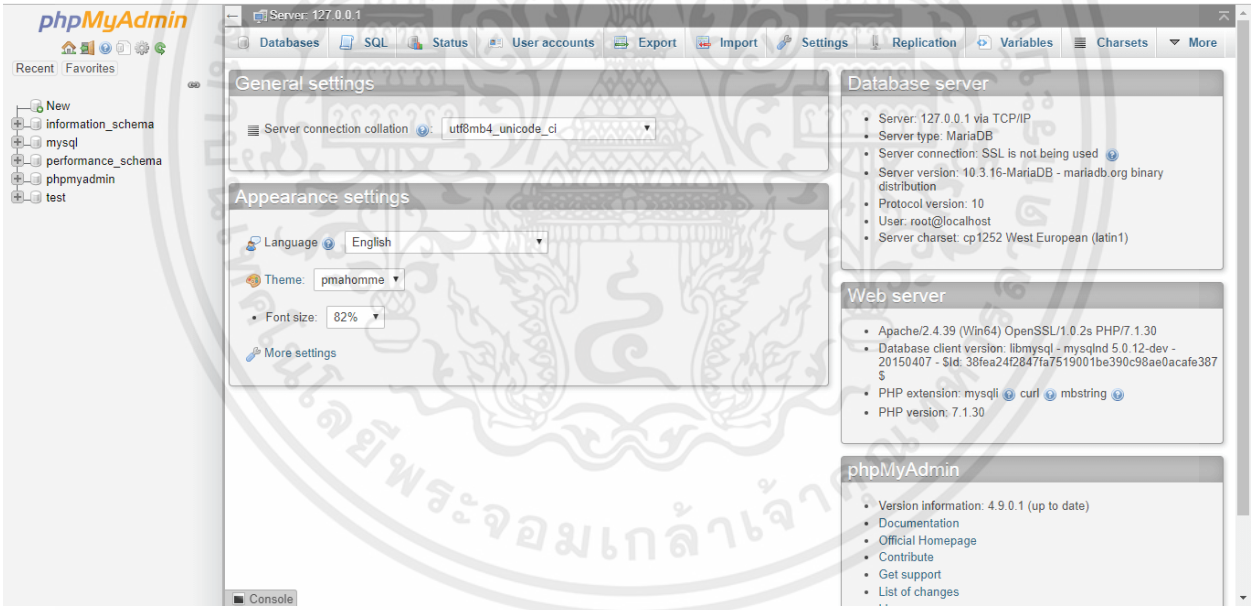
1) เครื่องที่นำมาทำ Server

2) โปรแกรม phpMyAdmin

3) โปรแกรม Grafana

3.3.2 ขั้นตอนการพัฒนา Dashboard

ลงโปรมแกรมที่ต้องใช้ทั้งหมด Node.js Node-red MySQL Grafana บน AWS Ec2

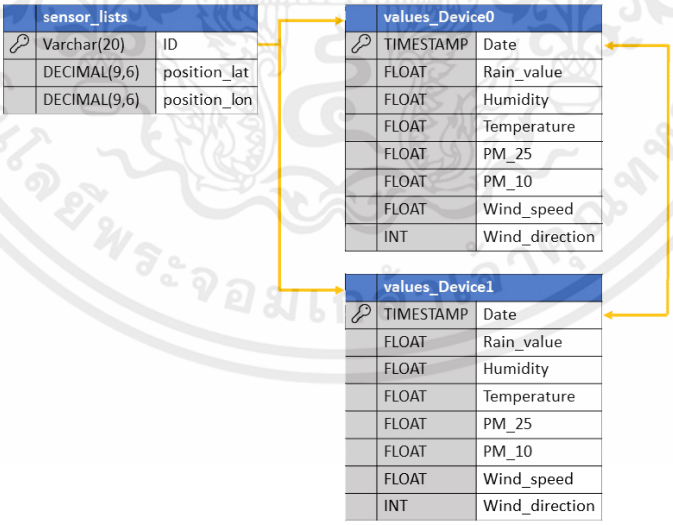


รูปที่ 3.14 PhpMyAdmin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.15 Grafana

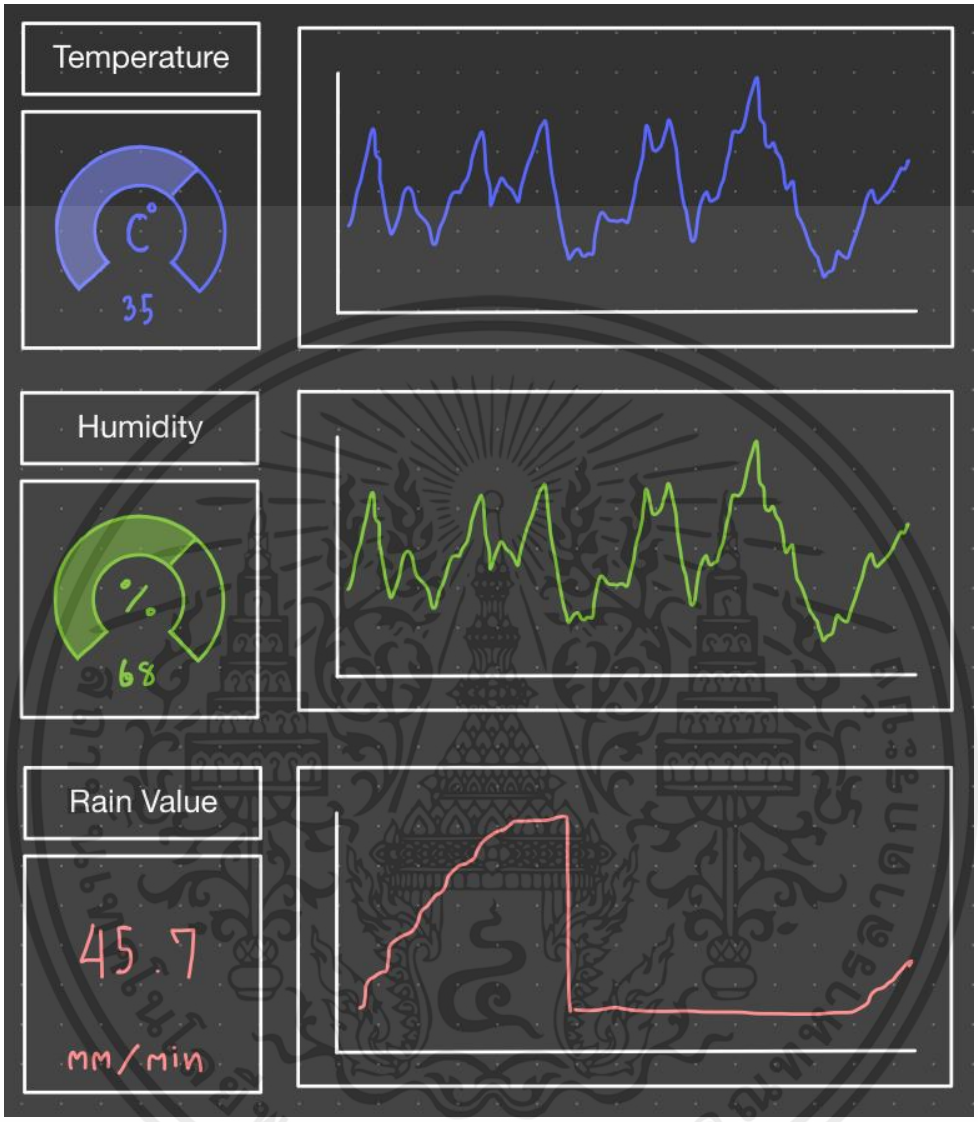
สร้างตารางข้อมูลและทดลอง query ข้อมูลออกมา โดยการใส่ด้วยมือ ทดลองการทำงาน ทำการออกแบบ Database และ ER Diagrams ให้สามารถทำงานได้อย่างมีประสิทธิภาพ



รูปที่ 3.16 ER Diagrams

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการใช้ Grafana ดึงค่าจาก Database มาแสดงผล โดยทำการออกแบบ Dashboard ไว้ดังนี้



รูปที่ 3.17 Dashboard Layout

จะใช้ Gauge ในการแสดงผลค่าปัจจุบัน เนื่องจากดูได้ง่ายและยังมีการใส่ช่วง Thresholds ความอันตรายได้ โดยแบ่งเป็นช่วงสีตามความอันตราย และด้านข้างจะมีกราฟแสดงค่าย้อนหลังได้ เพื่อแสดงผลในอดีตของค่าสภาพอากาศนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

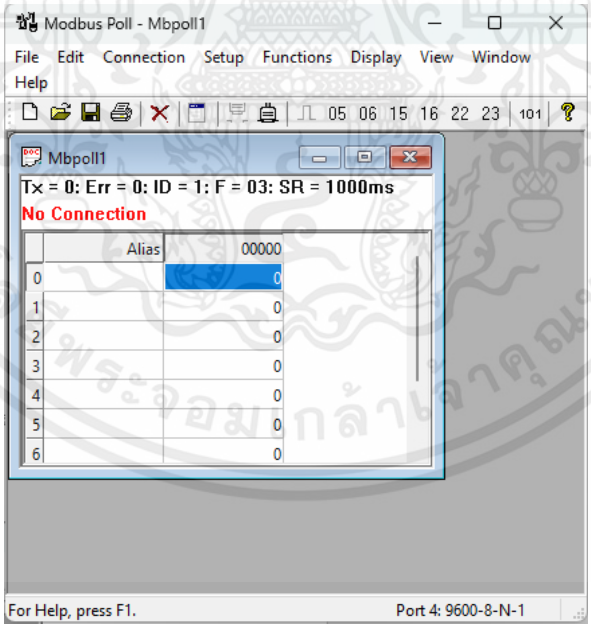
4.1 อ่านค่าและส่งค่าผ่าน โปรแกรม Modbus Poll

4.1.1 เชื่อมต่อ Sensor เข้ากับอุปกรณ์แปลงสัญญาณ RS485 to USB Serial และ power supply

รูปที่ 4.1 RS485 to USB Serial

4.1.2 เชื่อมต่อ USB และโปรแกรมเข้าด้วยกัน ให้สามารถอ่านค่าได้

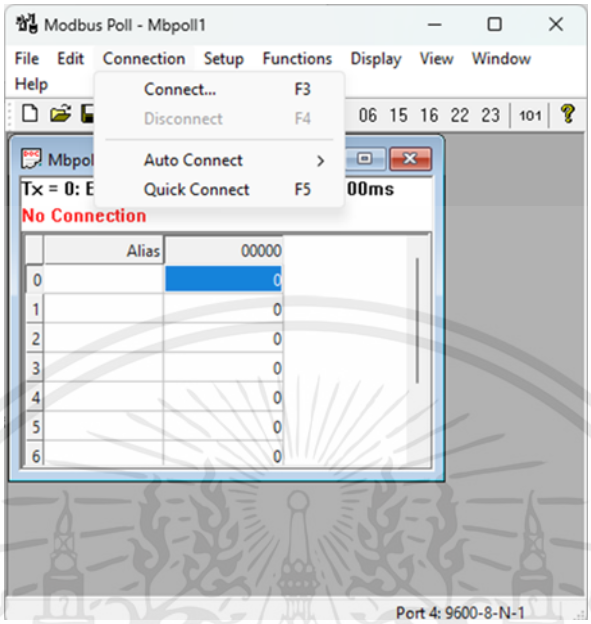
4.1.3 กดที่แถบ connection



รูปที่ 4.2 ModbusPoll (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 กด connect



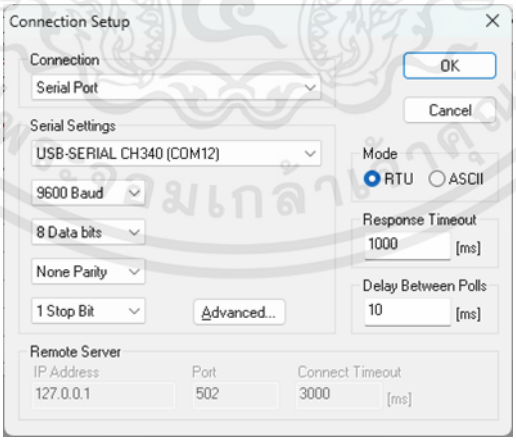
รูปที่ 4.3 ModbusPoll (2)

4.1.5 จะมีหน้าต่างการตั้งค่า ให้ตั้งค่าตาม Data sheet ที่บอกไว้โดยมีค่าที่ต้องดูคือ

1) เลือก Serial ที่ต่อไว้

2) กำหนด baud rate

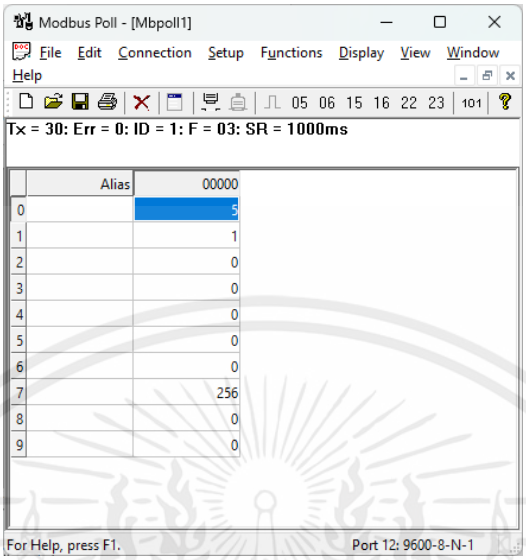
4.1.6 ตั้งค่าเรียบร้อยกด OK



รูปที่ 4.4 ModbusPoll (3)

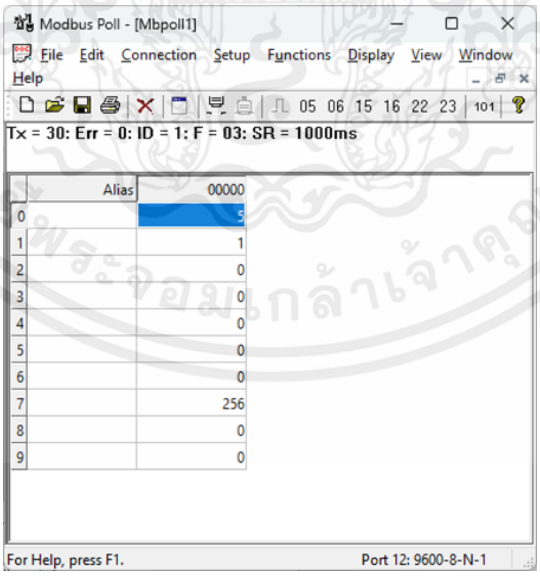
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.7 จะปรากฏค่าที่อ่านได้จากเซนเซอร์ตาม datasheet ค่าที่ได้ของ Sensor วัดฝนคือ address ที่ 0



รูปที่ 4.5 ModbusPoll (4)

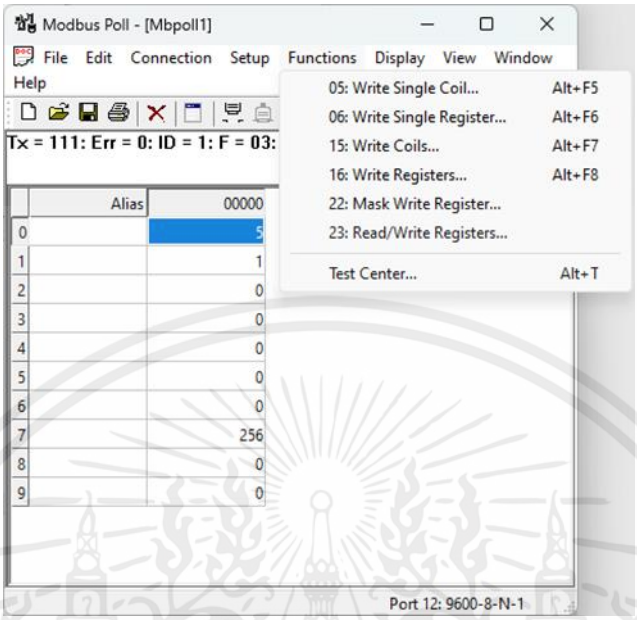
4.1.8 เซนเซอร์วัดฝนต้องการ reset ค่าทุกครั้งหลังจากฝนหยุด จึงต้องมีการส่งค่ากลับ Sensor โดยสามารถทำได้โดย กดแถบ function



รูปที่ 4.6 ModbusPoll (5)

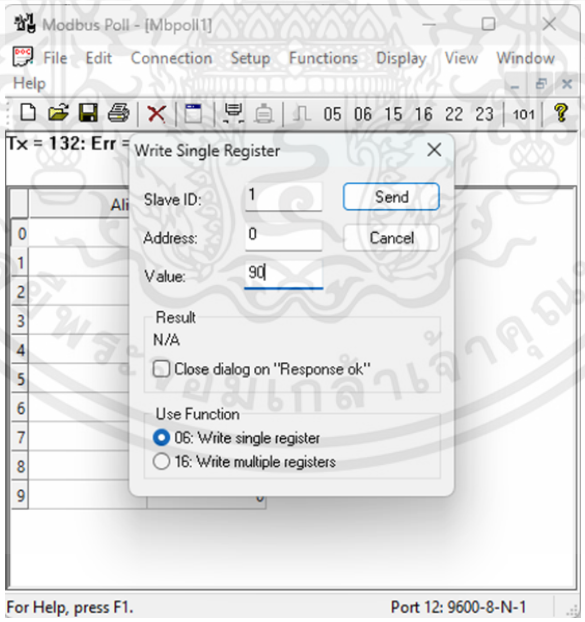
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.9 เลือก function 6 Write Single Register



รูปที่ 4.7 ModbusPoll (6)

4.1.10 ใส่ value 90 ที่ address 0 แล้วกด sent ทำให้ที่ address ถูก reset ค่ากลับไปเป็น 0



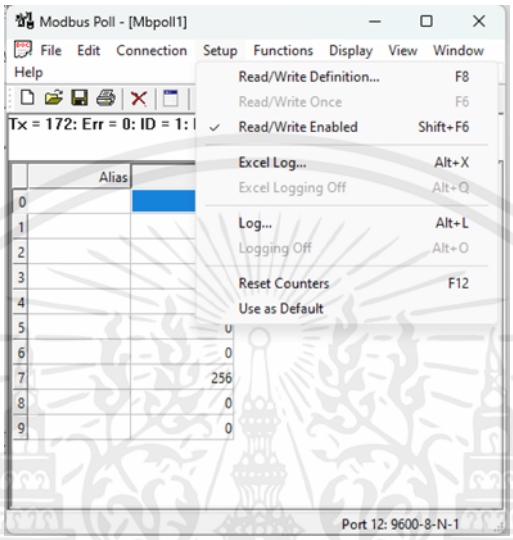
รูปที่ 4.8 ModbusPoll (7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ตั้งค่า Slave ID, Baud rate ของ Sensor ใหม่

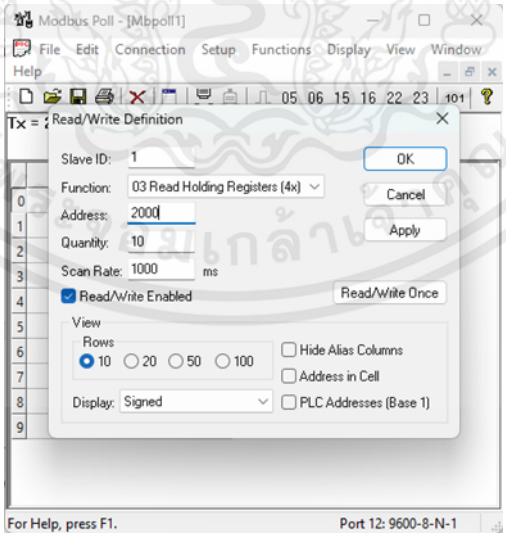
ตั้งค่า slave ID และ baud rate เพื่องายต่อการนำไปใช้งานต่อ โดยการใช้ Modbus Poll ในการตั้งค่า

4.2.1 กดที่แถบ Setup แล้วเลือก Read/Write Definition



รูปที่ 4.9 ModbusPoll (8)

4.2.2 จะปรากฏหน้าต่างการตั้งค่า หน้านี้สามารถแก้ไข slave ID, address ที่จะอ่านได้ ให้เลื่อนไปที่ address 2000

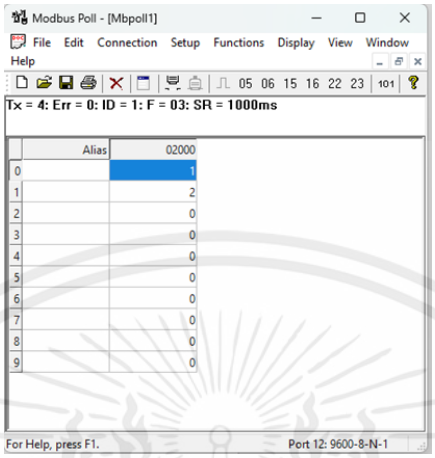


รูปที่ 4.10 ModbusPoll (9)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 Double Click ที่ช่องข้อมูลตรง address 2000 เพื่อตั้งค่า slave ID

4.2.4 Double Click ที่ช่องข้อมูลตรง address 2001 เพื่อตั้งค่า baud rate



รูปที่ 4.11 ModbusPoll (10)

4.2.5 โดยสุดท้าย ผู้จัดทำได้กำหนดค่าต่างๆ และรวมข้อมูลที่ต้องใช้ของ Sensor ดังนี้

ตาราง 4.1 การตั้งค่าเซนเซอร์

หัวข้อ	Sensor ปริมาณน้ำฝน	Sensor คุณภาพอากาศ	Sensor ความเร็วลม	Sensor ทิศทางลม
Baud rate	9600	9600	9600	9600
Slave ID	1	2	3	4
Data address	Address 0 (mm)	Address 500 (ความชื้น : %) Address 501 (อุณหภูมิ : C) Address 503 (µg/m^3) Address 504 (µg/m^3)	Address 0 (m/s)	Address 0
voltage	12V	12V	12V	12V

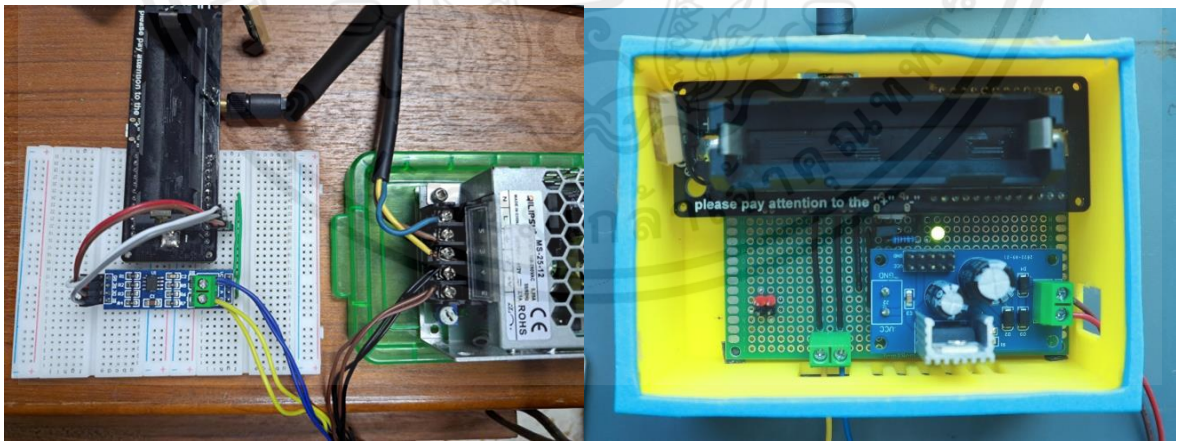
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 อ่านค่าและส่งค่า ผ่าน LoRa32

4.3.1. ทำการต่อวงจรทดลองตาม diagram ที่วางไว้ ต่อจ่ายไฟให้เซนเซอร์ 12 VDC นำสาย A และ B ต่อเข้า MAX485 ต่อไฟเลี้ยง MAX485 จาก LoRa32

ตาราง 4. 2 การเชื่อมต่อ Pin ระหว่าง LoRa32 และ MAX485

LoRa32	MAX485
3.3 V	VCC
GND	Gnd
Pin 15	RE
	DE
Pin 4	DI
Pin 0	RO



รูปที่ 4.12 RS485 wiring จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2. ใช้ซอฟต์แวร์ Arduino IDE ในการเขียนโปรแกรม เพื่อควบคุม LoRa32 และอ่านค่าจาก Sensor

4.3.3. Import library ModbusMaster เพื่อใช้งานการอ่าน RS485 และกำหนด Pin ตามข้างต้น 23 จะทำหน้าที่สั่งว่าให้ทำงานรับหรือส่ง ส่งค่า 0 เพื่อรอรับค่า ส่งค่า 1 เพื่อส่งค่าออก ประกาศ Object Modbus 2 ตัว สำหรับสอง sensor modbus1 สำหรับ sensor วัดฝน และ modbus2 สำหรับ sensor วัดอากาศ

```
#include "ModbusMaster.h"

#define MAX485_RE_NEG 23
#define RX_PIN 0
#define TX_PIN 4

ModbusMaster modbus1;
ModbusMaster modbus2;
```

รูปที่ 4.13 ตัวอย่างโปรแกรมที่เขียนใน Arduino (1)

4.3.4. สร้างตัวแปรสำหรับจัดการระบบต่างๆ และตัวแปรสำหรับเก็บค่าจาก Sensor

```
int base = 0;
int count = 0;
typedef struct
{
  float humidity;
  float temperature;
  float rain_value;
} value_type;
value_type value;
```

รูปที่ 4.14 ตัวอย่างโปรแกรมที่เขียนใน Arduino (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5. void setup ประกาศ serial ที่ใช้งาน 115200 สำหรับการทำงานในบอร์ด 9600 สำหรับการทำงานกับ Sensor ตั้งค่า Pin สั่งการ max485 set เป็น LOW สร้าง Object modbus แต่ละตัว

```
void setup() {
  Serial.begin(115200, SERIAL_8N1);
  Serial2.begin(9600, SERIAL_8N1, RX_PIN, TX_PIN);

  pinMode(MAX485_RE_NEG, OUTPUT);
  digitalWrite(MAX485_RE_NEG, LOW);

  modbus1.begin(0x01, Serial2);
  modbus1.preTransmission(preTransmission);
  modbus1.postTransmission(postTransmission);

  modbus2.begin(0x02, Serial2);
  modbus2.preTransmission(preTransmission);
  modbus2.postTransmission(postTransmission);
}
```

รูปที่ 4.15 ตัวอย่างโปรแกรมที่เขียนใน Arduino (3)

4.3.6. void loop ทำการอ่านค่าจาก sensor และ delay หลังการอ่านแต่ละครั้งเพื่อให้ max485 เปลี่ยน state ได้ทัน แล้วทำค่าที่ได้แสดงบน Serial monitor และทำการตรวจสอบเงื่อนไขฝนหยุดเพื่อ reset sensor วัดฝน

```
void loop() {
  Read_Rain_Sensor();
  delay(100);
  Read_Air_Sensor();
  delay(100);
  Serial.print("temperature: ");
  Serial.print(value.temperature);
  Serial.println(" C");
  Serial.print("humidity: ");
  Serial.print(value.humidity);
  Serial.println(" %");
  Serial.print("rain_value: ");
  Serial.print(value.rain_value);
  Serial.println(" mm/min");
  Serial.println();

  delay(800);
  check_reset();
}
```

รูปที่ 4.16 ตัวอย่างโปรแกรมที่เขียนใน Arduino (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7. Function สำหรับการเปิดปิดการอ่านค่าส่งค่าของ max485

```
void preTransmission() {
  digitalWrite(MAX485_RE_NEG, HIGH);
}
void postTransmission() {
  digitalWrite(MAX485_RE_NEG, LOW);
}
```

รูปที่ 4.17 ตัวอย่างโปรแกรมที่เขียนใน Arduino (5)

4.3.8. Function อ่านค่าจาก sensor โดยสั่งอ่านที่ Input register และส่งค่า address ที่เริ่มอ่านไป ตามด้วยจำนวนข้อมูลที่ต้องการอ่านไปอีกกี่ address และทำการตรวจสอบค่าที่ส่งกลับมาว่าสามารถอ่านได้หรือไม่ ถ้าอ่านได้จะดึงค่าจาก buffer มาใส่ในตัวแปร โดยการวัดฝนจะอ่านที address 0 ที่เดียว ส่วนการวัดอากาศจะอ่านที่ address 0 เป็นความชื้นและ address 1 เป็นอุณหภูมิ

```
void Read_Rain_Sensor() {
  uint8_t result = modbus1.readInputRegisters(0x00, 1);
  // Serial.print(result,HEX);
  if (modbus1.ku8MBSuccess == result) {
    value.rain_value = modbus1.getResponseBuffer(0) / 10;
  } else Serial.println(" Rain Error");
}

void Read_Air_Sensor() {
  uint8_t result = modbus2.readInputRegisters(0x00, 2);
  // Serial.print(result,HEX);
  if (modbus2.ku8MBSuccess == result) {
    value.temperature = modbus2.getResponseBuffer(1) / 10;
    value.humidity = modbus2.getResponseBuffer(0) / 10;
  } else Serial.println(" Air Error");
}
```

รูปที่ 4.18 ตัวอย่างโปรแกรมที่เขียนใน Arduino (6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

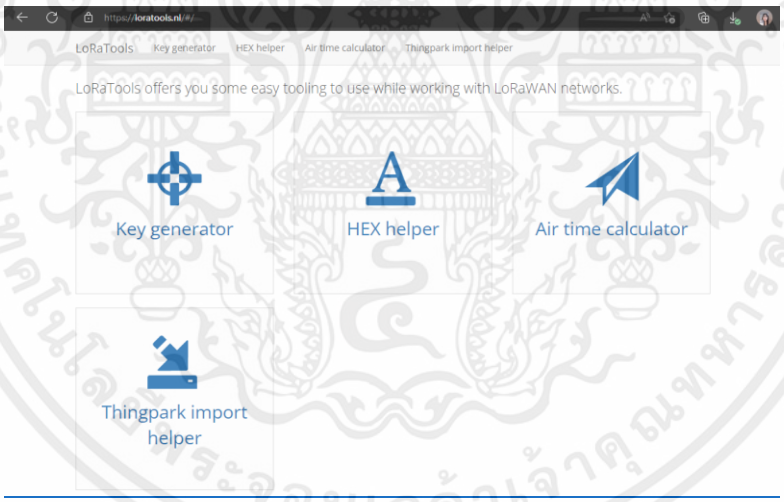
4.3.9. Function สำหรับการตรวจว่าฝนหยุดหรือไม่ โดนการคิดจากค่าฝนที่ได้ ถ้าไม่มีการเปลี่ยนแปลงเป็นเวลา 10 นาทีจะถือว่าฝนหยุดและสั่งการ reset ไปที่ sensor Ad

```
void check_reset() {
  if (value.rain_value == base && value.rain_value != 0) count += 1;
  if (count >= 900) {
    modbus2.writeSingleRegister(0x00, 90);
    count = 0;
  }
}
```

รูปที่ 4.19 ตัวอย่างโปรแกรมที่เขียนใน Arduino (7)

4.4 ส่งค่าที่อ่านได้ไปยัง Gateway และ Database

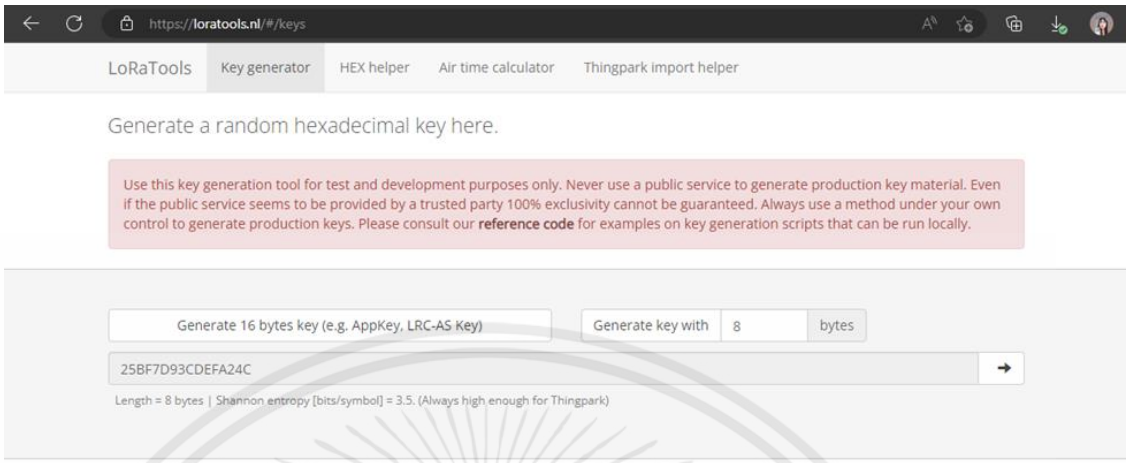
4.4.1. การสร้าง key สำหรับการสื่อสารผ่าน LoRa Class-A เข้าไปสร้าง key จากเว็บไซต์ https://loratools.nl



รูปที่ 4.20 ภาพจากเว็บ Loratool

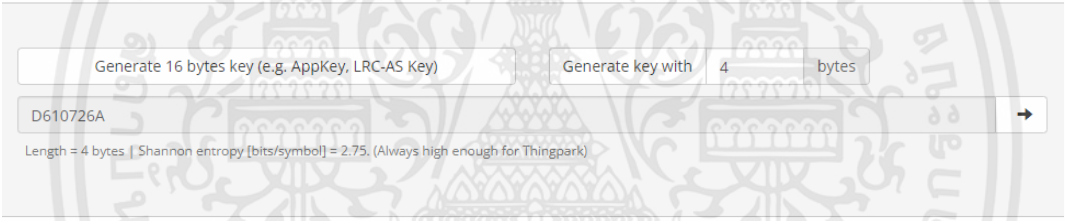
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2. ทำการสร้าง key 8 bytes มา 2 key เพื่อใช้เป็น APP Session Key และ Network Session Key



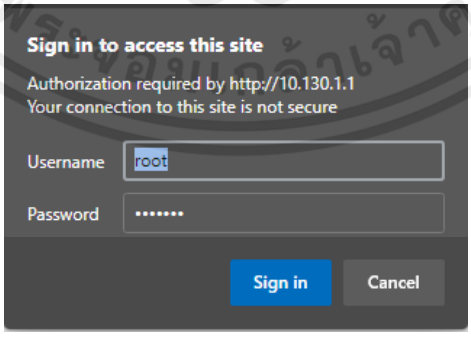
รูปที่ 4.21 ภาพการสร้าง Key 8 bytes

4.4.3. สร้าง key 4 bytes มา 1 key



รูปที่ 4.22 ภาพการสร้าง Key 4 bytes

4.4.4. การตั้งค่า LoRa Gateway เพื่อสร้างช่องทางรับข้อมูลที่ถูกส่งมาจากบอร์ด TTGO ผ่าน LoRa เข้าสู่ Default Gateway ของ LG308N LoRaWAN Gateway http://10.130.1.1

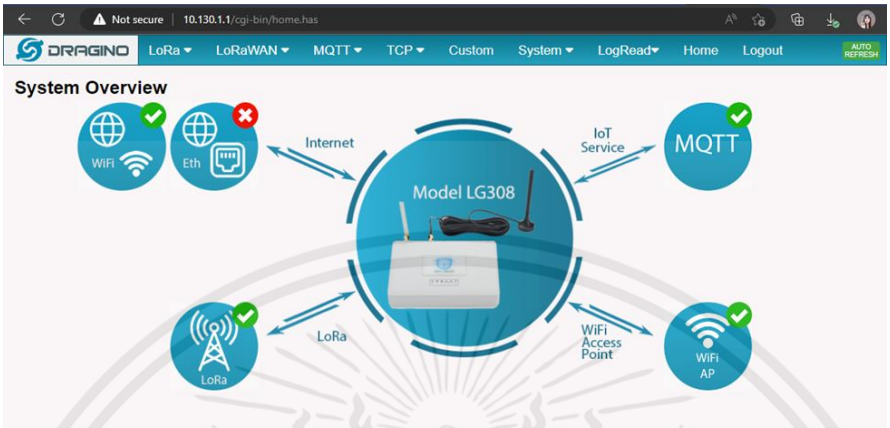


รูปที่ 4.23 ภาพการเข้าสู่ระบบของ Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

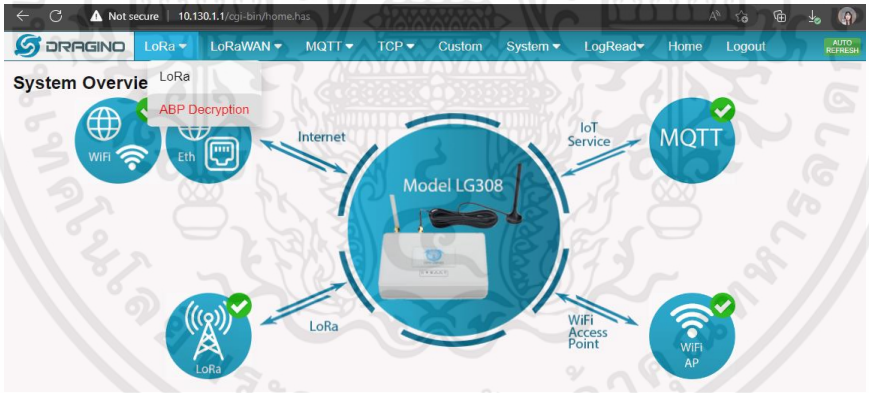
4.4.5. ทำการเข้าสู่ระบบของผู้ดูแลตามคู่มือของ Gateway ในที่นี้

Username: root Password: dragino



รูปที่ 4.24 ภาพ web gateway ของ dragino LoRa Gateway

4.4.6. เข้าไปที่เมนู LoRa >> ABP Decryption



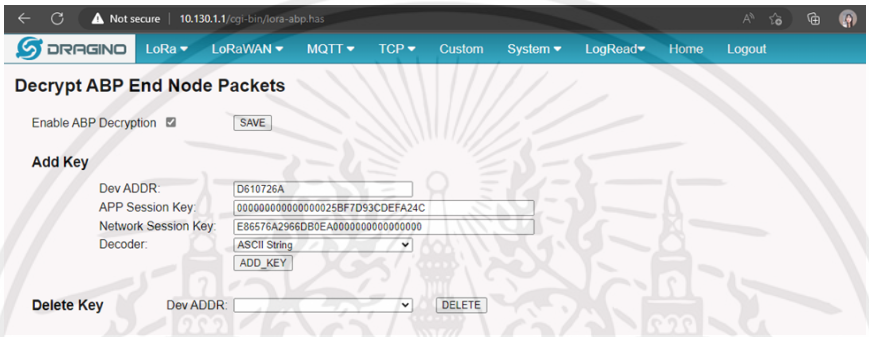
รูปที่ 4.25 ภาพการเข้าเมนู ABP Decryption

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.7. นำข้อมูลที่สร้างมาก่อนหน้านี้มาเติม

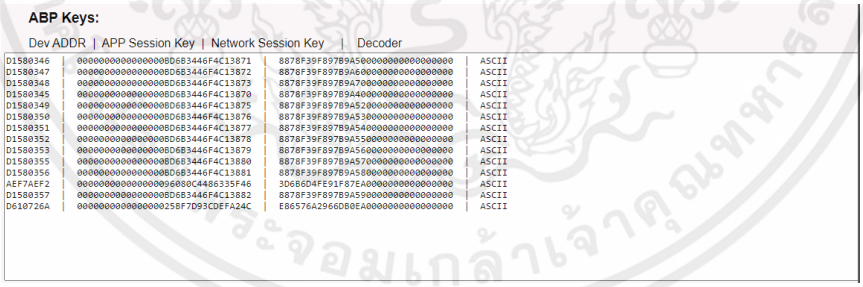
1. โดยนำข้อมูล 4 bytes เติม Dev ADDR
2. นำข้อมูล 8 bytes เติมในช่อง APP Session Key โดยเติม 0 ไปด้านหน้า 16 ตัว
3. และ นำข้อมูล 8 bytes เติมในช่อง Network Session Key โดยเติม 0 ไปด้านหน้า 16 ตัว
4. จากนั้นกด ADD_KEY เพื่อเพิ่มช่องทางการสื่อสารใหม่

■ หมายเหตุ : เป็นรูปแบบของการสื่อสาร LoRa Class-A



รูปที่ 4.26 ภาพการเข้าเมนู ABP Decryption

4.4.8. หลังจากนั้นช่องทางที่เพิ่มเข้าไปใหม่จะปรากฏขึ้นในช่อง ABP Keys



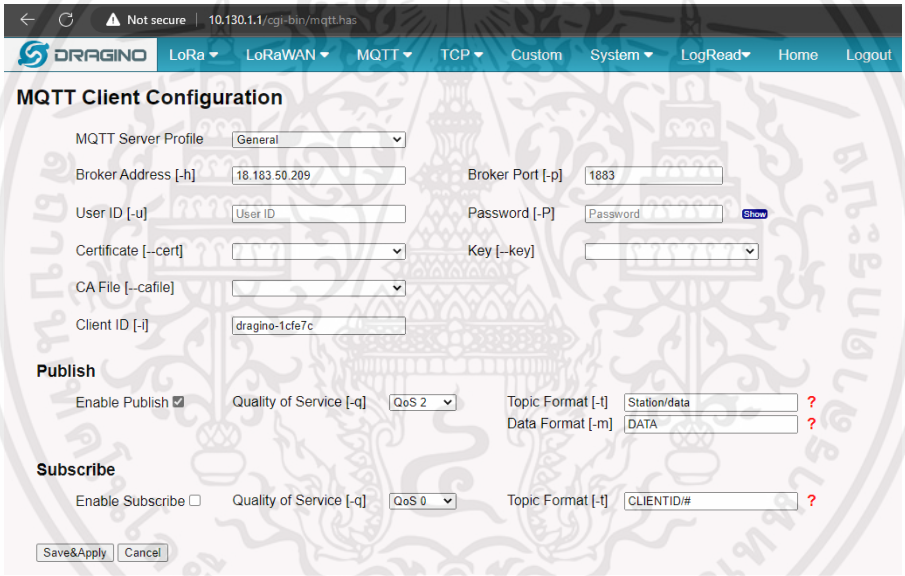
รูปที่ 4.27 Session Key ที่พร้อมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.9. หลังจากนั้นต้องทำการตั้งค่า Gateway ให้เป็น Wi-Fi Access Point เพื่อเชื่อมต่ออินเทอร์เน็ตสำหรับการส่งข้อมูลไปยัง Node-RED ด้วย MQTT ได้

ตั้งค่า MQTT ที่จะทำการ Forward ค่าจาก Gateway ออกไป

1) กำหนด MQTT broker ไปที่ broker.hivemq.com port 1883
2) ตั้ง Quality of Service เป็น 2 หมายถึงสำคัญที่สุด
3) กำหนด Publish Topic เป็น CLIENTID/CHANNEL/data
4) CLIENTID มาจาก ID ของ Gateway
5) CHANNEL มาจาก Device ADDR ที่สร้างไว้
6) data คือ address ของข้อมูล



รูปที่ 4.28 ภาพการตั้งค่า MQTT บน Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.10. หลังจากอ่านค่าจาก RS485 จะได้ข้อมูลที่เป็น อุณหภูมิ ความชื้น และปริมาณน้ำฝนแล้ว ในการส่งค่าต่อไปยัง Gateway ก่อนอื่น ต้องทำการเรียกใช้ library lorawan.h จากนั้นสร้างตัวแปรเก็บค่าเพื่อเข้าถึง Gateway ตามที่สร้างไว้ก่อนหน้านี้

```
*/
#include <lorawan.h>

//ABP Credentials
const char *devAddr = "D1580357";
const char *nwkSKey = "8878F39F897B9A590000000000000000";
const char *appSKey = "00000000000000000BD6B3446F4C13882";
```

รูปที่ 4.29 โปรแกรมสำหรับส่งค่าผ่าน LoRa บน Arduino (1)

4.4.11. ทำการเปิดใช้งาน LoRa Module ด้วยคำสั่งต่อไปนี้

```
if (!lora.init()) {
  Serial.println("RFM95 not detected");
  delay(2000);
  return;
}
lora.setDeviceClass(CLASS_A);
lora.setDataRate(SF12BW125);
lora.setFramePortTx(5);
lora.setChannel(MULTI);
lora.setTxPower(15);
lora.setNwkSKey(nwkSKey);
lora.setAppSKey(appSKey);
lora.setDevAddr(devAddr);
```

รูปที่ 4.30 โปรแกรมสำหรับส่งค่าผ่าน LoRa บน Arduino (2)

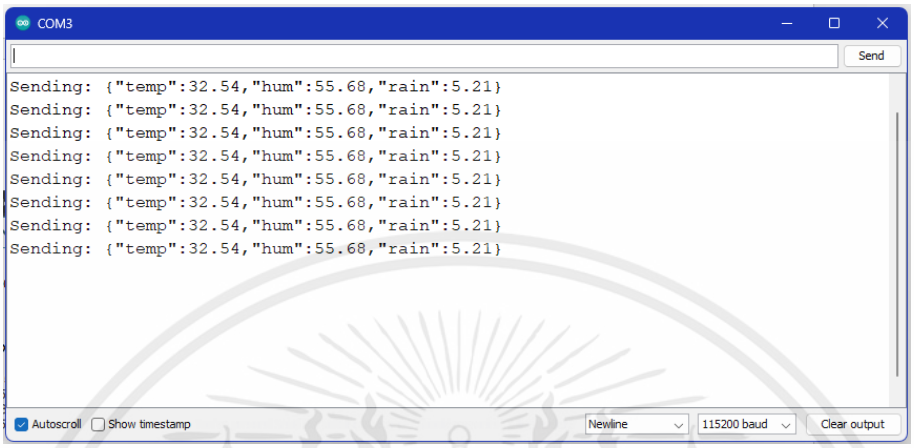
4.4.12. นำข้อมูลที่ผ่านมาจาก Sensor ทั้ง 3 มารวมเป็นข้อความ(String)ในรูปแบบ JSON ทำการส่งข้อมูลที่แปลงแล้วผ่าน Uplink จากนั้นรอการอัพเดทจาก RFM95

```
sprintf(myStr, "{\"temp\":%.2f,\"hum\":%.2f,\"rain\":%.2f}",temp,hum,rain);

Serial.print("Sending: ");
Serial.println(myStr);
lora.sendUplink(myStr, strlen(myStr), 0);


lora.update();
```

รูปที่ 4.31 โปรแกรมสำหรับส่งค่าผ่าน LoRa บน Arduino (3)

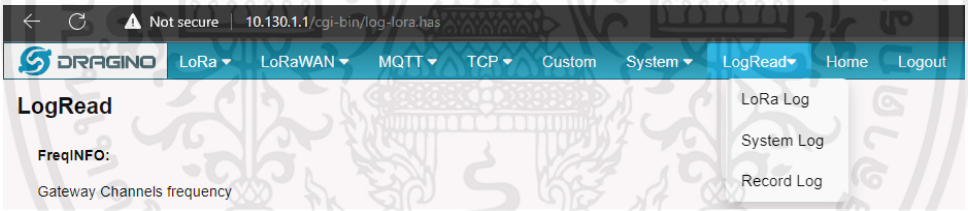
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.13. จากนั้นทำการอัพโหลด Code ลง บอร์ด TTGO หลังจากอ่านค่าจาก Sensor แล้ว บอร์ดจะส่งข้อมูลออกมา ดูผลได้จากทาง Serial Monitoring



รูปที่ 4.31 Serial Monitoring การส่งข้อมูลผ่าน LoRa

4.4.14. ข้อมูลจะถูกส่งไปยัง LoRa Gateway สามารถดูผลได้ผ่าน LoRa Log



รูปที่ 4.32 ภาพเมนู LoRa Log

รูปที่ 4.33 ภาพข้อมูลที่ผ่านเข้า Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

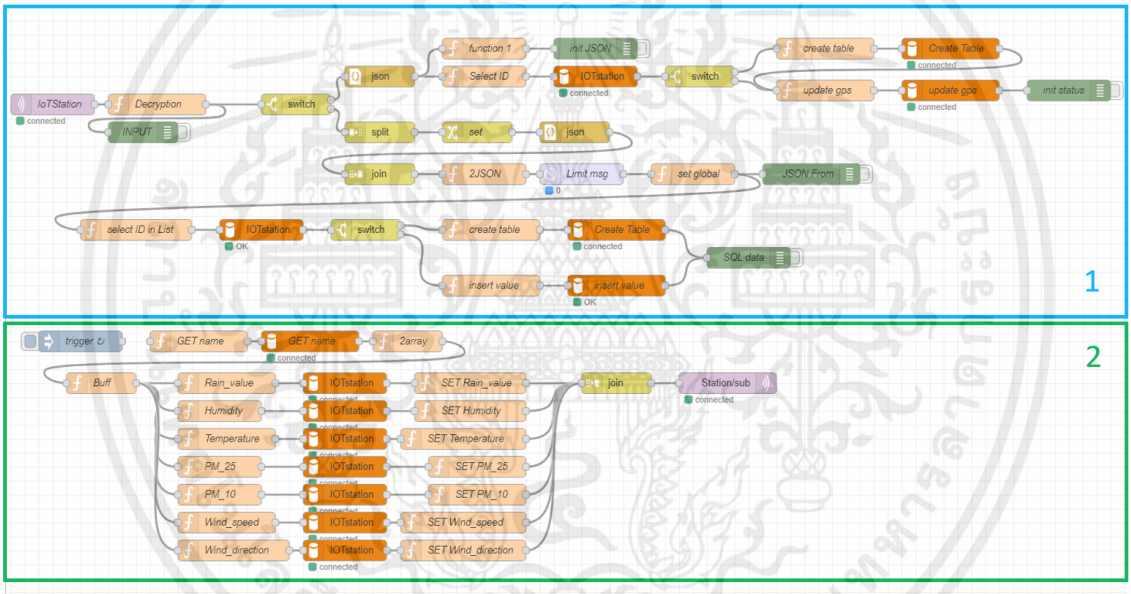
4.5 การแยกข้อมูลและแสดงผลขึ้นหน้า Dashboard

การแยกข้อมูลด้วย Node-RED หลังจากได้รับข้อมูลผ่าน MQTT จะทำการแยกข้อมูลและตรวจสอบข้อมูล โดยมีข้อมูล 2 ประเภทคือ ข้อมูลระบุตำแหน่งอุปกรณ์ผ่าน GPS และข้อมูลค่าสภาพอากาศ คัดแยกข้อมูลและทำงานใน 2 เงื่อนไข โดยจะแยกได้เป็นข้อมูลของอุปกรณ์ที่มีอยู่แล้ว และอุปกรณ์ใหม่ในระบบ

โดยการทำงานบน Node-RED จะถูกแบ่งเป็น 2 ส่วนหลักคือ

ส่วนที่ 1 ส่วนในการแยกและจัดเก็บข้อมูล

ส่วนที่ 2 ส่วนการดึงค่าออกจาก Database ไปเผยแพร่ผ่าน MQTT

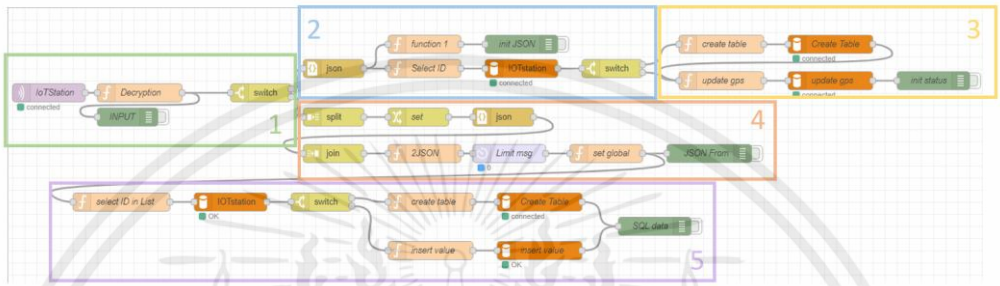


รูปที่ 4.34 การแยกข้อมูลด้วย Node-RED (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.1 ส่วนในการแยกและจัดเก็บข้อมูล

ส่วนนี้ทำหน้าที่รับข้อมูลจาก MQTT ที่ถูกส่งมาจาก LoRa Gateway และทำการแยกข้อมูลเป็น 2 ประเภทคือ ข้อมูลระบุตำแหน่งอุปกรณ์ผ่าน GPS และข้อมูลค่าสภาพอากาศ และทำการจัดการข้อมูลใน 2 เงื่อนไข คือ ข้อมูลของอุปกรณ์ที่มีอยู่แล้ว และอุปกรณ์ใหม่ในระบบ

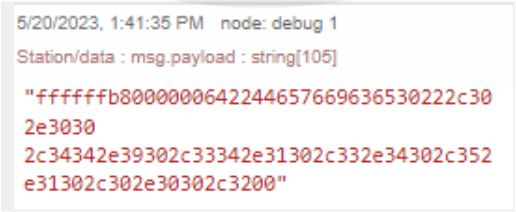


รูปที่ 4.35 การแยกข้อมูลด้วย Node-RED (2)

1). โดยจะทำการรับข้อมูลแล้วแปลงจากรหัส LoRa เป็นข้อมูลอักษรปกติ แล้วคัดแยกตามรูปแบบข้อมูลเป็นข้อมูล 2 ประเภท ที่จัดการข้อมูลตำแหน่งและจัดการข้อมูลค่าสภาพอากาศ

รูปที่ 4.36 การแยกข้อมูลด้วย Node-RED (3)

รับข้อมูลเข้ามาผ่าน MQTT เป็นข้อมูล 2 แบบ คือข้อมูลระบุตำแหน่งอุปกรณ์ผ่าน GPS และข้อมูลค่าสภาพอากาศ



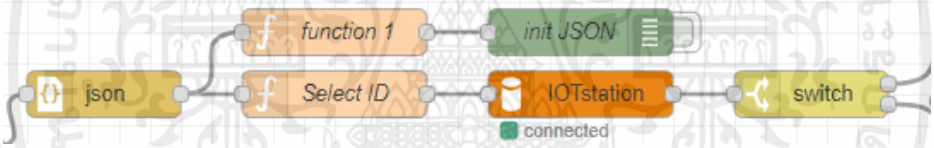
รูปที่ 4.37 ตัวอย่างข้อมูลที่ส่งจาก LoRa Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถอดรหัสข้อมูลจาก LoRa เป็นข้อมูลอักษรปกติ โดยข้อมูลตำแหน่งจะเป็นรูปแบบ JSON และข้อมูลค่าสภาพอากาศจะเป็น array ต่อเนื่องกันโดยมีลำดับตามที่กำหนด

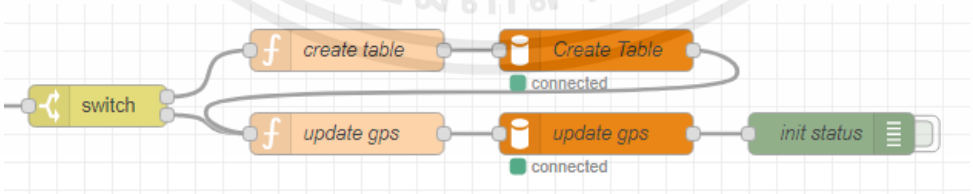
รูปที่ 4.38 ข้อมูลหลังจากถอดรหัส

2). รับค่าตำแหน่งมาจากส่วนแปลง plain text เป็น JSON และทำการส่งค่าตำแหน่งเป็นตัวแปร Global เพื่อไม่ให้ข้อมูลหายไปจากดึงค่าจาก Database มาทับใน Payload แล้วทำการดึงค่าจาก Database นำข้อมูลไปเทียบว่ามีอุปกรณ์นี้ใน Database หรือยัง แล้วทำการส่งไป 2 ทางในส่วนถัดไป



รูปที่ 4.39 การแยกข้อมูลด้วย Node-RED (4)

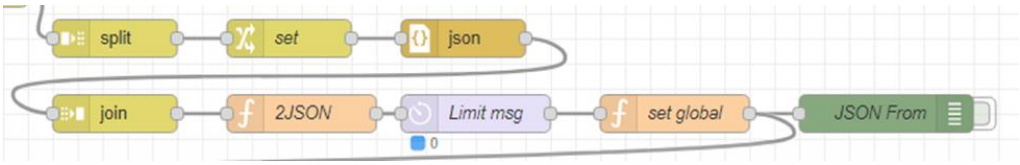
3). ถ้าข้อมูลไม่มีอุปกรณ์นี้ในระบบจะทำการสร้างตารางใหม่แล้ว update ค่าลงไปในฐานข้อมูล และถ้ามีอยู่ใน Database แล้วจะทำการ update ไปเลย



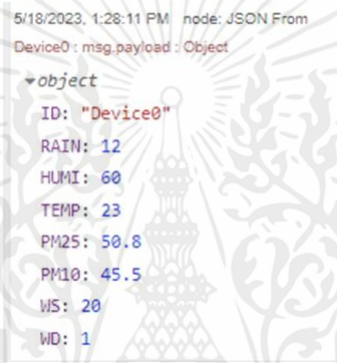
รูปที่ 4.40 การแยกข้อมูลด้วย Node-RED (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4). หากข้อมูลเป็นค่าสภาพอากาศ จะทำการแยกข้อมูลจาก array เป็นส่วนๆ แล้วแปลงเป็นข้อมูลในรูปแบบ JSON แล้วทำการส่งข้อมูลเป็นตัวแปร global เพื่อทำงานในส่วนถัดไป

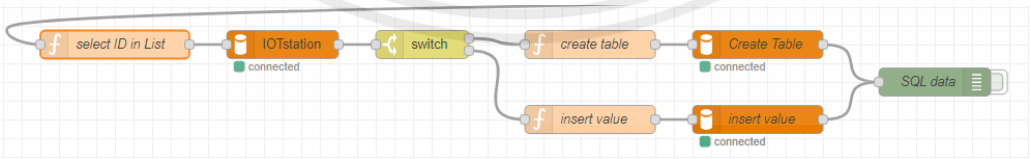


รูปที่ 4.41 การแยกข้อมูลด้วย Node-RED (6)



รูปที่ 4.42 การแยกข้อมูลด้วย Node-RED (7)

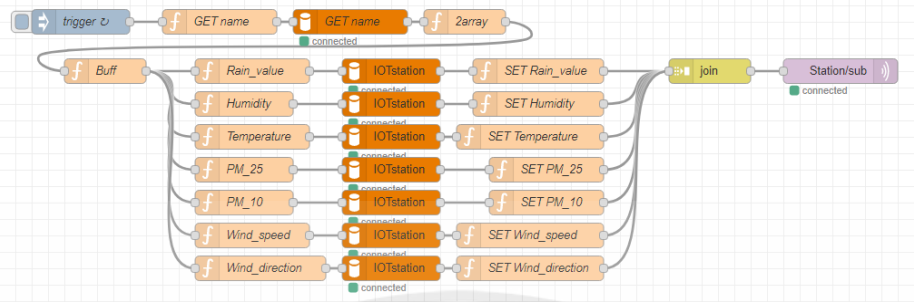
5). จากส่วนก่อนหน้าหลังจากได้ข้อมูลในรูป JSON แล้ว จะทำการตรวจสอบว่ามีอุปกรณ์ในฐานข้อมูลหรือยัง โดยการดึงค่าจาก Database แล้วทำส่งข้อมูลไป 2 ทาง ถ้ายังไม่มีในฐานข้อมูลจะทำการสร้างตารางใหม่ ถ้ามีแล้วจะทำการ insert ข้อมูลใหม่เข้าในตารางของอุปกรณ์นั้น



รูปที่ 4.43 การแยกข้อมูลด้วย Node-RED (8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

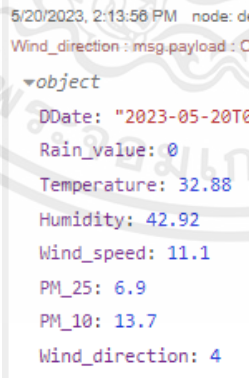
4.5.2 ส่วนการดึงค่าออกจาก Database ไปเผยแพร่ผ่าน MQTT



รูปที่ 4.44 การแยกข้อมูลด้วย Node-RED (9)

โดยจะมีตัว trigger ทุกๆ 1 นาที ทำการสั่งการทำงาน ให้มีการดึงชื่ออุปกรณ์ทั้งหมดใน database มาแล้วแปลงเป็นรายชื่อรูปแบบ CSV และส่งไปดึงข้อมูลค่าเฉลี่ยปัจจุบันของแต่ละค่าสภาพอากาศ หลังจากได้ข้อมูลทั้งหมดจะทำการรวมข้อมูลเป็นรูปแบบ JSON แล้ว publish ออกผ่าน MQTT

รูปที่ 4.45 ข้อมูลที่ดึงมาในรูปแบบ CSV

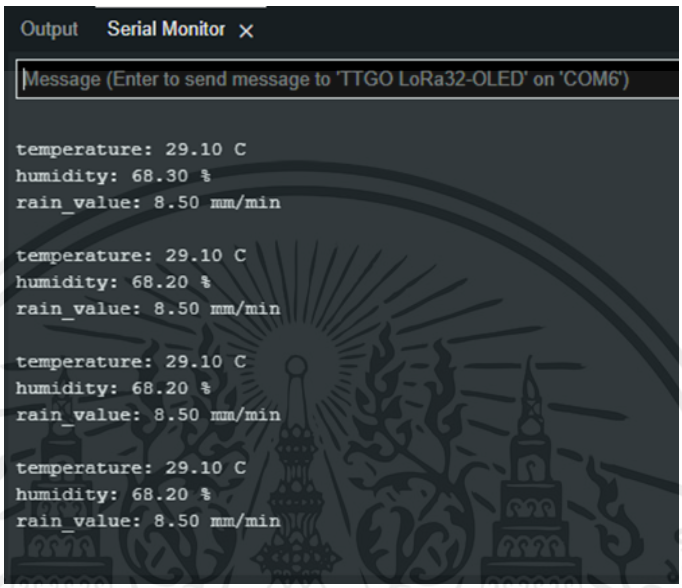


รูปที่ 4.46 ข้อมูล Publish บน MQTT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลทดลองกำหนดค่าและอ่านค่าจากเซนเซอร์

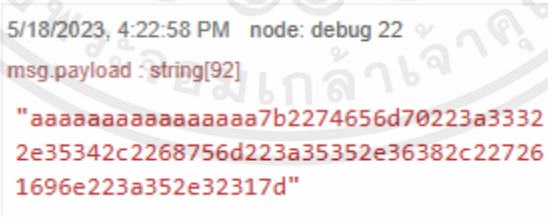
หลังจากทำการกำหนดค่าให้เซนเซอร์ 3 ประเภท คือ เซนเซอร์วัดอุณหภูมิ เซนเซอร์วัดความชื้น และเซนเซอร์วัดปริมาณน้ำฝน จากนั้นทำการวัดค่าจริงได้สำเร็จ ซึ่งทำให้ได้ผลลัพธ์ดังนี้



รูปที่ 4.47 ตัวอย่างผลลัพธ์การอ่านค่าเซนเซอร์

4.7 ผลการทดลองส่งข้อมูลที่อ่านได้จากเซนเซอร์ผ่าน LoRa ไปยัง Node-Red

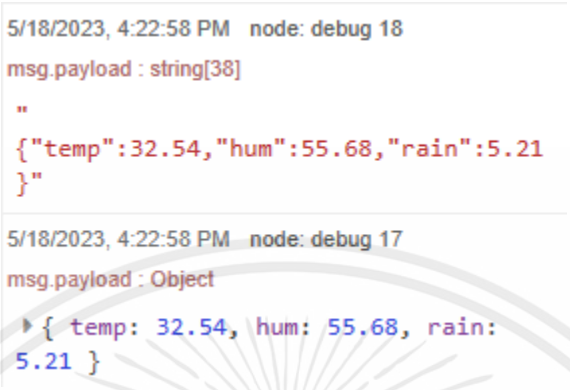
หลังจากส่งข้อมูลผ่าน LoRaWAN ได้สำเร็จแล้วใช้ Node-RED เข้าไป subscribe MQTT topic ของ LoRa Gateway เมื่อได้รับข้อมูลผ่าน MQTT จาก LoRa Gateway มาแล้ว ข้อมูลที่ได้จะออกมาเป็นข้อมูลฐาน 16



รูปที่ 4.48 ตัวอย่างข้อมูลที่ส่งมาจาก LoRa Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นหลังจากนำข้อมูลมาตัดส่วนที่ไม่จำเป็นออกและแปลงข้อมูลกลับมาเป็นข้อความ จะได้ผลลัพธ์ออกมาเป็น String รูปแบบ JSON ดังรูปที่ 4.49



รูปที่ 4.49 ตัวอย่างผลลัพธ์การถอดรหัสข้อมูลที่ส่งมาจาก LoRa Gateway

4.8 การจัดเก็บข้อมูลและดึงค่าขึ้นแสดงผลบน Dashboard

4.8.1 การจัดเก็บข้อมูล

จากการออกแบบข้อมูลจะถูกแบ่งเป็น 2 แบบคือข้อมูลระบุตำแหน่งของอุปกรณ์ และข้อมูลค่าสภาพอากาศ โดยจะจัดเก็บในตารางดังนี้

ข้อมูลระบุตำแหน่ง จะจัดเก็บ ID ซึ่งเป็นชื่อ Device ที่ถูกส่งมาจากเสาแทนชื่อของอุปกรณ์ และมีข้อมูล position เป็นข้อมูลตำแหน่ง GPS รูปแบบ latitude longitude

ID	position_lat	position_lon
Device0	13.727419	100.772843
Device1	13.727418	100.771848
NULL	NULL	NULL

รูปที่ 4.50 ผลการจัดเก็บข้อมูล (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลค่าสภาพอากาศ จะทำการจัดเก็บค่าเวลาที่ทำการบันทึกค่าของข้อมูลนั้น และค่าจากเซนเซอร์ทั้งหมด โดยจะเป็นรูปแบบของ 1 ตารางต่อ 1 อุปกรณ์ และจะทำการสร้างตารางใหม่อัตโนมัติหากมีอุปกรณ์ลงทะเบียนใหม่ ผ่านระบบบน Node-RED

Date	Rain_value	Humidity	Temperature	PM_25	PM_10	Wind_speed	Wind_direction
2023-05-20 07:09:39	0	44.7	34.4	3.3	4.8	0	2
2023-05-20 07:10:09	0	44.6	34.4	3.5	5.1	0	2
2023-05-20 07:10:42	0	44.5	34.4	3.4	5.1	0	2
2023-05-20 07:11:12	0	45.2	34.1	3.4	4.9	0	2
2023-05-20 07:11:39	0	45.3	34	3.5	5	0	2
2023-05-20 07:12:18	0	45.4	34	3.6	5.1	0	2
2023-05-20 07:12:44	0	45.7	33.9	3.5	5	0	2
2023-05-20 07:13:22	0	45.7	33.8	3.7	5.2	0	2
2023-05-20 07:13:43	0	46	33.7	3.7	5.4	0	2
2023-05-20 07:14:11	0	46.1	33.7	3.5	5	0	2
2023-05-20 07:14:50	0	46.4	33.5	3.6	5.4	0	2
2023-05-20 07:15:14	0	46.3	33.6	3.5	4.8	0	2
2023-05-20 07:15:44	0	46	33.7	3.8	5.5	0	2
2023-05-20 07:16:17	0	45.7	33.9	3.8	5.4	0	2
2023-05-20 07:16:56	0	45.4	34.1	3.5	5	0	2
2023-05-20 07:17:20	0	45.3	34.1	3.8	5.6	0	2
2023-05-20 07:17:56	0	45.4	34.1	3.5	5	0	2
2023-05-20 07:18:17	0	44.8	34.2	3.6	5.4	0	2
2023-05-20 07:18:50	0	44.4	34.3	3.4	5	0	2
2023-05-20 07:19:17	0	44.6	34.3	3.7	5.2	0	2
2023-05-20 07:19:50	0	44.7	34.3	3.7	5.2	0	2
2023-05-20 07:20:26	0	44.6	34.3	3.7	5.2	0	2
2023-05-20 07:20:50	0	44.6	34.4	3.5	5.2	0	2
2023-05-20 07:21:32	0	44.7	34.4	3.7	5.2	0	2

รูปที่ 4.51 ผลการจัดเก็บข้อมูล (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8.2 การแสดงผลข้อมูล

ใช้ Grafana Dashboard ในการแสดงผล โดยตั้งค่าให้รองรับการดึงค่าจาก MySQL ดึงข้อมูลผ่านการทำงานที่วางไว้และตั้งค่าแสดงผลเป็นแบบมีค่า Thresholds ระดับความอันตรายของแต่ละค่า

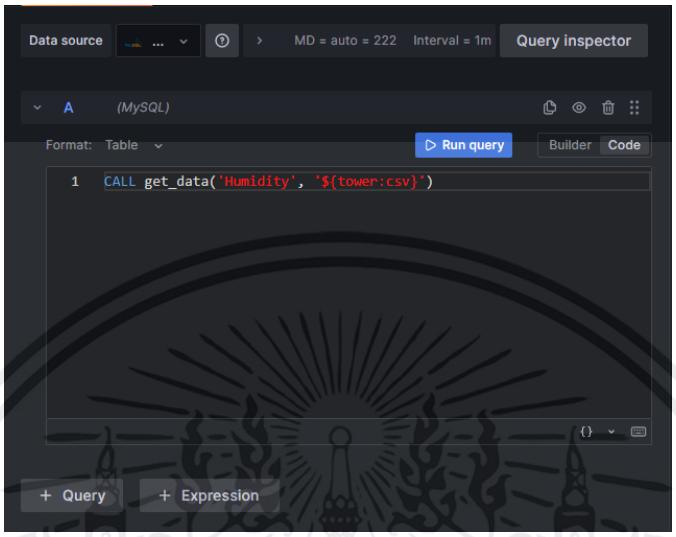
รูปที่ 4.52 หน้า Grafana

ในการตั้งค่าการดึงค่าข้อมูลจะมี 2 จัดสำคัญ คือคำสั่งดึงข้อมูลหรือ query และการตั้งค่าสภาพการแสดงผลข้อมูล ดังรูปส่วนการดึงจะอยู่ด้านล่าง และส่วนตั้งค่าแสดงผลจะอยู่ด้านขวา

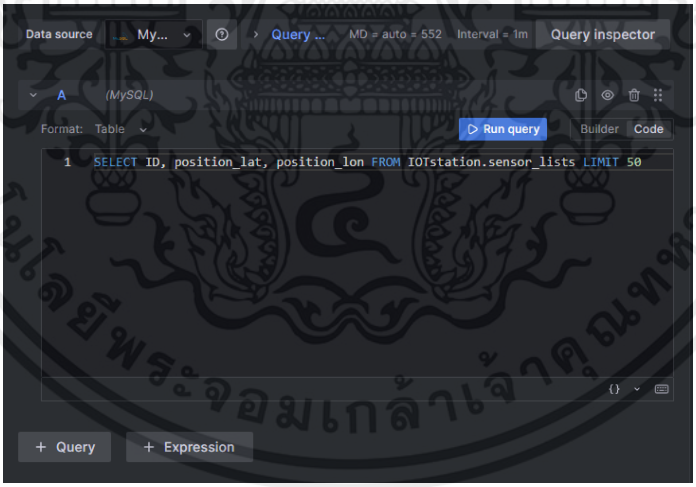
รูปที่ 4.53 หน้าตั้งค่า Grafana

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการใส่คำสั่ง query เป็นคำสั่งที่ดึงค่ามาจาก Database โดยเลือกเป็นการใส่คำสั่งแบบ Code ที่จะทำการเลือกคำสั่งได้เอง โดยใส่คำสั่ง query ลงไปได้เลย



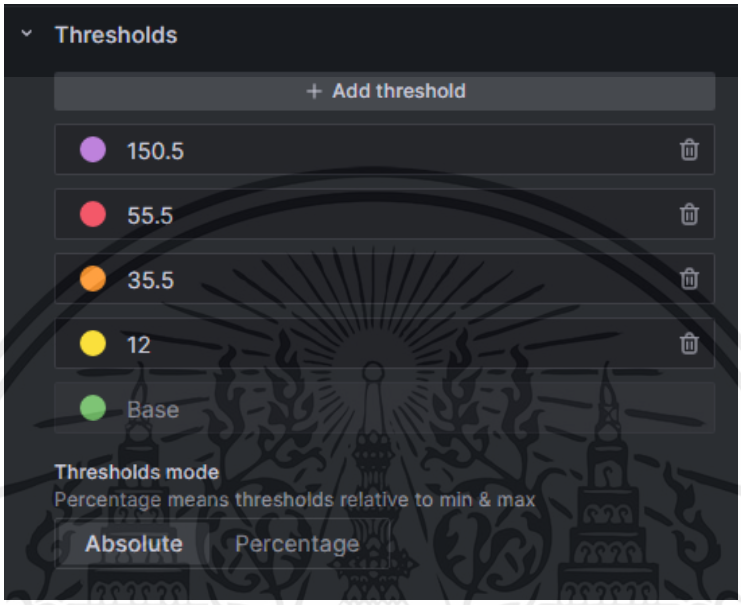
รูปที่ 4.54 หน้าตั้งค่า Grafana (1)



รูปที่ 4.55 หน้าตั้งค่า Grafana (2)

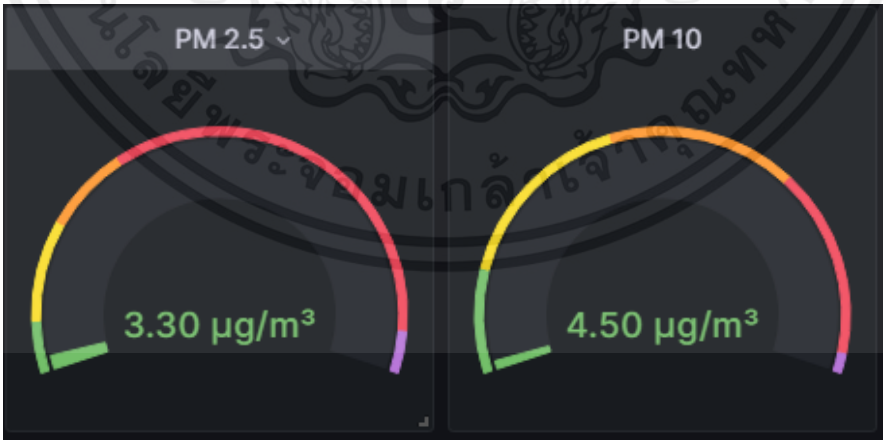
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการตั้งค่าแสดงผล จะถูกตั้งค่าทางด้านขวา เป็นรูปแบบในการแสดงผลของแต่ละข้อมูล ทั้งหน่วยของข้อมูล ค่า max min ของข้อมูลนั้น ชื่อ Title ของข้อมูล และตั้งค่า Thresholds ความอันตรายของข้อมูล โดยตั้งได้ทั้งตามค่าของข้อมูลหรือ เป็น % จาก min max



รูปที่ 4.56 Thresholds

จะทำการแสดงผลค่าอันตรายตามเขตสีรอบ Gauge



รูปที่ 4.57 หน้าตั้งค่า Grafana

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

เนื่องจากปัจจุบัน โลกได้มีการพัฒนาเข้าสู่สังคมระบบ IOT มากขึ้น หลายๆ อย่างสามารถทำผ่านโทรศัพท์เครื่องเดียวได้ และทางผู้ดำเนินงาน ได้เคยพัฒนาเสากระจายเสียงผ่าน VoIP (ตัวกลางการสื่อสารผ่าน Internet โดยใช้ IP) ทางผู้ดำเนินงานจึงเล็งเห็นการพัฒนาระบบเสาเดิมที่มีเพียงระบบกระจายเสียง ให้สามารถตรวจสภาพอากาศ และสภาพแวดล้อมต่างๆ ได้ โดยการออกแบบระบบและพัฒนาชุดตรวจวัดสภาพอากาศของเครื่องมือตรวจวัด (Sensor) มาตรฐาน RS485 เพื่อทำงานร่วมกับเสากระจายเสียงผ่าน VoIP เดิม โดยทำการอ่านค่าจากเครื่องมือวัด (Sensor) ด้วยอุปกรณ์ควบคุมขนาดเล็ก (Microcontroller) และทำการส่งค่าที่ได้ไปเก็บข้อมูลบนฐานข้อมูล (Database) และแสดงผลได้ ผ่านการสื่อสาร LoRaWAN เพื่อให้มีการใช้พลังงานต่ำและลดต้นทุนการผลิต ทำให้มีการเข้าถึงการใช้งานได้ง่ายขึ้น

โดยผลการดำเนินงานทำให้สามารถวัดสภาพอากาศได้หลากหลาย ทั้งปริมาณฝุ่น PM2.5 PM10 ปริมาณน้ำฝน อุณหภูมิ ความชื้น ตำแหน่งที่ตั้งของเสา ความเร็วและทิศทางลมได้ และนำข้อมูลที่วัดได้เหล่านั้นส่งผ่านคลื่นวิทยุ LoRa ไปยัง Gateway เพื่อนำข้อมูลเข้าสู่ระบบเก็บข้อมูลใน AWS และนำข้อมูลไปแสดงผลบน Dashboard ของ Grafana เพื่อให้ผู้ใช้สามารถเข้ามาดูสภาพอากาศในบริเวณเสาผ่านอินเตอร์เน็ตได้เพียงแค่มีลิงค์

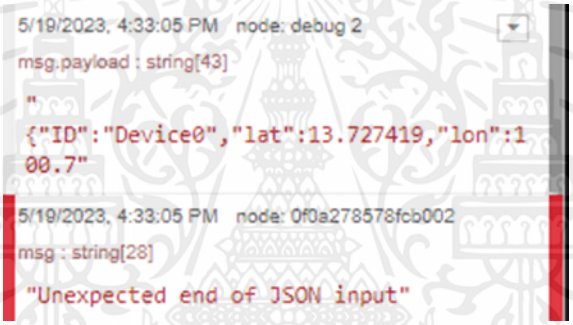
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาที่พบในการทำการทดลอง

5.2.1 การส่งข้อมูลด้วย LoRa ในบางครั้งเกิดความผิดพลาด

การส่งข้อมูลผ่าน LoRa ที่ในบางครั้งข้อมูลที่ได้มีการคลาดเคลื่อน ไม่ได้ข้อมูลตามที่ต้องการเช่น ข้อมูลเพี้ยน ข้อมูลขาดหาย ทำให้ระบบทำงานไม่ได้ตามผลที่ต้องการ

รูปที่ 5.1 ตัวอย่างข้อมูลที่ผิดเพี้ยนทำให้ระบุตัวอุปกรณ์ผิดพลาด



รูปที่ 5.2 ข้อมูลที่ขาดไม่เป็นไปตามรูปแบบ ทำให้ระบบขัดข้อง

5.2.2 ข้อจำกัดของ Grafana

ปัญหาของ Dashboard ที่ไม่สามารถเรียกข้อมูลจาก database ได้ละเอียดไม่พอ Grafana ทำงานเป็นคำสั่งชั้นเดียวบน MySQL ทำให้ระบบที่ออกแบบมาทำงานกับหลายตารางทำงานได้ยาก ไม่สามารถเรียกข้อมูลในรูปแบบที่ซับซ้อนได้ เช่น การดึงค่าจากหลายๆ ตารางมารวมกันแล้วเฉลี่ย ณ เวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 **แนวทางการแก้ไขปัญหา**

5.3.1 การส่งข้อมูลด้วย LoRa ในบางครั้งเกิดความผิดพลาด

ทำตัวกรองระบบให้ข้อความที่ไม่เป็นไปตามรูปแบบที่ว่าไว้ ถูกปัดตกไปก่อนที่จะไปส่งผลกระทบต่อส่วนอื่น และหากลดรอบการส่งลงให้เหมาะสม จะลดโอกาสที่จะทับซ้อนกับข้อมูลอื่นๆ ที่ส่งผ่านคลื่นวิทยุในย่านใกล้เคียงได้

5.3.2 ข้อจำกัดของ Grafana

ทำ function Stored Procedures ของ MySQL ให้รองรับกับการส่งคำขอของ Dashboard เป็น function ที่ทำงานแบบ dynamic ได้ และเรียกใช้งานในคำสั่งชั้นเดียว ทำให้ทำงานกับ Grafana ได้

```
CALL get_data_WD('Humidity', '${tower:csv}')
```

รูปที่ 5.3 คำสั่งเรียกใช้งานข้อมูลบน Grafana

```
CREATE DEFINER=`admin`@`%` PROCEDURE `get_data`(IN variable VARCHAR(255), IN tables VARCHAR(255))
BEGIN
    SET @sql = '';
    SELECT GROUP_CONCAT(
        DISTINCT CONCAT(
            'SELECT CAST(DATE_FORMAT(Date , \'%Y-%m-%d %H:%i:00\') as datetime) AS DDate , ROUND(AVG(',variable,'),2) AS `',variable,'` FROM IOTstation.values_',
            table_name,
            ' GROUP BY DDate'
        )
        SEPARATOR ' UNION '
    ) INTO @sql
    FROM (
        SELECT TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(tables, ',', n.n), ',', -1)) AS table_name
        FROM (
            SELECT a.N + b.N * 10 + 1 AS n
            FROM (
                SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3
                UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7
                UNION ALL SELECT 8 UNION ALL SELECT 9
            ) AS a
            CROSS JOIN (
                SELECT 0 AS N UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL SELECT 3
                UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7
                UNION ALL SELECT 8 UNION ALL SELECT 9
            ) AS b
            ORDER BY n
        ) AS n
        WHERE n <= 1 + LENGTH(tables) - LENGTH(REPLACE(tables, ',', ''))
    ) AS t;

    SET @sql = CONCAT(
        'SELECT DDate, ROUND(AVG(`',variable,'`), 2) AS `',variable,'` FROM (',
        @sql,
        ') AS t GROUP BY DDate ORDER BY DDate'
    );

    PREPARE stmt FROM @sql;
    EXECUTE stmt;
    DEALLOCATE PREPARE stmt;
END
```

รูปที่ 5.4 ตัวอย่างคำสั่งใน MySQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 **ข้อเสนอแนะในการพัฒนา**

เสาไฟทางเดินตรวจอากาศอัจฉริยะสามารถทำงานได้อย่างถูกต้องตามขอบเขตของโครงงานแล้ว แต่ยังสามารถพัฒนาเพื่ออำนวยความสะดวกให้ผู้ใช้งานได้มากขึ้นเช่น

1. ทำจอแสดงผลในบริเวณเสา เพื่อให้ผู้ใช้บริเวณใกล้เคียงสามารถดูค่าต่าง ๆ ได้สะดวกยิ่งขึ้น
2. ทำการแจ้งเตือนอันตรายให้ผู้ใช้บริเวณใกล้เคียงหากเกิดสถานการณ์อันตรายเช่น ปริมาณฝุ่นสูงกว่าปกติ อากาศร้อนผิดปกติ หรือลมแรงผิดปกติ จะทำการเตือนด้วยวิธีการใดวิธีการหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Miniature innovation. 2021, July 09. **Weather Station Sensor, VMS-300BYH-M** [On-line]. Available: https://www.miniature-solution.com/product/28/

Miniature innovation. 2021, July 16. **ABS RS485 ABS rain sensor, MI-RAINBOX485-ABS** [On-line]. Available: https://www.miniature-solution.com/product/55/

Miniature innovation. 2021, July 11. **Wind Direction, VMS-3000-FX-N01** [On-line]. Available: https://www.miniature-solution.com/product/39/

Miniature innovation. 2021, April 03. **Wind speed Three cups outdoor weather station, VMS-3000-FS-N01** [On-line]. Available: https://www.miniature-solution.com/product/12/

Nectec, ดร.ธีรเชษฐ์ สูรพันธุ์, ณัฐพล ตันสังวรณ์, ศศิวิภา หาสุข, วลัยลักษณ์ คงพระจันทร์. 2020 May 24. **การสื่อสารในงานอุตสาหกรรมด้วยโพรโทคอล Modbus** [On-line]. Available: https://www.nectec.or.th/news/news-public-document/modbus-protocol.html

What is RS485? [On-line]. 2017, May 10. Available: https://www.omi.co.th/th/article/rs485

What is LoRa? [On-line]. 2023, May 22. Available: https://en.wikipedia.org/wiki/LoRa

Kritsada Arjchariyaphat. 2018, Feb 27. **How LoRaWAN work?** [On-line]. Available: https://medium.com/deaware/lora-lorawan-คืออะไร-มารู้จักกันดีกว่า-98d20055a4ca

Antares LoRaWAN Library for ESP32. [On-line]. 2023, Feb 2. https://antares.id/id/lora-esp32-tutorial.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

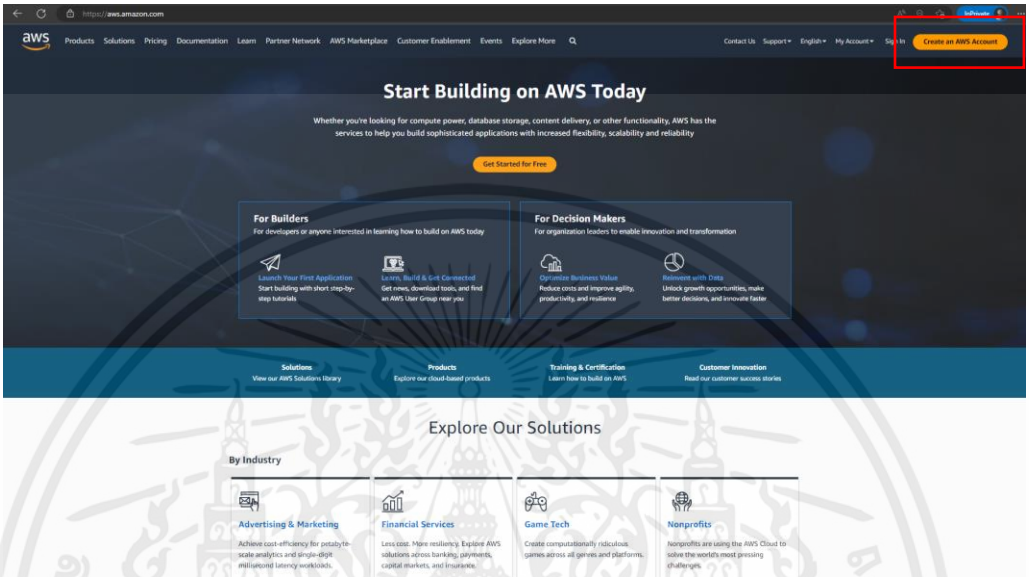
ภาคผนวก ก.

การติดตั้งโปรแกรมที่ใช้ในโครงงาน

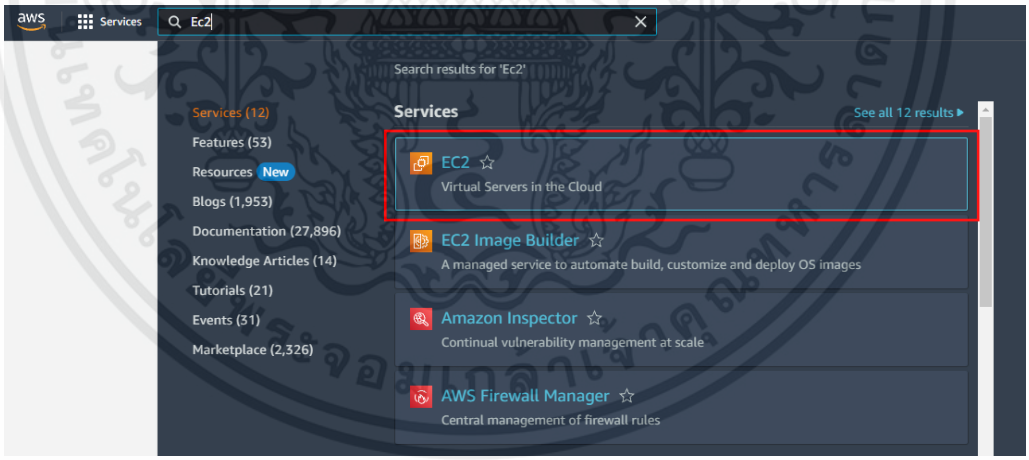
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างบัญชี AWS เปิดใช้งาน Ec2 สำหรับทำเป็น Cloud Server

1. เข้าไปที่เว็บไซต์ของ AWS https://aws.amazon.com และคลิกที่ Create an AWS Account หรือถ้ามีบัญชีอยู่แล้วให้กด Sign in to the console

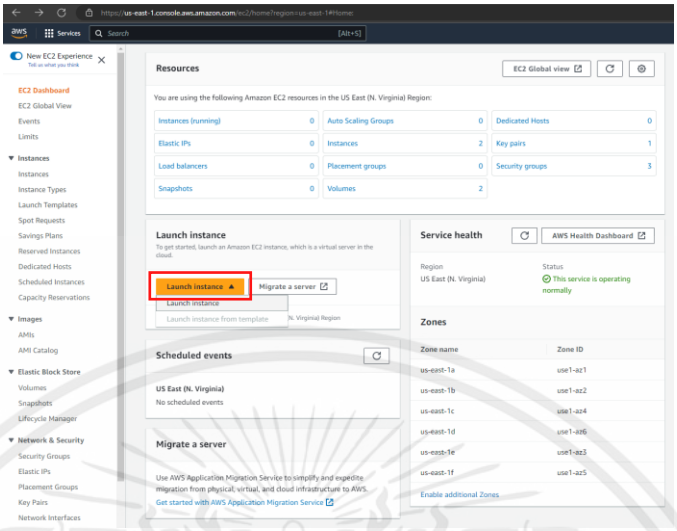


2. ทำการสร้างบัญชีและเข้าสู่ระบบให้เรียบร้อยและทำการค้นหา Service ที่ชื่อว่า Ec2

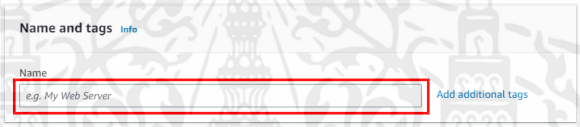


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

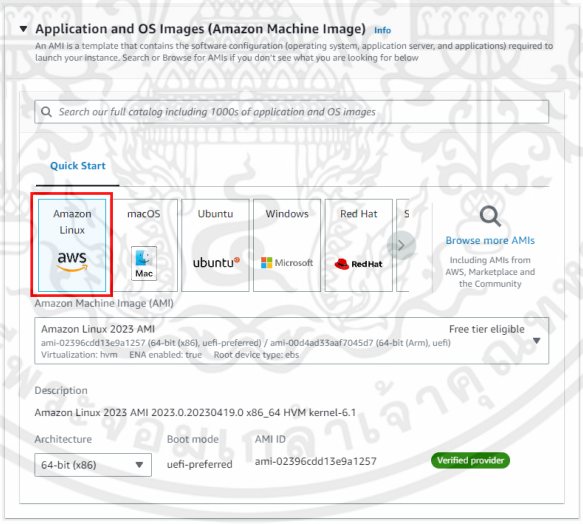
3. ไปที่ Launch instance และเลือกตัวเลือก Launch instance



4. ทำการตั้งชื่อ Instance ของเรา

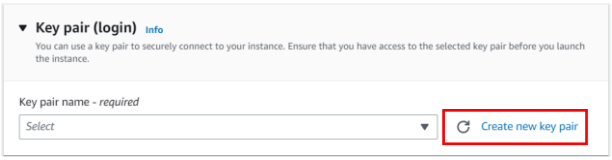


5. เลือกระบบปฏิบัติการของ Instanceในที่นี้เลือกเป็น Amazon Linux

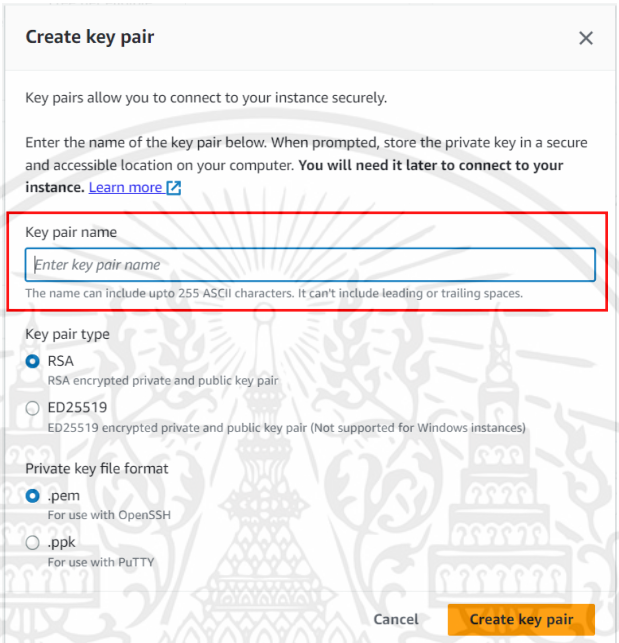


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

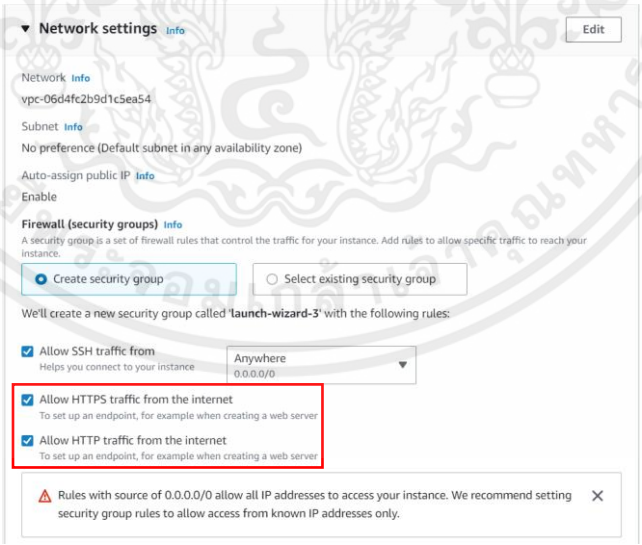
6. ทำการสร้าง key pair



7. ตั้งชื่อ Key pair ของเรา

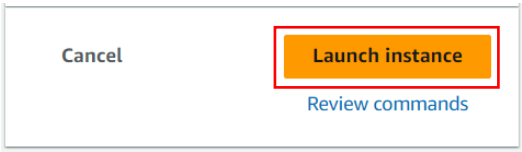


8. คลิก Checkbox เพื่ออนุญาตให้ใช้พอร์ต HTTP และ HTTPS

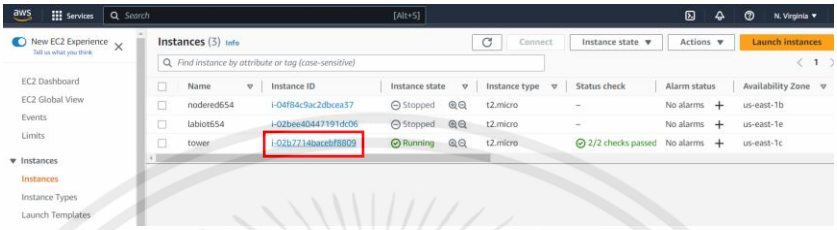


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

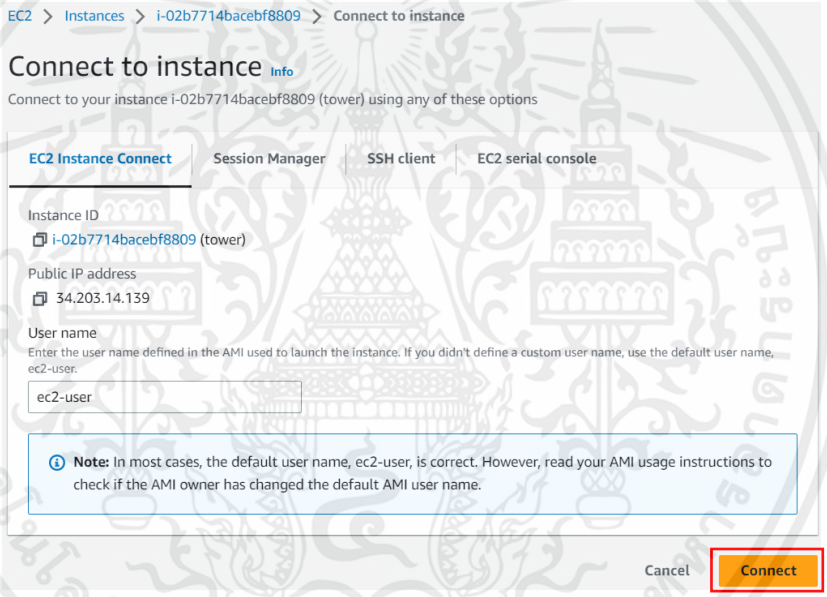
9. กด Launch instance เพื่อเปิดใช้งาน Instance



10. คลิกที่ Instance ID ของ Instance ที่เราสร้าง

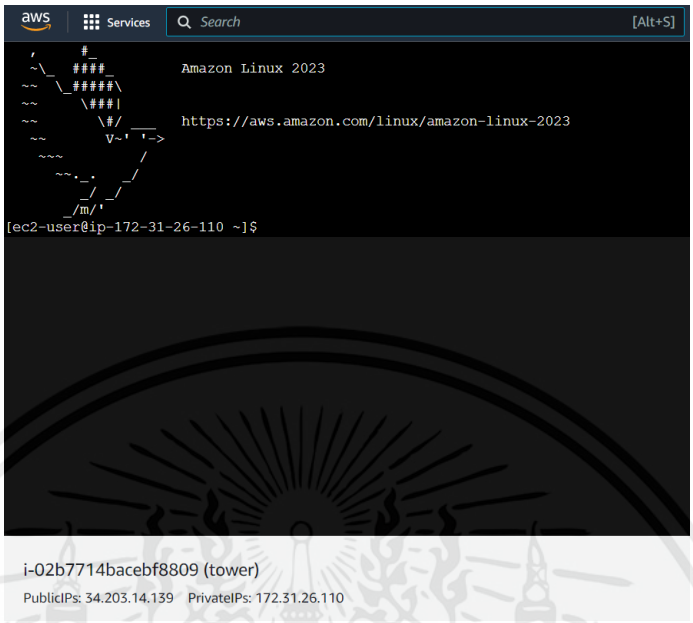


11. ไปที่ EC2 Instance Connect และคลิก Connect เพื่อเข้าใช้งาน SSH



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. เมื่อมีหน้าต่างตามภาพปรากฏขึ้น แสดงให้เห็นว่า SSH ของ Instance พร้อมใช้งานแล้ว

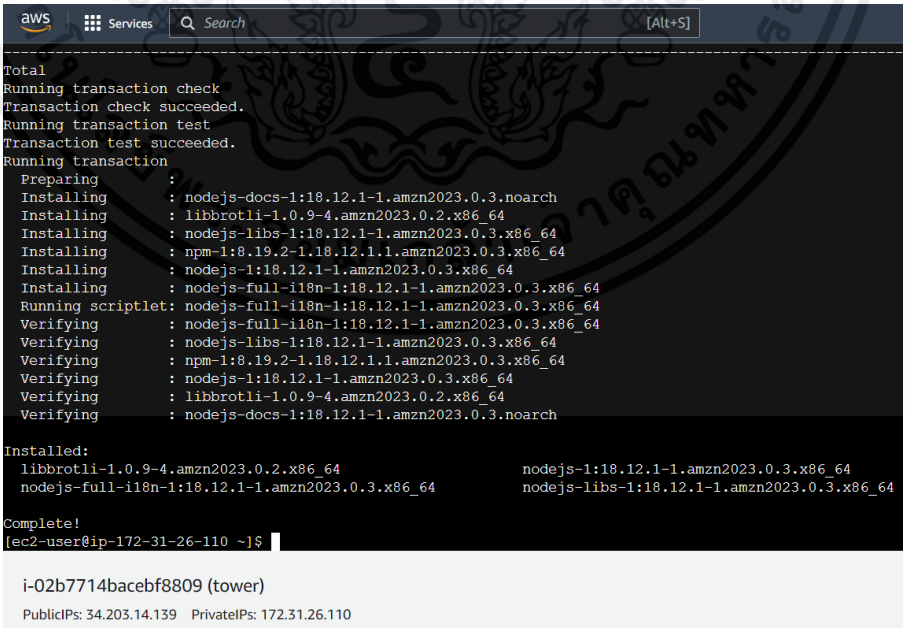


การติดตั้ง Node-RED ใน AWS Ec2

1. ทำการเปิดหน้าต่าง SSH ของ Ec2 ขึ้นมา จากนั้นพิมพ์คำสั่งเพื่อทำการติดตั้ง Node.js

   sudo yum install nodejs npm

   พิมพ์ Y และกด Enter เมื่อติดตั้งสำเร็จจะปรากฏข้อความดังภาพ



   ตรวจสอบว่า Node.js ติดตั้งสำเร็จหรือไม่ด้วยคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

node -v

```
[ec2-user@ip-172-31-26-110 ~]$ node -v
v18.12.1
[ec2-user@ip-172-31-26-110 ~]$
```

2. พิมพ์คำสั่งเพื่อติดตั้ง Node-red

sudo npm install -g --unsafe-perm node-red

```
[ec2-user@ip-172-31-26-110 ~]$ sudo npm install -g --unsafe-perm node-red

added 292 packages, and audited 293 packages in 15s

40 packages are looking for funding
  run `npm fund` for details

7 vulnerabilities (4 low, 3 moderate)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
npm notice
npm notice New major version of npm available! 8.19.2 -> 9.6.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.6.5
npm notice Run npm install -g npm@9.6.5 to update!
npm notice
[ec2-user@ip-172-31-26-110 ~]$
```

3. ทำการติดตั้ง pm2 เพื่อใช้ในการให้ Node-red ทำงานเบื้องหลังตลอดเวลาและเริ่มทำงานอัตโนมัติเมื่อเปิดเครื่อง Instance

sudo npm install -g pm2

pm2 start node-red

เช็คสถานะ การทำงานของ Node-red

pm2 status

บันทึกการทำงานของ pm2 ด้วยคำสั่ง

pm2 save

การติดตั้ง MQTT broker ใน AWS Ec2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Update ตัวติดตั้ง

   sudo apt update

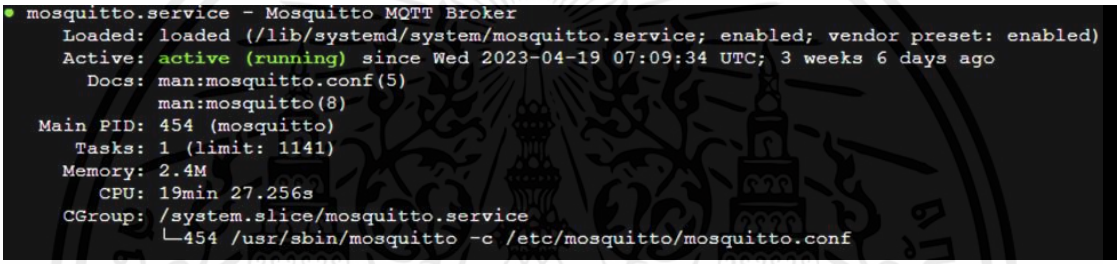
2. ติดตั้ง Mosquitto Broker

   sudo apt install -y mosquitto

3. ตรวจสอบสถานะของ mosquitto

   sudo systemctl status mosquitto

   สถานะทำงานปกติ



4. คำสั่งสำหรับจัดการ mosquitto

   หยุดการทำงาน

   sudo systemctl stop mosquitto

   เปิดการทำงาน

   sudo systemctl start mosquitto

   รีสตาร์ทการทำงาน

   sudo systemctl restart mosquito

   5 ติดตั้ง mosquitto client

   sudo apt install -y mosquitto-clients

   6 เปิด topic ที่ต้องการใช้งาน

   mosquitto_sub -t "Station/data"

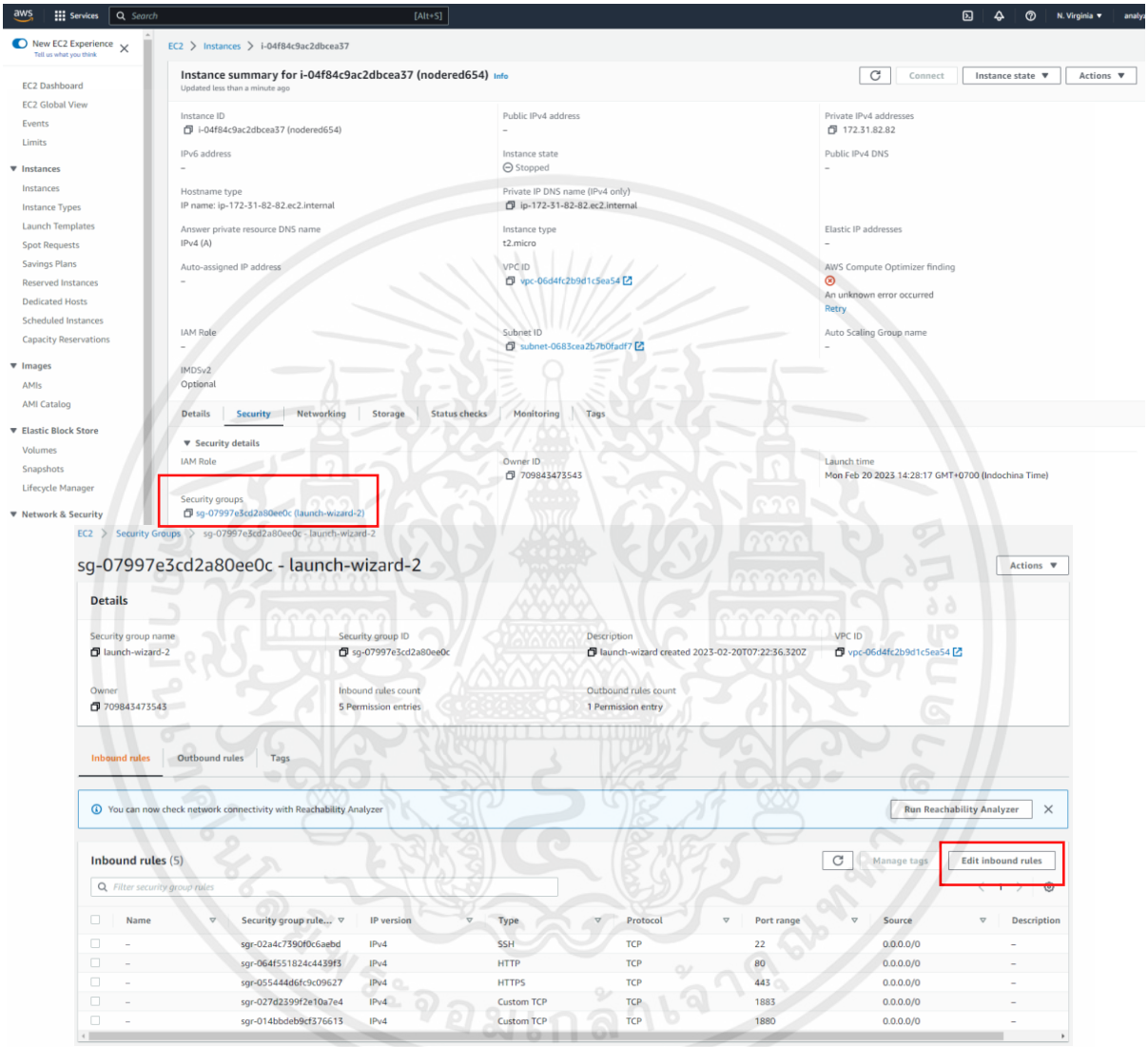
   7 ทดลองส่งค่าผ่าน MQTT

   mosquitto_pub -m "ON" -t " Station/data"

การเปิด port 1880 และ 1883 สำหรับ access node-red และ MQTT broker จากเครือข่ายภายนอก

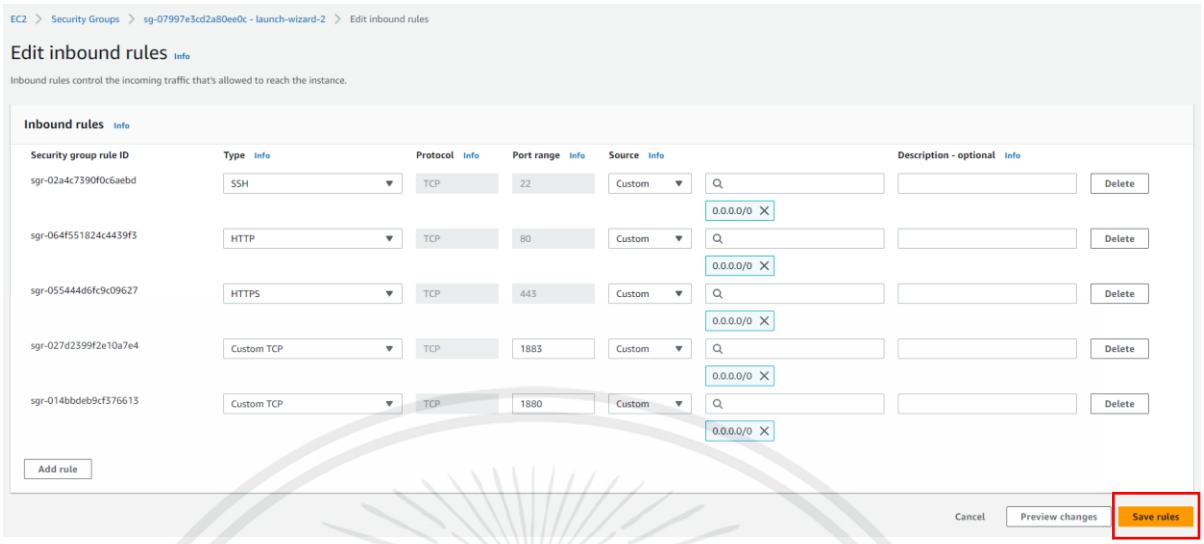
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เข้าไปที่ Instance ที่เราสร้าง เลื่อนลงมาด้านล่างจะพบข้อมูล Security Group กดที่ลิงค์เพื่อไปตั้งค่า Security Group
2. ไปที่ Edit Inbound rules เพื่อตั้งค่า Permission ต่างๆ ของแต่ละพอร์ต



3. จากนั้นกด Add Rule เพื่อสร้าง Rule ใหม่สำหรับ port 1880 และ 1883 โดยเลือก Source เป็นAny where เพื่อให้คนที่เข้ามาใช้ port 1880 และ 1883 เข้ามาจาก Global Network IP ใดบนโลกก็ได้

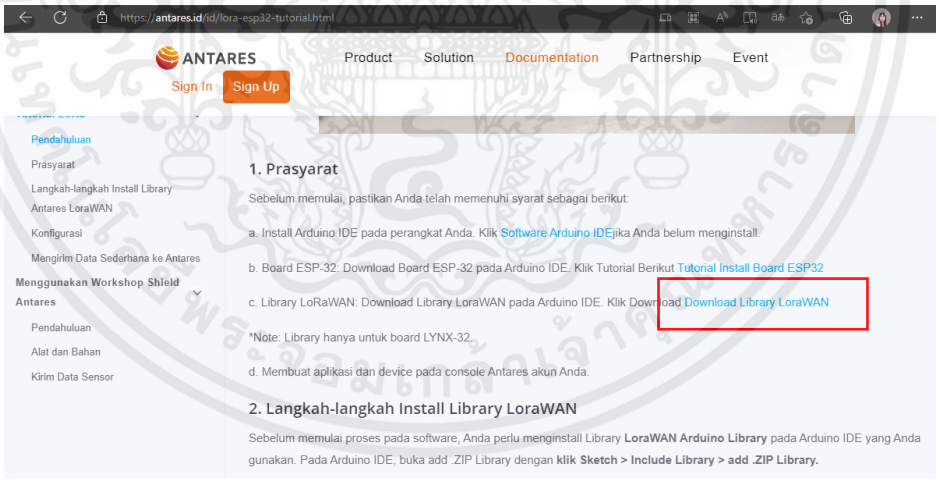
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



คลิกที่ Save rules เพื่อบันทึกการแก้ไข ทำให้พอร์ต 1880 และ 1883 พร้อมใช้งานแล้ว

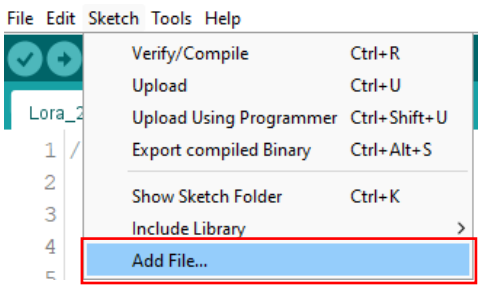
การติดตั้ง LoRaWAN Library

1. ทำการดาวน์โหลด LoRaWAN Library ของ Antares จากเว็บไซต์ https://antares.id/id/lora-esp32-tutorial.html

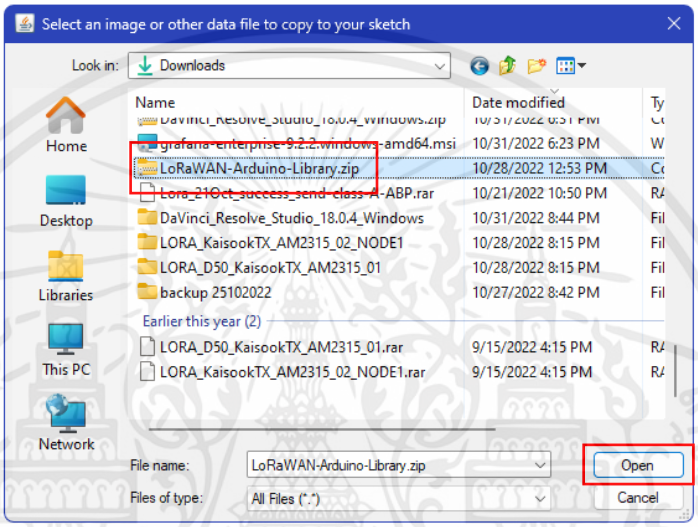


2. จากนั้นเปิดโปรแกรม Arduino และทำการ import library จากไฟล์ที่ดาวน์โหลดมา โดยไปที่ Sketch >> Add File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3. เลือกไฟล์ที่พึ่งดาวน์โหลดมาและกด open



Library LoRaWAN พร้อมใช้งานแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้